

**Universidad Jaume I**

Departamento de Ingeniería y Ciencia  
de los Computadores



**Hacia un Diseño Óptimo  
de la Arquitectura  
Multilayer Feedforward**

**Tesis Doctoral**

Realizada por: Mercedes Fernández Redondo

Dirigida por: Carlos A. Hernández Espinosa

Castellón, 11 de Septiembre de 2001



*Esta Tesis Doctoral ha sido realizada parcialmente con el apoyo de los siguientes proyectos:*

- *PIA97-19 de la Fundació Caixa-Castelló y la Universidad Jaume I, titulado “Redes neuronales: comparación de métodos de selección de información de entrada y algoritmos de aprendizaje que varían el número de unidades ocultas”*
- *GV99-75-1-14 de la Generalitat Valenciana, titulado “Optimización de la Capacidad de Generalización en redes neuronales de tipo Multilayer Feedforward”.*



## ***Agradecimientos***

*Quiero expresar mi agradecimiento a todas aquellas personas que me han apoyado y me ayudaron durante la realización de esta Tesis Doctoral.*

*En especial a Dr. Carlos Hernández Espinosa, director de esta Tesis Doctoral, sin cuyos consejos y colaboración no hubiese sido posible este trabajo. También quiero expresarle mi agradecimiento por su interés en la creación y desarrollo del Grupo de Investigación de Redes Neuronales y Computación Suave, al cual pertenezco.*

*Gracias también a mi familia y amigos.*



# ÍNDICE

## PRÓLOGO. Objetivos y Resumen de la Tesis Doctoral

### CAPÍTULO 1. Introducción

<i>1.1. Introducción</i>	1
<i>1.2. Definición de Red Neuronal y algoritmo de aprendizaje</i>	2
<i>1.3. Motivaciones de la Investigación</i>	5
<i>1.4. Arquitectura MF y algoritmo Backpropagation</i>	11
<i>1.5. Descripción de las Bases de Datos</i>	18
<i>1.6. Herramientas de trabajo: PC, SUN, Matlab</i>	24
<i>1.7. Bibliografía del Capítulo</i>	26

### CAPÍTULO 2. Procedimientos para el uso de entradas desconocidas

<i>2.1. Introducción</i>	28
<i>2.2. Procedimientos de utilización de entradas desconocidas:</i>	30
<i>2.2.1. Codificación mediante un valor</i>	30
<i>2.2.2. Codificación utilizando Aritmética de Intervalos</i>	30
<i>2.2.2.1 Operaciones con aritmética de intervalos</i>	32
<i>2.2.2.2 Multilayer Feedforward con aritmética de intervalos</i>	33
<i>2.2.2.3 Backpropagation con aritmética de intervalos</i>	37
<i>2.2.2.3.1 Algoritmo de Hernández</i>	38
<i>2.2.2.3.2 Algoritmo de Ishibuchi</i>	42
<i>2.2.2.3.3 Medidas de eficiencia</i>	46
<i>2.2.3. Codificación mediante Probabilidades</i>	47
<i>2.2.4. Codificación de entradas binarias, que no utiliza entradas desconocidas en el entrenamiento</i>	50
<i>2.3. Metodología</i>	51
<i>2.3.1. Elección de bases de datos</i>	51
<i>2.3.2. Elección de la arquitectura</i>	53
<i>2.3.3. Proceso de entrenamiento y test</i>	57
<i>2.4. Resultados experimentales</i>	58
<i>2.5. Conclusiones</i>	75
<i>2.6. Bibliografía del Capítulo</i>	77

**CAPÍTULO 3. Métodos de selección de la información de entrada a la red neuronal**

3.1. <i>Introducción</i>	80
3.2. <i>Métodos de selección de entradas basados en ARNMF</i>	85
3.2.1. <i>Primer grupo</i>	87
3.2.2. <i>Segundo grupo</i>	88
3.2.3. <i>Tercer grupo</i>	93
3.2.4. <i>Cuarto grupo</i>	94
3.2.5. <i>Quinto grupo</i>	95
3.2.6. <i>Sexto grupo</i>	99
3.2.7. <i>Séptimo grupo</i>	100
3.3. <i>Métodos de selección de entradas basados en ACE</i>	106
3.3.1. <i>Teoría de la Información</i>	106
3.3.2. <i>Distancias dentro y entre clases</i>	112
3.3.3. <i>Lógica difusa</i>	115
3.4. <i>Métodos de selección de entradas basados en ARE</i>	117
3.5. <i>Algoritmo de aprendizaje mediante gradientes conjugados</i>	117
3.6. <i>Uso de Weight Decay en los métodos ARNMF</i>	121
3.7. <i>Metodología</i>	123
3.8. <i>Resultados Experimentales</i>	131
3.9. <i>Aplicación a un problema real: patología de la voz</i>	145
3.10. <i>Conclusiones</i>	155
3.11. <i>Bibliografía del Capítulo</i>	157
3.12. <i>Apéndice del Capítulo</i>	161

**CAPÍTULO 4. Algoritmos de aprendizaje para la selección del número de unidades de la capa oculta de la red durante el proceso de aprendizaje**

4.1. <i>Descripción del Problema</i>	165
4.2. <i>Descripción de los diferentes métodos</i>	172
4.2.1. <i>Fase constructiva: determinación de nuevos pesos</i>	175
4.2.2. <i>Fase de podado: elección de la neurona a podar</i>	176
4.3. <i>Metodología</i>	188
4.4. <i>Desarrollo Experimental</i>	192
4.5. <i>Conclusiones</i>	198
4.6. <i>Bibliografía del Capítulo</i>	199
4.7. <i>Apéndice del Capítulo</i>	201

**CAPÍTULO 5. Estudio de la influencia en la capacidad de generalización de la red del número de capas ocultas**

5.1. <i>Descripción del Problema</i>	207
5.2. <i>Arquitectura Multilayer Feedforward con dos capas ocultas</i>	210
5.3. <i>Metodología</i>	213
5.4. <i>Desarrollo Experimental</i>	215
5.5. <i>Conclusiones</i>	218
5.6. <i>Bibliografía del Capítulo</i>	219



<b>CAPÍTULO 6. Métodos de inicialización de pesos para Redes Multilayer Feedforward</b>	
6.1. Descripción del Problema	221
6.2. Descripción de los diferentes métodos	226
6.2.1. Método 1	226
6.2.2. Método 2	226
6.2.3. Método 3	227
6.2.4. Método 4	230
6.2.5. Método 5	231
6.2.6. Método 6	233
6.2.7. Método 7	234
6.2.8. Método 8	236
6.2.9. Método 9	240
6.2.10. Método 10	243
6.2.11. Método 11	248
6.2.12. Método 12	253
6.2.13. Método 13	256
6.3. Metodología	261
6.4. Desarrollo Experimental	265
6.5. Conclusiones	273
6.6. Bibliografía del Capítulo	274
6.7. Apéndice del Capítulo	277
<b>CAPÍTULO 7. Conclusiones y trabajos futuros</b>	285
<b>CAPÍTULO 8. Referencias bibliográficas</b>	291
<b>APÉNDICES</b>	
I. Apéndices del Capítulo 2	305
I.1. Resultados de los métodos 1,4 y 5	305
I.2. Resultados de los métodos 2 y 3	312
I.3. Comparación entre los métodos 2 y 3	350
II. Apéndices del Capítulo 3	361
II.1. Ordenación de las entradas para ARNMF y ACE	361
II.2. Ordenación de las entradas para ARNMF&WD	378
II.3. Eficiencias para los métodos ARNMF y ACE	383
II.4. Eficiencias para los métodos ARNMF&WD	413
II.5. Comparación entre métodos ARNMF y ACE	424
II.6. Comparación entre métodos ARNMF con y sin WD	459

## *ÍNDICE*

# Prólogo

## OBJETIVOS Y RESUMEN DE LA TESIS DOCTORAL

### Objetivos y Resumen de la Tesis Doctoral

El objetivo de esta Tesis Doctoral ha sido básicamente el de realizar un estudio comparativo sobre los diferentes métodos existentes para resolver diversos aspectos referentes al diseño de la arquitectura de red neuronal Multilayer Feedforward, en problemas de clasificación. Hemos elegido la arquitectura Multilayer Feedforward por que es una de las más utilizadas a la hora de realizar aplicaciones en el campo de las Redes Neuronales.

Los aspectos de diseño de la arquitectura de red neuronal estudiados han sido:

- Codificación de la información o entradas desconocidas en la red.
- Selección de la información de entrada a la red.
- Selección del número de unidades ocultas de la red.
- Influencia en la capacidad de generalización del número de capas ocultas.
- Inicialización de la red.

En cada uno de los aspectos se ha realizado un estudio comparativo de los diferentes métodos existentes para resolver dicho problema. Todos los aspectos están ampliamente relacionados entre sí, puesto que en todos ellos el objetivo final es seleccionar una arquitectura de red idónea para realizar una determinada aplicación; entendiendo por idónea, la selección del número apropiado de capas ocultas, de unidades de la capa oculta, de unidades en la capa de entrada y la obtención de una buena capacidad de generalización.

En el primer capítulo, se encuentra una pequeña introducción a conceptos de Redes Neuronales (definición, problemas de diseño, arquitecturas usadas) y a las herramientas de trabajo utilizadas en el desarrollo de esta Tesis Doctoral, tales como bases de datos, equipamiento y software. En él se justifica de forma más detallada la motivación de esta investigación.

En el segundo capítulo, se describe la investigación realizada sobre el problema de codificación de entradas desconocidas. Se ha realizado un estudio comparativo de 5 métodos diferentes para la utilización de entradas desconocidas y tras un estudio experimental bastante meticuloso se ha propuesto uno de estos métodos como el más eficiente.

En el tercer capítulo, se realiza un estudio sobre la importancia de la información de entrada, analizándose diferentes métodos existentes de seleccionar de forma automática la información de entrada a la red. Se han comparado 26 métodos: 19 basados en el análisis de la red neuronal Multilayer Feedforward y 7 basados en el análisis del conjunto de entrenamiento. Se ha medido la eficiencia de los mismos aplicándolos a 15 problemas de clasificación. Como resultado, obtenemos la ordenación de los métodos según su eficiencia, recomendando finalmente el uso de los mejores a la hora de realizar una aplicación concreta. En el capítulo también se menciona un ejemplo de aplicación real.

En el capítulo 4, se muestra un estudio comparativo entre los algoritmos de aprendizaje avanzados, que seleccionan de forma dinámica el número de unidades (neuronas) de la capa oculta de la red durante el proceso de aprendizaje. Se diferencia entre dos tipos de procedimientos: constructivos y de podado. Experimentalmente, se combinó el uso de los mismos de forma que la selección final de la arquitectura se realizó añadiendo primero neuronas a la capa oculta y luego testeando si durante este proceso constructivo se excedió el número de neuronas idóneo y en su caso eliminándolas. También concluimos cuál es el mejor procedimiento a seguir para resolver este problema.

En el capítulo 5, se estudia de qué manera influye en la capacidad de generalización de la red el número de capas ocultas seleccionado para la arquitectura de la misma. La conclusión final fue que la introducción de más de una capa oculta deteriora la capacidad de generalización.

En el capítulo 6, se aborda el problema de inicialización de la red neuronal previa a su entrenamiento, en concreto de los pesos. Como veremos, la inicialización de los pesos influye en diferentes aspectos de la red como son: su capacidad de generalización, la velocidad de convergencia y el éxito en la convergencia. Mediante un estudio comparativo entre los diferentes métodos que

se encuentran en la bibliografía concluiremos qué procedimientos resultan más ventajosos desde los tres puntos de vista y recomendaremos su utilización.

En el capítulo 7, se encuentra un resumen más extenso de las conclusiones y aportaciones de esta Tesis Doctoral, así como posibles trabajos de investigación futuros y áreas de aplicabilidad experimental de los resultados obtenidos.

Finalmente, en el capítulo 8, se recogen las referencias bibliográficas que se aportaron el final de cada capítulo ordenadas alfabéticamente.

En el apéndice general de la Tesis se ha incluido debido a su extensión algunos resultados experimentales correspondientes al segundo y tercer capítulos.

*OBJETIVOS Y RESUMEN DE LA TESIS DOCTORAL*

# Capítulo 1

## INTRODUCCIÓN

- 1.1. Introducción*
- 1.2. Definición de Red Neuronal y algoritmo de aprendizaje*
- 1.3. Motivaciones de la Investigación*
- 1.4. Arquitectura MF y algoritmo Backpropagation*
- 1.5. Descripción de las Bases de Datos*
- 1.6. Herramientas de trabajo: PC, SUN, Matlab*
- 1.7. Bibliografía del Capítulo*

### 1.1. Introducción

En este capítulo nos ocuparemos sólo de aquellas cuestiones sobre redes neuronales que son imprescindibles para la investigación que pretendemos realizar, puesto que nuestro interés se centra en mejorar diversos aspectos del diseño de la red neuronal Multilayer Feedforward.

Veremos en primer lugar algunas definiciones clásicas de red neuronal con la intención de aportar una pequeña base sobre el tema. En segundo lugar, describiremos detalladamente las motivaciones de esta investigación destacando el gran interés de realizar estudios comparativos entre diversos métodos existentes para resolver un determinado problema. Posteriormente, se describirá la arquitectura Multilayer Feedforward y el algoritmo más empleado usualmente en su entrenamiento, Backpropagation.

Finalmente, se presentan las herramientas de trabajo empleadas en el desarrollo de la Tesis Doctoral: bases de datos, software y equipamiento (PCs).

Así mismo también, incluimos una selección bibliográfica básica sobre redes neuronales de interés general.

## 1.2. Definición de Red Neuronal y algoritmo de aprendizaje

En la bibliografía pueden encontrarse muchas y variadas definiciones de red neuronal. Citamos aquí dos de las que a nuestro parecer son más representativas.

La primera de ellas, podemos encontrarla en la referencia [4] y es la siguiente:

### **Una definición de Red Neuronal:**

*“ Una Red Neuronal es un sistema computacional, implementado bien por hardware o software, que trata de imitar las habilidades computacionales de los sistemas biológicos, mediante el uso de un gran número de simples neuronas artificiales interconectadas.*

*Las neuronas artificiales son emulaciones sencillas de las neuronas biológicas, toman la información de un sensor u otra neurona artificial, realizan operaciones muy elementales con estos datos y pasan los resultados de esta información a otras neuronas artificiales.*

*Las redes neuronales operan procesando los datos mediante sus neuronas en la manera descrita. Usan a la vez una propagación en paralelo y en serie de la información.*

*Las tres principales características que describen a una red neuronal son: estructura, dinámica y aprendizaje.”*

En esta definición, se observa el interés por modelar el comportamiento de nuestro cerebro mediante unidades que realicen operaciones sencillas y que estén muy interconectadas entre sí al igual que nuestras neuronas. También se da una noción de su uso, es decir, es necesario definir una estructura que indique de qué forma se comunican entre sí las unidades de la red, hay que especificar como realizan la comunicación y para que simulen el comportamiento del cerebro es necesario hacerlas aprender de alguna manera (algoritmo de aprendizaje).

En la referencia [5] podemos encontrar la segunda definición.



**Otra definición de Red Neuronal.**

*“Una red neuronal es una estructura de procesamiento de la información distribuida y en paralelo, compuesta de elementos que realizan el tratamiento de la información.*

*Dichos elementos pueden poseer una memoria local, y deben llevar a cabo operaciones de proceso de la información localizadas. Además, estarán interconectadas entre sí vía canales unidireccionales de propagación de la información, llamados conexiones.*

*Cada elemento procesador, tiene una única salida que se divide en muchas conexiones colaterales, transportando cada una de ellas la misma señal (la señal de salida del elemento procesador). Esta señal puede ser de cualquier tipo matemático; el tratamiento de la información que se lleva a cabo dentro de cada elemento procesador, puede ser definido arbitrariamente con la restricción de que debe ser completamente local. Es decir, debe depender sólo de los valores actuales de las señales de entrada que llegan a ese elemento procesador, vía conexiones, o de valores acumulados en la memoria del elemento procesador.”*

Esta definición es mucho más abstracta, tiene una visión más matemática. Se habla de unidades de proceso de información que tratan señales. Se insiste sobre la amplia interconexión, una unidad calcula su salida (la cual propaga a otras unidades) usando información interna (que le llegó de otras unidades).

En ambas definiciones se habla principalmente de la estructura de la red pero no se dice nada sobre su aprendizaje, podríamos utilizar como algoritmo de aprendizaje cualquiera que permita variar los parámetros libres de la red (valores de las conexiones o los pesos). Así como también, en ambas definiciones se habla de procesamiento serie y paralelo de la información.

Podemos deducir varias ideas a partir de estas definiciones.

Una red neuronal posee una estructura. Se entiende por estructura la descripción matemática de la red y su funcionamiento, no la manera en que se implementa (que puede ser por software o hardware).

En la descripción matemática, se debe determinar la configuración de las unidades (neuronas) e interconexiones entre ellas y describir de qué forma se propagan las señales en la estructura indicando las operaciones que realizan.

Además, en la estructura habrá unidades de entrada (por las que se recibe información externa) y unidades de salida (por las que se proporciona la información resultante de la red).

## *1.2. DEFINICIÓN DE RED NEURONAL Y ALGORITMO DE APRENDIZAJE*

Internamente, la red puede poseer otras neuronas o unidades que se agruparán en diferentes capas (transmisión paralela de la información) según sea la función matemática que realicen.

Las unidades de la red se comunican entre sí mediante conexiones, a las cuales como veremos más adelante, se les asocia un valor denominado peso.

Con esto, tenemos básicamente determinados los elementos de una red neuronal.

Sin embargo, necesitamos un mecanismo que ayude a que la red neuronal aprenda a realizar o resolver una cierta tarea, como ocurre con las neuronas de nuestro cerebro. Para ello, se utiliza lo que se llama algoritmo de aprendizaje. La tarea del algoritmo será adaptar los valores de las interconexiones de las neuronas (pesos) durante el llamado proceso de aprendizaje para que al final del proceso la red sea capaz de solucionar un problema de forma más o menos eficiente.

El proceso de aprendizaje de la red puede ser generalmente de varios tipos: supervisado o no supervisado. En el primer caso la red adapta sus parámetros utilizando en el proceso de aprendizaje lo que se llaman ejemplos de entrenamiento, es decir, un conjunto de datos de entrada de los cuales se conoce la respuesta (o salida) que deseamos que proporcione la red. En el segundo, se tienen patrones de entrada a la red pero no se usa ninguna respuesta o salida deseada (objetivo del patrón).

En esta tesis, nos hemos centrado en el uso de la red neuronal Multilayer Feedforward generalmente usando el algoritmo Backpropagation. Ambos, están descritos en el apartado 1.4 de este capítulo.

## 1.3. Motivaciones de la Investigación

La investigación sobre Redes Neuronales ha experimentado un gran auge desde el año 1987 en el que se celebró la primera conferencia internacional abierta sobre redes neuronales (International Joint Conference on Neural Networks, IJCNN'87).

En los últimos años, se han realizado con éxito un gran número de aplicaciones basadas en Redes Neuronales en campos tan variados como comercio, ciencia, industria, medicina, etc.

Una revisión amplia sobre estos aspectos puede encontrarse en la referencia [4]. Se trata de un texto bastante antiguo (del año 1990) pero da una idea de la gran variedad y cantidad de aplicaciones sobre este tema.

Las aplicaciones sobre Redes Neuronales van adquiriendo progresivamente gran interés comercial de forma que se van transfiriendo a los sectores productivos o las áreas socioeconómicas. Al mismo tiempo que la calidad y corrección de las soluciones basadas en redes neuronales es en general igual o superior a la que puedan aportar sistemas basados en otras tecnologías.

Si hacemos una clasificación muy general, mediante redes neuronales pueden resolverse básicamente dos tipos de problemas.

- ***Problemas de clasificación***
- ***Problemas de aproximación funcional.***

En los primeros, se pretende distinguir entre categorías, clases de objetos o situaciones. Ejemplos de problemas de clasificación son la identificación de fallos o defectos, el diagnóstico médico de enfermedades, etc.

Ejemplos del segundo tipo son la predicción de series temporales (de utilidad en la predicción de demandas como la energía eléctrica, o la predicción de valores bursátiles), la identificación de sistemas, la realización de sistemas de control, etc.

Como vemos, el ámbito de aplicación de las redes neuronales es muy amplio, puesto que casi cualquier problema se puede encuadrar como un caso de clasificación o de aproximación funcional.

Las redes neuronales (en sistemas que utilizan algoritmos supervisados) tienen además la característica de ser capaces de llegar a resolver un problema tan sólo basándose en ejemplos concretos (información concreta) de resolución de

### 1.3. MOTIVACIONES DE LA INVESTIGACIÓN

dicho problema. Solo necesitan una muestra representativa de los ejemplos de resolución del problema, esto es lo que se denomina conjunto de entrenamiento.

Tras un proceso de aprendizaje con dichas muestras las redes neuronales tienen la capacidad de responder correctamente a otros ejemplos del problema relativamente similares a los del conjunto de entrenamiento pero que jamás han sido utilizados antes por la red. A esa capacidad de aprendizaje de la red es a lo que se denomina capacidad de generalización, y como veremos más adelante, es probablemente la característica más importante de una red neuronal y lo que las hace verdaderamente útiles. A mayor capacidad de generalización más eficiente se considera la solución suministrada por la red neuronal.

Como estudiaremos más adelante, los factores que influyen en la capacidad de generalización de una red neuronal son muchos, aunque básicamente se pueden resumir en tres: su *arquitectura* (o estructura interna), el *algoritmo de aprendizaje* que se emplee para realizar el entrenamiento y el conjunto de datos (ejemplos) que se incluyan en *el conjunto de entrenamiento*.

El algoritmo de aprendizaje es el que hace que la red adapte de una determinada forma sus parámetros para llegar a una cierta solución con los ejemplos que se le suministran. Parece evidente que la elección de estos parámetros influirá en el comportamiento final de la red y por tanto en su eficiencia.

Ocurrirá lo mismo con la elección del conjunto de entrenamiento. Este debe contener datos verdaderamente representativos para la resolución del problema. Si la calidad de los datos (información sobre el problema) es baja, los datos no serán representativos de todos los aspectos del problema, es decir, se estará incluyendo información no útil, por lo que no podemos esperar que la red resuelva bien el problema.

El otro factor que influye en la eficiencia que proporcione la red es la arquitectura de la misma. Dicha estructura fija el modelo que a priori hemos decidido que usaremos para resolver el problema. Según el modelo, se obtendrán resultados diferentes y por tanto la eficiencia obtenida por la red será distinta.

Si se revisa la bibliografía sobre redes neuronales, la arquitectura más utilizada es Multilayer Feedforward (MF). Por ello, esta investigación se ha centrado en el estudio detallado sobre varios aspectos de la misma.

La red neuronal Multilayer Feedforward está constituida básicamente por tres capas de neuronas, una capa de entrada, una capa oculta y otra de salida. Más adelante realizaremos una descripción detallada de la misma.

Por otra parte, si revisamos la bibliografía, se puede ver que en la mayoría de casos se utilizan en el diseño de la red técnicas de las primeras que se propusieron, como por ejemplo algoritmos de aprendizaje muy básicos. Sin embargo, a lo largo de los años se han propuesto muchos algoritmos de aprendizaje más avanzados con el objetivo de aumentar la capacidad de generalización de las redes neuronales.

¿Por qué entonces se utilizan en muchas aplicaciones actuales algoritmos de aprendizaje o procedimientos ya superados (los más básicos)? La respuesta es múltiple.

En primer lugar, para una persona que solo tenga unos conocimientos mínimos sobre redes neuronales puede ser imposible utilizar algo avanzado, simplemente por desconocimiento de su existencia. Sin embargo, aún suponiendo que la persona que realiza la aplicación sea un experto en redes neuronales, puede encontrarse con el problema de no saber que procedimiento escoger entre la multitud de los existentes. Si dicha persona revisa la bibliografía se encontrará que en general no existen comparaciones entre los diferentes procedimientos que le indiquen cual debe ser su elección.

En definitiva, se encontrará ante una multitud de alternativas, con escasa información que le permita seleccionar la más idónea o más eficiente. Como mucho, puede seleccionar uno o varios métodos avanzados y esperar que alguno funcione bien, o simplemente usar los procedimientos más clásicos, ampliamente conocidos. Esta es una situación bastante habitual en el campo de redes neuronales.

Actualmente, en este campo se están proponiendo nuevas arquitecturas, nuevos algoritmos, nuevas implementaciones físicas, etc. Sin embargo, cuando se propone, por ejemplo un nuevo algoritmo de aprendizaje o un procedimiento para aumentar la eficiencia, *las comparaciones* que se realizan con algoritmos o procedimientos conocidos anteriores, son bastante exiguas o prácticamente inexistentes.

Además, en el caso de que se incluya alguna comparación se suele realizar respecto a procedimientos básicos y no respecto a procedimientos avanzados que persigan el mismo objetivo que lo que se propone.

***Por ello, pensamos que el interés de proponer nuevos algoritmos se reduce bastante si no se sabe exactamente cuál es su eficiencia con respecto a procedimientos ya existentes.***

### 1.3. MOTIVACIONES DE LA INVESTIGACIÓN

*Así, este es el momento apropiado para realizar un estudio comparativo amplio sobre métodos y algoritmos ya existentes, ya que todo procedimiento nuevo presentará los problemas que hemos comentado anteriormente.*

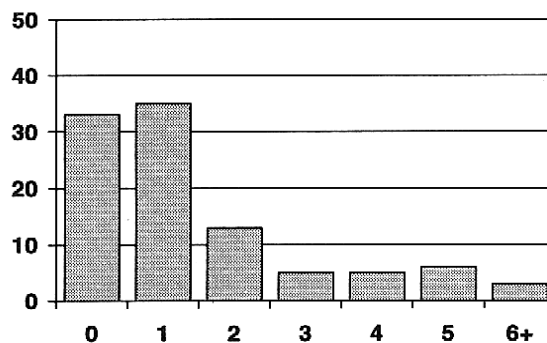
En la bibliografía [2], podemos encontrar algunos aspectos interesantes en este sentido, los cuales avalan plenamente lo que se ha expuesto aquí.

En la referencia, se presenta un estudio detallado de 190 artículos referentes a nuevos algoritmos de aprendizaje de redes neuronales que fueron publicados en los años 1993 y 1994. En dicho estudio, se mide por una parte la cantidad de comparaciones realizadas con otros algoritmos ya existentes y por otro lado, se evalúa el número de problemas utilizados para medir la eficiencia de los nuevos algoritmos.

Los artículos utilizados en el estudio de la referencia se obtuvieron de cuatro conocidas revistas dedicadas a la investigación sobre redes neuronales: *Neural Networks*, *Neural Computation*, *Neurocomputing* y *IEEE Transactions on Neural Networks*.

En la figura 1 se presentan algunos resultados de dicho estudio.

En ella, se clasifican de forma porcentual los algoritmos de aprendizaje que se proponen en dichos años en función del número de comparaciones realizadas por el autor de la referencia, con otros algoritmos ya existentes. Se puede ver que aproximadamente un 33% de los nuevos algoritmos propuestos en estos años no se compararon con ningún otro algoritmo conocido.

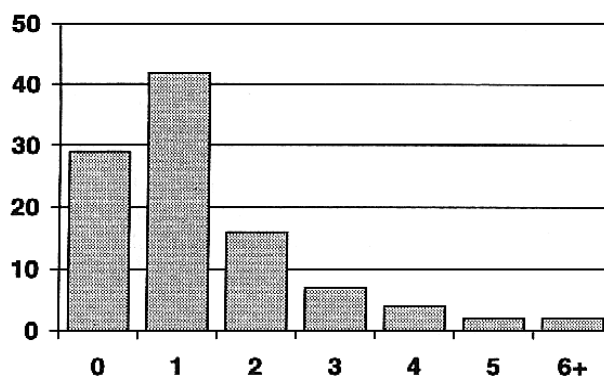


**Fig. 1. Porcentaje de algoritmos de aprendizaje en función del número de comparaciones con otros algoritmos conocidos.**

La comparación con otros algoritmos de aprendizaje conocidos y la propia evaluación experimental de los algoritmos propuestos según la calidad y cantidad de problemas que se utiliza para evaluarlo, resultan exiguas.

En la referencia [2], como hemos mencionado antes, se realiza también un estudio cuantitativo sobre las evaluaciones experimentales que se suelen emplear para los nuevos algoritmos propuestos.

En la figura 2 se muestra una clasificación porcentual de los nuevos algoritmos de aprendizaje propuestos en los años 1993-1994, en función del número de problemas reales que se usaron para su evaluación experimental. Como se aprecia en la figura tan sólo en un 2% de los nuevos algoritmos que se proponen utilizaron seis o más problemas reales. Todo



**Fig.2. Porcentaje de algoritmos de aprendizaje en función del número de problemas reales usados para su comparación.**

esto, lleva a la situación de tener varios procedimientos para realizar una determinada cosa, y no saber cuál es el más eficiente o idóneo, dificultando la elección de uno por aquella persona que pretenda realizar una aplicación. La determinación de cuáles son los procedimientos más eficientes o idóneos, es de especial importancia tanto para el transvase de tecnología avanzada existente al ámbito industrial, como para el progreso en sí del campo de redes neuronales.

Hemos usado la referencia [2], del año 1996, para mostrar que la situación respecto al número de comparaciones realizadas y el número de problemas a los que se aplican los nuevos algoritmos no ha variado mucho a lo largo de los años, la situación pues está estancada, por lo que consideramos de gran interés realizar comparaciones entre diferentes técnicas existentes para el diseño de aspectos concretos de la red.

También consideraremos importante utilizar un número suficiente de problemas para medir la eficiencia de los métodos que estudiemos.

Esta Tesis Doctoral se titula: “Hacia un diseño óptimo de la Arquitectura Multilayer Feedforward”. Como indica el título, se trata en ella de intentar optimizar algunos aspectos sobre el diseño de la red. Con la palabra “optimizar” no pretendemos expresar que se realizarán nuevos procedimientos para mejorar el diseño de la red, sino que se hará un estudio detallado de procedimientos ya existentes y se justificará cual de ellos resulta óptimo en cada uno de los aspectos de diseño que trataremos. Nuestro principal interés, será estudiar por separado diversos aspectos sobre el diseño de la red, realizando para ello un estudio de los diferentes métodos existentes para solucionar ese aspecto concreto. Posteriormente, se llevará a cabo un análisis comparativo entre los métodos estudiados.

### 1.3. MOTIVACIONES DE LA INVESTIGACIÓN

Para comparar las eficiencias de los métodos se aplicarán los mismos a un número considerable de problemas de clasificación, la mayor parte de ellos reales, cuyos datos han sido obtenidos del repositorio UCI [1].

Por último, se concluirá que métodos resultan más eficientes para cada uno de los aspectos del diseño de la red que hemos estudiado.

Como introducción a nuestra investigación se describe en el apartado siguiente la arquitectura de la red *Multilayer Feedforward (MF)* y el algoritmo de aprendizaje más básico y más usado en su entrenamiento, *Backpropagation*.



## 1.4. Arquitectura MF y algoritmo Backpropagation

Como hemos dicho antes en esta investigación nos centraremos en la arquitectura *Multilayer Feedforward*, en general de tres capas y realizaremos un estudio detallado sobre varios aspectos relativos a su diseño.

Existen muchos algoritmos de aprendizaje para esta red neuronal, pero uno de los más básicos, descrito y usado en la bibliografía es el conocido como *Backpropagation*. Por ello, en este primer capítulo vamos a describir cuál es la arquitectura (estructura) de la red neuronal Multilayer Feedforward y obtendremos las ecuaciones y funcionamiento del algoritmo de aprendizaje Backpropagation.

La red Multilayer Feedforward está constituida por tres capas diferentes de neuronas (o unidades), la capa de entrada, la capa oculta y la capa de salida.

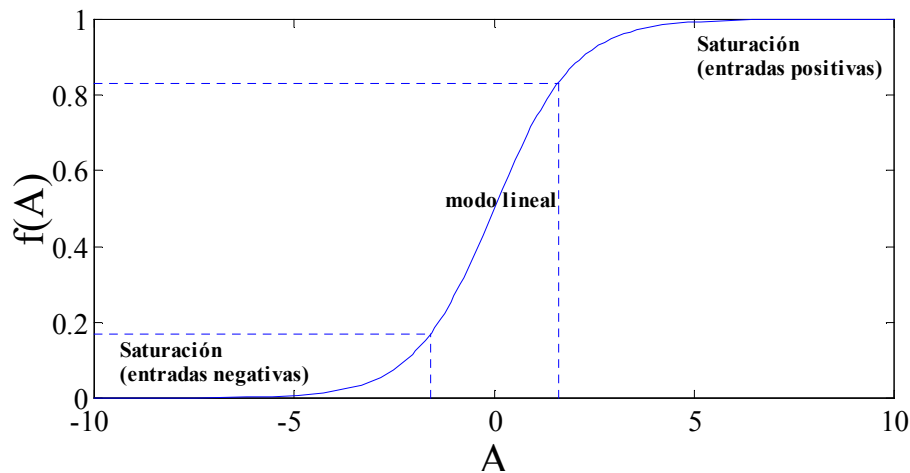
La capa de entrada está formada por un número de neuronas fijado por el número o cantidad de información que queremos procesar con la red (características de la información). Esta capa simplemente transmite la información de entrada al interior de la red, sus neuronas están totalmente interconectadas con las de la capa oculta.

La capa oculta (formada por un cierto número de neuronas) realiza un procesamiento de la información de entrada a sus neuronas mediante una función de transferencia de tipo sigmoide (aunque en ocasiones puede usarse otro tipo de funciones). Las neuronas están totalmente interconectadas con las neuronas de la capa de entrada y cada conexión lleva asociada un peso, de forma que la entrada a una neurona oculta será una combinación lineal de las entradas ponderadas por los pesos que entran en dicha neurona oculta. La salida de cada neurona se obtendrá aplicando la función de transferencia sobre cada combinación.

Por último, la capa de salida está totalmente interconectada con la capa oculta, hay un peso asociado a cada conexión y el procesamiento de la información de las neuronas de salida es la suma de las señales de salida de la capa oculta ponderada por los pesos. Las entradas a las neuronas de salida serán también una combinación lineal de la salida de las neuronas ocultas y la salida de la red se obtendrá aplicando la función de transferencia sobre dicha combinación.

En la figura 3 se observa el comportamiento de la función de transferencia de tipo sigmoide y los modos de activación de la misma.

#### 1.4. ARQUITECTURA MF Y ALGORITMO BACKPROPAGATION



**Fig.3. Función de transferencia sigmoide y sus modos de activación.**

La expresión de la función de transferencia sigmoide viene dada por la ecuación siguiente.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1.1)$$

Para cada capa tendremos un vector formado por los valores de entrada y otro por los de salida de las neuronas de dicha capa.

#### **RED MULTILAYER FEEDFORWARD CON UNA CAPA OCULTA**

En la figura 4 se encuentra dibujada la estructura de una red Multilayer Feedforward de tres capas.

Llamemos a los vectores de salida de las diferentes capas como sigue.

$$\text{Capa de entrada} \quad \{i_1, i_2, \dots, i_N\} = \{io_1, io_2, \dots, io_N\} = \{a_1, a_2, \dots, a_N\}$$

$$\text{Capa oculta} \quad \{ho_1, ho_2, \dots, ho_{Nh}\} = \{f(hi_1), f(hi_2), \dots, f(hi_{Nh})\}$$

$$\text{Capa de salida} \quad \{oo_1, oo_2, \dots, oo_{Ns}\} = \{f(oi_1), f(oi_2), \dots, f(oi_{Ns})\}$$

Veamos cuales son las ecuaciones de la red.

En la ecuación (1.2) mostramos la expresión de la entrada a una neurona  $j$  de la capa oculta como una combinación de las entradas a la red ponderadas con los pesos.

$$hi_j = \sum_{i=1}^N wi_{ij} \cdot a_i + \theta_j \quad (1.2)$$

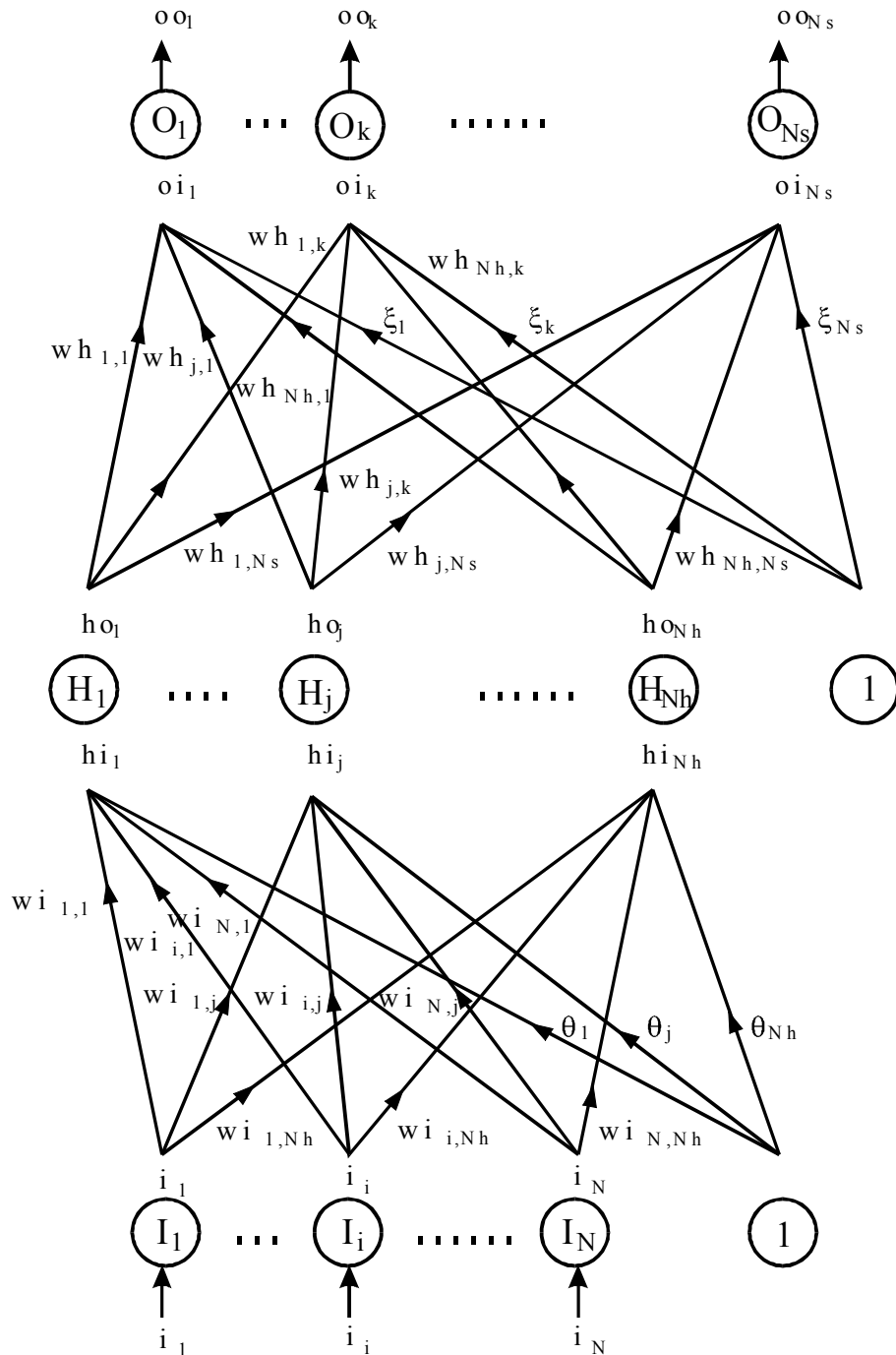


Fig.4. Estructura de Red Neuronal Multilayer Feedforward.

#### 1.4. ARQUITECTURA MF Y ALGORITMO BACKPROPAGATION

La ecuación (1.3) expresa la entrada a una neurona  $k$  de la capa de salida en función de una combinación lineal de las salidas de las neuronas de la capa oculta ponderadas con sus pesos.

$$oi_k = \sum_{j=1}^{Nh} wh_{jk} \cdot ho_j + \xi_k \quad (1.3)$$

La ecuación (1.4) nos da la salida de una neurona  $j$  de la capa oculta, obtenida aplicando la función de transferencia sobre la entrada (1.2) a la misma.

$$ho_j = \frac{1}{1 + \exp(-hi_j)} \quad (1.4)$$

La ecuación (1.5) describe la salida de una neurona  $k$  de la capa de salida, obtenida aplicando la función de transferencia sobre la entrada (1.3) a la misma.

$$oo_k = \frac{1}{1 + \exp(-oi_k)} \quad (1.5)$$

En las ecuaciones anteriores  $\theta$  y  $\xi$  son los vectores de “threshold” o umbrales que intervienen en la capa oculta y en la capa de salida de la red. Estos se consiguen introduciendo una neurona con salida constante de valor 1 en la capa inferior a la estudiada.

En las ecuaciones anteriores  $wi_{ij}$  y  $wh_{jk}$  son los pesos que representan las conexiones entre capas diferentes (ver figura 4).

Una vez definida la arquitectura de la red para poder utilizarla en la resolución de un problema concreto es necesario entrenar la red para que aprenda a resolver dicho problema.

Para ello, será necesario disponer de un conjunto suficiente de ejemplos (patrones) de entrenamiento con sus correspondientes objetivos. Es decir, cada patrón de entrada lleva asociado un objetivo o resultado que sería la solución correcta al problema.

Para realizar el entrenamiento se van presentando los diferentes ejemplos a la red y mediante un algoritmo de aprendizaje esta debe de acercarse cada vez más al objetivo que corresponde al ejemplo que se le presenta. Cuando los resultados que suministre la red sean lo suficientemente cercanos a los objetivos podremos considerar como acabado el entrenamiento.

Llamamos  $I$  al vector que representa un ejemplo y  $T$  al vector del objetivo asociado a dicho ejemplo. Sea  $P$  el número total de ejemplos,  $N$  el número de

neuronas de entrada a la red (dimensión de  $I$ ),  $N_s$  el número de neuronas de salida de la red (dimensión de  $T$ ) y  $N_h$  el número de neuronas ocultas.

Veamos el funcionamiento del algoritmo *Backpropagation*. En primer lugar se define el “*Error Cuadrático Medio*” que mide la diferencia entre el resultado que proporciona la red para un cierto ejemplo  $P$  y el objetivo de dicho ejemplo.

$$E_P = \frac{1}{2} \cdot \sum_{k=1}^{N_s} (t_{P,k} - oo_{P,k}) \quad (1.6)$$

El error total será la suma para todos los patrones de los errores obtenidos en la ecuación (1.6).

El proceso de aprendizaje consiste en minimizar (respecto de los pesos) la función de error para todos los patrones del conjunto de entrenamiento, los cuales se van presentando a la red en iteraciones sucesivas hasta conseguir la convergencia de la función de error.

Para cada iteración los pesos de la red se irán modificando y modificarán las entradas y salidas de las neuronas y el error en la siguiente iteración.

El procedimiento a seguir será:

1) *Inicializar aleatoriamente o mediante algún método los pesos de la red.*

2) *Introducir un patrón y modificar los pesos de la red según las ecuaciones:*

$$\Delta wh_{j,k}(t+1) = \eta \cdot \left( -\frac{\partial E_P}{\partial wh_{j,k}} \right) + \beta \cdot \Delta wh_{j,k}(t); \quad (1.7)$$

$$wh_{j,k}(t+1) = wh_{j,k}(t) + \Delta wh_{j,k}(t+1)$$

$$\Delta wi_{i,j}(t+1) = \eta \cdot \left( -\frac{\partial E_P}{\partial wi_{i,j}} \right) + \beta \cdot \Delta wi_{i,j}(t); \quad (1.8)$$

$$wi_{i,j}(t+1) = wi_{i,j}(t) + \Delta wi_{i,j}(t+1)$$

*Donde  $\eta$  es el paso de adaptación y  $\beta$  es el momento.*

3) *Calcular el Error después de acabada la iteración. Tomar el resultado final de los pesos obtenidos como la nueva inicialización de pesos para la siguiente iteración.*

4) *Repetir los pasos 2 y 3 durante un número determinado de iteraciones que permita la convergencia de la función de error (hacia un mínimo), es decir, que el Error calculado en 3 alcance un mínimo.*

#### 1.4. ARQUITECTURA MF Y ALGORITMO BACKPROPAGATION

Para realizar el cálculo de los pesos en cada iteración, vemos que es necesario obtener las expresiones de las derivadas de la función de error respecto de los pesos.

Primero calculamos las derivadas para los pesos entre la capa oculta y la de salida. A partir de la ecuación (1.6) vemos la dependencia de  $E_p$  con  $oo_{P,k}$  y la de esta (ecuación (1.5)) con la variable  $oi_{P,k}$ .

Aplicando la regla de la cadena para derivar obtenemos:

$$\frac{\partial E_p}{\partial wh_{j,k}} = \frac{\partial E_p}{\partial oo_{P,k}} \cdot \frac{\partial oo_{P,k}}{\partial oi_{P,k}} \cdot \frac{\partial oi_{P,k}}{\partial wh_{j,k}} \quad (1.9)$$

A partir de la ecuación (1.6) se tiene:

$$\frac{\partial E_p}{\partial oo_{P,k}} = -(t_{P,k} - oo_{P,k}) \quad (1.10)$$

De la ecuación (1.1) podemos deducir la derivada de la función de transferencia de nuestra red.

$$f'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = f(x) \cdot (1 - f(x)) \quad (1.11)$$

De las ecuaciones (1.5), (1.11) y (1.3) se puede deducir que:

$$\frac{\partial oo_{P,k}}{\partial oi_{P,k}} = oo_{P,k} \cdot (1 - oo_{P,k}) \quad (1.12)$$

$$\frac{\partial oi_{P,k}}{\partial wh_{j,k}} = ho_{P,j} \quad (1.13)$$

Definimos la siguiente variable

$$\delta_{P,k} = -(t_{P,k} - oo_{P,k}) \cdot oo_{P,k} \cdot (1 - oo_{P,k}) \quad (1.14)$$

Sustituimos en (1.9) y el resultado final de la derivada respecto de los pesos  $wh_{j,k}$  será:

$$\frac{\partial E_p}{\partial wh_{j,k}} = \delta_{P,k} \cdot ho_{P,k} \quad (1.15)$$

Calcularemos ahora la derivada respecto de los pesos entre las unidades de la capa oculta y la de entrada. Para ello vemos que depende de las variables  $oo_{P,k}$ ,  $oi_{P,k}$  y  $ho_{P,j}$ . Tendremos:

$$\frac{\partial E_P}{\partial wi_{i,j}} = \sum_k \frac{\partial E_P}{\partial oo_{P,k}} \cdot \frac{\partial oo_{P,k}}{\partial oi_{P,k}} \cdot \frac{\partial oi_{P,k}}{\partial wi_{i,j}} \quad (1.16)$$

A su vez:

$$\frac{\partial oi_{P,k}}{\partial wi_{i,j}} = \frac{\partial oi_{P,k}}{\partial ho_{P,k}} \cdot \frac{\partial ho_{P,k}}{\partial hi_{P,k}} \cdot \frac{\partial hi_{P,k}}{\partial wi_{i,j}} \quad (1.17)$$

Sustituyendo la ecuación anterior en la derivada que queremos calcular se obtiene la siguiente ecuación.

$$\frac{\partial E_P}{\partial wi_{i,j}} = \sum_k \left\{ \frac{\partial E_P}{\partial oo_{P,k}} \cdot \frac{\partial oo_{P,k}}{\partial oi_{P,k}} \cdot \frac{\partial oi_{P,k}}{\partial ho_{P,k}} \cdot \frac{\partial ho_{P,k}}{\partial hi_{P,k}} \cdot \frac{\partial hi_{P,k}}{\partial wi_{i,j}} \right\} \quad (1.18)$$

Solo queda calcular la última derivada que será:

$$\frac{\partial hi_{P,k}}{\partial wi_{i,j}} = i_{P,k} \quad (1.19)$$

Sustituyendo las derivadas en la ecuación (1.18) el valor final será:

$$\frac{\partial E_P}{\partial wi_{i,j}} = \sum_k \left\{ -(t_{P,k} - oo_{P,k}) \cdot oo_{P,k} \cdot (1 - oo_{P,k}) \cdot wh_{j,k} \cdot ho_{P,j} \cdot (1 - ho_{P,j}) \cdot i_{P,i} \right\} \quad (1.20)$$

Definimos la siguiente variable:

$$\delta h_{P,j} = \left( \sum_k \delta_{P,k} \cdot wh_{j,k} \right) \cdot ho_{P,j} \cdot (1 - ho_{P,j}) \quad (1.21)$$

Sustituyendo se obtiene finalmente la derivada respecto de los pesos  $wi_{i,j}$  :

$$\frac{\partial E_P}{\partial wi_{i,j}} = \delta h_{P,j} \cdot i_{P,j} \quad (1.22)$$

## 1.5. Descripción de las Bases de Datos

En las investigaciones que se han llevado a cabo en esta Tesis se han utilizado problemas obtenidos del “UCI repository of machine learning databases” (<http://www.ics.uci.edu/~mllearn/MLRepository.html>) que están a disposición pública.

Se trata en general de problemas reales de clasificación. A continuación damos una descripción detallada de dichos problemas. Para poder utilizarlos en nuestros experimentos se han transformado los datos de los mismos convirtiéndolos en datos numéricos normalizados en el intervalo [0,1].

En cada capítulo explicaremos con más detalle qué bases de datos o problemas hemos utilizado y el tipo de transformación sobre los datos que ha sido necesario aplicar.

Veamos la descripción de las bases de datos escogidas.

### **Abalone Data:**

El problema consiste en predecir la edad de un “abalone”. Bajo dicho nombre se agrupa a un conjunto de moluscos gastrópodos marinos comestibles<sup>1</sup>.

Se debe determinar si la edad está entre 2’5 y 9’5 años, entre 10’5 y 11’5, o es 12’5 o más años. Hay pues tres clases.

Para ello se utilizan ocho entradas, como son su sexo, la longitud y el diámetro de la concha, el peso de la concha, el peso de la parte comestible, el peso total, etc. Los datos son reales y fueron obtenidos por los Laboratorios de Investigación de la Marina de Tasmania, en Australia.

El número de datos disponibles es de 4177. Hemos usado 3300 en el entrenamiento y 877 en el test.

### **Balance Scale Weight & Distance Database:**

Se trata de predecir si una balanza estará equilibrada, inclinada a la izquierda o a la derecha, por lo tanto es un problema con tres clases.

Las cuatro entradas que se utilizan para resolver el problema son: los dos pesos situados a la derecha e izquierda de la balanza y las dos distancias de dichos pesos al centro de la balanza.

---

<sup>1</sup> En Castellano "abulón", caracol marino de California de concha grande, gruesa, auriculada y muy nacarada.



Los datos disponibles son 625. Hemos usado 500 ejemplos para el entrenamiento y 125 para el test.

### **Cylinder Bands:**

En este problema se trata de predecir si se producirán retardos en el proceso conocido en inglés como “cylinder bands”, en una máquina de impresión de rotograbado.

La predicción se realiza en función de 39 parámetros distintos que intentan describir las condiciones de operación de la máquina. Los datos son reales y el problema tiene pues 39 atributos de entrada numéricos.

Se utilizarán dos clases, una para indicar que hay retardos en el proceso y la otra para indicar que no los hay.

Aunque la base de datos dispone de 512 ejemplos gran parte de ellos tienen atributos desconocidos, por lo que al final nos quedamos con solo 277 ejemplos. De ellos utilizaremos 190 para el entrenamiento y 87 para el test.

### **BUPA Liver Disorders:**

El problema es determinar si un paciente tiene algún problema de hígado o no. Las entradas son los resultados de cinco tests realizados en la sangre y el consumo medio de alcohol por día. Los datos son reales y fueron obtenidos por la compañía BUPA Medical Research Ltd.

Tenemos seis atributos o entradas numéricas y dos clases: si hay o no problemas en el hígado.

Los datos disponibles son 345. Hemos usado 245 ejemplos para el entrenamiento y 100 ejemplos para el test.

### **Credit Approval:**

El problema consiste en determinar si una petición de una tarjeta de crédito debe ser concedida o no. Se trata de un problema de clasificación de dos clases.

El número de entradas es 15 y su significado se desconoce, puesto que se trata de datos reales en los que se ha pretendido garantizar la confidencialidad.

La base de datos consta de 690 ejemplos pero 37 de ellos tienen entradas desconocidas o perdidas. Hemos usado 653 ejemplos, 453 para el entrenamiento y 200 para el test.

## 1.5. DESCRIPCIÓN DE LAS BASES DE DATOS

### **Display1:**

El problema es determinar el número que está representando en un display led de 7 segmentos, sabiendo si cada uno de los leds está encendido o apagado.

Se trata de un problema de clasificación con diez clases (cada uno de los diez dígitos) y siete entradas binarias, una por cada led. Para complicar el problema se han seleccionado aleatoriamente un 10% de los valores de las entradas y se ha invertido su valor binario.

La base de datos generada consta de 2900 ejemplos, 900 para el entrenamiento y 2000 para el test.

### **Display2:**

Es exactamente el mismo problema que Display1, solo que se han añadido un total de 7 entradas binarias adicionales cuyos valores son totalmente aleatorios y por lo tanto absolutamente irrelevantes. En total tendremos 14 entradas.

La base de datos generada consta de 2900 ejemplos, 900 para el entrenamiento y 2000 para el test.

### **Glass Identification Database:**

El problema consiste en clasificar trozos de vidrio como pertenecientes a una de las seis clases siguientes: artículos de mesa, recipientes, faros, ventanas de edificios, ventanas de edificios tipo “float processed” y ventanas de vehículos tipo “float processed” (no existen ventanas de vehículos que no sean tipo “float processed” por lo que no se considera como una clase). Parece ser que esta clasificación tiene mucho interés en ámbitos de investigación criminológica.

Los datos son reales y para realizar la clasificación se utiliza diez entradas, entre ellas el valor del índice de refracción y la cantidad porcentual de diversos elementos en su composición, Na, Mg, Al, Si, K, Ca, Ba, Fe.

El problema consta de 214 ejemplos, hemos usado 150 ejemplos para el entrenamiento y 64 para el test.

### **Heart Disease Database:**

El problema consiste en determinar si un paciente tiene alguna enfermedad de corazón, será pues un problema de dos clases. Para predecirlo, se utilizan 13 entradas de índole diversa, desde algún parámetro del ECG, hasta el hecho de que el paciente sea o no fumador.

Los datos son reales y fueron recopilados por varias clínicas especializadas internacionales.

El número de ejemplos total de la base de datos es 920 pero entre ellos hay muchos que tienen entradas desconocidas, por lo que finalmente usamos 297, 210 como ejemplos para el entrenamiento y 87 para el test.

#### **Image Segmentation Data:**

Los ejemplos de esta base de datos fueron obtenidos aleatoriamente de un conjunto de imágenes exteriores. Las imágenes fueron segmentadas manualmente para obtener la clasificación de cada ejemplo. Un ejemplo representa una región de 3x3 puntos de la imagen. El objetivo es determinar a cuál de las siete siguientes clases pertenece el ejemplo: césped, ventana, cemento, follaje, nieve, ladrillo o camino, es decir siete clases.

Para la clasificación se utilizan un total de 19 entradas que representan características del conjunto de puntos que forman el ejemplo. Los datos son reales.

Utilizamos un total de 2311 ejemplos, 1500 para el entrenamiento y 811 para el test.

#### **Mushroom Database:**

El problema consiste en clasificar una seta como comestible o venenosa (dos clases).

Para ello se utilizan un total de 22 características generales de la seta y su hábitat, siendo los datos reales.

Se dispone en la base de datos de 8124 ejemplos, de los cuales 2480 tienen entradas desconocidas o perdidas. Nos quedamos finalmente con 5644 ejemplos y se usaron 4144 para el entrenamiento y 1500 para el test.

#### **The Monk's Problems 1, 2 y 3:**

Se trata de tres problemas artificiales con entradas binarias. No obstante, tienen interés porque fueron utilizados para realizar la primera comparación internacional entre algoritmos de aprendizaje de máquinas. En los problemas hay dos clases y seis entradas.

Se dispone de 432 ejemplos, 332 para el entrenamiento y 100 para el test.

## *1.5. DESCRIPCIÓN DE LAS BASES DE DATOS*

### **Musk “Clean2” Database:**

El problema consiste en clasificar si una molécula sintetizada artificialmente será de tipo “musk” o no, es decir, tiene dos clases. Como “musk” se conoce a una secreción glandular de fuerte olor del venado macho que es usada en perfumería, también a cualquier otra sustancia que tenga un olor similar. Dado que es de gran utilidad en perfumería, se trata de sintetizar dicho olor artificialmente.

Como entradas se utiliza un total de 166 características de la molécula que se va a sintetizar. Los datos son reales y fueron obtenidos por la corporación farmacéutica Arris de USA.

Se dispone de 6598 ejemplos, hemos utilizado 5098 para entrenamiento y 1500 para test.

### **Pima Indians Diabetes Database:**

El problema consiste en diagnosticar si un paciente tiene o no diabetes. Se trata pues de un problema con dos clases. En concreto lo que se pretende es distinguir si el paciente muestra o no signos de diabetes de acuerdo con los criterios de la Organización Mundial de la Salud. Para el diagnóstico se usan un total de ocho entradas y los datos son reales.

El problema consta de 768 ejemplos, hemos empleado 518 para entrenamiento y 250 para test.

### **Thyroid Domain:**

El problema consiste en clasificar si el funcionamiento de la glándula tiroides del paciente es normal, tiene una hiperfunción o una hipofunción, tendremos tres clases.

El número de entradas que se utilizan son 21 (15 atributos binarios y 6 continuos) y los datos son reales.

Se tiene un total de 7200 ejemplos, de los cuales se utilizan 5700 para el entrenamiento y 1500 para el test.

### **Voting Records:**

El problema es determinar si una persona es republicana o demócrata a partir de las contestaciones a un total de 16 preguntas. Los datos son reales y fueron recogidos de las votaciones del congreso de los Estados Unidos, USA, en el año 1987.

El problema consta de dieciseis entradas y dos clases. Se dispone de 435 ejemplos, hemos utilizado 335 para el entrenamiento y 100 para el test.

**Wisconsin Diagnostic Breast Cancer:**

El problema consiste en determinar si un tumor de mama es maligno o benigno, tendremos dos clases.

La determinación se hace con 32 entradas que describen las características de los núcleos de las células encontradas en una biopsia. Los datos son reales y fueron recogidos en la Universidad de Wisconsin.

Se tienen un total de 569 ejemplos y usaremos 409 para el entrenamiento y 160 para el test.

## 1.6. Herramientas de trabajo: PC, SUN, Matlab

La investigación sobre Redes Neuronales implica una gran carga computacional en máquinas (PC's u otras). Es necesario ejecutar simulaciones de los diferentes algoritmos y métodos estudiados, entrenando también las redes neuronales que se quieran finalmente utilizar.

A lo largo de nuestro trabajo hemos necesitado un número elevado de procesadores en donde poder realizar dicha tarea.

Gracias al material disponible ya en la Universidad Jaume I y al apoyo de algunos proyectos como: *PIA97-19* de la Fundació Caixa-Castelló y la Universidad Jaume I y *GV99-75-I-14* de la Generalitat Valenciana, hemos dispuesto del material necesario para realizar nuestra investigación.

La primera necesidad fue la de máquinas con el mayor número de procesadores posible, donde ejecutar nuestros programas.

La siguiente necesidad fue la de un lenguaje de programación que posibilitará un desarrollo de programas muy rápido y con códigos fuentes sencillos, de forma que se redujera considerablemente la posibilidad de errores en la programación. Respecto a esta última necesidad optamos por la utilización de MATLAB. Toda la programación realizada en esta Tesis se ha realizado con MATLAB.

Respecto a la necesidad de máquinas, durante el primer año de trabajo se dispuso tan sólo de una estación de trabajo SUN UltraSPARC. Para dicha máquina compramos las licencias para *MATLAB*, *Compilador de C* (Matlab C Compiler) y *biblioteca matemática de C de Matlab* (Matlab Math C Library). Con dicho software teníamos resultado el problema de la programación.

Posteriormente, se llevó a cabo la compra de PC's con dos procesadores cada uno y de licencias de *MATLAB* para los mismos. Con ello el trabajo se fue agilizando bastante, cada procesador (de cada PC) realizaba una tarea concreta de forma mucho más rápida y el reparto de trabajo fue mucho más productivo.

En la actualidad el "Grupo de Redes Neuronales y Computación Suave" de la Universidad Jaume I, dispone de una estación de trabajo SUN UltraSPARC y de 15 PC's (con dos procesadores cada uno) con diferentes velocidades, ya que su compra se realizó en diferentes años.

Otro aspecto importante en la investigación sobre Redes Neuronales es el almacenamiento de los resultados que se obtienen.

Los resultados de las simulaciones realizadas en Redes Neuronales suelen ser ficheros de gran tamaño, por ello a lo largo de la realización de la investigación nos hemos encontrado con el problema de tener que almacenar toda la información que generábamos. Para ello, hemos dispuesto de una unidad de discos magnetoópticos removibles Sony de 2,6 Gb, de una unidad de cinta DAT DDS-3 de HP de capacidad 12GB, CD-ROM (aquí grabamos datos de especial importancia que ya han sido analizados para poder disponer de ellos con rapidez) y discos ZIP de 100Mb (en ellos solemos guardar programas, artículos, etc.).

Actualmente también disponemos de una unidad RAID-5 con capacidad de 50Gb.

## 1.7. Bibliografía del Capítulo

- [1] Merz, C. J., Murphy, P. M., (1998). UCI repository of machine learning databases. Available at: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- [2] Prechelt, L., "A quantitative study of experimental evaluations of neural network learning algorithms: current research practice". *Neural Networks*, vol. 9, no. 3, pp. 457-462, 1996.
- [3] Rumelhart, D. E., McClelland, J. L., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol. I, II y III. Cambridge, MA: MIT Press, 1986, 1987.
- [4] Maren, A., Harston, C., Pap, R., *Handbook of Neural Computing Applications*. Academic Press, Inc. 1990.
- [5] Hecht-Nielsen, R., *Neurocomputing*, Addison Wesley, 1991.
- [6] Haykin, S., *Neural Networks. A comprehensive Foundation*. IEEE Press. Macmillan College Publishing Company, Inc. 1994.
- [7] Yoh-Han Pao, *Adaptative Pattern Recognition and Neural Networks*. Addison-Wesley Publishing Company, Inc. 1989.
- [8] Devroye, L., Györfi, L., Lugosi, G., *A Probabilistics Theory of Pattern Recognition*. Springer-Verlag New York, Inc. 1996.
- [9] Schalkoff, R. J., *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons, Inc. 1992.
- [10] Golden, R. M., *Mathematical methods for Neural Network Analysis and Design*. Cambridge, MA: MIT Press. 1996.
- [11] Deco, G., Obradovic, D., *An information Theoretic Approach to Neural Computing*. Springer-Verlag New York, Inc. 1996.
- [12] Reed, R. D., Marks, R. J., *Neural Smithing. Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA: MIT Press. 1999.
- [13] Fine, T.L., *Feedforward Neural Network Methodology*. Springer-Verlag New York, Inc. 1999.
- [14] Müller, B., Reinhardt, J., Strickland, M.T., *Neural Networks. An Introduction*. Springer-Verlag Berlin Heidelberg. 1<sup>st</sup> Ed. 1990-2<sup>nd</sup> Ed. 1995.



# Capítulo 2

## PROCEDIMIENTOS PARA EL USO DE ENTRADAS DESCONOCIDAS

- 2.1. *Introducción*
- 2.2. *Procedimientos de utilización de entradas desconocidas*
  - 2.2.1. *Codificación mediante un valor*
  - 2.2.2. *Codificación utilizando Aritmética de Intervalos*
    - 2.2.2.1 *Operaciones con aritmética de intervalos*
    - 2.2.2.2 *Multilayer Feedforward con aritmética de intervalos*
    - 2.2.2.3 *Backpropagation con aritmética de intervalos*
      - 2.2.2.3.1 *Algoritmo de Hernández*
      - 2.2.2.3.2 *Algoritmo de Ishibuchi*
      - 2.2.2.3.3 *Medidas de eficiencia*
  - 2.2.3. *Codificación mediante Probabilidades*
  - 2.2.4. *Codificación de entradas binarias, que no utiliza entradas desconocidas en el entrenamiento*
- 2.3. *Metodología*
  - 2.3.1. *Elección de bases de datos*
  - 2.3.2. *Elección de la arquitectura*
  - 2.3.3. *Proceso de entrenamiento y test*
- 2.4. *Resultados experimentales*
- 2.5. *Conclusiones*
- 2.6. *Bibliografía del Capítulo*

## 2.1. Introducción

El objetivo de este capítulo es realizar, para la arquitectura de Red Neuronal Multilayer Feedforward, un estudio comparativo entre los procedimientos que existen para utilizar entradas desconocidas.

Veamos en primer lugar, cuál es el significado de una entrada desconocida.

Supongamos que se quiere resolver un cierto problema utilizando redes neuronales. Lo primero que necesitamos conocer es qué información queremos que procese la red (información de entrada en las neuronas de la red). En general la información de entrada a la red (ejemplos) suele estar bien determinada. Sin embargo, puede ocurrir que tras el entrenamiento, nos encontremos algún caso (ejemplo de entrada) en el cual no toda la información de entrada esté definida, es decir, la entrada a una o más neuronas de la red no se conozca, y, aún así, queramos obtener cuál será la clasificación más probable que proporcione la red con la información de que disponemos.

Este problema es un caso muy típico en aplicaciones de diagnóstico médico, en donde uno o varios de los síntomas o pruebas en un determinado paciente (entradas a ciertas neuronas de entrada de la red) pueden no conocerse. En [1] se describen algunos ejemplos.

Como se describe en la referencia [1], es de esperar que la red neuronal sea capaz de proporcionar algún diagnóstico para ese paciente (clasificación de ejemplos con entradas desconocidas), a pesar de que la información de entrada sea incompleta. El problema es cómo obtener la clasificación de un ejemplo cuya información es incompleta, es decir, cómo codificar estas entradas desconocidas o cómo decirle a la red neuronal que una cierta entrada se desconoce. La solución más simple podría ser codificar las entradas desconocidas con un valor, pero ¿cuál?

Existen diferentes métodos en la bibliografía para abordar el problema de codificación de las entradas desconocidas y no se conoce cuál es el más eficiente. El objetivo de este capítulo es realizar una comparación experimental entre dichos métodos para ver cuál de ellos resuelve mejor el problema.

Para algunos de los métodos los autores proponen incluir entradas desconocidas dentro de los ejemplos del conjunto de entrenamiento. Así, razonan que no será posible que la red responda adecuadamente ante entradas desconocidas si no se han usado antes en el entrenamiento. Sin embargo, la inclusión de entradas desconocidas en el conjunto de entrenamiento puede generar diversos problemas.

En primer lugar, el tamaño del conjunto de entrenamiento crece considerablemente, por lo que el esfuerzo computacional para desarrollar una aplicación será mayor.

En segundo lugar, la eficiencia de la red puede verse afectada. Si fijamos un valor determinado para la entrada desconocida de un ejemplo el cual está asociado a un cierto objetivo (clase de clasificación correcta para el ejemplo), estamos forzando a la red a que nos dé como resultado una clase concreta para ese valor de la entrada desconocida. Esto puede suponer una pérdida de eficiencia en la red cuando no se usen entradas desconocidas.

Nosotros pensamos que es interesante medir de qué manera puede afectar la introducción de entradas desconocidas tanto en el entrenamiento como en el test. Para ello, en nuestros experimentos, a partir de varios problemas de clasificación generaremos otros nuevos introduciendo en los problemas iniciales diferentes porcentajes de entradas desconocidas en los conjuntos de datos para el entrenamiento y para el test.

Para medir la eficiencia de los métodos de codificación de entradas desconocidas con los nuevos problemas, se entrenarán y testearán diferentes redes y al final podremos ver en que medida influye el uso de entradas desconocidas en el entrenamiento, para mejorar o no la eficiencia de la red (capacidad de generalización).

Es el siguiente apartado se describen detalladamente los métodos sobre codificación de entradas desconocidas que vamos a estudiar.

## 2.2. Procedimientos de utilización de entradas desconocidas

Se han propuesto diversos métodos para abordar el problema de la codificación de entradas desconocidas para la red Multilayer Feedforward. Algunos de ellos utilizan entradas desconocidas en el proceso de entrenamiento de la red y otros no. A continuación describimos los métodos que consideramos más representativos y que son los que se usarán finalmente para realizar la comparación.

### 2.2.1. Codificación mediante un valor

En la bibliografía podemos ver la descripción de este método en [1].

La codificación para entradas binarias se suele tomar, “sí” o “presente” como 1 y “no” o “ausente” como 0. Según este método una entrada desconocida se codificará con el valor 0.5.

En el caso de entradas no binarias el problema se complica [1]. En los problemas que usaremos los valores de las entradas están normalizadas en el intervalo [0,1]. Para este caso, lo que se propone en este método es sustituir las entradas desconocidas por el valor 0 y desplazar el intervalo de los valores conocidos a [0.25, 1].

Este método requiere que se genere un conjunto de entrenamiento con entradas desconocidas y por lo tanto que se incluyan entradas desconocidas en el proceso de aprendizaje.

En [18], [19], [20] y [21], se encuentran algunos ejemplos de utilización de este método de codificación de entradas desconocidas.

### 2.2.2. Codificación utilizando Aritmética de Intervalos

Con este método, se propone un concepto diferente para tratar las entradas desconocidas. En primer lugar, se hace una Generalización de la red Multilayer Feedforward y el algoritmo de aprendizaje Backpropagation a Aritmética de Intervalos. Esta generalización permitirá mezclar en un vector de entrada, entradas representadas por un intervalo y entradas puntuales dadas por un valor. Así las entradas desconocidas se codifican usando el intervalo [0,1] si el conjunto de

posibles valores de la entrada está contenido en  $[0,1]$  y las entradas conocidas pueden ser representadas por su valor.

El método además requiere que se realice un aprendizaje con un conjunto de entrenamiento que contenga entradas desconocidas.

En la bibliografía, podemos encontrar dos generalizaciones de Multilayer Feedforward y Backpropagation a Aritmética de Intervalos [3], [4] y [5], [6]. En ambas, la extensión de Multilayer Feedforward a Aritmética de Intervalos es la misma pero el algoritmo de aprendizaje Backpropagation que se propone difiere en la función de error utilizada, más adelante describiremos ambas con detalle.

Veamos en que consiste la generalización de la arquitectura Multilayer Feedforward a Aritmética de Intervalos.

En el capítulo primero, se describió brevemente la arquitectura Multilayer Feedforward y el algoritmo de aprendizaje Backpropagation. Partiendo de ella surge una primera aproximación a la utilización de aritmética de intervalos tal y como se describe en la referencia [2]. La introducción de aritmética de intervalos en la red permite que se empleen vectores puntuales e intervalos vectoriales como entradas en los ejemplos para el entrenamiento de la red neuronal.

Se entiende por vector puntual:  $\{a_1, a_2, \dots, a_N\}$ ,  $a_i \in \mathfrak{R}$

Un intervalo vectorial se definirá como:

$$\{[a_1^I, a_1^S], [a_2^I, a_2^S], \dots, [a_N^I, a_N^S]\}, a_i^I, a_i^S \in \mathfrak{R} \text{ y } a_i^I \leq a_i^S \quad \forall i$$

Donde los superíndices  $I$  y  $S$  denotan los extremos inferior y superior del intervalo respectivamente.

Tenemos así un vector compuesto por intervalos como entradas a la red (ejemplos de entrada). Si en algún intervalo el extremo inferior y superior coinciden, tendremos como ejemplo de entrada a la red un vector que será una mezcla de puntos e intervalos.

La red propuesta en [2] es capaz utilizar dichos vectores como ejemplos de entrada a la red, tanto en el entrenamiento como después.

Sin embargo, la red descrita en [2] era bastante limitada, solo poseía una unidad de salida, con lo que solo servía para resolver problemas de clasificación con dos clases. Además, los objetivos de la salida de la red eran valores reales, por lo que no existía la posibilidad de utilizar también intervalos como objetivos del entrenamiento de la misma.

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

Por ello, surge una nueva propuesta descrita en las referencias [3] y [4]. Se trata de introducir la aritmética de intervalos, tanto en la estructura Multilayer Feedforward, como en el algoritmo de aprendizaje Backpropagation de forma más eficiente y general que la propuesta en [2].

En dichas referencias, se propone una forma de permitir la introducción de intervalos vectoriales tanto en la capa de entrada (ejemplos de entrada), como en la de salida (objetivos), por lo que la propagación de la señal de una capa a otra se hará también con intervalos vectoriales.

El algoritmo de aprendizaje deberá de adaptarse a esta estructura, de forma que se salvarán los inconvenientes que presentaba el método descrito en [2]. El algoritmo final será una verdadera generalización de Backpropagation a aritmética de intervalos, ya que si utilizamos vectores de entrada y salida puntuales todas las ecuaciones se reducirán a las clásicas utilizadas en Multilayer Feedforward con Backpropagation.

### 2.2.2.1. Operaciones con Aritmética de intervalos

Primero veremos algunas operaciones que pueden realizarse con aritmética de intervalos y que nos serán de utilidad. En la referencia [7] puede encontrarse una exposición más detallada.

Sean dos intervalos cerrados  $A$  y  $B$  en los cuales denotaremos con superíndices  $I$  y  $S$  a los extremos inferior y superior respectivamente.

$$A = [a^I, a^S] \quad B = [b^I, b^S]$$

Definiremos las siguientes operaciones:

**Suma de intervalos:**

$$A + B = [a^I, a^S] + [b^I, b^S] = [a^I + b^I, a^S + b^S] \quad (2.1)$$

**Producto de un número real positivo o negativo por un intervalo:**

$$m \cdot A = m \cdot [a^I, a^S] = \begin{cases} [m \cdot a^I, m \cdot a^S], & \text{si } m \geq 0 \\ [m \cdot a^S, m \cdot a^I], & \text{si } m < 0 \end{cases} \quad (2.2)$$

**Función exponencial de un intervalo:**

$$\exp(A) = \exp([a^I, a^S]) = [\exp(a^I), \exp(a^S)] \quad (2.3)$$

**Derivada parcial respecto a variables de los límites de los intervalos:**

$$\frac{\partial}{\partial x} ([a \cdot x^2, c \cdot x^2 + d \cdot y]) = \begin{cases} [2 \cdot a \cdot x, 2 \cdot c \cdot x + d \cdot y], & \text{si } 2ax \leq 2cx + dy \\ [2 \cdot c \cdot x + d \cdot y, 2 \cdot a \cdot x], & \text{si } 2cx + dy \leq 2ax \end{cases} \quad (2.4)$$

### 2.2.2.2. Multilayer Feedforward con aritmética de intervalos

En el primer capítulo describimos con detalle la arquitectura de la red Multilayer Feedforward.

Ahora vamos a ver cuál es la generalización de Multilayer Feedforward que se propone en las referencias [3] y [4].

Partimos de la función de transferencia o de activación utilizada en Multilayer Feedforward para una neurona  $x$ :

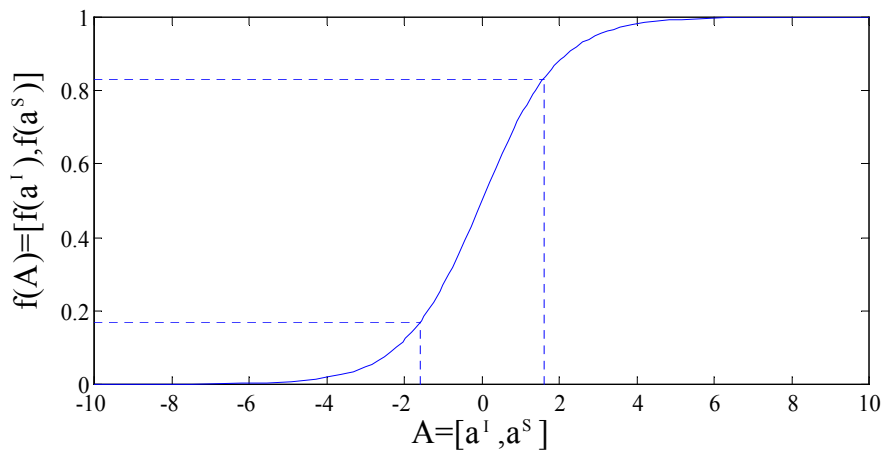
$$f(x) = \frac{1}{1 + \exp(-x)} \quad (2.5)$$

La generalización de la misma a aritmética de intervalos será:

$$f(A) = f([a^I, a^S]) = [f(a^I), f(a^S)] = \left[ \frac{1}{1 + \exp(-a^I)}, \frac{1}{1 + \exp(-a^S)} \right] \quad (2.6)$$

Esta definición es consistente ya que se trata de una función monótona creciente, por lo cual, la nueva función de transferencia transformará un intervalo  $A=[a^I, a^S]$  a la entrada de la neurona en otro intervalo  $f(A)=[f(a^I), f(a^S)]$  a la salida de la misma.

En la figura 1 podemos ver la representación de la nueva función de transferencia.



**Fig.1. Función de transferencia para aritmética de intervalos.**

Veamos ahora como se transforman el resto de parámetros de la estructura de la red neuronal.

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

Para Multilayer Feedforward teníamos que los vectores de salida de las diferentes capas son:

$$\text{Capa de Entrada: } \{io_1, io_2, \dots, io_N\} = \{a_1, a_2, \dots, a_N\} \quad (2.7)$$

$$\text{Capa Oculta: } \{ho_1, ho_2, \dots, ho_{Nh}\} = \{f(hi_1), f(hi_2), \dots, f(hi_{Nh})\} \quad (2.8)$$

$$\text{Capa de Salida: } \{oo_1, oo_2, \dots, oo_{Ns}\} = \{f(oi_1), f(oi_2), \dots, f(oi_{Ns})\} \quad (2.9)$$

Las ecuaciones para la red eran:

$$hi_j = \sum_{i=1}^N wi_{ij} \cdot a_i + \theta_j \quad (2.10)$$

$$oi_k = \sum_{j=1}^{Nh} wh_{jk} \cdot ho_j + \xi_k \quad (2.11)$$

$$ho_j = \frac{1}{1 + \exp(-hi_j)} \quad (2.12)$$

$$oo_k = \frac{1}{1 + \exp(-oi_k)} \quad (2.13)$$

Veamos como se transforman estas ecuaciones mediante el uso de Aritmética de Intervalos.

Ahora tendremos vectores formado por intervalos, denotaremos por  $P$  a un patrón de entrada concreto, por  $I$  al límite inferior de un intervalo y por  $S$  al límite superior de un intervalo.

Sea  $[i_{P,i}^I, i_{P,i}^S]$  el intervalo que representa a la coordenada  $i$ -ésima del vector de entrada a la red del patrón  $P$  (neurona de entrada  $i$ ). La dimensión de dicho vector es  $N$  y puede escribirse:

$$\{[i_{P,1}^I, i_{P,1}^S], [i_{P,2}^I, i_{P,2}^S], \dots, [i_{P,N}^I, i_{P,N}^S]\} \quad (2.14)$$

Igual que en la estructura Multilayer Feedforward, los pesos entre las capas de la estructura de la red neuronal que relacionan la salida de una capa con la entrada de la siguiente tendrán valores reales.

La entrada a una neurona que no pertenezca a la capa de entrada será la suma de las contribuciones (salidas ponderadas por los pesos) de las neuronas de la capa anterior. Estas operaciones siguen siendo iguales. La diferencia radica en que ahora se multiplicarán los pesos (números reales) entre dos neuronas por el intervalo que representa la salida de la neurona de la capa anterior, por lo que aparecerán también multiplicaciones y sumas de intervalos vectoriales.



2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

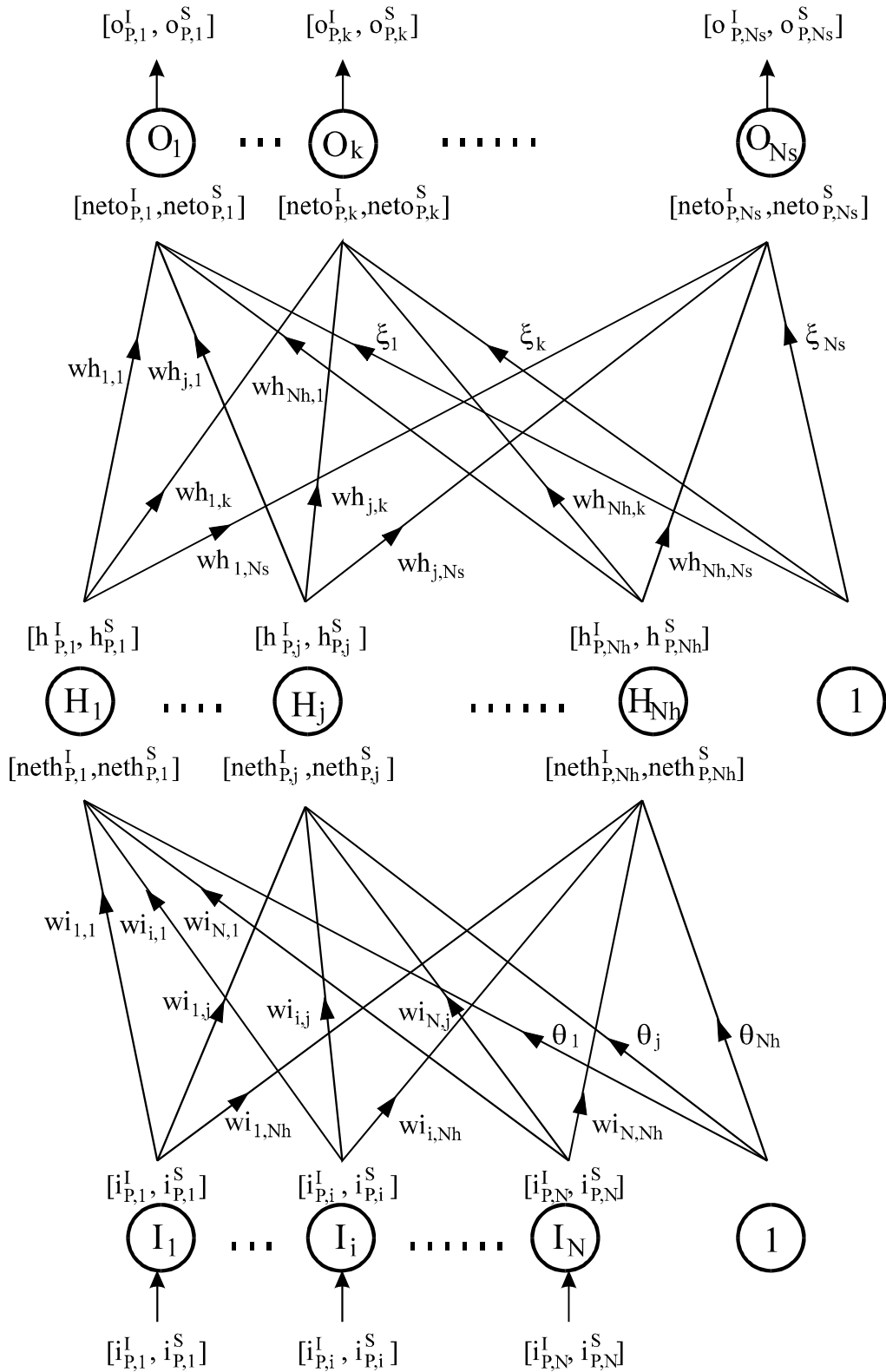


Fig.2. Estructura Multilayer Feedforward con aritmética de intervalos.

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

Utilizaremos la estructura de la red con una sola capa oculta. El interés o no de usar más de una capa oculta en la red se tratará en el capítulo 5.

En la figura 2 puede verse la generalización de la estructura Multilayer Feedforward a aritmética de intervalos que hemos utilizado. Llamaremos:

*Salida de la neurona i de la capa de entrada:*

$$I_{P,i} = [i_{P,i}^I, i_{P,i}^S] \quad (2.15)$$

*Salida de la neurona j de la capa oculta:*

$$H_{P,j} = [h_{P,j}^I, h_{P,j}^S] = f(Neth_{P,j}) = [f(neth_{P,j}^I), f(neth_{P,j}^S)] \quad (2.16)$$

*Salida de la neurona k de la capa de salida:*

$$O_{P,k} = [o_{P,k}^I, o_{P,k}^S] = f(Neto_{P,k}) = [f(neto_{P,k}^I), f(neto_{P,k}^S)] \quad (2.17)$$

Llamamos  $\theta$  y  $\xi$  a los vectores de “threshold” o umbrales que intervienen en la capa oculta y en la capa de salida respectivamente. Estos se consiguen introduciendo una neurona de salida constante con valor 1 en la capa inferior a la que se este estudiando.

La entrada a la unidad  $j$  de la capa oculta será:

$$Neth_{P,j} = [neth_{P,j}^I, neth_{P,j}^S] = \sum_{i=1}^N wi_{i,j} \cdot I_{P,i} + \theta_j \quad (2.18)$$

Donde:

$$neth_{P,j}^I = \sum_{\substack{i=1 \\ wi_{i,j}>0}}^N wi_{i,j} \cdot i_{P,i}^I + \sum_{\substack{i=1 \\ wi_{i,j}<0}}^N wi_{i,j} \cdot i_{P,i}^S + \theta_j \quad (2.19)$$

$$neth_{P,j}^S = \sum_{\substack{i=1 \\ wi_{i,j}>0}}^N wi_{i,j} \cdot i_{P,i}^S + \sum_{\substack{i=1 \\ wi_{i,j}<0}}^N wi_{i,j} \cdot i_{P,i}^I + \theta_j \quad (2.20)$$

En las ecuaciones anteriores  $wi_{i,j}$  son los pesos que conectan la neurona  $j$  de la capa oculta con la neurona  $i$  de la capa de entrada. La aparición en las ecuaciones de dos sumatorios se debe a la propiedad que vimos antes sobre el producto de un intervalo por un número real, ecuación (2.2).

La entrada a la unidad  $k$  de la capa de salida será:

$$Neto_{P,k} = [neto_{P,k}^I, neto_{P,k}^S] = \sum_{j=1}^{N_h} wh_{j,k} \cdot H_{P,j} + \xi_k \quad (2.21)$$

Donde:

$$neto_{P,k}^I = \sum_{\substack{j=1 \\ wh_{j,k} > 0}}^{N_h} wh_{j,k} \cdot h_{P,j}^I + \sum_{\substack{j=1 \\ wh_{j,k} < 0}}^{N_h} wh_{j,k} \cdot h_{P,j}^S + \xi_k \quad (2.22)$$

$$neto_{P,k}^S = \sum_{\substack{j=1 \\ wh_{j,k} > 0}}^{N_h} wh_{j,k} \cdot h_{P,j}^S + \sum_{\substack{j=1 \\ wh_{j,k} < 0}}^{N_h} wh_{j,k} \cdot h_{P,j}^I + \xi_k \quad (2.23)$$

En las ecuaciones anteriores  $wh_{j,k}$  son los pesos que conectan la neurona  $k$  de la capa de salida con la neurona  $j$  de la capa oculta.

Cabe destacar que si en las ecuaciones anteriores nos restringimos a vectores puntuales, es decir, las variables etiquetadas con los índices  $I$  y  $S$  representan el mismo valor, obtendremos exactamente las mismas ecuaciones que vimos en el capítulo primero para Multilayer Feedforward.

### 2.2.2.3. Backpropagation con aritmética de intervalos

Una vez definida la estructura de la red será necesario un entrenamiento de la misma antes de poder usarla. Para ello, debemos disponer de un conjunto suficiente de ejemplos de entrenamiento con sus correspondientes objetivos. Los ejemplos (patrones) y objetivos como vimos en la sección anterior vendrán representados por vectores de intervalos.

Para un cierto patrón  $P$  tendré que el ejemplo y su objetivo podrán expresarse por las ecuaciones (2.24) y (2.25) respectivamente, siendo  $N$  el número de atributos de entrada y  $Ns$  el número de clases o unidades de salida.

$$I_P = \{ [i_{P,1}^I, i_{P,1}^S], [i_{P,2}^I, i_{P,2}^S], \dots, [i_{P,N}^I, i_{P,N}^S] \} \quad (2.24)$$

$$T_P = \{ [t_{P,1}^I, t_{P,1}^S], [t_{P,2}^I, t_{P,2}^S], \dots, [t_{P,Ns}^I, t_{P,Ns}^S] \} \quad (2.25)$$

Ahora tenemos que realizar la generalización del algoritmo de aprendizaje Backpropagation a aritmética de intervalos. Como vimos, dicho algoritmo consiste básicamente en ajustar los pesos de la red durante iteraciones consecutivas, a la vez que se minimiza una cierta función de error.

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

La generalización consiste, igual que el apartado anterior, en transformar las ecuaciones teniendo en cuenta que tanto las señales de las neuronas de entrada como las de salida vienen representadas por intervalos y que si estos fuesen puntuales deberíamos obtener las ecuaciones tradicionales de Backpropagation.

En la bibliografía se pueden encontrar dos generalizaciones diferentes de Backpropagation a aritmética de intervalos. La diferencia básica entre ellas es la función de error de partida.

En los siguientes apartados describiremos ambas generalizaciones y desarrollaremos sus ecuaciones.

### 2.2.2.3.1. Algoritmo de Hernández

En las referencias [3] y [4] Hernández describe una generalización a aritmética de intervalos desarrollada tanto para la red Multilayer Feedforward como para el algoritmo Backpropagation.

El primer paso para definir la generalización del algoritmo es definir una función de error para el uso de intervalos. Para ello, se fija como objetivo que tanto el límite inferior como superior de las unidades de salida coincidan con los límites de los intervalos que representan los objetivos.

Se define así el error cuadrático medio para el patrón  $P$  como:

$$E_P = \frac{1}{4} \cdot \sum_{k=1}^{N_S} \left\{ (t_{P,k}^I - o_{P,k}^I)^2 + (t_{P,k}^S - o_{P,k}^S)^2 \right\} \quad (2.26)$$

El proceso de aprendizaje consiste en minimizar la función de error para todos los ejemplos del conjunto de entrenamiento, los cuales se presentan a la red en sucesivas iteraciones hasta conseguir la convergencia de la función de error.

La minimización del error se hará respecto de los pesos de la red. Dado  $wh_{jk}$  uno de los pesos entre las neuronas  $j$  de una capa y  $k$  de la capa siguiente, el algoritmo consistirá en adaptar los pesos a medida que hacemos que converja la función de error. Después de cada iteración los pesos serán modificados según las siguientes ecuaciones:

$$\Delta wh_{j,k}(t+1) = \eta \cdot \left( -\frac{\partial E_P}{\partial wh_{j,k}} \right) + \beta \cdot \Delta wh_{j,k}(t) \quad (2.27)$$

$$\Delta wi_{i,j}(t+1) = \eta \cdot \left( -\frac{\partial E_P}{\partial wi_{i,j}} \right) + \beta \cdot \Delta wi_{i,j}(t) \quad (2.28)$$

Donde  $\eta$  es el paso de adaptación y  $\beta$  es el momento.

Las salidas de las neuronas están limitadas entre los valores 0 y 1 (positivos), por lo que al multiplicar por un peso según el signo de este puede variar el orden de los límites de los intervalos.

### Cálculo de las derivadas de la capa oculta

Calculamos las derivadas de los pesos entre la capa oculta y la de salida.

A partir de la ecuación (2.26) vemos la dependencia de  $E_p$  con  $o_{P,k}^I$  y  $o_{P,k}^S$  y la de estas (ecuación (2.17)) con las variables  $neto_{P,k}^I$   $neto_{P,k}^S$ .

Aplicando la regla de la cadena para derivar obtenemos:

$$\frac{\partial E_p}{\partial wh_{j,k}} = \frac{\partial E_p}{\partial o_{P,k}^I} \cdot \frac{\partial o_{P,k}^I}{\partial neto_{P,k}^I} \cdot \frac{\partial neto_{P,k}^I}{\partial wh_{j,k}} + \frac{\partial E_p}{\partial o_{P,k}^S} \cdot \frac{\partial o_{P,k}^S}{\partial neto_{P,k}^S} \cdot \frac{\partial neto_{P,k}^S}{\partial wh_{j,k}} \quad (2.29)$$

$$\frac{\partial E_p}{\partial wh_{j,k}} = \frac{\partial E_p^I}{\partial wh_{j,k}} + \frac{\partial E_p^S}{\partial wh_{j,k}} \quad (2.30)$$

A partir de la ecuación (2.26) obtenemos:

$$\frac{\partial E_p}{\partial o_{P,k}^I} = -\frac{1}{2} \cdot (t_{P,k}^I - o_{P,k}^I) \quad (2.31)$$

De la ecuación (2.5) podemos deducir la derivada de la función de transferencia utilizada en la red.

$$f'(x) = \frac{\exp(-x)}{(1 + \exp(-x))^2} = f(x) \cdot (1 - f(x)) \quad (2.32)$$

De las ecuaciones (2.17) y (2.32) se puede deducir que:

$$\frac{\partial o_{P,k}^I}{\partial neto_{P,k}^I} = o_{P,k}^I \cdot (1 - o_{P,k}^I) \quad (2.33)$$

La última derivada va a depender del signo del peso. A partir de la ecuación (2.22) se obtiene:

$$\frac{\partial neto_{P,k}^I}{\partial wh_{j,k}} = \begin{cases} h_{P,j}^I & \text{si } wh_{j,k} \geq 0 \\ h_{P,j}^S & \text{si } wh_{j,k} < 0 \end{cases} \quad (2.34)$$

Del mismo modo, a partir de la ecuación (2.23) se obtiene:

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

$$\frac{\partial \text{neto}_{P,k}^S}{\partial w_{j,k}} = \begin{cases} h_{P,j}^S & \text{si } wh_{j,k} \geq 0 \\ h_{P,j}^I & \text{si } wh_{j,k} < 0 \end{cases} \quad (2.35)$$

Definimos las siguientes variables:

$$\delta_{P,k}^I = -\frac{1}{2} \cdot (t_{P,k}^I - o_{P,k}^I) \cdot o_{P,k}^I \cdot (1 - o_{P,k}^I) \quad (2.36)$$

$$\delta_{P,k}^S = -\frac{1}{2} \cdot (t_{P,k}^S - o_{P,k}^S) \cdot o_{P,k}^S \cdot (1 - o_{P,k}^S) \quad (2.37)$$

El resultado final de la derivada será:

$$\frac{\partial E_P}{\partial w_{j,k}} = \begin{cases} \delta_{P,k}^I \cdot h_{P,j}^I + \delta_{P,k}^S \cdot h_{P,j}^S & \text{si } wh_{j,k} \geq 0 \\ \delta_{P,k}^I \cdot h_{P,j}^S + \delta_{P,k}^S \cdot h_{P,j}^I & \text{si } wh_{j,k} < 0 \end{cases} \quad (2.38)$$

### Cálculo de las derivadas de la capa de entrada

Calcularemos ahora la derivada respecto de los pesos entre las unidades de la capa oculta y la de entrada.

$$\frac{\partial E_P}{\partial w_{i,j}} = \frac{\partial E_P^I}{\partial w_{i,j}} + \frac{\partial E_P^S}{\partial w_{i,j}} \quad (2.39)$$

Vemos que ahora dependerá de las variables  $o_{P,k}$ ,  $\text{neto}_{P,k}$  y  $h_{P,j}$ . Tendremos:

$$\frac{\partial E_P^I}{\partial w_{i,j}} = \sum_k \frac{\partial E_P}{\partial o_{P,k}^I} \cdot \frac{\partial o_{P,k}^I}{\partial \text{neto}_{P,k}^I} \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial w_{i,j}} \quad (2.40)$$

$$\frac{\partial E_P^S}{\partial w_{i,j}} = \sum_k \frac{\partial E_P}{\partial o_{P,k}^S} \cdot \frac{\partial o_{P,k}^S}{\partial \text{neto}_{P,k}^S} \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial w_{i,j}} \quad (2.41)$$

De la ecuación (2.22) y teniendo en cuenta el signo de los pesos, tenemos:

$$\frac{\partial \text{neto}_{P,k}^I}{\partial w_{i,j}} = \begin{cases} \frac{\partial \text{neto}_{P,k}^I}{\partial h_{P,j}^I} \cdot \frac{\partial h_{P,j}^I}{\partial \text{neth}_{P,k}^I} \cdot \frac{\partial \text{neth}_{P,k}^I}{\partial w_{i,j}} & \text{si } wh_{j,k} \geq 0 \\ \frac{\partial \text{neto}_{P,k}^I}{\partial h_{P,j}^S} \cdot \frac{\partial h_{P,j}^S}{\partial \text{neth}_{P,k}^S} \cdot \frac{\partial \text{neth}_{P,k}^S}{\partial w_{i,j}} & \text{si } wh_{j,k} < 0 \end{cases} \quad (2.42)$$

Del mismo modo a partir de la ecuación (2.23):

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

$$\frac{\partial \text{neto}_{P,k}^S}{\partial w_{i,j}} = \begin{cases} \frac{\partial \text{neto}_{P,k}^S}{\partial h_{P,j}^S} \cdot \frac{\partial h_{P,j}^S}{\partial \text{neto}_{P,k}^S} \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial w_{i,j}} & \text{si } wh_{j,k} \geq 0 \\ \frac{\partial \text{neto}_{P,k}^S}{\partial h_{P,j}^I} \cdot \frac{\partial h_{P,j}^I}{\partial \text{neto}_{P,k}^S} \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial w_{i,j}} & \text{si } wh_{j,k} < 0 \end{cases} \quad (2.43)$$

Sustituyendo las ecuaciones anteriores en la derivada que queremos calcular se obtiene la siguiente ecuación.

$$\begin{aligned} \frac{\partial E_P}{\partial w_{i,j}} = & \sum_{k, wh_{j,k} \geq 0} \left\{ \frac{\partial E_P}{\partial o_{P,k}^I} \cdot \frac{\partial o_{P,k}^I}{\partial \text{neto}_{P,k}^I} \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial h_{P,j}^I} \cdot \frac{\partial h_{P,j}^I}{\partial \text{neto}_{P,k}^I} \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial w_{i,j}} \right\} + \\ & \sum_{k, wh_{j,k} < 0} \left\{ \frac{\partial E_P}{\partial o_{P,k}^I} \cdot \frac{\partial o_{P,k}^I}{\partial \text{neto}_{P,k}^I} \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial h_{P,j}^S} \cdot \frac{\partial h_{P,j}^S}{\partial \text{neto}_{P,k}^S} \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial w_{i,j}} \right\} + \\ & \sum_{k, wh_{j,k} \geq 0} \left\{ \frac{\partial E_P}{\partial o_{P,k}^S} \cdot \frac{\partial o_{P,k}^S}{\partial \text{neto}_{P,k}^S} \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial h_{P,j}^S} \cdot \frac{\partial h_{P,j}^S}{\partial \text{neto}_{P,k}^S} \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial w_{i,j}} \right\} + \\ & \sum_{k, wh_{j,k} < 0} \left\{ \frac{\partial E_P}{\partial o_{P,k}^S} \cdot \frac{\partial o_{P,k}^S}{\partial \text{neto}_{P,k}^S} \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial h_{P,j}^I} \cdot \frac{\partial h_{P,j}^I}{\partial \text{neto}_{P,k}^I} \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial w_{i,j}} \right\} \end{aligned} \quad (2.44)$$

Utilizando las ecuaciones (2.32), (2.36) y (2.37) podemos sustituir en la ecuación anterior el valor de las derivadas.

$$\begin{aligned} \frac{\partial E_P}{\partial w_{i,j}} = & \sum_{k, wh_{j,k} \geq 0} \left\{ \delta_{P,k}^I \cdot wh_{j,k} \cdot h_{P,j}^I \cdot (1 - h_{P,j}^I) \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial w_{i,j}} \right\} + \\ & \sum_{k, wh_{j,k} < 0} \left\{ \delta_{P,k}^I \cdot wh_{j,k} \cdot h_{P,j}^S \cdot (1 - h_{P,j}^S) \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial w_{i,j}} \right\} + \\ & \sum_{k, wh_{j,k} \geq 0} \left\{ \delta_{P,k}^S \cdot wh_{j,k} \cdot h_{P,j}^S \cdot (1 - h_{P,j}^S) \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial w_{i,j}} \right\} + \\ & \sum_{k, wh_{j,k} < 0} \left\{ \delta_{P,k}^S \cdot wh_{j,k} \cdot h_{P,j}^I \cdot (1 - h_{P,j}^I) \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial w_{i,j}} \right\} \end{aligned} \quad (2.45)$$

Solo queda calcular la última derivada que depende del signo de los pesos:

$$\frac{\partial \text{neto}_{P,j}^I}{\partial w_{i,j}} = \begin{cases} i_{P,i}^I & \text{si } w_{i,j} \geq 0 \\ i_{P,i}^S & \text{si } w_{i,j} < 0 \end{cases} \quad (2.46)$$

$$\frac{\partial \text{neto}_{P,j}^S}{\partial w_{i,j}} = \begin{cases} i_{P,i}^S & \text{si } w_{i,j} \geq 0 \\ i_{P,i}^I & \text{si } w_{i,j} < 0 \end{cases} \quad (2.47)$$

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

Definimos las siguientes variables:

$$\delta h_{P,j}^I = \left( \sum_{k, wh_{j,k} \geq 0} \delta_{P,k}^I \cdot wh_{j,k} + \sum_{k, wh_{j,k} < 0} \delta_{P,k}^S \cdot wh_{j,k} \right) \cdot h_{P,j}^I \cdot (1 - h_{P,j}^I) \quad (2.48)$$

$$\delta h_{P,j}^S = \left( \sum_{k, wh_{j,k} \geq 0} \delta_{P,k}^S \cdot wh_{j,k} + \sum_{k, wh_{j,k} < 0} \delta_{P,k}^I \cdot wh_{j,k} \right) \cdot h_{P,j}^S \cdot (1 - h_{P,j}^S) \quad (2.49)$$

Utilizando (2.46), (2.47), (2.48) y (2.49) y sustituyendo en la ecuación (2.45) queda:

$$\frac{\partial E_P}{\partial w_{i,j}} = \begin{cases} \delta h_{P,j}^I \cdot i_{P,i}^I + \delta h_{P,j}^S \cdot i_{P,i}^S & \text{si } w_{i,j} \geq 0 \\ \delta h_{P,j}^S \cdot i_{P,i}^I + \delta h_{P,j}^I \cdot i_{P,i}^S & \text{si } w_{i,j} < 0 \end{cases} \quad (2.50)$$

### 2.2.2.3.2. Algoritmo de Ishibuchi

En esta generalización a aritmética de intervalos, el vector de los objetivos no es un vector de intervalos, sino que cuando la entrada se clasifique en una cierta clase el objetivo será 1 y si no será cero.

La función de error definida por Ishibuchi en las referencias [5] y [6] es también diferente.

Ishibuchi define la función de error que hay que minimizar como:

$$E_P = \sum_{k=1}^{N_S} \max \left\{ \frac{(t_{P,k} - o_{P,k})^2}{2} \quad \text{tal que: } o_{P,k} \in O_{P,k} \right\} \quad (2.51)$$

Donde  $o_{P,k}$  es un número real, y  $O_{P,k}$  es el intervalo de salida para la  $k$ -ésima unidad de salida.

Utilizaremos aquí la misma notación que vimos para aritmética de intervalos desde la ecuación (2.15) hasta la (2.23).

Dada una clase (unidad de salida) le restamos al objetivo de dicha clase el valor  $o_{P,k}$  del intervalo y nos quedamos con el máximo. Es decir, sea  $r$  una clase:

Si el patrón es de la clase  $r$   $t_{P,k}=1$  y  $o_{P,k}$  debe ser mínimo, será  $o_{P,k}^I$ .

Si el patrón no es de la clase  $r$   $t_{P,k}=0$  y  $o_{P,k}$  debe ser máximo, será  $o_{P,k}^S$ .



## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

La función *max* de la ecuación (2.51) equivale simplemente a distinguir en la suma si la salida pertenece o no a la clase. Podemos reescribir así dicha ecuación como:

$$E_P = \left\{ \sum_{\substack{k=1 \\ k \neq r}}^{N_S} \frac{(t_{P,k} - o_{P,k}^S)^2}{2} \right\} + \frac{1}{2} \cdot (t_{P,r} - o_{P,r}^I)^2 \quad (2.52)$$

Igual que en el algoritmo de Hernández, después de cada iteración los pesos serán modificados según las siguientes ecuaciones:

$$\Delta wh_{j,k}(t+1) = \eta \cdot \left( -\frac{\partial E_P}{\partial wh_{j,k}} \right) + \alpha \cdot \Delta wh_{j,k}(t) \quad (2.53)$$

$$\Delta wi_{i,j}(t+1) = \eta \cdot \left( -\frac{\partial E_P}{\partial wi_{i,j}} \right) + \alpha \cdot \Delta wi_{i,j}(t) \quad (2.54)$$

Donde  $\eta$  es el paso de adaptación y  $\alpha$  el momento.

Veamos como calcular las derivadas en las expresiones anteriores.

### Cálculo de las derivadas de la capa oculta

Calculamos las derivadas para los pesos entre la capa oculta y la de salida.

A partir de la ecuación (2.52) vemos que hay que distinguir entre sí la entrada se clasifica en una cierta clase  $r$  o no.

$$\frac{\partial E_P}{\partial wh_{j,k}} = \begin{cases} \frac{\partial E_P}{\partial o_{P,k}^I} \cdot \frac{\partial o_{P,k}^I}{\partial wh_{j,k}} = -(t_{P,k} - o_{P,k}^I) \cdot \frac{\partial o_{P,k}^I}{\partial wh_{j,k}} & \text{si } k = r \\ \frac{\partial E_P}{\partial o_{P,k}^S} \cdot \frac{\partial o_{P,k}^S}{\partial wh_{j,k}} = -(t_{P,k} - o_{P,k}^S) \cdot \frac{\partial o_{P,k}^S}{\partial wh_{j,k}} & \text{si } k \neq r \end{cases} \quad (2.55)$$

Para calcular las derivadas de  $o_{P,k}^I$  y  $o_{P,k}^S$  usamos las siguientes ecuaciones:

$$\frac{\partial o_{P,k}^I}{\partial wh_{j,k}} = \frac{\partial o_{P,k}^I}{\partial \text{neto}_{P,k}^I} \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial wh_{j,k}} = o_{P,k}^I \cdot (1 - o_{P,k}^I) \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial wh_{j,k}} \quad (2.56)$$

$$\frac{\partial o_{P,k}^S}{\partial wh_{j,k}} = \frac{\partial o_{P,k}^S}{\partial \text{neto}_{P,k}^S} \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial wh_{j,k}} = o_{P,k}^S \cdot (1 - o_{P,k}^S) \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial wh_{j,k}} \quad (2.57)$$

Sustituyendo en la ecuación (2.55) queda:

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

$$\frac{\partial E_P}{\partial wh_{j,k}} = \begin{cases} -(t_{P,k} - o_{P,k}^I) \cdot o_{P,k}^I \cdot (1 - o_{P,k}^I) \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial wh_{j,k}} & \text{si } k = r \\ -(t_{P,k} - o_{P,k}^S) \cdot o_{P,k}^S \cdot (1 - o_{P,k}^S) \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial wh_{j,k}} & \text{si } k \neq r \end{cases} \quad (2.58)$$

Definimos las siguientes variables:

$$\delta_{P,k}^I = -(t_{P,k} - o_{P,k}^I) \cdot o_{P,k}^I \cdot (1 - o_{P,k}^I) \quad (2.59)$$

$$\delta_{P,k}^S = -(t_{P,k} - o_{P,k}^S) \cdot o_{P,k}^S \cdot (1 - o_{P,k}^S) \quad (2.60)$$

Sustituimos las ecuaciones (2.34) y (2.35) en las ecuaciones (2.56) y (2.57) y para obtener el resultado final de la derivada sustituimos el resultado anterior junto con las ecuaciones (2.59) y (2.60) en la ecuación (2.58). Así tenemos:

$$\frac{\partial E_P}{\partial wh_{j,k}} = \begin{cases} \delta_{P,k}^I \cdot h_{P,j}^I & \text{si } k = r \text{ y } wh_{j,k} \geq 0 \\ \delta_{P,k}^S \cdot h_{P,j}^S & \text{si } k \neq r \text{ y } wh_{j,k} \geq 0 \\ \delta_{P,k}^I \cdot h_{P,j}^S & \text{si } k = r \text{ y } wh_{j,k} < 0 \\ \delta_{P,k}^S \cdot h_{P,j}^I & \text{si } k \neq r \text{ y } wh_{j,k} < 0 \end{cases} \quad (2.61)$$

### Cálculo de las derivadas de la capa de entrada

Calcularemos ahora la derivada respecto de los pesos entre las unidades de la capa oculta y la de entrada.

Hay que distinguir de nuevo entre sí los pesos son positivos o negativos.

$$\frac{\partial E_P}{\partial wi_{i,j}} = \begin{cases} \frac{\partial E_P}{\partial o_{P,k}^I} \cdot \frac{\partial o_{P,k}^I}{\partial wi_{i,j}} = -(t_{P,k} - o_{P,k}^I) \cdot \frac{\partial o_{P,k}^I}{\partial wi_{i,j}} & \text{si } k = r \\ \frac{\partial E_P}{\partial o_{P,k}^S} \cdot \frac{\partial o_{P,k}^S}{\partial wi_{i,j}} = -\sum_{\substack{k=1 \\ k \neq r}}^{Ns} (t_{P,k} - o_{P,k}^S) \cdot \frac{\partial o_{P,k}^S}{\partial wh_{i,j}} & \text{si } k \neq r \end{cases} \quad (2.62)$$

Teniendo en cuenta el signo de los pesos tenemos:

$$\frac{\partial o_{P,k}^I}{\partial wi_{i,j}} = \begin{cases} \frac{\partial o_{P,k}^I}{\partial \text{neto}_{P,k}^I} \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial h_{P,j}^I} \cdot \frac{\partial h_{P,j}^I}{\partial \text{neth}_{P,j}^I} \cdot \frac{\partial \text{neth}_{P,j}^I}{\partial wi_{i,j}} & \text{si } wh_{j,k} \geq 0 \\ \frac{\partial o_{P,k}^I}{\partial \text{neto}_{P,k}^I} \cdot \frac{\partial \text{neto}_{P,k}^I}{\partial h_{P,j}^S} \cdot \frac{\partial h_{P,j}^S}{\partial \text{neth}_{P,j}^S} \cdot \frac{\partial \text{neth}_{P,j}^S}{\partial wi_{i,j}} & \text{si } wh_{j,k} < 0 \end{cases} \quad (2.63)$$

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

$$\frac{\partial o_{P,k}^S}{\partial w_{i,j}} = \begin{cases} \frac{\partial o_{P,k}^S}{\partial \text{neto}_{P,k}^S} \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial h_{P,j}^S} \cdot \frac{\partial h_{P,j}^S}{\partial \text{neth}_{P,j}^S} \cdot \frac{\partial \text{neth}_{P,j}^S}{\partial w_{i,j}} & \text{si } wh_{j,k} \geq 0 \\ \frac{\partial o_{P,k}^S}{\partial \text{neto}_{P,k}^S} \cdot \frac{\partial \text{neto}_{P,k}^S}{\partial h_{P,j}^I} \cdot \frac{\partial h_{P,j}^I}{\partial \text{neth}_{P,j}^I} \cdot \frac{\partial \text{neth}_{P,j}^I}{\partial w_{i,j}} & \text{si } wh_{j,k} < 0 \end{cases} \quad (2.64)$$

Utilizando las ecuaciones (2.46) y (2.47), la derivada del límite inferior será:

$$\frac{\partial o_{P,k}^I}{\partial w_{i,j}} = \begin{cases} o_{P,k}^I \cdot (1 - o_{P,k}^I) \cdot wh_{j,k} \cdot h_{P,j}^I \cdot (1 - h_{P,j}^I) \cdot i_{P,i}^I & \text{si } w_{i,j} \geq 0 \text{ y } wh_{j,k} \geq 0 \\ o_{P,k}^I \cdot (1 - o_{P,k}^I) \cdot wh_{j,k} \cdot h_{P,j}^I \cdot (1 - h_{P,j}^I) \cdot i_{P,i}^S & \text{si } w_{i,j} < 0 \text{ y } wh_{j,k} \geq 0 \\ o_{P,k}^I \cdot (1 - o_{P,k}^I) \cdot wh_{j,k} \cdot h_{P,j}^S \cdot (1 - h_{P,j}^S) \cdot i_{P,i}^S & \text{si } w_{i,j} \geq 0 \text{ y } wh_{j,k} < 0 \\ o_{P,k}^I \cdot (1 - o_{P,k}^I) \cdot wh_{j,k} \cdot h_{P,j}^S \cdot (1 - h_{P,j}^S) \cdot i_{P,i}^I & \text{si } w_{i,j} < 0 \text{ y } wh_{j,k} < 0 \end{cases} \quad (2.65)$$

Y la del límite superior:

$$\frac{\partial o_{P,k}^S}{\partial w_{i,j}} = \begin{cases} o_{P,k}^S \cdot (1 - o_{P,k}^S) \cdot wh_{j,k} \cdot h_{P,j}^S \cdot (1 - h_{P,j}^S) \cdot i_{P,i}^S & \text{si } w_{i,j} \geq 0 \text{ y } wh_{j,k} \geq 0 \\ o_{P,k}^S \cdot (1 - o_{P,k}^S) \cdot wh_{j,k} \cdot h_{P,j}^S \cdot (1 - h_{P,j}^S) \cdot i_{P,i}^I & \text{si } w_{i,j} < 0 \text{ y } wh_{j,k} \geq 0 \\ o_{P,k}^S \cdot (1 - o_{P,k}^S) \cdot wh_{j,k} \cdot h_{P,j}^I \cdot (1 - h_{P,j}^I) \cdot i_{P,i}^I & \text{si } w_{i,j} \geq 0 \text{ y } wh_{j,k} < 0 \\ o_{P,k}^S \cdot (1 - o_{P,k}^S) \cdot wh_{j,k} \cdot h_{P,j}^I \cdot (1 - h_{P,j}^I) \cdot i_{P,i}^S & \text{si } w_{i,j} < 0 \text{ y } wh_{j,k} < 0 \end{cases} \quad (2.66)$$

El resultado final de la derivada sustituyendo en la ecuación (2.62) y usando las ecuaciones (2.59) y (2.60) será:

$$\frac{\partial E_P}{\partial w_{i,j}} = \begin{cases} \delta_{P,k}^I \cdot wh_{j,k} \cdot h_{P,j}^I \cdot (1 - h_{P,j}^I) \cdot i_{P,i}^I & \text{si } k = r \text{ } wh_{j,k} \geq 0 \text{ } w_{i,j} \geq 0 \\ \delta_{P,k}^I \cdot wh_{j,k} \cdot h_{P,j}^I \cdot (1 - h_{P,j}^I) \cdot i_{P,i}^S & \text{si } k = r \text{ } wh_{j,k} \geq 0 \text{ } w_{i,j} < 0 \\ \delta_{P,k}^I \cdot wh_{j,k} \cdot h_{P,j}^S \cdot (1 - h_{P,j}^S) \cdot i_{P,i}^S & \text{si } k = r \text{ } wh_{j,k} < 0 \text{ } w_{i,j} \geq 0 \\ \delta_{P,k}^I \cdot wh_{j,k} \cdot h_{P,j}^S \cdot (1 - h_{P,j}^S) \cdot i_{P,i}^I & \text{si } k = r \text{ } wh_{j,k} < 0 \text{ } w_{i,j} < 0 \\ \sum_{k=1 \wedge k \neq r}^{Ns} \delta_{P,k}^S \cdot wh_{j,k} \cdot h_{P,j}^S \cdot (1 - h_{P,j}^S) \cdot i_{P,i}^S & \text{si } k \neq r \text{ } wh_{j,k} \geq 0 \text{ } w_{i,j} \geq 0 \\ \sum_{k=1 \wedge k \neq r}^{Ns} \delta_{P,k}^S \cdot wh_{j,k} \cdot h_{P,j}^S \cdot (1 - h_{P,j}^S) \cdot i_{P,i}^I & \text{si } k \neq r \text{ } wh_{j,k} \geq 0 \text{ } w_{i,j} < 0 \\ \sum_{k=1 \wedge k \neq r}^{Ns} \delta_{P,k}^S \cdot wh_{j,k} \cdot h_{P,j}^I \cdot (1 - h_{P,j}^I) \cdot i_{P,i}^I & \text{si } k \neq r \text{ } wh_{j,k} < 0 \text{ } w_{i,j} \geq 0 \\ \sum_{k=1 \wedge k \neq r}^{Ns} \delta_{P,k}^S \cdot wh_{j,k} \cdot h_{P,j}^I \cdot (1 - h_{P,j}^I) \cdot i_{P,i}^S & \text{si } k \neq r \text{ } wh_{j,k} < 0 \text{ } w_{i,j} < 0 \end{cases} \quad (2.67)$$

### 2.2.2.3.3. Medidas de eficiencia

Una vez entrenada la red, hay que medir la eficiencia que proporciona en función del algoritmo utilizado en el entrenamiento para así poder comparar cuál es mejor. Para ello, necesitamos algún criterio que mida la eficiencia.

Dado un nuevo conjunto de ejemplos que contengan entradas desconocidas (conjunto de test), vemos en que medida son clasificados correctamente por la red, es decir, medimos el porcentaje de ejemplos bien clasificados (porcentaje de aciertos).

Sea un vector de entrada  $q$ , ecuación (2.14), cuyos elementos son intervalos, ecuación (2.15), la red nos proporcionará un vector de salida cuyos elementos son también intervalos, dados por la ecuación (2.17). Es necesario un criterio de comparación de dichos intervalos para decidir que clase se asocia con al vector de entrada.

$$\text{Sea la Entrada: } \{I_{q,1}, I_{q,2}, \dots, I_{q,N}\} = \{[i_{q,1}^I, i_{q,1}^S], [i_{q,2}^I, i_{q,2}^S], \dots, [i_{q,N}^I, i_{q,N}^S]\}$$

$$\text{Sea la Salida: } \{O_{q,1}, O_{q,2}, \dots, O_{q,Ns}\} = \{[o_{q,1}^I, o_{q,1}^S], [o_{q,2}^I, o_{q,2}^S], \dots, [o_{q,Ns}^I, o_{q,Ns}^S]\}$$

En nuestro trabajo hemos usado cinco criterios de comparación diferentes, algunos de los cuales han sido propuestos por Ishibuchi en la referencia [6].

$$\text{Definición 1: } O_{q,h} <_1 O_{q,k} \Leftrightarrow o_{q,h}^S < o_{q,k}^S \quad (2.68)$$

$$\text{Definición 2: } O_{q,h} <_2 O_{q,k} \Leftrightarrow o_{q,h}^I < o_{q,k}^I \quad (2.69)$$

$$\text{Definición 3: } O_{q,h} <_3 O_{q,k} \Leftrightarrow o_{q,h}^I < o_{q,k}^I \quad \text{y} \quad o_{q,h}^S < o_{q,k}^S \quad (2.70)$$

$$\text{Definición 4: } O_{q,h} <_4 O_{q,k} \Leftrightarrow o_{q,h}^S < o_{q,k}^I \quad (2.71)$$

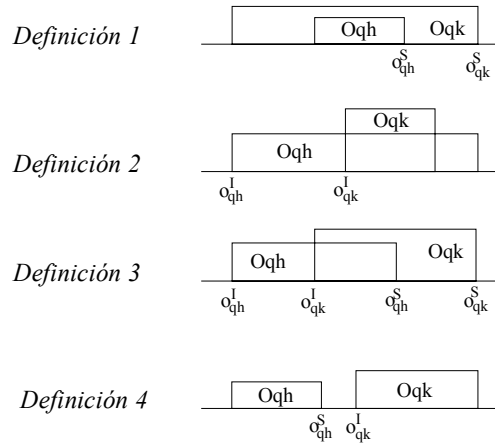
$$\text{Definición 5: } \text{Sea } \Psi = \{k \mid o_{q,k}^S \geq 0.5 \text{ y } o_{q,k}^I \geq 0.5 \text{ para } k = 1, \dots, Ns\} \quad (2.72)$$

*se asigna la clase  $k$  si el conjunto solo contiene un elemento*

Definimos la siguiente regla de clasificación, decimos que se asigna la clase  $k$  al vector de entrada  $I_q$  para cada una de las cuatro primeras definiciones anteriores (subíndice  $i$ ) si se cumple:

$$O_{q,h} <_i O_{q,k} \quad \text{para } h = 1, 2, \dots, Ns \quad \text{con } h \neq k \quad (2.73)$$

Para la definición 5, se asignará la clase  $k$  en caso de que sea el único intervalo cuyos límites superior e inferior sean mayores que 0.5. En cualquier otro caso, no se clasifica correctamente.



**Fig.3. Ilustración de los cuatro primeros criterios de comparación.**

En nuestros experimentos utilizaremos las cinco definiciones anteriores para medir el porcentaje de aciertos.

### 2.2.3. Codificación mediante Probabilidades

Este método fue descrito por Ahmad y Tresp en la referencia [9].

El método está basado en la estimación de las probabilidades asignadas a las clases dadas ciertas entradas desconocidas. Después de la estimación, se asigna el patrón con entradas desconocidas a la clase que tiene mayor probabilidad. Así, no es necesario un entrenamiento previo de la red con entradas desconocidas.

El cálculo de las probabilidades para el caso de Multilayer Feedforward implica el cálculo de una integral, la cual es obtenida utilizando técnicas de integración numérica. De esta forma, la complejidad computacional crece exponencialmente con el número de entradas desconocidas. Ahmad y Tresp en trabajos previos [8], demostraron que esta metodología sólo era eficiente cuando se usaba en redes con funciones Gaussianas normalizadas.

Los autores desarrollaron para este método un mecanismo de aproximación para el cálculo de la integral que fuese más eficiente y propusieron una solución que cumple, que la complejidad no aumenta con el número de entradas desconocidas.

A continuación describimos en qué consiste este método.

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

Asumimos que la red neuronal ha sido entrenada para clasificar un cierto patrón en una clase  $NN(x)$ , es decir, que puede estimar la expectación  $E(y|x)$  de la clase  $y$  para un cierto patrón de entrada  $x$ . La red debería ser capaz de calcular dicha expectación si el patrón posee entradas desconocidas. Sea  $x^c$  las entradas conocidas y  $x^u$  las entradas desconocidas de un patrón  $x$  dado. La expectación se calculará mediante la ecuación (2.74):

$$E(y|x^c) = \int E(y|x^c, x^u) \cdot P(x^u|x^c) \cdot dx^u \approx \frac{1}{P(x^c)} \cdot \int NN(x^c, x^u) \cdot P(x^c|x^u) \cdot dx^u \quad (2.74)$$

Para una red entrenada  $NN_i(x) = P(class_i|x)$  por lo que basta con sustituir en la ecuación anterior los valores de  $E(y|x^c)$  por  $P(class_i|x^c)$  y de  $NN(x^c, x^u)$  por  $NN_i(x^c, x^u) = P(class_i|x^c, x^u)$ .

Esta integración sigue siendo problemática. Se propone entonces, en el método de Ahmad y Tresp, una nueva aproximación utilizando “Ventanas de Parzen”.

Dado un conjunto de  $N$  patrones entrenados  $\{(x^k, y^k) | k=1, \dots, N\}$  podemos aproximar la probabilidad de que se de un cierto patrón  $x$  como:

$$P(x) \approx \frac{1}{N} \cdot \sum_{k=1}^N G(x; x^k, \sigma) \quad (2.75)$$

Donde  $G(x; x^k, \sigma)$  es una gaussiana normalizada centrada en el dato  $x^k$  del conjunto de entrenamiento y con varianza  $\sigma^2$ . Dicha gaussiana se puede expresar mediante la ecuación (2.76).

$$G(x; x^k, \sigma) = \frac{1}{(2 \cdot \pi \cdot \sigma^2)^{D/2}} \cdot \exp\left(-\frac{1}{2 \cdot \sigma^2} \cdot \|x - x^k\|^2\right) \quad (2.76)$$

$D$  es la dimensión del espacio de entrada (número de entradas de un patrón).

Por otra parte, podemos representar el conjunto de datos de entrenamiento en dos dimensiones tomando  $x_1$  (abscisas) como coordenadas conocidas del patrón  $x^k$  y  $x_2$  (ordenadas) las coordenadas desconocidas de dicho patrón.

Se puede estimar la distribución de patrones de entrada usando “Ventanas de Parzen” de forma que, para un patrón con  $x^c$ , vector de entradas conocidas, queda la ecuación (2.77):

$$P(x^c) \approx \frac{1}{N} \sum_{k=1}^N G(x^c; x^{c,k}, \sigma) \quad (2.77)$$

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

Donde  $G(x^c; x^{c,k}, \sigma)$  es una gaussiana proyectada en la dimensión de las entradas conocidas,  $x^c$  son los componentes conocidas del patrón considerado para el test y  $x^{c,k}$  son las correspondientes componentes de un patrón  $x^k$  del conjunto de entrenamiento.

La probabilidad de una clase dado un patrón con entradas conocidas  $x^c$  viene dada a través de la ecuación (2.78).

$$E(y | x^c) = \frac{\sum_{k=1}^N \int NN(x^c, x^u) \cdot G(x^c, x^u; x^k, \sigma) dx^u}{\sum_{k=1}^N G(x^c, x^{c,k}, \sigma)} \quad (2.78)$$

En este método se propone la siguiente aproximación para la integral:

$$\int NN(x^c, x^u) \cdot G(x^c, x^u; x^k, \sigma) dx^u \approx NN(x^c, x^{u,k}) \cdot G(x^c; x^{c,k}, \sigma) \quad (2.79)$$

Donde  $NN(x^c, x^{u,k})$  es la predicción de la red que se obtiene si sustituimos los componentes del patrón de entradas desconocidas por dichas componentes tomadas del conjunto de entrenamiento, la cuales son conocidas.

Al final, lo que se obtiene con esta aproximación como probabilidad de una clase es la ecuación (2.80)

$$E(y | x^c) \approx \frac{\sum_{k=1}^N NN(x^c, x^{u,k}) \cdot G(x^c; x^{c,k}, \sigma)}{\sum_{k=1}^N G(x^c; x^{c,k}, \sigma)} \quad (2.80)$$

Para realizar el cálculo de la ecuación anterior es necesario fijar previamente un valor adecuado de  $\sigma$  en las gaussianas.

En la referencia [9] se proponen dos métodos para realizar su cálculo. Ambos consisten en buscar el valor de  $\sigma$  probando diferentes valores y quedarnos con aquel que proporcione una mayor eficiencia, en el primer caso del logaritmo de la expectación de  $x^k$  y en el segundo del valor de la ecuación (2.80).

En nuestra investigación hemos usado el primer método. En él se parte de un intervalo inicial en el cual estará contenido el valor final de  $\sigma$ . Definimos una partición de dicho intervalo encontrando un conjunto de valores para  $\sigma$  igualmente espaciados. Para este conjunto se calcula, para los diferentes patrones del conjunto de entrenamiento, la expectación para los diferentes valores de  $\sigma$  mediante la ecuación (2.81).

## 2.2. PROCEDIMIENTOS DE UTILIZACIÓN DE ENTRADAS DESCONOCIDAS

$$\tilde{P}(x^k) \approx \frac{1}{N-1} \cdot \sum_{l=1, l \neq k}^N G(x^k; x^l, \sigma) \quad (2.81)$$

Finalmente se escoge el valor de  $\sigma$  que haga máxima la expresión de la ecuación (2.82):

$$\sum_k \log \tilde{P}(x^k) \quad (2.82)$$

### 2.2.4. Codificación de entradas binarias, que no utiliza entradas desconocidas en el entrenamiento

Este método descrito en [9], solo es aplicable para el caso de entradas binarias, por lo que su uso es bastante limitado.

El método utiliza un entrenamiento sin entradas desconocidas de la red Multilayer Feedforward con Backpropagation.

Después, en el conjunto de test se sustituyen las entradas desconocidas por un valor calculado a partir de la ecuación (2.83).

$$input_i = \frac{\sum_j r_{i,j} \cdot input_j \cdot m_j}{\sum_j r_{i,j} \cdot m_j} \quad (2.83)$$

Donde:

- $m_j$  es la componente de un vector que identifica si la entrada  $j$  de un cierto patrón de entrada,  $input_j$  es desconocida o no. Si es desconocida  $m_j$  valdrá 0 y si es conocida valdrá 1.

- $r$  es una matriz de “coeficientes de redundancia” (denominados así en la referencia) cuyos elementos  $r_{i,j}$  se calculan a partir de la ecuación (2.84).

$$r_{i,j} = \frac{\sum_{k=1}^n \overline{input_{k,i} \oplus input_{k,j}}}{n} \quad (2.84)$$

Donde  $n$  representa el número de patrones de entrada.



## 2.3. Metodología

En primer lugar se realizó una recopilación bibliográfica para seleccionar los métodos más representativos de mecanismos de codificación de entradas desconocidas, lo cuáles hemos descrito antes.

Para realizar el estudio hemos utilizado la siguiente metodología.

- *Primero escogemos las bases de datos del repositorio UCI, que utilizaremos e introduciremos en ellas entradas desconocidas.*
- *Escogemos para cada problema la arquitectura de red adecuada.*
- *Después realizamos un proceso de entrenamiento y test. Se entrenarán varias redes para cada arquitectura con diferentes inicialización de pesos y particiones de datos. Escogeremos el punto de convergencia (pesos) de cada red entrenada utilizando cros-validación y realizaremos el test de las diferentes redes para obtener el porcentaje de aciertos.*
- *Por último se analizarán los resultados y compararán los resultados de los diferentes métodos.*

### 2.3.1. Elección de bases de datos

Una vez seleccionados los diferentes métodos hicimos una selección de los problemas que utilizaríamos para evaluar la eficiencia de los mismos.

Dichos problemas fueron escogidos de entre los que están accesibles de forma pública en la Universidad de California en el “UCI repository of machine learning databases”. En el capítulo primero dimos una descripción detallada de dichos problemas.

Escogimos en total 19 bases de datos cuyos nombres son los siguientes:

*Abalone (ABALONE), Balance Scale Weight & Distance (BALANCE), Cylinder Bands (BANDS), BUPA Liver Disorders (BUPA), Credit Approval (CREDITO), Display1 (DISPLAY1), Display2 (DISPLAY2), Glass Identification (GLASS), Heart Disease (HEART), Image Segmentation (IMAGEN), Mushroom (MUSHROOM), The Monk’s Problems (MOK1, MOK2, MOK3), Musk (MUSK), Pima Indians Diabetes (PIMA), Thyroid Domain (THYROID), 1984 U.S. Congressional Voting Records (VOTE), Wisconsin Diagnostic Breast Cancer (WDBC).*

### 2.3. METODOLOGÍA

Las bases de datos públicas escogidas en general no contienen entradas desconocidas. Por ello, el primer paso fue generar a partir de las bases disponibles unas nuevas bases de datos con entradas desconocidas. El procedimiento seguido se detalla a continuación.

Para cada uno de los problemas hemos generado cinco nuevos problemas introduciendo aleatoriamente entradas desconocidas en los conjuntos de entrenamiento y test. El número de entradas desconocidas introducido en los datos varía según los porcentajes 5%, 10%, 20%, 30%, 40% sobre el total de entradas. Entendemos como total de entradas para un cierto conjunto (de entrenamiento o test) el número que se obtiene multiplicando el número de ejemplos de dicho conjunto por el número de entradas de cada ejemplo.

Las entradas que tomaremos como desconocidas entre el total de cada uno de los conjuntos (de entrenamiento y test) se seleccionarán de forma aleatoria y uniformemente distribuida entre el conjunto total de entradas. El número de entradas que tomaremos como desconocidas coincidirá con el porcentaje deseado.

Como resultado final, obtenemos por cada problema, cinco nuevos problemas diferentes con entradas desconocidas tanto en el conjunto de entrenamiento como en el de test:

- Con 5% de entradas desconocidas.
- Con 10% de entradas desconocidas.
- Con 20% de entradas desconocidas.
- Con 30% de entradas desconocidas.
- Con 40% de entradas desconocidas.

En total tendremos 95 problemas con entradas desconocidas y 19 sin entradas desconocidas (0% de entradas desconocidas), 114 problemas diferentes.

Así, tendremos para cada problema, seis posibles bases de datos para el entrenamiento y seis para el test, y si combinamos los diferentes porcentajes de entradas desconocidas obtendremos resultados para 36 diferentes problemas por cada problema original. Esto nos permitirá estudiar la influencia de las entradas desconocidas, tanto en el entrenamiento, como en el test. Para ello, veremos como influyen estos factores en el porcentaje de clasificación (porcentaje de aciertos).

Como hemos visto, para cada base de datos tenemos divididos los datos en un conjunto de datos para el entrenamiento y un conjunto de datos para el test. El conjunto de datos de entrenamiento, lo dividimos a su vez en dos, uno para el entrenamiento propiamente dicho y otro para escoger el punto de convergencia

mediante el método de cros-validación, que veremos más adelante. El porcentaje final de datos de cada uno de estos conjuntos sobre el total de datos fue de 60% para el conjunto de entrenamiento, 20% para el conjunto de cros-validación y 20% para el conjunto de test.

Cuando realicemos los entrenamientos de las redes, hablaremos de usar diferentes particiones para los datos. Con esto lo que queremos decir es que los porcentajes de ejemplos que tomaremos para cada conjunto de datos serán como se indicó antes, pero los ejemplos que asociaremos a cada conjunto se escogerán aleatoriamente entre el total de ejemplos de la base de datos.

### 2.3.2. Elección de la arquitectura

Necesitamos escoger la arquitectura de red que vamos a utilizar en cada problema. Utilizaremos una arquitectura Multilayer Feedforward con una sola capa oculta.

El número de unidades de la capa de entrada y el de la capa de salida lo determinará el problema concreto. Nos queda determinar el número de unidades en la capa oculta.

Tendremos que distinguir a la hora de entrenar las redes entre los métodos que codifican las entradas desconocidas con aritmética de intervalos y los que no utilizan intervalos. Básicamente, la diferencia radica en que en el primer caso, las entradas se codifican mediante un intervalo, y se utiliza como algoritmo de aprendizaje Backpropagation generalizado a intervalos, y en el segundo caso, no se usan intervalos sino que las entradas desconocidas se codifican según indique cada método concreto, y el algoritmo de entrenamiento será Backpropagation.

Para escoger el número de unidades ocultas de las redes utilizaremos el procedimiento de *prueba y error*. Para ello, partiremos de varias arquitecturas con diferente número de unidades en la capa oculta.

Entrenamos dichas redes con el algoritmo Backpropagation. Como ya hemos dicho antes, en los métodos de aritmética de intervalos usaremos para entrenar los algoritmos Backpropagation generalizados a intervalos que describimos anteriormente, y en el resto de métodos Backpropagation.

Durante el entrenamiento utilizaremos dos conjuntos de datos: el de entrenamiento propiamente dicho (60% de datos) y el de cros-validación (20 % de datos).

### 2.3. METODOLOGÍA

Si durante el proceso de entrenamiento alguna de las redes no converge se repite el entrenamiento utilizando diferente inicialización de la red, es decir, diferente inicialización de los pesos de la red.

Como resultado del entrenamiento de cada red analizaremos el error cuadrático medio, ECM, y el porcentaje de aciertos en la clasificación, POR.

Definimos el ECM como la suma para todos los patrones de entrenamiento del error  $E_p$  utilizado en el algoritmo de aprendizaje dividido por el número total de patrones y el número de neuronas de salida de la red:

$$ECM = \frac{1}{N \cdot N_{salida}} \cdot \sum_{i=1}^N E_p \quad (2.85)$$

El porcentaje de aciertos en la clasificación se calcula sumando el número de patrones de entrada bien clasificados y dividiendo por el número total de patrones. Para determinar cuántos patrones están bien clasificados, se estudian para un patrón de entrada, las salidas (valores de las neuronas de salida) de la red que éste proporciona. Si el máximo valor de salida coincide con el objetivo de dicho patrón, podemos decir que éste ha sido clasificado correctamente: hay un acierto. Así:

$$POR = \frac{\sum \text{aciertos}}{N} \quad (2.86)$$

Una vez acabado el entrenamiento hay que determinar el punto de convergencia de la red (iteración donde se alcanza el mínimo del error).

El método que utilizaremos durante toda la Tesis para la selección del punto de convergencia es *cross-validation*.

Dicho método consiste en medir el Error cuadrático medio (ECMCROS) y el Porcentaje de correcta clasificación (PORCROS) que se obtienen durante el entrenamiento utilizando el ***conjunto de datos para cross-validation***. El punto de convergencia será aquel que proporcione un valor de ECM-CROS mínimo.

Para finalizar el procedimiento de ***prueba y error*** falta, a partir de los resultados obtenidos para las diferentes redes, escoger aquella con un número de unidades oculta adecuado. Dicha red será aquella que proporcione el mayor valor para el porcentaje de acierto de *cross-validation*, es decir, PORCROS máximo.

En el entrenamiento de las redes se han escogido los siguiente parámetros: *Intervalo de inicialización de pesos*=0.1, *Momento*=0.05, *Paso de adaptación*=0.1.

Entendemos por intervalo de inicialización de pesos, el valor que utilizamos para definir el intervalo dentro del cual se escogerán aleatoriamente los valores iniciales de los pesos

El proceso de selección de la arquitectura que hemos descrito hasta aquí se realizará para cada uno de los problemas, es decir, para los problemas generados con los diferentes porcentajes de entradas desconocidas 0%, 5%, 10%, 20%, 30%, 40%, y para los casos de aritmética de intervalos, codificando en este caso las entradas desconocidas mediante un intervalo en lugar de mediante un valor. Al final tendremos las arquitecturas de redes idóneas para cada caso.

En la Tabla 1 tenemos la arquitectura de las redes obtenidas para diferentes bases de datos con un cierto porcentaje de entradas desconocidas (una red diferente por cada porcentaje y problema) para el caso de la utilización del método descrito en el apartado 2.2.1 (método 1).

**Tabla 1. Arquitectura de las redes con entradas desconocidas.**

PROBLEMA	Ninput	Nsalida	Nhidden(% de entradas desconocidas)					
			0%	5%	10%	20%	30%	40%
ABALONE	8	3	17	18	17	18	15	15
BALANCE	4	3	6	12	18	25	25	13
BANDS	39	2	13	13	16	14	12	13
BUPA	6	2	8	15	26	27	24	18
CREDITO	15	2	9	15	14	9	9	18
DISPLAY1	7	10	15	15	23	30	35	47
DISPLAY2	14	10	21	33	27	32	27	39
GLASS	10	6	6	12	13	17	16	15
HEAR	13	2	6	9	21	21	20	19
IMAGE	19	7	13	15	20	15	18	33
MUSHROOM	22	2	10	6	16	10	15	24
MOK1	6	2	5	10	14	17	24	24
MOK2	6	2	14	19	20	33	25	18
MOK3	6	2	2	6	9	15	18	15
MUSK	166	2	7	9	9	9	12	10
PIMA	8	2	15	14	28	18	24	29
THYROIDES	21	3	3	1	1	1	3	1
VOTING	16	2	13	10	6	9	8	12
WDBC	30	2	2	3	10	11	21	21

### 2.3. METODOLOGÍA

Por ejemplo, la red que tiene un porcentaje de 5% de entradas desconocidas en la base de datos ABALONE tendrá ocho unidades en la capa de entrada (Ninput), tres unidades en la capa de salida (Nsalida) y dieciocho unidades en la capa oculta (Nhidden con 5%).

En la Tabla 2, se muestra el número de unidades de la capa oculta de cada una de las redes seleccionadas para aritmética de intervalos usando en el entrenamiento el algoritmo de Hernández (método 2), y el algoritmo de Ishibuchi (método 3). El número de unidades en la capa de entrada y en la capa de salida de las redes es el mismo que figuraba en la Tabla 1.

**Tabla 2. Arquitectura de redes con entradas desconocidas y codificación con aritmética de intervalos para los métodos 2 y 3.**

PROBLEMA	Nhidden para Aritmética Ishibuchi (normal)					Nhidden para Aritmética Hernández				
	5%	10%	20%	30%	40%	5%	10%	20%	30%	40%
ABALONE	2	1	1	1	1	3	3	2	1	1
BALANCE	16	4	12	1	1	6	12	3	2	6
BANDS	6	1	1	1	1	22	13	13	2	1
BUPA	1	1	1	1	1	3	2	1	1	1
CREDITO	3	1	1	1	1	4	5	1	1	1
DISPLAY1	11	21	1	2	1	8	13	5	5	6
DISPLAY2	24	18	1	1	1	22	28	8	5	5
GLASS	6	9	15	1	1	7	8	6	19	6
HEAR	15	12	6	1	1	15	8	4	3	3
IMAGE	18	17	9	1	1	16	16	17	10	14
MUSHROOM	12	12	7	1	1	7	7	3	3	2
MOK1	1	1	1	1	1	3	9	5	1	1
MOK2	2	1	1	1	1	12	11	1	1	1
MOK3	5	3	2	1	1	5	3	2	3	7
MUSK	16	4	1	1	1	15	6	3	3	2
PIMA	2	1	1	1	1	12	3	3	3	1
THYROIDES	3	1	1	1	1	3	1	1	1	1
VOTING	7	8	5	3	1	7	10	8	3	3
WDBC	3	7	1	1	1	10	9	10	8	7

Tendremos una arquitectura para cada uno de los diferentes porcentajes de entradas desconocidas, Tabla 1, y lo mismo para cada uno de los 2 métodos de aritméticas de intervalos, Tabla 2.

### 2.3.3. Proceso de entrenamiento y test

El siguiente paso será realizar el entrenamiento de seis redes de cada una de las escogidas en el apartado anterior. Escogeremos para cada una de las seis redes diferente inicialización de pesos y partición de datos. De esta forma, se pretende que el resultado obtenido sea independiente de la inicialización de la red, ya que se calculará al final la media de la eficiencia de las seis redes.

Realizaremos el entrenamiento para cada uno de los problemas y con cada conjunto de entrenamiento (porcentajes 0%, 5%, 10%, 20%, 30%, 40%).

Para cada red se escoge el punto de convergencia utilizando el método de cros-validación descrito en el apartado anterior. El punto de convergencia nos proporcionará la arquitectura final de la red (valor de los pesos). Es decir, tendremos el resultado de la red entrenada (números de unidades de las diferentes capas y pesos) que hay que usar para resolver cada problema.

Conocida la red entrenada (estructura y pesos), para medir la eficiencia de los diferentes métodos realizamos el test de las mismas para cada método y con los diferentes conjuntos de test (diferentes porcentajes de entradas desconocidas). Esto se repite para las seis redes y se promedia el resultado obteniendo un error. Finalmente, obtenemos el porcentaje medio de correcta clasificación en el test con un error.

Para calcular dicho resultado se evalúa el valor medio de los porcentajes, la desviación estándar y a partir de esta el valor del error, según la ecuación (2.87).

$$\begin{aligned}
 PORMEDIO &= \frac{\sum POR}{Nredes} & (2.87) \\
 Desv &= \frac{\sqrt{\sum (POR - PORMEDIO)^2}}{\sqrt{Nredes - 1}} \\
 Error &= \frac{Desv}{\sqrt{Nredes}}
 \end{aligned}$$

De los resultados obtenidos podemos analizar cómo afecta en un método concreto, la introducción de entradas desconocidas en los conjuntos de entrenamiento y/o test. Después podemos comparar los resultados obtenidos y ver que método funciona mejor.

## 2.4. Resultados Experimentales

En este capítulo presentamos una comparación de los 5 métodos de codificación de entradas desconocidas.

Para el método 4 (descrito en el apartado 2.2.3) de codificación mediante probabilidades, sólo se consideran los porcentajes de entradas desconocidas en el conjunto de test, ya que el método no exige un entrenamiento previo con entradas desconocidas.

El método 5 de codificación (descrito en el apartado 2.2.4) como vimos, sólo es aplicable al caso de entradas binarias, por lo cual su uso es bastante restringido. Este método lo hemos aplicado solo a los problemas *DISPLAY1* y *DISPLAY2* y utilizando entradas desconocidas solo en el conjunto de test ya que tampoco requiere un entrenamiento con entradas desconocidas.

En la comparación se han utilizado un total de 19 problemas y se ha medido la eficiencia para diferentes porcentajes de entradas desconocidas en los problemas: 0%, 5%, 10%, 20%, 30% y 40%.

Veremos que los métodos basados en la generalización de Backpropagation a aritmética de intervalos y con codificación durante el entrenamiento de las entradas desconocidas en el intervalo  $[0,1]$  dan el mejor resultado.

Denominamos a los métodos según el orden en que los hemos descrito:

**Método 1:** Codificación mediante un valor.

**Método 2:** Aritmética de intervalos con el algoritmo de Hernández.

**Método 3:** Aritmética de intervalos con el algoritmo de Ishibuchi.

**Método 4:** Codificación mediante probabilidades.

**Método 5:** Codificación de entradas binarias.

Usaremos, como ya dijimos antes, 19 problemas en los cuales se ha introducido en el total de datos, los siguientes porcentajes de entradas desconocidas: 0 (problema original sin entradas desconocidas), 5, 10, 20, 30 y 40. Obtendremos a partir de aquí 114 problemas. Cada conjunto de datos de un problema se divide en un conjunto de datos para el entrenamiento (60% del total de datos), un conjunto de datos para cros-validación (20%) y un conjunto de datos para el test (20%).



Hemos entrenado seis redes diferentes usando Backpropagation para cada uno de los problemas, con cada uno de los conjuntos de datos de entrenamiento (0, 5, 10, 20, 30 y 40 % de entradas desconocidas) y utilizando la codificación por valor (método 1). Hemos calculado el punto de convergencia usando cros-validación, y realizado el test de dichas redes, con los diferentes conjuntos de datos de test, midiendo el porcentaje de correcta clasificación según la regla de clasificación que escoge la clase con mejor salida como la correcta.

También hemos realizado el test de las redes entrenadas, utilizando la codificación de entradas desconocidas en el conjunto de test dada por los métodos 4 y 5.

Por último, hemos entrenado y testeado 6 redes utilizando la extensión de Backpropagation a aritmética de intervalos con el algoritmo de aprendizaje descrito por Hernández (método 2) y otras 6 con el algoritmo utilizado por Ishibuchi (método 3).

En todos los casos, utilizamos la media de los resultados de las 6 redes entrenadas de forma que los resultados finales son calculados con un error, es decir, no dependerán de la inicialización de los pesos, datos, etc.

En todos los casos la elección de punto de convergencia de la red se realizó utilizando cros-validación.

En las Tablas de la 5 a la 23 del apéndice I de la Tesis Doctoral podemos encontrar los resultados para los métodos 1, 4 y 5. Cada tabla representa el resultado del proceso de entrenamiento y test de una cierta base de datos. Por ejemplo, en la Tabla 5 podemos ver los resultados para la base de datos ABALONE.

En las Tablas de la 24 a la 66 del apéndice I de la Tesis se pueden encontrar los resultados obtenidos para los métodos que utilizan aritmética de intervalos (métodos 2 y 3).

En el test de aritmética de intervalos se han usado 5 reglas de clasificación diferentes que se describieron en las ecuaciones (2.68), (2.69), (2.70), (2.71), (2.72). De ellas, las de porcentaje POR4 y POR5 son las más restrictivas y proporcionan un porcentaje de aciertos muy inferior.

En cuanto a las demás, POR1, POR2 y POR3 los resultados son muy parecidos, excepto para las bases de datos BALANCE, DISPLAY1 y GLASS en donde el porcentaje POR1 ofrece los mejores resultados. Por ello, nos centramos en esta primera regla de clasificación. En ella se utiliza el límite superior del

## 2.4. RESULTADOS EXPERIMENTALES

intervalo de salida y se considera que la entrada es clasificada en la clase cuyo límite superior es mayor:

$$[a,b]<[c,d] \Leftrightarrow b<d$$

En las Tablas de la 24 a la 42 del apéndice I de la Tesis pueden verse los resultados obtenidos utilizando el algoritmo de entrenamiento de Hernández y en las Tablas de la 43 a la 61 los resultados de aplicar el algoritmo de Ishibuchi.

Se dan los resultados para los cinco criterios de clasificación nombrados como POR1, POR2, POR3, POR4 y POR5.

Cada tabla corresponde a una base de datos diferente y en ella tenemos los 6 posibles porcentajes de entradas desconocidas usadas en el entrenamiento (6 problemas distintos) y los 6 posibles conjuntos de test.

Por ejemplo, para la Tabla 24 se dan los resultados para el problema ABALONE para los cinco criterios de clasificación. Para cada criterio, por ejemplo POR1, una cierta fila representa el entrenamiento de la red con un cierto porcentaje de entradas desconocidas. Para esa fila, cada columna corresponde al test realizado con los 6 posibles conjuntos de test (0, 5, 10, 20, 30 y 40 % de entradas desconocidas). De esa forma podemos ver cuál es la influencia de introducir diferentes porcentajes de entradas desconocidas tanto en los conjuntos de entrenamiento como en los de test.

En general, la primera conclusión que se puede obtener al mirar estos resultados, es que la eficiencia de las redes disminuye al aumentar el porcentaje de entradas desconocidas, tanto en el entrenamiento como en el test y que la rapidez del deterioro de la eficiencia depende en gran medida de la base de datos que estemos utilizando.

Esta situación ocurre también en el caso del 0% de entradas desconocidas en el test, es decir, en la respuesta de la red entrenada ante datos sin entradas desconocidas. Conforme se aumenta el porcentaje de entradas desconocidas en el entrenamiento se observa un decremento de la eficiencia de la respuesta de la red ante entradas normales (conocidas). Ello hace que debamos limitar el porcentaje de entradas desconocidas a utilizar en el entrenamiento. Consideramos que de los valores probados el más idóneo es el del 5%.

En las Tablas de la 62 a la 66 del apéndice I de la Tesis se da una comparación entre los métodos 2 y 3. Se compara la eficiencia de los métodos 2 y 3 para cada una de las 5 reglas de clasificación y para el caso en que los porcentajes de entradas desconocidas en el entrenamiento y test sean los mismos.

En principio, el uso del mismo porcentaje parece lógico. Además, es lo que sugieren algunos artículos de investigación publicados [17], en donde se muestra que las distribuciones del conjunto de entrenamiento y test deben ser lo más parecidas posible. Más adelante veremos un estudio más detallado de estos resultados.

Analizaremos a continuación los resultados obtenidos para cada método y finalmente compararemos unos métodos con otros.

En primer lugar, a partir de las Tablas de la 5 a 23, podemos observar que el método 1 presenta alta variabilidad en el test y la eficiencia depende del problema particular. En general, la eficiencia decrece al aumentar el número de entradas desconocidas, tanto en el test como en el entrenamiento. En algunos problemas este decrecimiento es lento pero en otro existe un porcentaje a partir del cual el método funciona muy mal.

Como ejemplo, tomemos los problemas PIMA y DISPLAY1.

A partir de las Tablas 20 (problema PIMA) y 10 (problema DISPLAY1) construimos las figuras 4 y 5 respectivamente, en las cuales representamos el porcentaje de correcta clasificación para las diferentes redes entrenadas con el método 1 (5, 10, 20, 30 y 40 % de entradas desconocidas) y testeado para cada uno de los diferentes conjuntos de test (5, 10, 20, 30 y 40 % de entradas desconocidas).

En la figura 4, vemos que para el problema PIMA utilizando el método 1, la eficiencia decrece suavemente al aumentar el porcentaje de entradas desconocidas en el entrenamiento y en el test, aunque este último cambio no es tan evidente (es aun más lento).

Sin embargo, para el problema DISPLAY1, figura 5, vemos que al pasar de 0% a 5% de entradas desconocidas en el test hay un cambio brusco, el método deja de funcionar. También se observa que el porcentaje en el entrenamiento prácticamente no afecta el resultado.

Llegamos pues a la conclusión que ya vimos, de que el método 1 presenta una alta variabilidad en el test al variar el porcentaje de entradas desconocidas y la eficiencia depende en gran medida del problema particular.

En cuanto a los métodos 2 y 3 podemos ver en algunos problemas que existe un porcentaje de entradas desconocidas en el entrenamiento a partir del cual el método deja de funcionar (la eficiencia disminuye súbitamente). El valor de dicho

## 2.4. RESULTADOS EXPERIMENTALES

porcentaje depende del problema y podemos observar que existen 3 tipos de situaciones.

Hay problemas en los cuales el método 2 funciona y el método 3 tiene un porcentaje límite en el entrenamiento. Por ejemplo, en la figura 6 vemos que el método 2 funciona bien para el problema MUSHROOM y en la figura 7 vemos que el método 3 no funciona (baja súbitamente su eficiencia) para el mismo problema a partir de un 30% de entradas desconocidas en el entrenamiento. En las figuras 8 y 9 tenemos la misma situación para el problema CREDITO. Ahora el método 3 deja de funcionar a partir de un 40% de entradas desconocidas.

Otro caso es, que tanto el método 2 como el método 3 tienen un límite en el número de entradas desconocidas en el entrenamiento a partir del cual dejan de funcionar. Las figuras 10 y 11 reflejan esta situación para el problema PIMA. El método 2 no funciona a partir del 20% y el método 3 no funciona a partir de un 10% de entradas desconocidas en el entrenamiento.

El último caso, es que los dos métodos funcionan bien. En la figura 12 y 13 podemos verlo para el problema VOTE. Hay un decrecimiento de la eficiencia al aumentar el porcentaje de entradas desconocidas pero los métodos funcionan. Posteriormente veremos una comparación más exhaustiva entre los métodos 2 y 3.

Comparemos ahora los diferentes métodos entre sí. Comparamos primero los 4 primeros métodos.

La aplicación de entradas desconocidas, como vimos al principio del capítulo para algunos métodos, solo tiene sentido si también se utilizan entradas desconocidas en el proceso de entrenamiento. Por ello, para comparar los métodos entre sí hemos escogido un porcentaje de entradas desconocidas en el entrenamiento del 5%. Hemos elegido este porcentaje tal y como comentamos antes, porque valores superiores suelen producir un empeoramiento de la respuesta ante entradas conocidas, deteriorando el funcionamiento normal de la red.

Representamos como antes para el método 1 (con 5% de entradas desconocidas en el entrenamiento), el método 2 y 3 (con 5% de entradas desconocidas en el entrenamiento) y el método 4, los diferentes valores de la eficiencia en función del porcentaje de entradas desconocidas en el test.

En la figura 14 y 15 podemos ver la comparación de los 4 métodos para los problemas PIMA e IMAGEN respectivamente.

Vemos que el método 4 funciona peor que los demás. Nosotros pensamos que el mal funcionamiento del método de probabilidad se puede deber a que las aproximaciones que hacen sus autores a la hora de calcular las integrales no son muy adecuadas.

En las figuras 16 y 17 se ha representado la eficiencia de los 5 métodos para los problemas DISPLAY1 y DISPLAY2 respectivamente y en las mismas condiciones de antes (con 5% de entradas desconocidas en el entrenamiento).

Vemos que el método 4 sigue funcionando mal y que el método 5 funciona bastante bien. Sin embargo a pesar de funcionar bien, el método 5 presenta la desventaja de que solo puede aplicarse a problemas con entradas binarias por lo que su uso es bastante restringido.

Para estas dos bases de datos vemos que el método 1 funciona muy mal. Ya dijimos que la eficiencia de este método dependía mucho de la base de datos a la que se aplicará.

Como conclusión, podemos decir que el método 1 da mejor eficiencia que el método 4 y que los métodos que utilizan aritmética de intervalos son los que mejor resuelven la mayoría de los problemas.

Por ello proponemos a continuación una comparación más exhaustiva de los mismos.

### **Comparación entre métodos de aritmética de intervalos.**

Como ya vimos, la diferencia básica entre los métodos 2 (de Hernández) y 3 (de Ishibuchi) reside en que en el algoritmo de aprendizaje las funciones de error propuestas son diferentes, y por lo tanto también lo son los algoritmos de aprendizaje.

Nos referimos al método de Hernández como “Algoritmo de Hernández” y al método de Ishibuchi lo denotaremos como “Algoritmo de Ishibuchi”.

Como vimos anteriormente en el test de aritmética de intervalos se han usado 5 reglas de clasificación diferentes que se describen en las ecuaciones (2.68), (2.69), (2.70), (2.71), (2.72).

Usaremos los 19 problemas que teníamos. Para cada problema, como dijimos, se generaron 5 nuevos problemas con los diferentes porcentajes de entradas desconocidas: 5, 10, 20, 30, 40. Tendremos 95 problemas a los que aplicar estos métodos.

2.4. RESULTADOS EXPERIMENTALES

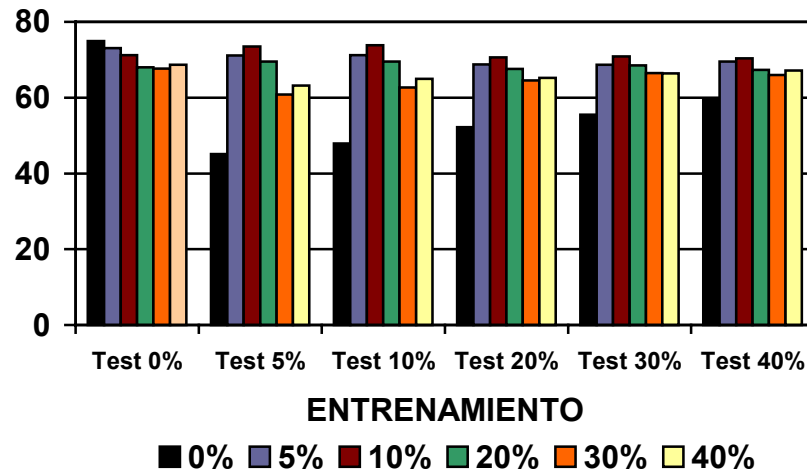


Fig.4. Porcentaje de correcta clasificación en función del % de entradas desconocidas para el problema PIMA. Método 1.

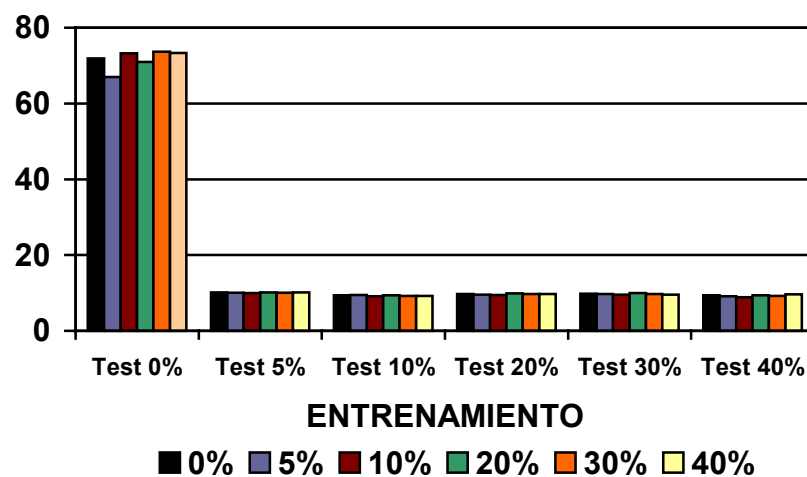


Fig.5. Porcentaje de correcta clasificación en función del % de entradas desconocidas para el problema DISPLAY1. Método 1.

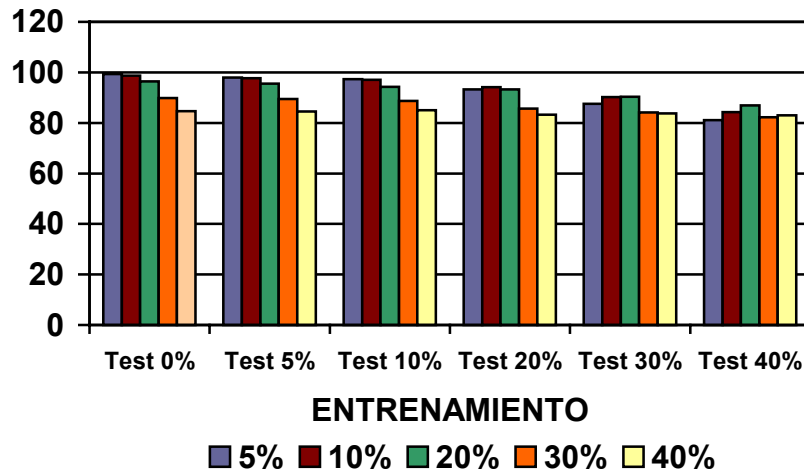


Fig.6. Porcentaje de correcta clasificación en función del % de entradas desconocidas para el problema MUSHROOM. Método 2.

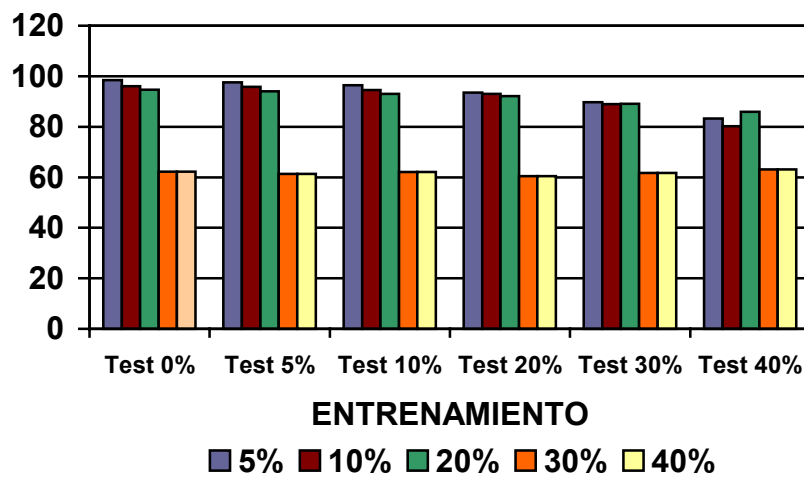
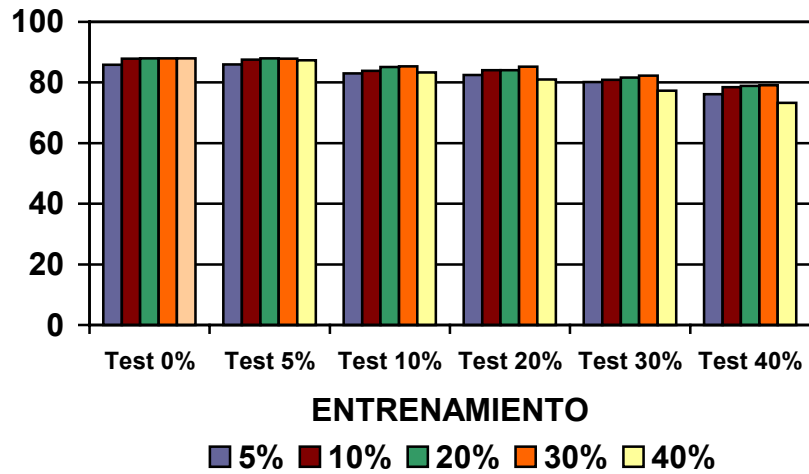
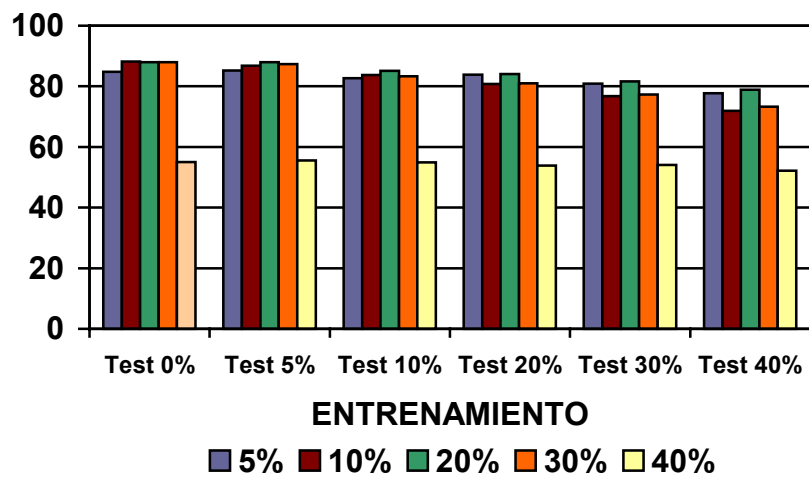


Fig.7. Porcentaje de correcta clasificación en función del % de entradas desconocidas para el problemas MUSHROOM. Método 3.

## 2.4. RESULTADOS EXPERIMENTALES



*Fig.8. Porcentaje de correcta clasificación en función del % de entradas desconocidas para el problema CREDITO. Método 2.*



*Fig.9. Porcentaje de correcta clasificación en función del % de entradas desconocidas para el problema CREDITO. Método 3.*



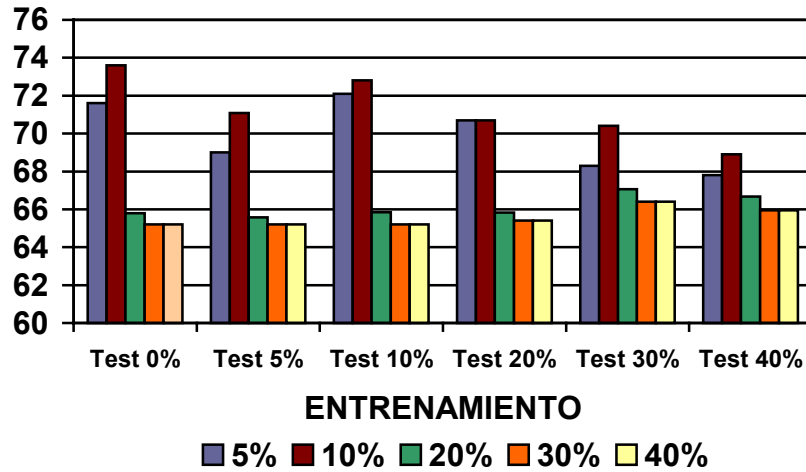


Fig.10. Porcentaje de correcta clasificación en función del % de entradas desconocidas para el problema PIMA. Método 2.

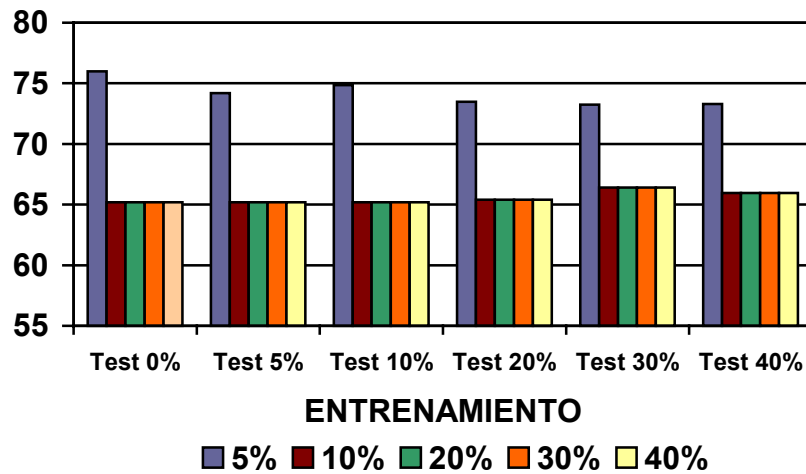


Fig.11. Porcentaje de correcta clasificación en función del % de entradas desconocidas para el problema PIMA. Método 3.

2.4. RESULTADOS EXPERIMENTALES

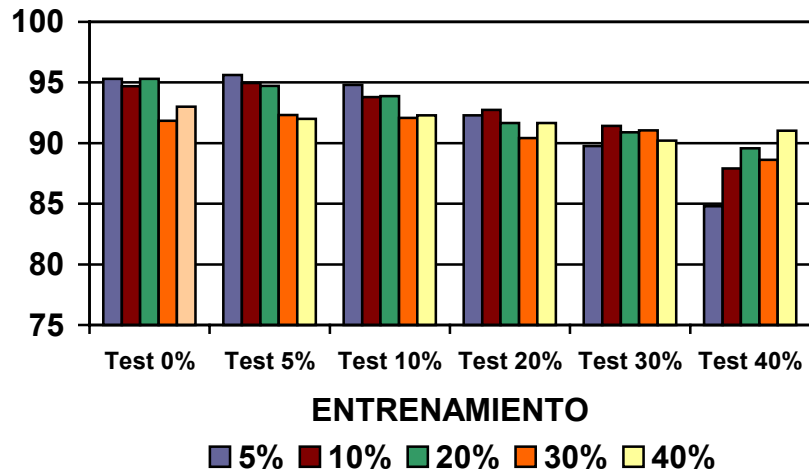


Fig.12. Porcentaje de correcta clasificación en función del % de entradas desconocidas para el problema VOTE. Método 2.

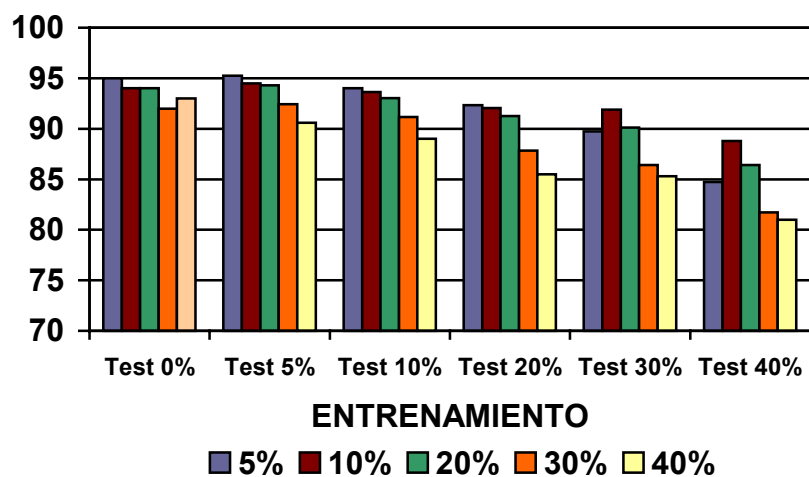
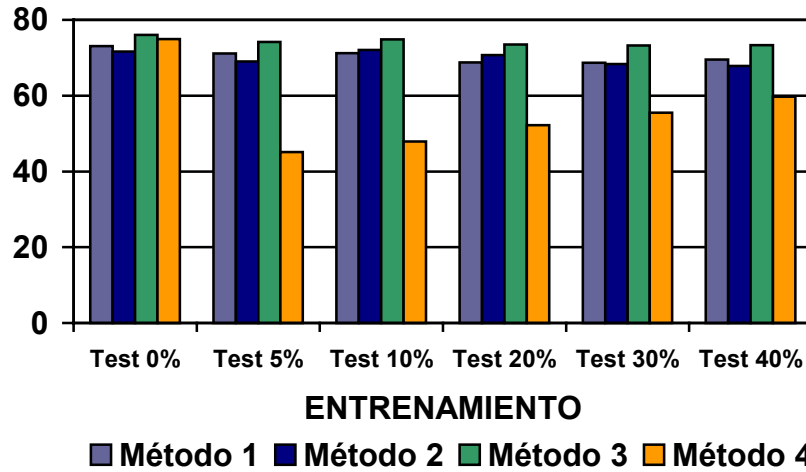
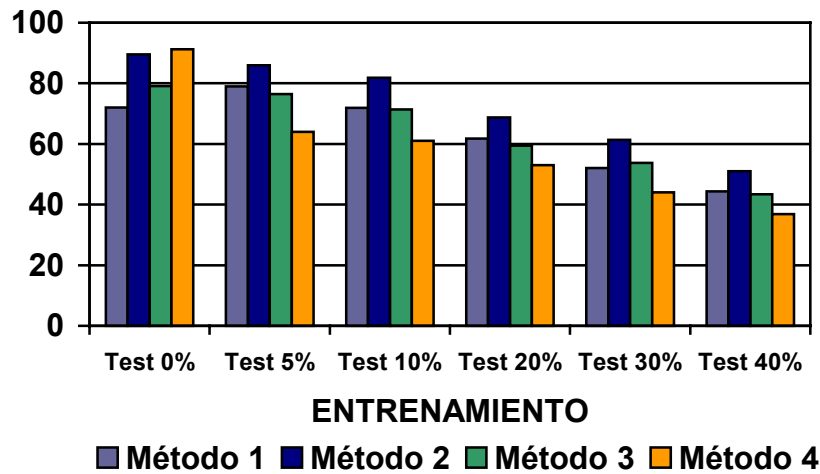


Fig.13. Porcentaje de correcta clasificación en función del % de entradas desconocidas para el problema VOTE. Método 3.



*Fig.14. Porcentaje de correcta clasificación en función del % de entradas desconocidas en el test (5% en entrenamiento) para el problema PIMA. Método 1, 2, 3 y 4.*



*Fig.15. Porcentaje de correcta clasificación en función del % de entradas desconocidas en el test (5% en entrenamiento) para el problema IMAGEN. Método 1, 2, 3 y 4.*

2.4. RESULTADOS EXPERIMENTALES

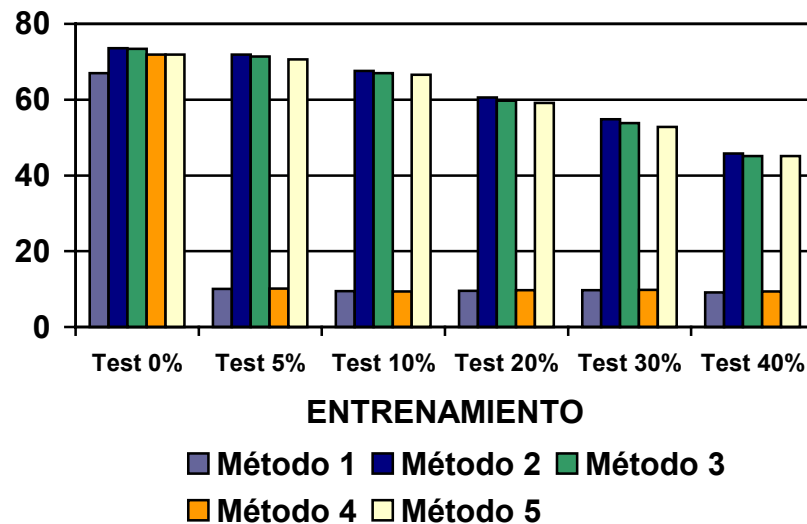


Fig.16. Porcentaje de correcta clasificación en función del % de entradas desconocidas en el test (5% en entrenamiento) para el problema DISPLAY1. Método 1, 2, 3, 4 y 5.

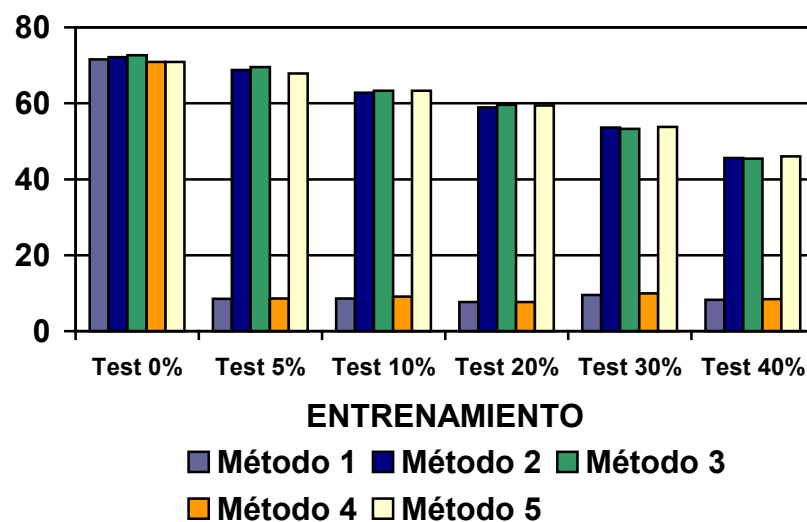


Fig.17. Porcentaje de correcta clasificación en función del % de entradas desconocidas en el test (5% en entrenamiento) para el problema DISPLAY2. Método 1, 2, 3, 4 y 5.

Para cada problema se entrenaron, como dijimos antes, 6 redes con diferente inicialización de pesos y se promediaron los resultados obteniendo un error.

Se estudio la eficiencia obtenida utilizando en el entrenamiento el “Algoritmo de Hernández” y el “Algoritmo de Ishibuchi”.

En las tablas de la 62 a la 66 del apéndice I de la Tesis Doctoral se compara la eficiencia de los algoritmos de Hernández y Ishibuchi para el caso de igual porcentaje de entradas desconocidas en el entrenamiento y test, y para las diferentes reglas de clasificación.

Para una cierta regla de clasificación, por ejemplo POR1 (tabla 62) y una cierta base de datos, se dan los diferentes porcentajes de entradas desconocidas y la eficiencia obtenida para los algoritmos de Hernández y Ishibuchi.

Para poder comparar los resultados, agruparemos los mismos en diferentes categorías según la eficiencia de un algoritmo sea mejor o peor que la del otro. Realizamos la siguiente clasificación:

- 1:** Algoritmo de Hernández igual que el de Ishibuchi.
- 2:** Algoritmo de Hernández mejor que el de Ishibuchi en menos del 2%.
- 3:** Algoritmo de Hernández mejor que el de Ishibuchi entre 2 y 10%.
- 4:** Algoritmo de Hernández mejor que el de Ishibuchi en más del 10%.
- 5:** Algoritmo de Hernández peor que el de Ishibuchi en menos del 2%.
- 6:** Algoritmo de Hernández peor que el de Ishibuchi entre 2 y 10%.
- 7:** Algoritmo de Hernández peor que el de Ishibuchi en más del 10%.

Por ejemplo, para el problema ABALONE y la regla de clasificación POR1, un 5% quiere decir que se ha usado 5% de entradas desconocidas en el conjunto de entrenamiento y en el conjunto de test y se da la medida de la eficiencia del algoritmo de Hernández, 57.6, y la del algoritmo de Ishibuchi, 54.9. Como la eficiencia del algoritmo de Hernández es mayor que la de Ishibuchi en más de un 2%, clasificamos este resultado en la categoría 3.

Esta operación se realizó con todos los problemas y reglas de clasificación y una vez obtenidas todas las categorías se procedió a contar los resultados.

## 2.4. RESULTADOS EXPERIMENTALES

En la Tabla 3, para cada una de las reglas de clasificación hemos contado cuantos problemas hay en cada categoría (hay 95 problemas en total).

**Tabla 3. Número de problemas en cada categoría para las 5 reglas.**

	Categoría						
	1	2	3	4	5	6	7
<b>POR1</b>	17	23	17	34	2	2	0
<b>POR2</b>	16	22	17	34	4	2	0
<b>POR3</b>	16	23	20	29	4	3	0
<b>POR4</b>	17	10	10	8	21	22	7
<b>POR5</b>	20	7	7	16	19	20	6

Estudiando esta clasificación podemos ver qué algoritmo funciona mejor y en que casos.

A partir de la Tabla 3, hemos dibujado un histograma para cada regla de clasificación, figura 18, en el cual representamos como barra central el número de problemas en los cuales el algoritmo de Hernández funciona igual que el de Ishibuchi (categoría 1), hacia la derecha las barras representan que el algoritmo de Hernández funciona mejor (categorías 2, 3 y 4) y hacia la izquierda representan que funciona peor (categorías 5, 6 y 7).

Se observa que el algoritmo de Hernández funciona mejor en la mayoría de los casos para las reglas de clasificación POR1, POR2 y POR3 y sólo en las reglas POR4 y POR5, funciona mejor Ishibuchi.

Como ya se comentó anteriormente, las reglas de clasificación POR4 y POR5 son demasiado estrictas y su porcentaje de aciertos es muy inferior a las demás. Pensamos pues que no es muy interesante su uso.

En la Tabla 4 y la figura 19 se resumen el total de problemas que funcionan mejor, peor o igual.

**Tabla 4. Número de problemas finales que funcionan mejor, peor o igual.**

	Mejor Hernández	Mejor Ishibuchi	Iguales
<b>POR1</b>	74	4	17
<b>POR2</b>	73	6	16
<b>POR3</b>	72	7	16
<b>POR4</b>	28	50	17
<b>POR5</b>	30	45	20

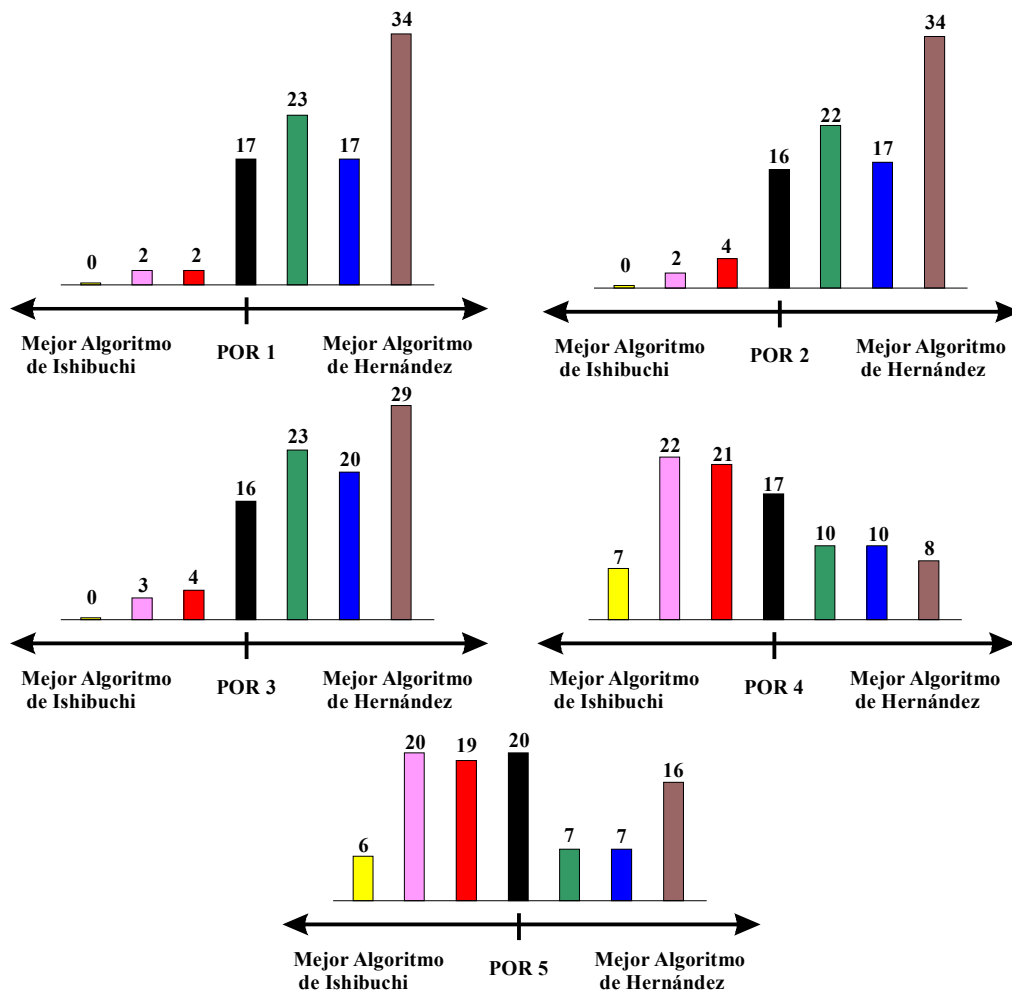


Fig. 18. Comparación entre el algoritmo de Hernández y el algoritmo de Ishibuchi utilizando cinco criterios diferentes de clasificación.

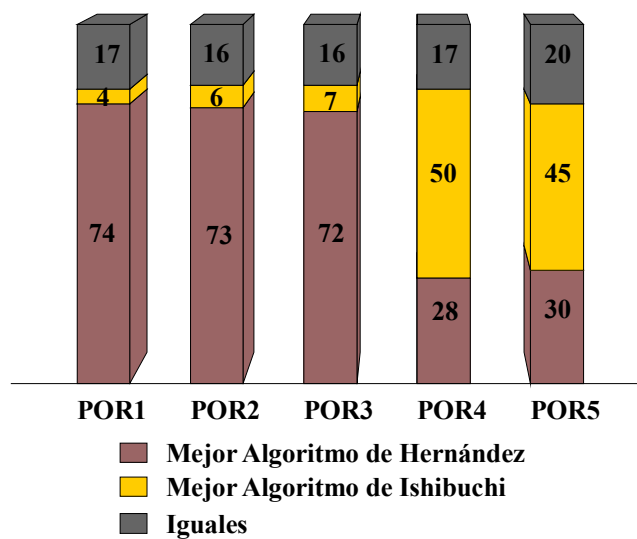


Fig. 19. Comparación entre el Método 2 y el Método 3.

#### *2.4. RESULTADOS EXPERIMENTALES*

Para el primer criterio de clasificación POR1, que anteriormente comentamos que era el más idóneo, vemos que hay 74 problemas en los que funciona mejor el algoritmo de Hernández, 4 en los que es mejor el algoritmo de Ishibuchi y 17 en los que funcionan igual.

Podemos concluir que si consideramos el primer criterio POR1 como el usual y más aconsejable, el algoritmo de Hernández funciona mejor que el de Ishibuchi.

Como conclusión final, podemos recomendar el uso del algoritmo de Hernández (método 2) como el más eficiente para la codificación de entradas desconocidas.



## 2.5. Conclusiones

En este capítulo hemos presentado una comparación de 5 métodos de codificación de entradas desconocidas.

La comparación se realizó partiendo de 19 problemas en los cuales se introdujeron 6 diferentes porcentajes de entradas desconocidas, 0, 5, 10, 20, 30 y 40 %, obteniendo así un total de 95 problemas sobre los que medir la eficiencia de los métodos.

La comparación final entre diversos métodos nos llevó a las siguientes conclusiones.

Si analizamos los resultados para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test podemos ver la influencia de los mismos sobre los diferentes métodos.

Así, para el método 1 se observa que funciona en general bien, pero presenta alta variabilidad en el test al variar el porcentaje de entradas desconocidas. La eficiencia dependerá en gran medida del problema particular. En algún problema hemos encontrado un porcentaje de entradas desconocidas en el entrenamiento a partir del cual el método deja de funcionar bien.

En los métodos con aritmética de intervalos hemos encontrado que en general el método 2 (Hernández) funciona mejor que el 3 (Ishibuchi) para las 3 primeras reglas de clasificación que hemos utilizado. Las otras 2 reglas son muy restrictivas (pueden quedar ejemplos sin clasificar) por lo que creemos que las conclusiones que se puedan sacar de su funcionamiento no son muy relevantes.

Para los métodos con aritmética de intervalos, también hemos encontrado 3 tipos de situaciones, una en la que ambos métodos funcionan bien (mejor el 2), otra en la que funciona bien el método 2 y en el 3 hay un porcentaje de entradas desconocidas en el entrenamiento a partir del cual deja de funcionar, y la última en la cual ambos métodos poseen un porcentaje de entradas desconocidas en el entrenamiento, a partir del cual funcionan claramente peor.

También hemos comparado los cuatro primeros métodos entre sí. Para ello hemos usado el caso de 5% de entradas desconocidas en el entrenamiento, ya que es aconsejable usar un pequeño porcentaje de entradas desconocidas, si no se quiere que haya un deterioro de la eficiencia de la red ante entradas conocidas.

## 2.5. CONCLUSIONES

Hemos observado que el método 4 funciona claramente peor que los demás. Nosotros pensamos que se podría mejorar su eficiencia modificando la aproximación realizada para el cálculo de las integrales.

Los métodos 2 y 3 son de nuevo los que mejor funcionan.

Finalmente, hemos comparados los 5 métodos para los problemas con entradas binarias ( $DISPLAY1$  Y  $DISPLAY2$ ) obteniendo los mismos resultados. Para el método 5 hemos obtenido buenos resultados, pero presenta la desventaja de que sólo es aplicable a problemas con entradas binarias por lo que su uso es bastante restringido.

Como conclusión final, podemos decir que el método 1 da mejor eficiencia que el método 4 y que los métodos que utilizan aritmética de intervalos son los que mejor resuelven la mayoría de los problemas, es especial el método 2.

Los resultados de nuestra investigación en este campo fueron publicados en 4 conferencias internacionales, cuyas referencias son [10], [11], [12] y [13].

## 2.6. Bibliografía del capítulo

- [1] Maren, C. Harston, R. Pap. *Handbook of Neural Computing Applications*. Academic Press Inc. 1990.
- [2] Ishibuchi, H., Tanaka, H., “An extension of the BP-Algorithm to Interval Input Vectors -Learning from Numerical Data and Expert’s Knowledge-”. Proceedings of the 1991 International Joint Conference on Neural Networks, Singapore, pp. 1588-1593, 1991.
- [3] Hernández, C., Espí, J., Nakayama, K.,”A generalization of Backpropagation to Interval Arithmetic”. Proceedings of the World Congress on Neural Networks (WCNN’93), vol. 4, pp. 131-134, 1993.
- [4] Hernández C.A., Espí, J., Nakayama, K., Fernández, M., “Interval Arithmetic Backpropagation”. Proceedings of the 1993 International Joint Conference on Neural Networks (IJCNN’93), vol. 1, pp. 375-378, 1993.
- [5] Ishibuchi, H., Miyazaki, A., Kwon, K., Tanaka, H., “Learning from incomplete training data with missing values and medical application”. Proceedings of 1993 IEEE International Conference on Neural Networks (Nagoya), pp. 1871-1874, 1993.
- [6] Ishibuchi, H., Miyazaki, A., Tanaka, H., “Neural-network-based diagnosis systems for incomplete data with missing inputs”. Proceedindgs of 1994 IEEE International Conference on Neural Networks, pp. 3457-3460, 1994.
- [7] Alefeld, G., Herzberger, J., *Introduction to Interval Computations*. Academic Press. New York. 1983.
- [8] Tresp, V., Neuneier, R., Ahmad, S., “Efficient methods for dealing with missing data in supervised learning”. Advances in Neural Information Processing Systems, pp. 689-696, 1995.
- [9] Armitage, W.D., Jien-Chung Lo, “Enhancing the robustness of a Feedforward neural network in the presence of missing data”. Proceedings of IEEE International Conference on Neural Networks, pp. 836-839, 1994.
- [10] Fernández, M., Hernández, C.A., “An experimental evaluation of Different Methods for Handling Missing Inputs”. Proceeding of the International Joint Conference on Neural Networks (IJCNN’98), vol. 2, pp. 1450-1453, 1998.
- [11] Fernández, M., Hernández, C.A., “How to handle unknown Inputs”. Proceeding of the International Computer Symposium (ICS’ 98), vol. B, pp. 272-276, 1998.
- [12] Fernández, M., Hernández, C.A., “On the error function of interval arithmetic Backpropagation”. Proceeding of the European Symposium on Artificial Neural Networks (ESANN’ 98), pp. 135-139, 1998.
- [13] Fernández, M., Hernández, C.A., “A comparison between two interval arithmetic learning algorithms”. Proceedings of the International Joint Conference on Neural Networks (IJCNN’99), pp. 1293-1298, 1999.
- [14] Ahmad, S., Tresp, V., “Classification with missing and uncertain inputs”. Proceedings of 1993 IEEE International Conference on Neural Networks, pp. 1949-1954, San Francisco, California, March 1993.
- [15] Anguita D., Ridella S., Rovetta S., “Limiting the Effects of Weight Errors in Feedforward Networks using Interval Arithmetic”. Proceedings of the International Conference on Neural Networks, vol. 1, pp. 414-417, 1996

## 2.6. BIBLIOGRAFÍA DEL CAPÍTULO

- [16] Haung L., Zhang B., “A Novel-Network-Related Approach for Regression Analysis with Interval Model”. Proceedings of the International Conference on Neural Networks, vol. 1, pp. 611-616, 1996.
- [17] Hernández, C., Espí, J., Nakayama, K.,”Effect of training data distribution on Backpropagation Generalization”. Proceedings of the 1993 IEEE International Symposium on Circuits and Systems (ISCAS’93), pp. 2148-2151, 1993.
- [18] Bounds, D. G., Lloyd, P. J., “A multilayer perceptron for the diagnosis of low back pain”. Proceedings of the IEEE International Conference on Neural Networks, vol. 2, pp. 481-489, 1988.
- [19] Mulsant, B. H., Servan-Sheriber, E., “A connectionist approach to the diagnosis of dementia”, Proceedings of the Twelfth Annual Symposium on Computer Application in Medical Care, pp. 245-250, 1988.
- [20] Scalia, F., Marconi, L., Ridella, S., Arrigo, P., Mansi, C., Mela, G. S., “An example of back propagation: diagnosis of dyspepsia”, Proceedings of the First IEEE International Conference on Artificial Neural Networks, pp. 332-335. 1989.
- [21] Harvey, R. M., “A neural network as a potential means for aiding nursing diagnosis”, Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care, pp. 875-879, 1990.

# Capítulo 3

## MÉTODOS DE SELECCIÓN DE LA INFORMACIÓN DE ENTRADA A LA RED NEURONAL

- 3.1. *Introducción*
- 3.2. *Métodos de selección de entradas basados en ARNMF*
  - 3.2.1. *Primer grupo*
  - 3.2.2. *Segundo grupo*
  - 3.2.3. *Tercer grupo*
  - 3.2.4. *Cuarto grupo*
  - 3.2.5. *Quinto grupo*
  - 3.2.6. *Sexto grupo*
  - 3.2.7. *Séptimo grupo*
- 3.3. *Métodos de selección de entradas basados en ACE*
  - 3.3.1. *Teoría de la Información*
  - 3.3.2. *Distancias dentro y entre clases*
  - 3.3.3. *Lógica difusa*
- 3.4. *Métodos de selección de entradas basados en ARE*
- 3.5. *Algoritmo de aprendizaje mediante gradientes conjugados*
- 3.6. *Uso de Weight Decay en los métodos ARNMF*
- 3.7. *Metodología*
- 3.8. *Resultados Experimentales*
- 3.9. *Aplicación a un problema real: patología de la voz*
- 3.10. *Conclusiones*
- 3.11. *Bibliografía del Capítulo*
- 3.12. *Apéndice del Capítulo*

## 3.1. Introducción

A lo largo de este trabajo se pretende conseguir una arquitectura óptima para la red neuronal Multilayer Feedforward. La arquitectura habitual de la red neuronal Multilayer Feedforward tiene una capa de salida, una capa de entrada y una o más capas ocultas.

En este capítulo consideramos el caso de una única capa oculta. En capítulos posteriores, se estudiará cómo escoger el número de capas y el número de unidades en las mismas.

En adelante nos referiremos a un vector de entrada como ejemplo o patrón de entrenamiento y llamaremos entrada o característica de entrada a cada una de las componentes de dicho vector.

Puesto que tratamos de resolver problemas de clasificación, el número de unidades de la capa de salida se suele elegir de forma que coincida con el número de clases del problema de clasificación. Se asigna una neurona a cada clase y con ello se supone que se consigue una mayor inmunidad al ruido en las entradas.

En cuanto a la capa de entrada de la red, el problema principal es seleccionar la información que se utilizará para resolver un determinado problema o aplicación.

La información que seleccionemos será codificada numéricamente e introducida en las neuronas de entrada a la red. De esta forma, el número de neuronas de entrada vendrá determinado en función de la información que seleccionemos.

Por ejemplo, en una aplicación de diagnóstico médico hay que seleccionar el conjunto de síntomas y pruebas útil y mínimo que es necesario para resolver el problema: diagnóstico de una enfermedad. Cada síntoma, será codificado numéricamente y será una entrada en la red neuronal (una entrada en una neurona de la capa de entrada).

Normalmente en aplicaciones reales complejas se tiene una gran cantidad de información que puede ser medida y utilizada como entrada de una red neuronal. Cuanto más complicado sea el problema más variables y medidas serán necesarias para resolver el problema. Sin embargo, a priori, se desconoce la importancia relativa de una información para resolver un problema.

El primer paso para realizar una aplicación en redes neuronales será precisamente resolver este problema, es decir, decidir qué información se utilizará

como entradas de la red. Esta selección influirá en gran medida en la eficiencia final de la red. Por ejemplo, es posible que estemos seleccionando información ruidosa, redundante o incluso irrelevante, y que ello suponga un deterioro de la capacidad de generalización.

Generalmente, a la hora de desarrollar una determinada aplicación se recurre a la ayuda de uno o varios expertos humanos que deciden qué información es necesaria para resolver el problema.

Este procedimiento tiene varios inconvenientes.

En primer lugar, está sujeto a la experiencia y conocimientos del experto. Es posible que se introduzca información no útil que provoque un deterioro en la eficiencia de la red. También puede ocurrir, que se omita información importante porque el experto no la considere como tal.

Otro inconveniente es que en problemas muy complejos es posible que ni siquiera el experto humano sea capaz de decidir cuál es la información útil. En la bibliografía podemos encontrar ejemplos que ilustran esta situación.

En [1], el problema que se pretendía resolver era la modelización de parámetros críticos de la Máquina Principal de la Lanzadera Espacial (Space Shuttle Main Engine). Los autores comentan que las relaciones funcionales entre los parámetros que se midieron eran muy complejas y desconocidas. Además, el número de posibles parámetros de entrada era enorme. Por ello, utilizaron un método avanzado de selección de entradas.

En la referencia [2], el problema consiste en tomar decisiones sobre selección de personal y sobre cuando realizar una determinada acción formadora para una persona. Estas decisiones se realizaban basándose en las respuestas de un cuestionario. La utilidad de las preguntas fue cuestionada por el propio autor. Finalmente, se usó una red neuronal para tratar de extraer las preguntas que eran más adecuadas.

Otros ejemplos pueden encontrarse en los campos de medicina, diagnóstico de fallos de una máquina compleja, etc.

A este problema se le conoce como “selección de entradas o características” (en inglés “feature selection”). Se trata de un problema tópico muy importante en el campo de reconocimiento de patrones y por ello debemos de darle la misma importancia en el campo de redes neuronales.

Para resolver este problema, parece lógico, escoger un pequeño subconjunto de características (entradas) a partir de un conjunto más grande de información

### 3.1. INTRODUCCIÓN

potencialmente útil o candidata. El subconjunto de información que escojamos debe cumplir que sea el mínimo posible y suficiente para realizar bien la tarea de clasificación (sin deteriorar la eficiencia).

La forma inicial de abordar el problema sería, recoger la información de las entradas que se consideran potencialmente interesantes y posteriormente evaluar su utilidad con algún criterio, descartando las entradas inútiles y seleccionando un subconjunto de entradas a partir del conjunto original.

¿Por qué elegimos un subconjunto y no utilizamos directamente todo el conjunto inicial de información?

Para una red neuronal haciendo una selección del número de unidades de entrada (subconjunto útil de información) podemos conseguir varias ventajas. En primer lugar, al reducir el número de entradas reducimos el tamaño final de la red neuronal usada y de esta forma se simplifica el modelo utilizado para resolver el problema. En segundo lugar, reducimos la cantidad de datos que necesitamos obtener y procesar después del entrenamiento, es decir, cuando la red ya está funcionando. Esto reducirá el coste para la utilización de la red en una aplicación práctica.

En tercer lugar, se reducirá el tiempo de entrenamiento ya que la arquitectura de la red es menor (menor número de pesos y unidades de entradas). En definitiva, tendremos menor coste computacional.

Y por último, podremos conseguir un incremento en la capacidad de generalización de la red, es decir, un incremento en la eficiencia de la aplicación construida. Este resultado ha sido observado en este y otros trabajos de investigación anteriores. Podemos pensar que se debe a que al reducir el tamaño de la red proporcionando solo información útil (no redundante) se hace más fácil la tarea de aprendizaje.

Como hemos dicho antes, para resolver el problema de selección de entradas necesitamos algún tipo de criterio para poder medir la importancia de las entradas y este criterio va a depender del clasificador que se utilice para resolver el problema.

Por ejemplo, un subconjunto de variables de entrada puede ser óptimo para un determinado clasificador (por ejemplo, una red Multilayer Feedforward entrenada con Backpropagation) y muy ineficiente para otro (la misma red pero entrenada con un algoritmo distinto).



En la bibliografía se han propuesto diversos métodos para seleccionar la información de entrada a la red.

Normalmente se parte de un conjunto amplio de información y el objetivo es conseguir un subconjunto mínimo y útil de dicha información.

Las aproximaciones que se están utilizando para resolver este problema son muy variadas.

En [2] y [3] los autores proponen realizar un entrenamiento de la red con un conjunto de información amplio, para posteriormente realizar un análisis de la sensibilidad de la salida de la red con respecto a las entradas. Así, las entradas que poseen menor sensibilidad se pueden considerar como las menos importantes y se pueden omitir en entrenamientos posteriores.

Otros autores, en [1] y [4], proponen el uso de algoritmos genéticos para seleccionar las entradas de la red neuronal. Partiendo de un conjunto amplio de información, existen muchos subconjuntos que pueden usarse como entrada a la red. Evaluar la eficiencia de una red neuronal para todos los subconjuntos es un problema computacionalmente muy costoso. Puesto que los algoritmos genéticos son apropiados para realizar una búsqueda en problemas en los que hay un espacio de parámetros grande, fueron empleados por los autores en esta situación.

En las referencias [5] y [6], se propone un algoritmo de aprendizaje que simultáneamente selecciona las entradas de la red y el número de unidades de la capa oculta. Para ello, definen una función que intenta establecer la importancia de cada unidad dentro de la estructura de la red. Dicha definición está basada en conceptos de teoría de la información como el análisis de interdependencias.

Por último, en la referencia [14], se propone un método heurístico simple para la estimación de la importancia de las entradas en una red ya entrenada. Este método se basa en juzgar la contribución a las unidades de salida de cada una de las entradas. Las que menos contribuyen se consideran como menos importantes. Este método en cierto modo puede verse como un análisis sencillo de sensibilidad.

Para elegir un método concreto es importante considerar el coste computacional que conlleva.

El entrenamiento de una red conlleva un alto coste computacional por lo que parece imposible poder analizar todos los posibles subconjuntos de información de entrada para seleccionar el más idóneo.

Por ello, en redes neuronales, se suele preferir algún método de estimación directa.

### 3.1. INTRODUCCIÓN

Dentro de esta categoría nosotros hemos realizado la siguiente clasificación:

- Métodos basados en el análisis de una red neuronal entrenada de tipo Multilayer Feedforward (ARNMF).
- Métodos basados en el análisis del conjunto de entrenamiento (ACE).
- Métodos basados en el análisis de otras redes neuronales distintas de Multilayer Feedforward, redes específicas (ARE).

Todos estos métodos permiten obtener una ordenación de las entradas candidatas a la red neuronal acorde con la importancia de las mismas, pero no nos suministran cuál debe ser el subconjunto mínimo y útil de entradas que debemos seleccionar. En otras palabras, podemos saber qué entrada se descartaría en primer lugar, la menos importante, y también en que orden debemos ir descartándolas, pero desconocemos donde debemos de parar, cuantas entradas habrá en el subconjunto final.

Este será uno de los objetivos de esta Tesis: a partir de las diferentes ordenaciones de entradas que proporcionan los diferentes métodos conseguir una metodología que nos permita saber cuál será el subconjunto óptimo para cada método, y comparar los resultados para concluir que método funciona mejor.

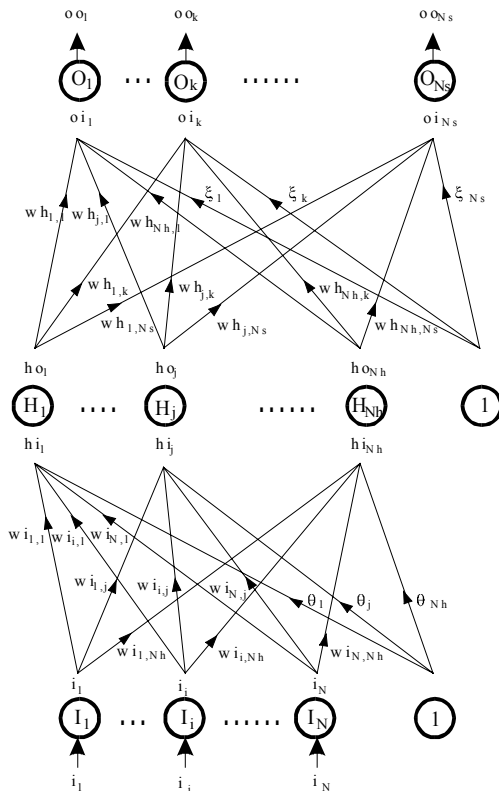
En el siguiente apartado estudiaremos cada uno de los métodos con más detalle.

### 3.2. Métodos de selección de entrada basados en ARNMF

Primero nos centraremos en los métodos de selección de entradas basados en el análisis de una red neuronal entrenada de tipo Multilayer Feedforward, que a partir de aquí denotaremos como ARNMF.

Para poder aplicar estos métodos necesitamos en primer lugar entrenar al menos una red neuronal con el conjunto completo de entradas candidatas. En segundo lugar, se deberá analizar la red entrenada (el análisis dependerá de cada uno de los métodos) para determinar la ordenación de las entradas según su importancia. Finalmente, a partir de las ordenaciones se tratará de encontrar el conjunto óptimo de entradas que se puede obtener a partir de las ordenaciones dadas por cada método.

En todos los casos, partimos de una red neuronal de tipo Multilayer Feedforward entrenada con el algoritmo Backpropagation. La función de transferencia o activación usada a la salida de la capa oculta y la de salida será de tipo sigmoide. La función de error usada será la clásica que mide la diferencia entre la salida deseada y la salida real obtenida.



La estructura y ecuaciones de la red neuronal usada son las clásicas de Multilayer Feedforward, que ya vimos en el primer capítulo. Veamos aquí un recordatorio.

$$hi_j = \sum_{i=1}^N wi_{ij} \cdot a_i + \theta_j \quad (3.1)$$

$$oi_k = \sum_{j=1}^{Nh} wh_{jk} \cdot ho_j + \xi_k \quad (3.2)$$

$$ho_j = \frac{1}{1 + \exp(-hi_j)} \quad (3.3)$$

$$oo_k = \frac{1}{1 + \exp(-oi_k)} \quad (3.4)$$

Fig 1. Arquitectura Multilayer Feedforward.

### 3.2. MÉTODOS DE SELECCIÓN DE ENTRADA BASADOS EN ARNMF

Donde  $i_i$  son las entradas a las neuronas en la capa de entrada a la red,  $hi_j$  y  $ho_j$  son respectivamente las entradas y salidas en las neuronas de la capa oculta y por último  $oi_k$  y  $oo_k$  serán las entradas y salidas respectivamente, en las neuronas de la capa de salida.

Todos estos métodos están basados en el principio de que durante la fase de entrenamiento la red neuronal debería seleccionar la importancia de las entradas y este resultado se debería de reflejar en la estructura final de la red, por ejemplo los pesos conectados a las unidades de entrada.

A continuación, vamos a ver una descripción detallada de los métodos más representativos de esta categoría que hemos encontrado en la bibliografía.

Todos ellos proponen como seleccionar las características útiles, es decir, dan una ordenación en función de su importancia. Nosotros utilizaremos la ordenación que proporcionan los métodos para seleccionar el subconjunto óptimo de características. Luego, estudiaremos qué método proporciona la ordenación más eficiente.

Estos métodos están basados como hemos dicho en el análisis de una red entrenada Multilayer Feedforward. Algunos de ellos, definen la relevancia de una entrada  $i$ ,  $S_i$ . La manera de calcular dicha relevancia puede ser muy variada. Por ejemplo, utilizando la ecuación (3.5) podemos calcular la relevancia  $S_i$  a partir de la relevancia  $s_{ij}$  de cada peso  $w_{ij}$  conectado a la unidad de entrada  $i$  y a la unidad de la capa oculta  $j$ . Una entrada es considerada más importante si su relevancia es mayor.

$$S_i = \sum_{j=1}^{Nh} s_{ij} \quad (3.5)$$

Podemos agrupar los diferentes métodos de ARNMF que vamos a estudiar en función de la idea básica de medida de la relevancia. Con ello obtenemos, siete grupos que son los que estudiaremos a continuación con más detalle.

### 3.2.1. Primer grupo

En este primer grupo el criterio para definir la relevancia de los pesos está basado en la estimación directa de la magnitud de los pesos.

Hemos utilizado dos métodos basados en esta idea que están descritos en las referencias [9] y [10].

#### **MÉTODO BL2**

El primero, propuesto en [9], lo denominaremos a partir de aquí BL2. Propone calcular la relevancia para una característica de entrada  $i$  como la suma de las relevancias de los pesos entre la capa de entrada y la capa oculta. Dichas relevancias pueden calcularse por la ecuación (3.6).

$$s_{ij} = (w_{ij})^2 \quad (3.6)$$

Donde  $w_{ij}$  es el peso entre la entrada  $i$  y la neurona de la capa oculta  $j$ .

La relevancia total para de una característica de entrada  $i$  se calculará mediante la ecuación (3.5). Una entrada será más importante cuanto mayor sea su relevancia, con lo cual, a partir de los valores de las relevancias  $S_i$  podemos realizar una ordenación de las características según su importancia.

#### **MÉTODO TEKA**

En la referencia [10], se definen varios métodos de selección de características. Dentro del primer grupo que estamos estudiando, podemos encontrar un método basado en la estimación directa de las magnitudes de los pesos conectados entre las neuronas de la capa de entrada y la capa oculta de la red neuronal. Denominaremos a este método como TEKA.

Se miden los valores absolutos de las magnitudes de los pesos  $w_{ij}$  entre la capa oculta y la de entrada y se obtiene la relevancia de cada peso según la ecuación (3.7).

$$s_{ij} = |w_{ij}| \quad (3.7)$$

De nuevo la relevancia de una entrada  $i$  se calcula a partir de la ecuación (3.5). Una entrada será más importante si su relevancia  $S_i$  es mayor.

### 3.2.2. Segundo grupo

En este grupo el criterio de relevancia está basado en la estimación del incremento en el error cuadrático medio  $E$  (E.C.M), cuando se podan los pesos, es decir, cuando se sustituye su valor por el valor cero.

#### MÉTODO CIB

El primer método de este grupo, está descrito en la referencia [11] y lo denominaremos CIB. Esta basado en la idea de definir la relevancia de un peso en función de la variación en la función de error  $E$  de la red cuando se suprime dicho peso, es decir se hace cero.

Cuando perturbamos el valor de un peso se producirá un cambio en la función de error de la red neuronal. La importancia relativa de un peso se puede definir como la perturbación que produce en la función de error de la red cuando se elimina dicho peso.

$$s_k = s_{ij} = E(w_k = 0) - E(w_k = w_k^f) \quad (3.8)$$

La ecuación anterior se puede expresar como la aproximación por serie de Taylor de la función de error. Su expresión vendrá dada por la ecuación (3.9).

$$s_k = \partial E = \sum_i g_i \partial w_i + \frac{1}{2} \sum_i h_{ii} \partial w_i^2 + \frac{1}{2} \sum_{i \neq j} h_{ij} \partial w_i \partial w_j + O(\|\partial W\|^3) \quad (3.9)$$

Donde  $g_i$  son las primeras derivadas del error  $E$  respecto de los pesos y  $h_{ij}$  son las segundas derivadas del error que constituirán la llamada matriz hesiana  $H$ .

En este método se asume que el podado de los pesos es realizado en un mínimo de la función de error, por lo cual, se descartan las derivadas de primer orden.

Además, se asume que las perturbaciones son pequeñas y los valores de las segundas derivadas grandes, por lo cual, se desprecian términos de tercer orden. También se supone una matriz hesiana diagonal.

Con todo esto, la expresión final que obtenemos para la relevancia de un peso vendrá dada en la ecuación (3.10).

$$s_k = s_{ij} = \frac{1}{2} \cdot h_{kk} \cdot w_k^2 \quad (3.10)$$

Donde hemos denotado por  $w_k$  al peso  $w_{ij}$  y  $h_{kk}$  es el correspondiente elemento de la matriz hesiana  $H$ .

Para realizar dicho cálculo se supone que la matriz hesiana es diagonal y su expresión puede calcularse según se describe en la referencia [8]. En el apéndice del capítulo hemos resumido dicho cálculo y damos las ecuaciones finales para el cálculo de la matriz hesiana  $H$ .

### **MÉTODO TEKE**

En la referencia [10], se describe un método que denominaremos TEKE basado en la misma idea que el anterior. La diferencia fundamental es, que ahora no se asume que la matriz hesiana sea diagonal y se calcula su inversa.

La relevancia de un peso  $w_{ij}$  se define según la ecuación (3.11).

$$s_k = s_{ij} = \frac{1}{2} \cdot \frac{w_k^2}{[H^{-1}]_{kk}} \quad (3.11)$$

Donde  $[H^{-1}]_{kk}$  son los elementos de la diagonal principal de la matriz inversa de la hesiana.

El coste computacional de este método será mayor ya que ahora hay que calcular la matriz hesiana completa y su inversa.

De nuevo podemos ver su cálculo en el apéndice del capítulo.

### **MÉTODO TEKC**

Por último en [10], se describe un método que denominaremos TEKC y que se basa también en el desarrollo de Taylor de la función de error. Sin embargo, ahora la aproximación consiste en quedarse con el término de primer orden (gradiente descendente) y despreciar términos superiores.

La relevancia de un peso  $w_{ij}$  se calcula por la ecuación (3.12).

$$s_{ij} = \sum_{t=0} \frac{\partial E}{\partial w_{ij}}(t) \cdot \Delta w_{ij}(t) \cdot \frac{w_{ij}(t)}{w_{ij}^i - w_{ij}^f} \quad (3.12)$$

Donde la suma sobre  $t$  se realiza sobre todas las iteraciones del entrenamiento, desde la inicial,  $t=0$ , hasta la de convergencia de la red. Los valores  $w_{ij}^i$  y  $w_{ij}^f$  son los iniciales y finales (en la iteración de convergencia) de los pesos  $w_{ij}$ . Por último,  $\Delta w_{ij}(t)$  es el cambio en los pesos calculado por el algoritmo de aprendizaje en la iteración  $t$ .

### MÉTODO MAO

Por último describiremos un método al que denominaremos MAO y que está descrito en [18].

La función de error cuadrático usada en el entrenamiento se define como la suma para todos los patrones de los errores cuadráticos de cada patrón.

$$E = \sum_{p=1}^N E_p = \frac{1}{2} \sum_{p=1}^N \sum_{k=1}^{N_s} (y_k^L - t_k)^2 \quad (3.13)$$

En esta ecuación,  $N$  es el número de patrones de entrenamiento,  $N_s$  el número de neuronas de salida, y  $L$  la capa de salida.

Se calcula la relevancia de una entrada como el incremento en el error cuadrático medio (ECM),  $E$ , cuando se elimina dicha entrada.

$$S_i = \sum_{p=1}^N \Delta E_p(I_i) \quad (3.14)$$

Para calcular el incremento en el ECM se realiza el desarrollo de Taylor, según la ecuación (3.15).

$$\Delta E_p(I_i) = \frac{\partial E_p}{\partial I_i} \cdot \Delta I_i + \frac{1}{2} \cdot \frac{\partial^2 E_p}{\partial I_i^2} \cdot (\Delta I_i)^2 \quad (3.15)$$

En ella  $\Delta I_i$  toma el valor  $(0-I_i)$  y las derivadas se pueden calcular recursivamente haciendo uso de las relaciones siguientes:

$$\frac{\partial E_p}{\partial y_i^l} = \sum_{j=1}^{N_{l+1}} \frac{\partial E_p}{\partial y_j^{l+1}} \cdot f' \cdot w_{ij} \quad (3.16)$$

$$\frac{\partial^2 E_p}{\partial (y_i^l)^2} = \sum_{j=1}^{N_{l+1}} \frac{\partial^2 E_p}{\partial (y_j^{l+1})^2} \cdot (f' \cdot w_{ij})^2 + \sum_{j=1}^{N_{l+1}} \frac{\partial E_p}{\partial y_j^{l+1}} \cdot f'' \cdot w_{ij}^2 \quad (3.17)$$

En donde  $f$  representa la función de transferencia sigmoide habitual,  $N_{l+1}$  es el número de unidades de la capa  $l+1$  y  $y_j^{l+1}$  es la salida  $j$  de la capa  $l$ .

Para una red de tres capas las derivadas con respecto a las salidas serán:

$$\frac{\partial E_p}{\partial o_k} = (o_k - t_k) \quad (3.18)$$



$$\frac{\partial^2 E_p}{\partial o_k^2} = 1 \quad (3.19)$$

Las derivadas respecto a las unidades de la capa oculta:

$$\frac{\partial E_p}{\partial h_j} = \sum_{k=1}^{Ns} \frac{\partial E_p}{\partial o_k} \cdot o_k \cdot (1 - o_k) \cdot wh_{jk} = \sum_{k=1}^{Ns} (o_k - t_k) \cdot o_k (1 - o_k) \cdot wh_{jk} \quad (3.20)$$

$$\begin{aligned} \frac{\partial^2 E_p}{\partial h_j^2} &= \sum_{k=1}^{Ns} \frac{\partial^2 E_p}{\partial o_k^2} \cdot (o_k \cdot (1 - o_k) \cdot wh_{jk})^2 + \sum_{j=1}^2 \frac{\partial E_p}{\partial o_k} \cdot o_k \cdot (1 - o_k) \cdot (1 - 2o_k) \cdot wh_{jk}^2 = \\ &= \sum_{k=1}^{Ns} (o_k \cdot (1 - o_k) \cdot wh_{jk})^2 + \sum_{j=1}^2 (o_k - t_k) \cdot o_k \cdot (1 - o_k) \cdot (1 - 2o_k) \cdot wh_{jk}^2 \end{aligned} \quad (3.21)$$

Y las derivadas con respecto a las unidades de entrada serán:

$$\frac{\partial E_p}{\partial I_i} = \sum_{j=1}^{Nh} \frac{\partial E_p}{\partial h_j} \cdot h_j \cdot (1 - h_j) \cdot wi_{ij} \quad (3.22)$$

$$\frac{\partial^2 E_p}{\partial I_i^2} = \sum_{j=1}^{Nh} \frac{\partial^2 E_p}{\partial h_j^2} \cdot (h_j \cdot (1 - h_j) \cdot wi_{ij})^2 + \sum_{j=1}^{Nh} \frac{\partial E_p}{\partial h_j} \cdot h_j \cdot (1 - h_j) \cdot (1 - 2h_j) \cdot wi_{ij}^2 \quad (3.23)$$

Podemos sustituir los valores de las ecuaciones de la (3.18) a la (3.21) en las ecuaciones (3.22) y (3.23).

Sustituyendo el resultado de las ecuaciones (3.22) y (3.23) en la ecuación (3.15) y después el resultado, en la ecuación (3.14) se puede calcular las relevancias para las diferentes características de entrada.

### 3.2.3. Tercer grupo

Otros métodos definen la relevancia usando la varianza de los pesos  $w_{ij}$  conectados a una entrada  $i$ . Están basados en la idea de que una pequeña varianza hace que la entrada se comporte como un umbral y su importancia será menor.

#### MÉTODO DEV

El primer método descrito en [12] y que llamamos DEV, está basado en la idea de encontrar las características más relevantes capaces de discriminar objetos. Cada patrón o ejemplo de entrada en la red neuronal es representado por un punto del hiperespacio de características. Los patrones que pertenecen a las

### 3.2. MÉTODOS DE SELECCIÓN DE ENTRADA BASADOS EN ARNMF

diferentes clases serán agrupados en diferentes regiones del espacio y separados por hiperplanos.

Desde el punto de vista geométrico, la selección de las características más relevantes que discriminen entre las clases de patrones de entrada, se puede hacer investigando los hiperplanos que separan las regiones del hiperespacio. Si una característica no es relevante para distinguir los patrones considerados, todos los hiperplanos que dividen las regiones del hiperespacio deben ser paralelos al vector unidad correspondiente a esa característica.

La identificación de los hiperplanos puede realizarse suponiendo una red neuronal de dos capas entrenada con Backpropagation.

Sea  $y_k$  la salida de la neurona  $k$  en la capa de salida de la red y  $net_k$  la entrada en la neurona  $k$  de la dicha capa. Tendremos:

$$y_k = f(net_k); \quad net_k = \sum_{i=0}^N w_{ik} x_i \quad (3.24)$$

Al final del entrenamiento, una entrada  $x_i$  es clasificada en la clase  $k$  si su valor de salida es mayor que el valor de salida en las demás clases:

$$\begin{aligned} \text{Si } y_k > y_q \quad \forall q \neq k \\ \text{Entonces } \sum_{i=0}^N w_{ik} x_i - w_{iq} x_i > 0 \quad \forall q \neq k \end{aligned} \quad (3.25)$$

La contribución de una característica  $x_s$  a la salida de la clase  $j$  será:

$$w_{sj} - w_{sq} \quad \forall q \neq j \quad (3.26)$$

Diremos que una característica no es relevante, si dicho valor es el mismo para todas las clases. Esto equivale a que la varianza de los pesos conectados con esa entrada sea cero para todas las salidas.

Basándose en esta idea, para una red neuronal de tres capas se puede calcular la relevancia de una característica  $i$  de entrada, a partir de la varianza proporcionada por los pesos conectados a ella como se indica en la ecuación (3.27).

$$S_i = \text{var}_i = \sum_j^{Nh} (w_{ij} - E(w_{ij}))^2 = \sum_j^{Nh} \left( w_{ij} - \frac{\sum_j^{Nh} w_{ij}}{Nh} \right)^2 \quad (3.27)$$

Donde  $Nh$  es el número de neuronas de la capa oculta.

### **MÉTODO DER3**

El segundo método, está descrito en [13] y lo llamaremos DER3.

Esta basado en la misma idea que el método anterior, es decir, definir la relevancia en función del valor de la varianza de los pesos entre la capa de entrada y la capa oculta de la red.

Sin embargo, ahora, para poder comparar las varianzas de los diferentes pesos entre sí, se normaliza respecto del valor esperado de dichos pesos.

La relevancia de una entrada  $i$  se medirá por la ecuación (3.28).

$$S_i = \frac{\text{var}_i}{E(w_{ij})} = \frac{N_h \cdot \text{var}_i}{\sum_j^{N_h} w_{ij}} \quad (3.28)$$

Donde  $w_{ij}$  es el peso desde la unidad  $i$  de entrada a la  $j$  de la capa oculta,  $N_h$  es el número de unidades en la capa oculta,  $E(w_{ij})$  es la expectación de dicho peso y  $\text{var}_i$  es la varianza que se calculará igual que en la ecuación (3.27).

### **3.2.4. Cuarto grupo**

Otros métodos evalúan la contribución de una entrada a la salida teniendo en cuenta el valor de los pesos en toda la estructura de la red neuronal.

### **MÉTODO TEKB**

El primer método de este grupo, está descrito en la referencia [10] y lo llamaremos TEKB.

Propone evaluar la contribución de una entrada en las salidas de la red teniendo en cuenta toda la estructura de la misma, es decir, todos los pesos.

Este método está basado en que las neuronas pueden trabajar en diferentes modos dependiendo del valor de la ecuación (3.29).

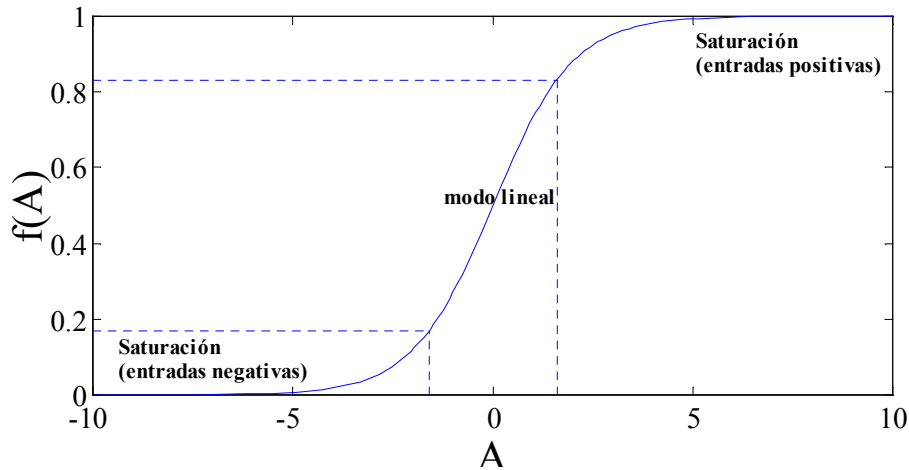
Los modos de activación dependen de las entradas en la capa inferior. La entrada  $A_j$  a una neurona  $j$  se puede expresar según la ecuación siguiente:

$$A_j = \sum_i w_{ij} \cdot a_i \quad (3.29)$$

### 3.2. MÉTODOS DE SELECCIÓN DE ENTRADA BASADOS EN ARNMF

Donde  $a_i$  es la entrada en una neurona de la capa inferior y  $w_{ij}$  es el peso que une la neurona  $j$  de la capa superior con la neurona  $i$  de la capa anterior (inferior).

Si la suma es grande (positiva o negativa) la neurona trabaja cerca de saturación. Si la suma es pequeña la neurona trabaja en el modo lineal. Esto puede observarse en la figura 2.



**Fig. 2. Función de transferencia y modos de activación de la función sigmoide.**

Una neurona de entrada conectada a neuronas que trabajen en modo lineal en la capa oculta, tendrá siempre menor relevancia que aquellas neuronas de entrada que estén conectadas a las neuronas ocultas que trabajen en modo de saturación.

Se puede calcularla relevancia de las entradas a una capa, en función de las relevancias de la capa superior, como vemos en la ecuación siguiente.

$$S_i^s = \sum_{j=1}^M \left( \frac{w_{ij}}{\max |w_{aj}| \forall a} \right)^2 \cdot S_j^{s+1} \quad (3.30)$$

Donde  $S_i^s$  es la relevancia de la neurona  $i$  en la capa  $s$ ,  $S_j^{s+1}$  es la de la neurona  $j$  de la capa  $s+1$  y  $w_{ij}$  es el peso entre la neurona  $i$  en la capa  $s$  y la neurona  $j$  en la capa  $s+1$ .

En el denominador, se calcula el peso que tiene un valor más alto de todos los pesos que acaban en la neurona  $j$  de la capa  $s+1$  (así se normaliza el valor de la relevancia a 1).

Esta ecuación es recursiva, primero se fijan las relevancias de las neuronas de la capa de salida iguales a 1, y recursivamente se estima el valor de las relevancias de las unidades de entrada de las capas inferiores hasta llegar a la capa de entrada a la red.

### **MÉTODO TEK**

Y por último en [14], tenemos un método que denominaremos TEK.

Este método es parecido al anterior pero mide la relación entre las neuronas  $i$  y  $j$  según la ecuación:

$$s_{ij} = \frac{|w_{ij} \cdot a_i^s|}{\max_k |w_{kj} \cdot a_k^s|} \quad (3.31)$$

El máximo se toma sobre todos los pesos que acaban en la neurona  $j$ .

Utilizando la norma euclídea y sumando para todos los patrones de entrada, se define la relevancia de la unidad  $i$  dada por los pesos que salen de ella según la ecuación (3.32).

$$S_i^s = \sum_j (s_{ij}^s)^2 \cdot S_j^{s+1} = \sum_j \frac{|w_{ij} \cdot a_i^s|^2}{\max_k |w_{kj} \cdot a_k^s|^2} \cdot S_j^{s+1} = \sum_{j=1}^M \frac{(w_{ij})^2 \cdot E[a_i^s]^2}{\max\{(w_{kj})^2 \cdot E[a_k^s]^2, \forall k\}} \cdot S_j^{s+1} \quad (3.32)$$

La expectación de  $a_i^s$  se calcula mediante la ecuación (3.33).

$$E(a_k^s) = \frac{\sum_{p=1}^N a_k^s}{N} \quad (3.33)$$

Donde  $N$  es el número de patrones.

Esta fórmula recurrente, calcula la sensibilidad de una unidad de la capa  $s$  a través de la sensibilidad de la capa  $s+1$ . Las sensibilidades de las neuronas de la capa de salida se toman iguales a 1. Todas las sensibilidades en una capa son normalizadas a 1.

Si asumimos que  $E[a_i^s]$  es prácticamente el mismo para las neuronas de la misma capa  $s$ , se obtendrá la ecuación (3.30) del método TEKB.

### **3.2.5. Quinto grupo**

Hay varios métodos que usan la sensibilidad de las salidas con respecto a una entrada para definir su relevancia. Están basados en la idea de que una alta sensibilidad significa una gran variación de los valores de salida debido a un cambio en una entrada, por lo que, se puede suponer que la entrada es más importante.

### **MÉTODO BL1**

En la referencia [9] se describe un método que denominaremos a partir de aquí como BL1.

Para definir la importancia de las entradas se mide la sensibilidad de la salida respecto de las entradas.

Como primer paso, se define la derivada de la salida respecto de una entrada específica que dependerá de las entradas en la red y de los pesos en ella.

Si consideramos la red Multilayer Feedforward y las ecuaciones de la misma descritas al principio de esta sección 3.2, obtendremos:

$$\begin{aligned} \frac{\partial o_k}{\partial I_i}(x, w) &= \frac{\partial o_k}{\partial o_i} \cdot \frac{\partial o_i}{\partial h o_j} \cdot \frac{\partial h o_j}{\partial h i_j} \cdot \frac{\partial h i_j}{\partial I_i} \\ &= o_k \cdot (1 - o_k) \cdot \sum_{j=1}^{N_h} w h_{jk} \cdot h o_j \cdot (1 - h o_j) \cdot w i_{ij} \end{aligned} \quad (3.34)$$

Donde  $h o_j$  es la salida de la neurona  $j$  de la capa oculta,  $o_k$  es la salida de la neurona de salida  $k$  de la red,  $w h_{jk}$  es el peso entre la unidad  $j$  de la capa oculta y la unidad  $k$  de la capa de salida,  $w i_{ij}$  es el peso entre la unidad  $i$  de la capa de entrada y la unidad  $j$  de la capa oculta y  $I_i$  es la entrada en la neurona  $i$  de la capa de entrada.

La relevancia de una entrada  $i$  se definirá a partir de las sensibilidades según la ecuación (3.35), que representa la sensibilidad media dentro del espacio de variabilidad de las entradas.

$$S_i = \frac{1}{N} \cdot \sum_{x \in S} \sum_{k=1}^{N_s} \sum_{x \in D} \left| \frac{\partial o_k}{\partial I_i}(x, w) \right| \quad (3.35)$$

Donde  $x$  es un vector de características de entrada,  $S$  es el conjunto de entrenamiento,  $D$  es un conjunto de puntos uniformemente distribuidos en el rango de variabilidad de las entradas,  $N$  es la suma de los cardinales de  $S$  y  $D$  y  $N_s$  es el número de unidades en la capa de salida de la red.

### **MÉTODO CLO**

Otro método similar a BL1 que denominaremos CLO fue propuesto en [15].

Este método mide la relevancia de una unidad de entrada en función de la sensibilidad de las salidas respecto a las entradas.

Igual que en el método anterior, se define la sensibilidad de una neurona  $j$  de la capa de salida respecto una unidad  $i$  de la capa de entrada para un cierto patrón  $x$ , mediante la ecuación (3.36) y su expresión final vendrá dada por la ecuación (3.34).

$$s_{ij}^x = \frac{\partial o_j}{\partial I_i}(x, w) \quad (3.36)$$

Cada pareja de valores  $i, j$  produce una matriz de sensibilidades diferente.

La sensibilidad media de la unidad  $j$  en la capa de salida, para cambios en la unidad  $i$  de la capa de entrada para todo el conjunto de entrenamiento, se obtendrá sumando para todos los patrones mediante la expresión siguiente.

$$A_{ij} = \sqrt{\frac{\sum_{x \in S} (s_{ij}^x)^2}{N}} = \sqrt{\frac{\sum_{x \in S} \left( \frac{\partial o_j}{\partial I_i}(x, w) \right)^2}{N}} \quad (3.37)$$

A partir de la ecuación (3.37), se puede definir la relevancia de la unidad  $i$  de la capa de entrada, como el máximo de las sensibilidades entre el elemento  $i$  y cualquier elemento  $j$  de la capa de salida.

$$S_i = \max(A_{ij}) \quad \forall j \quad (3.38)$$

### **MÉTODO PRI**

Otro método que denominamos PRI, se describe en la referencia [16].

Está basado en la medida del error en la clasificación correcta de un vector de entrada. Se mide la probabilidad a posteriori de una clase dado un vector de entrada  $x$ . La probabilidad de error en la clasificación para un nodo de salida dado un vector de entrada  $x$  puede ser calculada como:

$$P_{error}(j, x) = 1 - P(C_j, x) = \sum_{k \neq j} P(C_k, x) = \sum_{k \neq j} o_k \quad (3.39)$$

La relevancia de una entrada  $i$  se obtiene a partir de la medida de las sensibilidades, es decir de la derivada de las salidas respecto de las entradas.

Para ello, se calcula la derivada de la ecuación (3.39) respecto de una cierta entrada o característica  $i$ . Con ello podemos definir la relevancia de una entrada  $i$  por la ecuación (3.40).

$$S_i = \sum_{x \in S} \sum_{j=1}^{Ns} \sum_{x \in D} \left| \sum_{k \neq j} \frac{\partial o_k}{\partial I_i}(x, w) \right| \quad (3.40)$$

### 3.2. MÉTODOS DE SELECCIÓN DE ENTRADA BASADOS EN ARNMF

Donde  $j$  varía entre 1 y el número de neuronas  $N_s$  de salida de la red,  $D$  es un conjunto de puntos uniformemente distribuidos en el rango de variabilidad de las entradas,  $S$  es el conjunto de entrada usadas en el entrenamiento,  $I_k$  y  $o_k$  son respectivamente las entradas y las salidas de la red.

Igual que antes, la expresión de la derivada de la salida respecto de la entrada a la red vendrá dada por la ecuación (3.34).

#### **MÉTODO SAN**

El siguiente método descrito en [17] y que denominamos SAN, también está basado en medir la dependencia de una salida respecto de una entrada.

Para medir dicha dependencia, habría que realizar la integración sobre todas las entradas. Sin embargo, aquí se propone un método más simple, se mide el grado de dependencia de una unidad de salida  $k$  en la unidad de entrada  $i$  por la ecuación (3.41).

$$D(k,i) = \max \left\{ \left| \frac{\partial o_k}{\partial I_i}(x,w) \right| \quad \forall x \in S \right\} \quad (3.41)$$

En la referencia [17], no se especifica como decidir la importancia de una entrada respecto de otra, cuál es más importante.

En la ecuación anterior  $D(k,i)$  es una matriz de sensibilidades, el índice  $k$  varía entre 1 y el número de salidas y el índice  $i$  entre 1 y el número de entradas.

Nosotros hemos considerado que la relevancia de la entrada  $j$  es mayor que la de la entrada  $i$  si se cumple que  $D(k,j) > D(k,i)$  para un mayor número de valores de  $k$ .

#### **MÉTODO DER2**

En la referencia [13] tenemos un nuevo método que llamamos DER2.

Este método utiliza sensibilidades logarítmicas para evitar que aparezca el término de saturación  $o_k(1-o_k)$  de la sensibilidad normal, ecuación (3.34).

Se define la sensibilidad  $B_{ik}$  de la salida  $k$  con respecto a la entrada mediante la ecuación siguiente.

$$B_{ik} = \frac{\partial \ln ( | t_k - o_k | )}{\partial \ln I_i} \quad (3.42)$$

Aquí  $t_k$  es el objetivo de la salida  $o_k$ .



El método requiere que la red haya convergido y que los patrones estén bien clasificados. Después de que la red haya convergido se calculan las relevancias para todas las clases y se define la relevancia  $S_i$ , como el máximo de las sensibilidades según la ecuación (3.43).

$$S_i = \max(B_{ik}) \quad \forall k \quad (3.43)$$

### 3.2.6. Sexto grupo

Algunos métodos se basan en la estimación del decremento de la eficiencia de la red cuando sustituimos la entrada por un valor fijo apropiado. La relevancia de una entrada se considera mayor si el decremento en la eficiencia aumenta.

#### MÉTODO UTA

En la referencia [19], se describe el método que denominaremos UTA.

Está basado en el cálculo del incremento en el error cuadrático medio cuando se sustituye una característica de entrada  $i$ , por el valor medio de los valores que esa característica toma en todos los patrones de entrada.

Para calcular la relevancia de una característica, primero se sustituye en todos los patrones de entrada los valores  $x_i$  de una característica, por el valor medio obtenido según la ecuación (3.44).

$$\hat{x}_i = \frac{\sum_{p=1}^N x_i^p}{N} \quad (3.44)$$

Una vez realizada la sustitución se calculan los nuevos valores de salida en la capa oculta y a partir de ellos las salidas en las unidades de la capa de salida de la red. A partir de las salidas y los objetivos de la red se puede calcular el error cuadrático medio según la ecuación (3.13).

El incremento en el error cuadrático para un patrón será:

$$\Delta E_p(i) = E_p(i = \hat{i}) - E_p(i) \quad (3.45)$$

La relevancia de una característica se puede calcular por la ecuación:

$$S_i = \frac{1}{N} \cdot \sum_{p=1}^N \Delta E_p(i) \quad (3.46)$$

### 3.2. MÉTODOS DE SELECCIÓN DE ENTRADA BASADOS EN ARNMF

Para las relevancias del resto de características se repetirá el proceso.

Cuanto menor sea la variación en el error (menor relevancia), menos importante será la característica de entrada.

#### **MÉTODO LEE**

El último método, que denominamos LEE, está descrito en [20].

El método calcula el decremento en el porcentaje de aciertos en el test cuando se sustituye el valor de una entrada por su valor medio.

Es decir, igual que en el método UTA, se sustituye la característica por el valor medio de la ecuación (3.44), se calculan las salidas de la capa de salida y después las salidas de la red. Conocidas las nuevas salidas de la red se pueden calcular los nuevos valores de los porcentajes de acierto que provocan dichas sustituciones.

### 3.2.7. Séptimo grupo

Por último, hemos encontrado dos métodos que no podemos incluir en ninguno de los grupos o categorías anteriores.

#### **MÉTODO BOW**

En la referencia [21] se describe el método que denominaremos BOW.

Generalmente, en el proceso de entrenamiento para redes Multilayer Feedforward hemos usado el algoritmo Backpropagation.

El funcionamiento de este algoritmo consiste en que la diferencia (o error) entre la salida actual de la red y la deseada, es propagada hacia atrás a lo largo de la red neuronal, de forma que durante el proceso de entrenamiento se van modificando los pesos. La variación en los pesos se puede expresar por las fórmulas siguientes.

$$\Delta w(t+1) = w(t+1) - w(t) \quad (3.47)$$

Para los pesos entre la capa oculta y de salida.

$$\Delta w_{j,k}(t+1) = \eta \cdot \left( -\frac{\partial E_P}{\partial w_{j,k}} \right) + \alpha \cdot \Delta w_{j,k}(t) = -\eta \cdot \delta_k \cdot h_{o_k} + \alpha \cdot \Delta w_{j,k}(t) \quad (3.48)$$

Para los pesos entre la capa de entrada y la oculta.

$$\Delta w_{i,j}(t+1) = \eta \cdot \left( -\frac{\partial E_P}{\partial w_{i,j}} \right) + \alpha \cdot \Delta w_{i,j}(t) = -\eta \cdot \delta h_j \cdot x_j + \alpha \cdot \Delta w_{i,j}(t) \quad (3.49)$$

Donde  $\eta$  es el paso de adaptación,  $\alpha$  es el momento (entre 0 y 1),  $h_k$  es la salida de la neurona  $k$  de la capa oculta y  $x_j$  es entrada de la neurona de entrada  $j$ .

En el método BOW se va acumulando durante el proceso de entrenamiento un error  $e_{it}$  (en inglés “accumulated error index”), que será el que después nos defina la relevancia de las entradas.

Es decir, al entrenar las redes Multilayer Feedforward se van acumulando los valores de  $\delta h_j$  en cada iteración de entrenamiento. Dicho valor depende del valor de  $\delta_k$  y en el capítulo primero dimos su expresión.

La relevancia de una entrada se define como la suma que aparece en la ecuación (3.50) para todas las iteraciones del entrenamiento, hasta la iteración  $T$  de convergencia.

$$S_i = \sum_{t=0}^T |e_{it}| = \sum_{t=0}^T \left| \sum_{j=1}^{Nh} \delta h_j \cdot w_{ij} \right| \quad (3.50)$$

En la ecuación,  $w_{ij}$  es el peso entre la entrada  $i$  y la neurona oculta  $j$  y  $\delta h_j$  es, como hemos visto antes, el error de la neurona oculta  $j$  propagado hacia atrás (backpropagated) por el algoritmo de aprendizaje en la iteración  $t$ .

Cuanto mayor sea el error, más importante será la entrada y su relevancia será mayor.

### MÉTODO YOU

Por último, en [22], se describe un método que denominaremos YOU.

El método está basado en calcular el efecto de una entrada en la salida de la red neuronal. Se determina el grado de participación que cada neurona de entrada tiene en una neurona de salida.

El procedimiento a seguir es el siguiente.

- *Para cada neurona de la capa oculta se determina su participación en la activación de una neurona de salida.*
- *Se computa la participación de cada neurona de entrada en la activación de una neurona oculta.*

### 3.2. MÉTODOS DE SELECCIÓN DE ENTRADA BASADOS EN ARNMF

- Se deduce la participación de una neurona de entrada en la activación de una salida a través de una neurona de la capa oculta.
- Se suman las participaciones para todas las unidades ocultas, para determinar la participación total de una neurona de entrada en la activación de una neurona de salida.
- Se pesa esta participación en función de la importancia de cada neurona de salida.

Definimos en primer lugar algunas variables.

$f(x)$	Función de activación sigmoide usada en la red neuronal
$x_i$	Entradas y salidas de la capa de entrada a la red
$w_{ij}$	Los pesos entre la unidad $i$ y la unidad $j$ (capa entrada-oculta)
$w_{jk}$	Los pesos entre la unidad $j$ y la unidad $k$ (capa oculta-salida)
$A_{hid_j}$	Entrada en la neurona $j$ de la capa oculta
$x_{hid_j}$	Salida de la neurona $j$ de la capa oculta
$A_{out_k}$	Entrada en la neurona $k$ de la capa de salida
$x_{out_k}$	Salida de la neurona $k$ de la capa de salida
$inc\_hid$	Variación en el estado de la neurona debido a $A_{hid}$
$inc\_out$	Variación en el estado de la neurona debido a $A_{out}$

Se cumplirán las siguientes expresiones:

$$A_{hid_j} = \sum_i x_i \cdot w_{ij}; \quad x_{hid_j} = f(A_{hid_j}) = \frac{1}{1 + \exp(-A_{hid_j})} \quad (3.51)$$

$$A_{out_k} = \sum_j x_{hid_j} \cdot w_{jk}; \quad x_{out_k} = f(A_{out_k}) = \frac{1}{1 + \exp(-A_{out_k})} \quad (3.52)$$

Se pueden descomponer la entrada a las neuronas en dos términos, uno que indique la cooperación sobre la entrada  $i$  y otro que indique la no-cooperación.

Para la entrada en una neurona oculta  $j$  tendremos:

$$A_{hid_j} = A_{hid_j}^C + A_{hid_j}^N \quad (3.53)$$

Donde  $A_{hid_j}^C$  define la cooperación en el total de la entrada  $i$ :

$$A_{hid_j}^C = \sum_i x_i \cdot w_{ij}; \quad \forall i \text{ que cumple } (x_i \cdot w_{ij} \cdot A_{hid_j}) \geq 0 \quad (3.54)$$

Y  $Ahid_j^N$  define la no-cooperación en el total de la entrada  $i$ :

$$Ahid_j^N = \sum_i x_i \cdot wi_{ij}; \quad \forall i \text{ que cumple } (x_i \cdot wi_{ij} \cdot Ahid_j) < 0 \quad (3.55)$$

Para la entrada en la neurona  $k$  de salida tendremos:

$$Aout_k = Aout_k^C + Aout_k^N \quad (3.56)$$

Donde  $Aout_k^C$  define como:

$$Aout_k^C = \sum_j xhid_j \cdot wh_{jk}; \quad \forall j \text{ que cumple } (xhid_j \cdot wh_{jk} \cdot Aout_k) \geq 0 \quad (3.57)$$

Y  $Ahid_j^N$  se define como:

$$Aout_k^C = \sum_j xhid_j \cdot wh_{jk}; \quad \forall j \text{ que cumple } (xhid_j \cdot wh_{jk} \cdot Aout_k) < 0 \quad (3.58)$$

Una operación similar se puede realizar sobre las salidas de las neuronas.

Definimos el siguiente valor:

$$\Psi(a, b) = \int_a^b f(x) dx \quad (3.59)$$

Definimos para las salidas de la capa oculta:

$$inc\_hid_j^C = \Psi(0, Ahid_j^C); \quad (3.60)$$

$$inc\_hid_j^N = \Psi(0, Ahid_j) - \Psi(0, Ahid_j^C);$$

$$\tau hid = \frac{inc\_hid_j^N}{inc\_hid_j^C};$$

$$xhid_j^C = (1 - \tau hid) \cdot xhid_j;$$

$$xhid_j^N = \tau hid \cdot xhid_j;$$

$$xhid_j = xhid_j^C + xhid_j^N$$

Para las salidas de las neuronas de la capa de salida de la red:

$$inc\_out_k^C = \Psi(0, Aout_k^C); \quad (3.61)$$

$$inc\_out_k^N = \Psi(0, Aout_k) - \Psi(0, Aout_k^C);$$

$$\tau out = \frac{inc\_out_k^N}{inc\_out_k^C};$$

$$xout_k^C = (1 - \tau out) \cdot xout_k;$$

$$xout_k^N = \tau out \cdot xout_k;$$

$$xout_k = xout_k^C + xout_k^N$$

**Desde la entrada a la capa oculta**

Podemos definir el grado de participación que una neurona de entrada  $i$  tiene sobre una unidad  $j$  de la capa oculta dado un cierto patrón  $p$ , como  $I_i(j|p)$  según la ecuación:

$$I_i(j|p) = \begin{cases} \frac{x_i \cdot w_{ij}}{A_{hid_j^C}} \cdot x_{hid_j^C} & \text{si } (x_i \cdot w_{ij} \cdot A_{hid_j}) \geq 0 \\ \frac{x_i \cdot w_{ij}}{A_{hid_j^N}} \cdot x_{hid_j^N} & \text{si } (x_i \cdot w_{ij} \cdot A_{hid_j}) < 0 \end{cases} \quad (3.62)$$

**Desde la entrada a la salida a través de la capa oculta.**

Se define el grado de participación de una neurona de entrada  $i$  sobre una neurona de salida, a través de una neurona  $j$  oculta para un cierto patrón de entrenamiento  $p$ ,  $I_{ij}(k|p)$  según la ecuación:

$$I_{ij}(k|p) = \begin{cases} \frac{I_i(j|p) \cdot w_{jk}}{A_{out_k^C}} \cdot x_{out_k^C} & \text{si } (I_i(j|p) \cdot w_{jk} \cdot A_{out_k}) \geq 0 \\ \frac{I_i(j|p) \cdot w_{jk}}{A_{out_k^N}} \cdot x_{out_k^N} & \text{si } (I_i(j|p) \cdot w_{jk} \cdot A_{out_k}) < 0 \end{cases} \quad (3.63)$$

**Desde la entrada a la capa de salida.**

El grado total de participación de una neurona  $i$  de entrada sobre la salida total de la neurona  $k$  para un patrón dado  $p$ , vendrá dada por la contribución de todas las unidades ocultas en esa unidad de salida.

$$I_i^{Total}(k|p) = \sum_j I_{ij}(k|p) \quad (3.64)$$

En la tarea de clasificación con una red de  $N_{salida}$  salidas de la red, la decisión del sistema se basa en las unidades más activas al presentar el patrón  $p$  de la clase  $s$ . Por ello, se utiliza un factor de peso  $(N_{salida} - 1)$ , ya que ese número de neuronas de salida, toman otra decisión.

Para un cierto patrón  $p$ , se puede expresar la participación de una neurona  $i$  de entrada sobre las salidas de la red por la ecuación:

$$I_i(p) = (N_{salida} - 1) \cdot I_i^{Total}(s|p) - \sum_{k=1, k \neq s}^{N_{salida}} I_i^{Total}(k|p) \quad (3.65)$$

***Fase de selección.***

Usando una red neuronal entrenada para cada patrón, o ejemplo de entrenamiento, se puede calcular el grado de participación que una neurona  $i$  tiene sobre la decisión que toma la red, ecuación (3.65).

Para cada entrada  $i$  se puede calcular para cada patrón ( $N$  patrones de entrenamiento) el valor  $I_i(p)$ , a partir de todo el conjunto de entrenamiento y se obtiene la importancia media para cada neurona de entrada  $i$  a partir de:

$$I_i^{Media} = \frac{1}{N} \cdot \sum_{p=1}^N I_i(p) \quad (3.66)$$

Finalmente, cuanto mayor sea esta medida más importante será la entrada.

### 3.3. Métodos de selección de entradas basados en ACE

Los métodos basados en el análisis del conjunto de entrenamiento los hemos agrupado en tres categorías, en función del tipo de técnicas que utilizan: conceptos de la Teoría de la Información, medida de distancias dentro y entre las clases y por último lógica difusa (“fuzzy”).

A priori, estos métodos presentan la ventaja, frente a los métodos ARNMF, de que no hay que entrenar ninguna red neuronal para aplicarlos. Por ello, su carga computacional suele ser mucho menor.

#### 3.3.1. Teoría de la Información

##### MÉTODO BA

En [23] se describe el método basado en conceptos de Teoría de la Información y que denominaremos BA.

El algoritmo utilizado en este método es el siguiente:

- 1) Definir  $F$  como un conjunto que incluya a todas las entradas candidatas a la red neuronal y sea  $S$  un conjunto vacío.
- 2) Calcular la información mutua  $I(C,f)$  entre cada entrada y el conjunto de clases  $C=(c_1, \dots, c_M)$ .
- 3) Encontrar la entrada  $g$  que maximice  $I(C,f)$ . Incluir  $g$  en el conjunto  $S$  y eliminarla de  $F$ .  $S=S \cup \{f\}$ ,  $F=F - \{f\}$ .
- 4) Repetir los siguientes pasos hasta que el cardinal de  $S$  sea  $k$  o el conjunto  $F$  este vacío.

*i) Calcular la información mutua  $I(f,s)$  entre las entradas de  $F$  y de  $S$ , ( $f \in S$  y  $s \in S$ ).*

*ii) Escoger la entrada  $g \in F$  que maximice la ecuación:*

$$I(C, f) - \beta \cdot \sum_{s \in S} I(f, s) \quad (3.67)$$

*En donde  $\beta$  es un parámetro dentro del intervalo  $[0.5, 1]$ .*

*iii) Incluir  $g$  en  $S$  y eliminarla de  $F$ .*

$I(C,f)$  mide la Información Mutua de una entrada  $f$  con las clases. Cuanto mayor sea más importante se considera la entrada.



### 3.3. MÉTODOS DE SELECCIÓN DE ENTRADAS BASADOS EN ACE

$I(f,s)$  mide la redundancia, dependencia entre dos entradas diferentes. Cuanto más grande sea, más relacionadas estarán entre sí y por tanto una de ellas es redundante.

El parámetro  $\beta$  regula la importancia relativa entre la Información Mutua de una entrada con las clases y la Información Mutua de dicha entrada candidata y las demás entradas (las ya seleccionadas como importantes).

El algoritmo intenta a partir del conjunto original  $F$  de entradas encontrar un subconjunto ordenado con las  $k$  entradas más relevantes. Sin embargo es imposible saber a priori el valor apropiado de  $k$ .

Lo que se obtiene es una ordenación de la importancia de las entradas, la más importante será la primera que el algoritmo seleccione para el subconjunto  $S$  final, y la menos importante será la última seleccionada.

En la práctica, nosotros aplicaremos el algoritmo hasta que hayamos cubierto todo el conjunto  $F$ , de forma que obtendremos una ordenación, por su importancia, de todas las entradas.

Para el cálculo de la información mutua  $I(C,f)$  entre una entrada  $f$  y el conjunto de clases se usará la ecuación (3.54).

$$I(C, f) = \sum_{\substack{\forall i=1 \dots M \\ \forall k=1 \dots N}} p(c_i | r_k(f)) \cdot \log_2 \frac{p(r_k(f) | c_i)}{p(c_i) \cdot p(r_k(f))} \quad (3.68)$$

En ella  $r_1(f), \dots, r_N(f)$  es una partición del rango de variabilidad de los valores de la entrada  $f$  en partes iguales. La suma se realiza sobre todos estos intervalos y sobre todas las clases. Las entradas están normalizadas en el rango  $[0,1]$  por lo que la generación de los intervalos consiste en dividir el intervalo  $[0,1]$  en un número  $N$  de particiones iguales. El número de intervalos  $N$  de la partición se fija habitualmente entre 16 y 32. Nosotros lo hemos tomado como 24 en nuestros experimentos.

En la ecuación  $P(c_i)$  es la probabilidad de que una observación sea de la categoría o clase  $c_i$ .  $P(r_k(f))$  es la probabilidad de que una observación tenga la entrada  $f \in r_k(f)$  y  $P(r_k(f_j) | c_i)$  es la probabilidad de que una observación con  $f_j \in r_k(f_j)$  y que se clasifique en la clase  $c_i$ .

Análogamente, para el cálculo de la información mutua  $I(f,s)$  entre dos entradas  $f$  y  $s$  se usará la ecuación (3.69).

En ella,  $P(r_k(f))$  es la probabilidad de que una observación tenga la entrada  $f \in r_k(f)$ ,  $P(r_j(s))$  es la probabilidad de que una observación tenga la entrada  $s \in r_j(s)$

### 3.3. MÉTODOS DE SELECCIÓN DE ENTRADAS BASADOS EN ACE

y  $P(r_k(f)/r_j(s))$  es la probabilidad de que una observación tenga la entrada  $f \in r_k(f)$  y la entrada  $s \in r_j(s)$ .

$$I(f, s) = \sum_{\substack{\forall j=1 \dots N \\ \forall k=1 \dots N}} p(r_k(f) | r_j(s)) \cdot \log_2 \frac{p(r_k(f) | r_j(s))}{p(r_j(s)) \cdot p(r_k(f))} \quad (3.69)$$

El cálculo de las probabilidades se hace contando la frecuencia de aparición en el conjunto de entrenamiento, de forma semejante a la que se describe en el siguiente método.

#### **MÉTODO CHI**

Otro método diferente fue propuesto en [24] y lo denominaremos CHI.

El método está basado en el cálculo de la entropía de una entrada.

Se define la entropía de una fuente de información  $X$  con posibles salidas  $x_i$  ( $i=1, \dots, n$ ) como:

$$H(X) = \sum_{i=1}^n P(x_i) \cdot I(x_i) = - \sum_{i=1}^n P(x_i) \cdot \log_2 P(x_i) \quad (3.70)$$

Donde  $P(x_i)$  es la probabilidad de que se dé la salida  $x_i$ .

Sea  $C(M, L)$  el clasificador en  $M$  categorías de las características de entrada. La entropía será:

$$H(C) = - \sum_{i=1}^M P(c_i) \cdot \log_2 P(c_i) \quad (3.71)$$

Donde  $P(c_i)$  es la probabilidad de que una observación sea de la categoría o clase  $c_i$ . Esta probabilidad se calculará por la siguiente ecuación:

$$P(c_i) = \frac{N(c_i)}{N_o} = \frac{N(c_i)}{\sum_{i=1}^M N(c_i)} \quad (3.72)$$

Donde  $N_o$  es el número total de observaciones consideradas (en nuestro caso número de patrones del conjunto de entrenamiento), y  $N(c_i)$  es el número de observaciones que pertenece a la clase  $c_i$ .

El objetivo es medir la contribución de cada característica o entrada  $f_j$  en la tarea de clasificación. Para ello, se divide el rango de valores que pueden tomar cada una de las entradas en partes iguales,  $N$  particiones:  $r_1(f_j), r_2(f_j), \dots, r_N(f_j)$ .

Igual que en el método anterior, en nuestros experimentos tomaremos 24 particiones en el rango de variabilidad del intervalo [0,1].

Sea  $f$  una entrada y  $P(c_i|r_k(f_j))$  la probabilidad de una observación con  $f_j \in r_k(f_j)$  y que se clasifique en la clase  $c_i$ . Entonces:

$$P(c_i | r_k(f_j)) = \frac{N(r_k(f_j), c_i)}{N(r_k(f_j))} \quad (3.73)$$

Donde  $N(r_k(f_j))$  es el número de observaciones con  $f_j \in r_k(f_j)$  y  $N(r_k(f_j), c_i)$  es el número de observaciones de  $N(r_k(f_j))$  que son de la clase  $c_i$  ( $i=1, \dots, M$ ). Se cumplirá entonces:

$$N(r_k(f_j)) = \sum_{i=1}^M N(r_k(f_j), c_i) \quad (3.74)$$

La probabilidad de que una observación tenga  $f_j \in r_k(f_j)$  será:

$$P(r_k(f_j)) = \frac{N(r_k(f_j))}{N_o}; \quad N_o = \sum_{k=1}^N N(r_k(f_j)) \quad (3.75)$$

Finalmente, se puede definir una medida similar a la entropía para una entrada  $f_j$  mediante la ecuación siguiente:

$$CH(f_j) = - \sum_{k=1}^N p(r_k(f_j)) \cdot \sum_{i=1}^M p(c_i | r_k(f_j)) \cdot \log_2 p(c_i | r_k(f_j)) \quad (3.76)$$

Esta medida es siempre positiva y una entrada  $f_j$  es considerada más relevante si su valor  $CH$  es menor.

### **MÉTODO SET**

En [25] se propone un nuevo método basado en el cálculo de ganancias que llamaremos SET.

En primer lugar, se define la información esperada para la clasificación como:

$$I(S) = - \sum_{j=1}^M \frac{n_j}{n} \cdot \log_2 \frac{n_j}{n} \quad (3.77)$$

Donde  $n_j$  es el número de ejemplo de entrenamiento de la clase  $j$  y  $n$  es el número total de ejemplos de entrenamiento, conjunto  $S$ .

Consideramos para cada entrada  $f_i$  una partición en su rango de variabilidad  $r_1(f_i), r_2(f_i), \dots, r_N(f_i)$ , donde  $N$  es el número de particiones (igual a 24).

### 3.3. MÉTODOS DE SELECCIÓN DE ENTRADAS BASADOS EN ACE

Sea  $S_{ik}$  un subconjunto de  $S$ , en el cual el valor de la característica  $f_i$  en todos los vectores del conjunto de entrenamiento, está en la partición  $r_k(f_i)$ .

$$I(S_{ik}) = - \sum_{j=1}^M \frac{n_{ikj}}{n_{ik}} \cdot \log_2 \frac{n_{ikj}}{n_{ik}} \quad (3.78)$$

Donde  $n_{ikj}$  es el número de ejemplos del subconjunto  $S_{ik}$  que son de la clase  $j$  (número de muestras  $x$  para las que  $f_i \in r_k(f_i)$  y  $x \in c_j$ ) y  $n_{ik}$  es el número total de elementos del subconjunto  $S_{ik}$ .

Para una entrada  $f_i$  la información esperada se puede calcular por la ecuación siguiente:

$$E_i = \sum_{k=1}^N \frac{n_{ik}}{n} \cdot I(S_{ik}) \quad (3.79)$$

La contribución de una entrada  $f_i$  a la ganancia será:

$$G_i = I(S) - E_i \quad (3.80)$$

Posteriormente, se normaliza la contribución de cada entrada y se define la ganancia normalizada  $G_i'$  como:

$$G_i' = \frac{G_i}{I_i}; \quad I_i = - \sum_{k=1}^N \frac{n_{ik}}{n} \cdot \log_2 \frac{n_{ik}}{n} \quad (3.81)$$

En ellas  $n$  es el número total de muestras del conjunto de entrenamiento, y  $n_{ik}$  es el número de muestras para las cuales la entrada  $f_i$  toma un valor dentro del intervalo de la partición  $r_k(f_i)$  ( $f_i \in r_k(f_i)$ ).

Cuanto mayor sea la ganancia normalizada más relevante se considera la entrada.

#### **MÉTODO GD-DS**

Por último, en [26] se describe un método que llamaremos GD\_DS.

Está basado en la medida de la llamada Distancia GD que viene a ser una generalización de la Distancia de Máttaras, la cual permite recoger la interdependencias entre las entradas y definir una ordenación de las mismas según su importancia.

El cálculo de la Distancia GD entre un conjunto de entradas  $F$  y el conjunto de las clases  $C$  se realiza como indica la ecuación (3.82).

$$d_{GD}(F, C) = D(F, C)^T \cdot T^{-1} \cdot D(F, C) \quad (3.82)$$

En ella  $T$  es la matriz de transinformación, dada en la ecuación (3.83), compuesta por las informaciones mutuas  $I(f_i, f_j)$  entre dos entradas  $f_i$  y  $f_j$ . Para calcularlas se utiliza la ecuación que vimos antes (3.69).

$$T = \begin{bmatrix} I(f_1, f_1) & I(f_1, f_2) & \dots & I(f_1, f_p) \\ I(f_2, f_1) & I(f_2, f_2) & \dots & I(f_2, f_p) \\ \dots & \dots & \dots & \dots \\ I(f_p, f_1) & I(f_p, f_2) & \dots & I(f_p, f_p) \end{bmatrix} \quad (3.83)$$

Esta matriz tiene las siguientes propiedades:

- *Los elementos de la diagonal principal son siempre mayores o iguales al resto de elementos de la misma fila o columna.*
- *Todos los elementos son no negativos.*
- *La matriz es simétrica.*

Por otra parte,  $D(F, C)$  es un vector de distancias de Matarás que tendrá la siguiente expresión:

$$D(F, C) = \begin{pmatrix} d(f_1, C) \\ d(f_2, C) \\ \dots \\ d(f_p, C) \end{pmatrix} \quad (3.84)$$

Aquí, la distancia  $d(f_i, C)$  entre una entrada y el conjunto de clases  $C$  se calcula con la ecuación (3.85).

$$d(f_i, C) = H(f_i, C) - I(f_i, C) \quad (3.85)$$

Su primer término, es una entropía que puede obtenerse con la ecuación (3.86) y el segundo término, es de nuevo la información mutua entre una entrada y el conjunto de clases dada en la ecuación (3.68).

$$H(f_i, C) = - \sum_{k=1}^N \sum_{j=1}^M p(r_k(f_i) | c_j) \cdot \log_2 \frac{p(r_k(f_i) | c_j)}{p(c_j) \cdot p(r_k(f_i))} \quad (3.86)$$

Por tanto, cuando el conjunto de entradas proporcione bastante información acerca de una clase, sus informaciones mutuas serán altas y por tanto  $d(f_i, C)$  serán bajas. Si un elemento de  $T$  aumenta (aumenta la interacción entre dos entradas) la distancia también aumenta.

Así, los valores de  $d_{GD}$  serán más bajos cuanto mayor sea la información que proporcione el conjunto de entradas a la clase, y menos interacción exista entre las entradas del conjunto.

### 3.3. MÉTODOS DE SELECCIÓN DE ENTRADAS BASADOS EN ACE

La ecuación (3.82) que se propone en este método es semejante a la definición de distancia de Mahalanobis (*Duda & Hart 1973*), sustituyendo la matriz de covarianzas por la matriz de transinformación.

Finalmente, el método utiliza un algoritmo muy similar al que utilizaba el método denominado BA que expusimos anteriormente.

- 1) Definir  $F$  como un conjunto con todas las entradas candidatas y sea  $S$  un conjunto vacío.
- 2) Para todas las entradas  $f \in F$ , calcular  $d_{GD}(F-\{f\}, C)$ , la Distancia GD entre el conjunto de entradas  $F-\{f\}$  y el conjunto de clases  $C$ .
- 3) Encontrar la entrada  $g$  que minimice las distancias del paso 2. Incluir  $g$  en el conjunto  $S$  y eliminarla de  $F$ .  $S=S \cup \{g\}$ ,  $F=F-\{g\}$ .
- 4) Repetir los pasos 2 y 3 hasta que el cardinal de  $S$  sea  $k$ .

De nuevo podremos obtener una ordenación de entradas, será más importante aquella entrada que haya sido incluida primero en el conjunto  $S$ .

#### 3.3.2. Distancias dentro y entre clases

##### MÉTODO RLF

El primer método incluido se denomina Relief y es muy popular en la literatura sobre reconocimiento de patrones, se describe en [27] y lo denotaremos como RLF.

Describiremos aquí igual que en la referencia [27] una versión simplificada para problemas que sólo tengan dos clases,  $c_1$  y  $c_2$ .

Sea  $S$  el conjunto de ejemplos de entrenamiento. Sea un ejemplo de entrenamiento  $x = \{x_1, x_2, \dots, x_p\}$  de  $p$  componentes de entrada. De cada ejemplo, conocemos su clasificación, es decir, si es de la clase  $c_1$  o de la  $c_2$ .

El algoritmo usado en el método RLF será el siguiente:

- 1) Inicializar aleatoriamente en los alrededores de 0.5 un vector de relevancias de los pesos  $W = (w_1, \dots, w_p)$ .
- 2) Repetir los siguientes pasos un número apropiado de veces  $m$  superior al número de ejemplos de entrenamiento.

i) Escoger aleatoriamente un ejemplo  $x$  de  $S$ .

$$x = (x_1, x_2, \dots, x_p).$$

ii) Escoger también aleatoriamente dos ejemplos de  $S$ ,

$z \in c_1$  e  $y \in c_2$ , cercanos a  $x$ .

iii) Si  $(x \in c_1)$  entonces  $N\_hit=z$ ;  $N\_miss=y$ ;

Si no  $N\_hit=y$ ;  $N\_miss=z$ .

iv) Actualizar los valores de las relevancias de los pesos:

Para  $i=1$  hasta  $p$ ;

$$w_i = w_i - \text{diff}(x_i, N\_hit_i)^2 + \text{diff}(x_i, N\_miss_i)^2;$$

3) Normalizar las relevancias dividiendo por  $m$ .

En la actualización de las relevancias, la diferencia entre las componentes de dos ejemplos,  $\text{diff}(f_i, s_i)$ , depende de la naturaleza de estas componentes.

Para el caso de componentes nominales (información nominal) usaremos:

$$\text{diff}(f_i, s_i) = \begin{cases} 0, & \text{si } f_i \text{ y } s_i \text{ son los mismos} \\ 1, & \text{si } f_i \text{ y } s_i \text{ son diferentes} \end{cases} \quad (3.87)$$

Para componentes numéricas (las utilizadas en nuestros experimentos) utilizaremos la ecuación (3.88).

$$\text{diff}(f_i, s_i) = \frac{f_i - s_i}{m_k} \quad (3.88)$$

Aquí,  $m_k$  es un factor que normaliza los valores de  $\text{diff}$  en el intervalo  $[0,1]$ . En los problemas que nosotros hemos utilizado en nuestros experimentos, los datos están normalizados en el intervalo  $[0,1]$ , por lo cual, no será necesario el factor  $m_k$  en la ecuación (3.88).

De nuevo, una entrada es considerada más importante, si su correspondiente componente en el vector de relevancias es mayor.

### **MÉTODO SCH**

Otro método de este tipo se describe en [28] y lo denominaremos SCH.

Aquí se define la distancia entre dos entradas como la “Distancia Euclídea sopesada” dada por la ecuación (3.89).

$$d_{ij} = \sqrt{\sum_{q=1}^p w_q \cdot (f_q^i - f_q^j)^2} \quad (3.89)$$

La ecuación calcula la distancia entre dos ejemplos (de dimensión  $p$ ) de entrenamiento,  $F^i = (f_1^i, \dots, f_p^i)$  y  $F^j = (f_1^j, \dots, f_p^j)$  asignando pesos a sus componentes.

### 3.3. MÉTODOS DE SELECCIÓN DE ENTRADAS BASADOS EN ACE

Sea  $N$  el número de ejemplos en el conjunto de entrenamiento. Se definen los valores de  $\delta^S$ ,  $\delta^D$ ,  $N_D$  y  $N_S$  según las siguientes ecuaciones.

$$\delta^S(i, j) = \begin{cases} 1, & \text{si } F^i \text{ y } F^j \text{ son de la misma clase} \\ 0, & \text{en otro caso} \end{cases} \quad (3.90)$$

$$\delta^D(i, j) = \begin{cases} 1, & \text{si } F^i \text{ y } F^j \text{ no son de la misma clase} \\ 0, & \text{en otro caso} \end{cases}$$

$$N_S = \sum_{i=1}^N \sum_{j=i+1}^N \delta^S(i, j); \quad N_D = \sum_{i=1}^N \sum_{j=i+1}^N \delta^D(i, j);$$

Posteriormente, se define, en la ecuación (3.91), una función criterio  $J$  que mide la relación entre las distancias ínter e intraclases.

$$J = \frac{1}{N_S} \cdot \sum_{i=1}^N \sum_{j=i+1}^N \delta^S(i, j) \cdot d_{ij} - \frac{1}{N_D} \cdot \sum_{i=1}^N \sum_{j=i+1}^N \delta^D(i, j) \cdot d_{ij} \quad (3.91)$$

El siguiente paso consistirá en inicializar los pesos de forma aleatoria en un cierto intervalo, e ir modificándolos en las diferentes iteraciones mediante gradiente descendente. Se debe encontrar un conjunto de pesos  $w_q$  que minimicen  $J$  y para ello se utiliza la técnica del gradiente descendente.

La variación de la función  $J$  respecto de los pesos se puede calcular mediante la siguiente ecuación:

$$\frac{\partial J}{\partial w_q} = \frac{1}{N_S} \cdot \sum_{i=1}^N \sum_{j=i+1}^N \frac{\delta^S(i, j) \cdot (f_q^i - f_q^j)^2}{2 \cdot \sqrt{\sum_{t=1}^p w_t \cdot (f_t^i - f_t^j)^2}} - \frac{1}{N_D} \cdot \sum_{i=1}^N \sum_{j=i+1}^N \frac{\delta^D(i, j) \cdot (f_q^i - f_q^j)^2}{2 \cdot \sqrt{\sum_{t=1}^p w_t \cdot (f_t^i - f_t^j)^2}} \quad (3.92)$$

La adaptación de los pesos en cada iteración se realizará según:

$$w_q(t+1) = w_q(t) + \lambda \cdot \frac{\partial J}{\partial w_q} \quad (3.93)$$

En nuestros experimentos hemos tomado un valor para el paso de adaptación  $\lambda$  de 0.1.

Los valores finales de los pesos se restringirán al intervalo  $[0,1]$ .

Una entrada será más importante, si su componente correspondiente en el vector final de pesos es mayor.

Esto nos permitirá de nuevo ordenar las entradas en función de su importancia.



### 3.3.3. Lógica difusa

En este tipo de métodos la bibliografía es menos extensa. De hecho, solo hemos estudiado un método que realiza un análisis de regiones difusas.

#### **MÉTODO FUZ**

Denominaremos FUZ al método que se describe en [29].

Este método realiza un análisis de regiones difusas y para ello es necesario generar en primer lugar un conjunto de reglas difusas que nos delimiten dichas regiones.

Sea  $X_i$  el subconjunto de ejemplos del conjunto de entrenamiento  $X$ , catalogados como pertenecientes a la clase  $c_i$ . Supongamos que tenemos un problema con  $M$  clases diferentes.

Primero, generamos las reglas de difusas para los cuales, dado un vector de entrada  $x$  puede ser clasificado en una de las  $M$  clases.

Podemos generar varios “hiper-rectángulos” del nivel 1 (para las diferentes clases), que llamaremos  $A_{ii}$ ,  $i=1,\dots,M$ , cuyos bordes son los máximos y mínimos de cada entrada de los ejemplos del conjunto  $X_i$ . A dichos hiper-rectángulos los llamaremos **regiones de activación**.

Si dos regiones de activación se solapan, por ejemplo  $A_{ii}(1)$  y  $A_{jj}(1)$ , podemos definir otro hiper-rectángulo de nivel 1,  $I_{ij}(1)$ , como la zona de intersección de las regiones de activación. A estos nuevos hiper-rectángulos los denominaremos **regiones de inhibición**.

Con estas definiciones, un primer conjunto de reglas difusas sería:

**Regla:** Sea un ejemplo  $x$ , si  $x \subset A_{ii}(1)$  y  $x \not\subset I_{ij}(1) \quad \forall j, j \neq i$  entonces  $x \in c_i$

Este proceso puede continuar y podemos definir regiones de activación de nivel 2,  $A_{ij}(2)$  y  $A_{ji}(2)$  de las clases  $c_i$  y  $c_j$ , cuando existen datos en la región de inhibición  $I_{ij}(1)$ . Por ejemplo, para definir la región  $A_{ji}(2)$  utilizaremos los ejemplos  $X_i$  que pertenecen a la clase  $c_i$  y están contenidos en la región de inhibición de nivel 1,  $I_{ij}(1)$ .

De nuevo, si las regiones de activación de nivel 2,  $A_{ij}(2)$  y  $A_{ji}(2)$ , se solapan podríamos definir una región de inhibición de nivel 2,  $I_{ij}(2)$ .

### 3.3. MÉTODOS DE SELECCIÓN DE ENTRADAS BASADOS EN ACE

La continuación del proceso se realizaría hasta que las nuevas regiones de activación encontradas no se solapan. Es ese caso no se podrán definir nuevas regiones de inhibición y el proceso acabará.

Una vez obtenidas todas las regiones de activación e inhibición, podemos definir y calcular las relaciones de excepción,  $o_{ij}(F)$ , para un conjunto de entradas  $F$ . Dicha relación es la suma para todos los niveles  $n$ , del resultado de dividir el volumen de la región de inhibición,  $I_{ij}(n)$ , entre el volumen de la región de activación,  $A_{ij}(n)$ , multiplicando dicho cociente por la probabilidad de encontrar un ejemplo dentro de la región de inhibición (esta probabilidad se estima experimentalmente utilizando el conjunto de entrenamiento).

La probabilidad será el número de datos de entrenamiento de la clase  $i$  en la región de inhibición, dividido entre el número total de ejemplos de entrenamiento.

A su vez, podemos definir la relación de excepción total,  $O(F)$ , para un conjunto  $F$  de entradas, como la suma, para todos los valores de  $i$  y  $j$  con  $i \neq j$  de la relaciones de excepción  $o_{ij}(F)$ .

Finalmente, el algoritmo para ordenar las entradas es similar a los de los métodos ya vistos BA y DS\_GD, sólo que en este caso debemos de minimizar la relación de excepción total.

Sea el conjunto total de entradas  $F$ . Una entrada será eliminada (la más irrelevante) si al quitarla se minimiza la excepción total.

Llamamos,  $F_{org}$  al conjunto original de  $M$  entradas  $M \geq 2$  y  $F^m$  a un conjunto originalmente vacío. Se define el siguiente algoritmo:

- 1) Inicialmente ponemos  $F^m$  igual al conjunto original  $F_{org}$ ,  $m=M$ .
- 2) Repetir los siguientes pasos hasta que  $m=1$ .
  - i) Calcular  $O(F_i^{m-1})$  para  $i=1, \dots, m$ , donde  $F_i^{m-1}$  es el conjunto  $F_i^m$  en el cual se ha eliminado la entrada  $f_i^m$ .
  - ii) Encontrar la entrada  $f_j^m$  que haga mínima la excepción total  $O(F_j^{m-1})$ .
  - iii) Poner el conjunto  $F^{m-1} \leftarrow F_j^{m-1}$ , donde  $f_j^m$  ha sido eliminado.
  - iv) Poner  $m=m-1$ .

Con este algoritmo obtendremos una ordenación de las características por su importancia. La primera eliminada será la menos importante.

### **3.4. Métodos de selección de entradas basados en ARE**

La última categoría no la trataremos en esta Tesis Doctoral por tratarse de métodos que utilizan arquitecturas diferentes de Multilayer Feedforward y bastante diferentes a los descritos. Dejaremos su estudio para trabajos futuros.

### **3.5. Algoritmo de aprendizaje mediante gradientes conjugados**

Durante nuestra investigación el algoritmo de entrenamiento normalmente utilizado ha sido Backpropagation. Sin embargo, algunos autores han comentado que dicho algoritmo posee baja eficiencia temporal, es decir, la convergencia de las redes entrenadas con él suele ser bastante lenta.

Este hecho, en una investigación como la que aquí llevamos a cabo, tiene gran importancia debido al gran número de problemas sobre el que realizamos nuestros experimentos.

Por ello, pensamos que en ciertas ocasiones podría ser interesante acelerar mediante algún mecanismo el proceso de convergencia de Backpropagation.

En la referencia [40], se describe un nuevo algoritmo que utiliza gradientes conjugados y optimiza de forma dinámica el valor del paso de adaptación utilizado por Backpropagation.

Un valor fijo del paso de adaptación afecta al tiempo de convergencia de la red, por lo que en la referencia, se recomienda analizar el cambio de la función objetivo (o error) respecto del paso de adaptación.

En Backpropagation tenemos básicamente dos pasos. En el primero, se calcula el valor actual de una salida computado hacia delante, es decir, desde la capa de entrada de la red (usando las entradas y pesos) hacia la capa de salida. En el segundo paso se adaptan los valores de los pesos anteriores a unos nuevos valores, utilizando el método del gradiente descendente y recalculando tanto los pesos como el error, la computación es ahora hacia atrás. Esto se repite para todos los patrones de entrada y tantas veces como sea necesario hasta obtener el valor deseado del error (punto de convergencia).

### 3.5. ALGORITMO DE APRENDIZAJE MEDIANTE GRADIENTES CONJUGADOS

El nuevo algoritmo posee la misma base que Backpropagation, pero a la hora de adaptar los valores de los pesos usa gradientes conjugados y el parámetro llamado paso de adaptación se considera un valor variable que va siendo adaptado al tiempo que los pesos. De esta forma, se adapta su valor dinámicamente intentando acelerar la convergencia de la red.

La convergencia del nuevo algoritmo puede ser acelerada en un factor de 10 a 50 dependiendo de la arquitectura de la red y de la aplicación.

A continuación, describimos el nuevo algoritmo de entrenamiento. Las ecuaciones del mismo están descritas de forma general, es decir, para una red con cualquier número de capas.

Veamos en primer lugar cuál es la notación que usaremos.

- $P$  Número de ejemplos de entrenamiento.
- $J$  Función de error de Backpropagation.
- $N_m$  Número de unidades en la capa  $m$ .
- $y_{s,i}^m$  Salida de la  $i$ -ésima unidad de la capa  $m$  con el patrón  $X_S$  presente en la capa de entrada.  $y_{s,i}^m = f(W_i^{mt} \cdot Y_s^{m-1}) \quad i = 1, \dots, N_m \quad m = 1, \dots, M$
- $Y_s^{m-1}$  Vector de salidas de la capa  $m-1$ .  $Y_s^{m-1} = [1, y_{s,i}^{m-1}, \dots, y_{s,N_{m-1}}^{m-1}]^T$
- $W_i^{mt}$  Vector de pesos conectados entre la  $i$ -ésima unidad de la capa  $m$  y las unidades de la capa  $m-1$ .
- $f(\cdot)$  Función de transferencia de la red (sigmoide).

La función de error para el caso de Backpropagation y la adaptación de los pesos pueden expresarse mediante las ecuaciones:

$$J(W) = \frac{1}{P \cdot N_M} \cdot \sum_{s=1}^P \sum_{i=1}^{N_M} [d_{s,i} - y_{s,i}^M]^2 \quad (3.94)$$

$$W(k+1) = W(k) - \mu \cdot \nabla J(W)|_{W=W(k)} = W(k) - \mu \cdot \frac{\partial J(W)}{\partial W} \Big|_{W=W(k)} \quad (3.95)$$

Para el nuevo algoritmo, se deben transformar estas ecuaciones para que la adaptación de los pesos y el error dependan del valor del paso de adaptación  $\mu$ .

Así, se define una nueva función de error dependiente del paso de adaptación:

$$h(\mu) \equiv J(W(k) + \mu \cdot P(k)) \quad (3.96)$$

La ecuación correspondiente de adaptación de los pesos será:

$$W(k+1) = W(k) + \mu \cdot P(k) \quad (3.97)$$

En la referencia, el valor tomado para el paso de adaptación viene dado por:

$$\mu(k) = -\frac{h'(0)}{h''(0)} \quad (3.98)$$

Las ecuaciones de las derivadas se pueden obtener a partir de la definición dada para la nueva función de error  $h(\mu)$  y sus expresiones serán:

$$\begin{aligned} h'(\mu) &= \nabla J[W(k) + \mu \cdot P(k)]^T \cdot P(k) \\ h''(\mu) &= P(k)^T \cdot \nabla^2 J[W(k) + \mu \cdot P(k)] \cdot P(k) \end{aligned} \quad (3.99)$$

Sustituyendo en las ecuaciones anteriores y derivando se obtiene:

$$\begin{aligned} h'(\mu) &= -\frac{2}{P \cdot N_M} \cdot \sum_{s=1}^P \sum_{i=1}^{N_M} (d_{s,i} - y_{s,i}^M) \cdot \frac{\partial y_{s,i}^M}{\partial \mu} \\ h''(\mu) &= -\frac{2}{P \cdot N_M} \cdot \sum_{s=1}^P \sum_{i=1}^{N_M} \left\{ (d_{s,i} - y_{s,i}^M) \cdot \frac{\partial^2 y_{s,i}^M}{\partial \mu^2} - \left( \frac{\partial y_{s,i}^M}{\partial \mu} \right)^2 \right\} \end{aligned} \quad (3.100)$$

Las expresiones de las derivadas primera y segunda de las salidas de la capa  $m$ , con  $i=1,2,\dots,N_m$  y  $m=1,2,\dots,M$ , serán:

$$\text{Llamamos : } W_i^m(k+1) = W_i^m(k) + \mu \cdot P_i^m(k) \quad (3.101)$$

$$\begin{aligned} \frac{\partial y_{s,i}^m}{\partial \mu} &= f' \left\{ [W_i^m(k+1)]^T \cdot Y_s^{m-1} \right\} \cdot \left\{ P_i^m(k)^T \cdot Y_s^{m-1} + [W_i^m(k+1)]^T \cdot \frac{\partial Y_s^{m-1}}{\partial \mu} \right\} \\ \frac{\partial^2 y_{s,i}^m}{\partial \mu^2} &= f'' \left\{ [W_i^m(k+1)]^T \cdot Y_s^{m-1} \right\} \cdot \left\{ P_i^m(k)^T \cdot Y_s^{m-1} + [W_i^m(k+1)]^T \cdot \frac{\partial Y_s^{m-1}}{\partial \mu} \right\}^2 + \\ &+ f' \left\{ [W_i^m(k+1)]^T \cdot Y_s^{m-1} \right\} \cdot \left\{ 2 \cdot P_i^m(k)^T \cdot \frac{\partial Y_s^{m-1}}{\partial \mu} + [W_i^m(k+1)]^T \cdot \frac{\partial^2 Y_s^{m-1}}{\partial \mu^2} \right\} \end{aligned}$$

Donde, si  $m=0$  las derivadas primera y segunda de  $Y_s^m$ , serán cero.

Necesitamos también conocer  $P(k)$ , dicho valor vendrá dado por la ecuación:

$$P(k) = -\nabla J(W(k)) + \beta(k) \cdot P(k-1) \quad (3.102)$$

En el instante inicial  $P(k)$  se hace coincidir con el gradiente.

### 3.5. ALGORITMO DE APRENDIZAJE MEDIANTE GRADIENTES CONJUGADOS

El parámetro  $\beta(k)$  puede calcularse mediante la siguiente expresión:

$$\begin{aligned} \text{Llamamos : } \nabla_k &= \nabla J(W(k)) && (3.103) \\ \beta(k) &= \begin{cases} \frac{[\nabla_k - \nabla_{k-1}]^T \cdot \nabla_k}{\|\nabla_{k-1}\|^2} & \text{si } k \neq rN_r \text{ o } |\nabla_k^T \nabla_{k-1}| \leq 0.2 \cdot \|\nabla_k\| \cdot \|\nabla_{k-1}\| \\ 0 & \text{en otro caso} \end{cases} \end{aligned}$$

Para aplicar el nuevo algoritmo basta con utilizar las ecuaciones anteriores en el proceso de entrenamiento de la red.

Dichas ecuaciones están generalizadas para cualquier número de capas en la red. En nuestro caso, usaremos una red con tres capas (una sola capa oculta), por lo que basta con poner  $M=3$  y obtener simplemente las ecuaciones para la capa oculta y la de salida.

### 3.6. Uso de Weight Decay en los métodos ARNMF

En la sección 3.2 de este capítulo, vimos una amplia descripción de los métodos de selección de entradas basados en el análisis de una red neuronal Multilayer Feedforward.

Allí vimos, que para poder aplicar los métodos debíamos de partir de una red Multilayer Feedforward entrenada.

Una cuestión que ha surgido recientemente en la bibliografía, es qué algoritmo de entrenamiento debe ser usado para entrenar dicha red inicial.

Al principio (también en nuestros experimentos) se utilizó Backpropagation. La razón puede ser simplemente por simplicidad o porque es uno de los algoritmos más conocidos y usados.

Sin embargo, algunos autores [11] han usado y propuesto que se utilicen métodos de regularización (en inglés “regularization”) como el de “weight-decay”. La propuesta parece bastante razonable ya que el efecto de weight-decay es precisamente podar pesos irrelevantes, lo que en principio la llevaría a eliminar entradas innecesarias.

Podemos preguntarnos si hay alguna motivación concreta para utilizar técnicas de regularización como por ejemplo weight-decay en el proceso de entrenamiento. Veamos en que consiste básicamente un método de regularización.

Los métodos de regularización se propusieron inicialmente con el objetivo de aumentar la capacidad de generalización de una red neuronal [36]. El principio básico de los métodos de regularización es, introducir un término aditivo en la función de error que habitualmente se usa en el algoritmo de entrenamiento. Un ejemplo, es la ecuación:

$$E_{new} = Error + \lambda \cdot E_w \quad (3.104)$$

Donde *Error* es el criterio de error usual,  $E_w$  es el término de regularización y  $E_{new}$  será la nueva medida que debe ser minimizada en el aprendizaje.

La utilidad del término de regularización es diversa, pero en el caso de redes neuronales se suele utilizar para penalizar un exceso de parámetros (pesos, neuronas, etc.) en la estructura de la red neuronal. Como resultado de la penalización, durante el proceso de aprendizaje se producirá el efecto de disminuir el número de parámetros, por lo que, la estructura de la red resultante se puede considerar menos compleja. También, como resultado, se ha observado en bastantes ocasiones que aumenta la capacidad de generalización de la red.

### 3.6. USO DE WEIGHT DECAY EN LOS MÉTODOS ARNMF

Se pueden proponer multitud de términos diferentes de regularización pero para el caso de Multilayer Feedforward el término más simple es aquel que penaliza la magnitud de los pesos, más conocido como weight-decay.

El efecto de este término es que los pesos se aproximen progresivamente a cero. En este sentido, los pesos de la red que no sean útiles y el algoritmo de aprendizaje no incremente su magnitud, puede que sean finalmente eliminados, es decir, que su valor sea cero. Así podemos ver weight-decay como un método de podado de pesos.

En nuestro caso, queremos ver como afectaría el uso de weight-decay al caso de que tengamos entradas irrelevantes o redundantes. Por ejemplo, si una entrada es irrelevante no debería contribuir al proceso de clasificación, por ello, todos los pesos que estén conectados a ella deberían ser cero. Es decir, el algoritmo de aprendizaje que utilicemos debería de dirigir el valor de este tipo de pesos a cero y el efecto de usar weight-decay será incrementar esta tendencia.

A priori, podemos pues pensar que el efecto de podado de weight-decay debería ser beneficioso para los métodos de selección de entradas, ya que finalmente el número de pesos que se utilicen será menor, los más importantes. Puede incluso, que esto influya directamente sobre las entradas irrelevantes y sus pesos sean muy próximos a cero.

Así, algunos autores [11] y [36] han propuesto añadir weight-decay en el algoritmo de entrenamiento para realizar la tarea de selección de entradas.

Por ello, nosotros hemos considerado interesante comprobar esta hipótesis experimentalmente y evaluar la eficiencia de los diferentes métodos de selección de tipo ARNMF utilizando el algoritmo Backpropagation con y sin weight-decay.

La función de error que se utilizó para el caso de weight-decay fue la siguiente:

$$E = \frac{1}{P} \cdot \sum_{i=1}^P \sum_{j=1}^{Noutput} (t_{ij} - o_{ij})^2 + \lambda \cdot \sum_{k=1}^{Nweights} w_k^2 \quad (3.105)$$

Donde  $P$  es el número de patrones de entrenamiento,  $Noutput$  es el número de salidas o clases de la red y  $Nweights$  el total de pesos de la estructura.



### 3.7. Metodología

La metodología que emplearemos aquí es la habitual utilizada en redes neuronales.

El objetivo final es comparar las eficiencias que proporcionan los diferentes métodos y dar una ordenación de los métodos en función de su eficiencia relativa.

Los pasos generales a seguir en la investigación han sido los siguientes:

- *Recopilación bibliográfica y elección de los métodos de selección. Clasificación y estudio de los mismos.*
- *Recopilación y preparación de las bases de datos a utilizar.*
- *Obtención de la ordenación de entradas que suministra cada método.*
- *Obtención de los subconjuntos óptimos de entradas.*
- *Comparación de métodos entre sí.*

Vamos a ver que metodología se empleará en cada uno de los apartados.

#### ***Recopilación bibliográfica***

En primer lugar, se realizó una recopilación bibliográfica para determinar los métodos de selección de entrada más representativos que utilizaríamos en nuestros experimentos.

Una primera revisión se realizó con la ayuda de bases de datos como INSPEC ON DISK disponible en la Universidad de Valencia. En ella se puede acceder a la lectura de los abstract de los artículos para realizar una selección de aquellos que resultan de interés en nuestro estudio.

Tras la lectura y estudio detallado de los artículos que describían métodos de selección de entradas, se llegó a la selección de un total de 26 métodos diferentes: 19 basados en el análisis de la red neuronal Multilayer Feedforward y 7 basados en el conjunto de entrenamiento.

Posteriormente, se realizó una clasificación de los mismos atendiendo a su semejanza. En los apartados anteriores se ha realizado una descripción detallada de dichos métodos.

#### ***Selección y preparación de las bases de datos***

En segundo lugar, se escogieron las bases de datos o problemas sobre los cuales se iban aplicar dichos métodos de selección de entradas. Los problemas

### 3.7. METODOLOGÍA

utilizados son problemas del “UCI repository of machine learning databases” (<http://www.ics.uci.edu/~mllearn/MLRepository.html>) que están a disposición pública.

Se han utilizado un total de 15 problemas o bases de datos cuyos nombres hemos abreviado como sigue:

*Abalone (AB), Balance Scale Weight & Distance (BL), Cylinder Bands (BN), BUPA Liver Disorders (BU), Credit Approval (CR), Display1 (D1), Glass Identification (GL), Heart Disease (HE), Mushroom (LE), The Monk's Problems (M1, M2, M3), Pima Indians Diabetes (PI), 1984 U.S. Congressional Voting Records (VO), Wisconsin Diagnostic Breast Cancer (WD).*

En el primer capítulo dimos una descripción de los mismos.

Las bases de datos seleccionadas fueron adaptadas del siguiente modo. Primero se ha normalizado el rango de variabilidad de las entradas dentro del intervalo  $[0,1]$ . Esto es importante en los métodos ARNMF, ya que dicho rango tiene influencia en el valor final de los pesos conectados a las entradas y por tanto directamente sobre las medidas de relevancia dadas en los métodos utilizados.

Después, para todos los problemas, en todos los ejemplos, se introdujo una entrada adicional (la primera) generada aleatoriamente dentro del intervalo  $[0,1]$ . De esta forma, dicha entrada debería ser totalmente irrelevante, inútil, por lo que podría ser una medida de la eficiencia de los métodos, el método que considere esta entrada como muy importante será, en principio, bastante ineficiente para ese problema concreto.

Igual que en capítulo anterior, el conjunto de datos de cada problema para cada simulación que realicemos se divide en dos, un conjunto de datos para el entrenamiento y conjunto de datos para el test. Dentro de los datos de entrenamiento se ha realizado otra división entre los datos propiamente para el entrenamiento y los datos utilizados para *cross-validation*, los cuales nos permitirán escoger el punto de convergencia de las redes.

#### ***Ordenación de las entradas***

El primer resultado que queremos obtener es una ordenación de todas las entradas en función de su importancia para cada problema seleccionado y para cada método de selección de entradas utilizado.

Con estas ordenaciones podríamos estudiar cuál sería el subconjunto óptimo de entradas a usar para cada problema y método y a partir de este resultado

podríamos comparar la eficiencia de los métodos entre sí y dar una ordenación de la utilidad de los mismos (será más útil el que funcione mejor).

Para llegar a este resultado, primero hemos estudiado por separado los métodos de selección de los diferentes tipo: ARNMF y ACE.

En el caso de los métodos de tipo ARNMF, también se estudió el efecto de la introducción de weight-decay, obteniendo una ordenación de las entradas utilizando weight-decay y sin utilizarlo.

Para los métodos ARNMF con y sin weight-decay como ya vimos, es necesaria al menos una red neuronal entrenada en cada problema. Nosotros utilizamos para su entrenamiento el algoritmo Backpropagation con y sin weight-decay.

El número de unidades ocultas de la red utilizada para cada problema fue obtenido cuidadosamente mediante un procedimiento de prueba y error (igual que se hizo en el capítulo anterior).

La estructura final de una red después del procedimiento de aprendizaje, depende de ciertos factores “incontrolables”, como la inicialización de los pesos (que lo habitual es hacerla de forma aleatoria), la partición de los datos utilizada y el valor que otros parámetros tienen antes de comenzar el entrenamiento.

Ello supone que la selección de entradas también dependerá en última instancia de estos parámetros. Su eficiencia dependerá, en general, de la red neuronal concreta que se haya utilizado.

Esta situación es habitual en el campo de redes neuronales, y lo apropiado es realizar varios entrenamientos diferentes con distintas condiciones de inicialización de pesos y particiones de datos, calculando después la media de la medida deseada (eficiencia, relevancia de las entradas, etc.) con un error.

El error se evalúa utilizando la teoría de errores estándar:

$$Error = \frac{s}{\sqrt{n}} \quad (3.106)$$

En donde  $s$  es la estimación experimental de la desviación estándar de la población y  $n$  es el número de valores experimentales utilizado para calcular la media.

Entrenamos finalmente 30 redes para cada problema y base de datos con diferente inicialización de los pesos y partición de los datos.

### 3.7. METODOLOGÍA

Para cada problema, se aplicaron los diferentes métodos de ARNMF usando siempre 30 redes. Se obtienen así, para un determinado método 30 medidas de relevancia para cada entrada. El valor final de la relevancia de cada entrada se calcula hallando la media de los 30 valores, tal y como hemos dicho antes.

Pensamos que con esta metodología los resultados finales de la relevancia serán mucho más fiables.

Como resultado de la aplicación de los métodos, tendremos para cada problema y método un vector de relevancias de dimensión igual al número de entradas.

En la descripción de cada método, vimos si una entrada era más o menos importante según si la medida de la relevancia usada en el método, tuviese un valor mayor o menor.

Por ejemplo, en el método BL2 decíamos que una entrada era la más importante si el valor de la medida de la relevancia para ella era el mayor.

Con este razonamiento, podemos ordenar las componentes del vector de relevancias por orden de importancia, guardando en un nuevo vector el orden mediante los índices de la entrada a la que corresponde cada componente. Tendremos unos nuevos vectores que contienen la ordenación de las entradas en función de su importancia para cada método, la primera componente será el índice de la entrada menos importante y la última componente el índice de la más importante.

Finalmente, para cada problema y método de selección, tendremos un vector con la ordenación de todas las entradas candidatas de menor o mayor importancia.

El siguiente objetivo sería determinar el subconjunto óptimo de entradas necesario para resolver un cierto problema.

#### ***Obtención de los subconjuntos óptimos de entrada***

Los métodos de selección sólo nos permiten determinar el orden de importancia de las entradas.

El siguiente paso será determinar del conjunto total de entradas, cuáles son innecesarias, es decir, obtener el subconjunto de entradas que proporcione la mejor eficiencia o capacidad de generalización con el mínimo número de entradas.

Para ello, nosotros hemos empleado una metodología que nos permita determinar cuál será dicho subconjunto para cada problema y para cada método.

La idea es, a partir del vector de ordenación de entradas por su importancia, generar varios subconjuntos eliminando sucesivamente la entrada menos importante. Si tenemos  $N$  entradas obtendremos un total de  $N-1$  subconjuntos candidatos a ser el subconjunto óptimo.

El primer subconjunto se obtiene quitando la entrada menos importante (primera componente del vector). El segundo subconjunto se obtiene quitando las dos entradas menos importantes (primera y segunda componentes del vector de ordenación de entradas). El resto de subconjuntos se obtiene sucesivamente eliminando una entrada más en cada uno de ellos.

Para escoger el subconjunto óptimo de entre todos los generados necesitaremos medir la eficiencia que proporciona cada subconjunto (cómo de “buenas” son las entradas que incluye). Para ello, de nuevo necesitaremos entrenar una red para cada subconjunto generado y medir su capacidad de generalización.

Debemos recordar que, como hemos dicho antes, la eficiencia final de un subconjunto dependerá del clasificador que se vaya a utilizar. En nuestro caso, el clasificador de interés es Multilayer Feedforward y como ya hemos comentado antes la eficiencia obtenida en el test (medida de la capacidad de generalización) dependerá de la inicialización de los pesos y de la partición de los datos realizada en el entrenamiento.

De nuevo y por las mismas razones que antes, en lugar de entrenar una única red para cada subconjunto se realizaron varios entrenamientos diferentes con distintas condiciones de inicialización de pesos y particiones de datos, calculando después la media de la eficiencia con un error.

Finalmente, se ha entrenado un mínimo de diez redes (con diferente inicialización de pesos y particiones de datos) y se ha obtenido al final una eficiencia media (porcentaje medio de aciertos en el test) con un error, ecuación (3.106). La medida de la eficiencia resulta así mucho más significativa.

Hemos considerado que el error obtenido tenía un valor aceptable si no sobrepasaba el 3%. Si era superior, hemos entrenado 10 redes más, repitiendo este proceso hasta que el valor del error no sobrepasará dicho porcentaje.

Para el entrenamiento de estas redes se ha optado por utilizar un algoritmo diferente del estándar Backpropagation. Este algoritmo descrito en [40] y que

### 3.7. METODOLOGÍA

describimos en la sección 3.5, utiliza gradientes conjugados y optimiza de forma dinámica el valor del paso de adaptación  $\mu$  durante el proceso de entrenamiento, en lugar de tomar un valor fijo. Con esto, se consigue que la convergencia de las redes sea más rápida, por lo que se acelera entre un 10 y un 50 % el tiempo de computación.

Puesto que estamos utilizando 15 problemas diferentes y tenemos que entrenar un gran número de redes (un mínimo de 10 redes por subconjunto), el factor tiempo es bastante importante. Con este algoritmo de entrenamiento conseguimos acelerar en gran medida el proceso de aprendizaje de Backpropagation.

Una vez medida la eficiencia media que proporciona cada subconjunto para un problema y método dado, hay que comparar los resultados de todos los subconjuntos para determinar cuál es el óptimo.

Entendemos por óptimo, aquel subconjunto que proporciona la mejor capacidad de generalización, y en el caso de que haya dos subconjuntos con una eficiencia indistinguible, aquel que posee un menor número de entradas.

Por lo tanto, para obtener el subconjunto óptimo es importante poder asegurar que la diferencia entre dos eficiencias medias es estadísticamente significativa.

Con este objetivo utilizaremos el test de hipótesis *tests-t* descrito en la referencia [38]. Describimos a continuación cómo se realizará dicho proceso.

Se plantea la hipótesis nula de la igualdad de las medias de dos medidas de eficiencia (cada una correspondiente a un subconjunto de un método y problema).

Sea  $\mu_A$  y  $\mu_B$  las eficiencias medias. Planteamos la hipótesis de que son iguales.

$$H_o: \mu_A = \mu_B \quad (3.107)$$

En los casos en que dicha hipótesis pueda ser rechazada, se puede asegurar estadísticamente que la diferencia de valores es significativa.

El estadístico  $t$ , a calcular para dicho test es el correspondiente a la comparación de dos medias de poblaciones distintas y con varianzas desconocidas. Su valor depende de las dos medias y las diferencias se hacen más distinguibles, conforme disminuyen los errores.

Se calcula el estadístico  $t_o$  según la ecuación (3.108) y si su valor es mayor que un número que depende del grado de significancia, se concluye que los resultados son distinguibles.

$$t_o = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{Error_1^2 + Error_2^2}} \quad (3.108)$$

Los grados de libertad de la distribución  $t$  a utilizar se calculan de acuerdo con la siguiente ecuación:

$$v = \frac{(Error_1^2 + Error_2^2)^2}{\frac{Error_1^4}{n_1 + 1} + \frac{Error_2^4}{n_2 + 1}} - 2 \quad (3.109)$$

Como se comentó anteriormente, las diferencias entre dos medias se hacen más distinguibles, conforme disminuyen sus errores.

A su vez, los errores disminuyen conforme se toman más medidas. Como se puede ver en la ecuación (3.106), la desviación estándar permanecerá aproximadamente constante (puesto que es una estimación de la desviación estándar de la población) y el número  $n$  de medidas aumentará.

Estos factores son los que se utilizarán para determinar el número de redes a entrenar y utilizar en cada caso.

El objetivo es hacer las diferencias entre medias estadísticamente significativas, y para ello el número de medidas a tomar dependerá de la variabilidad (la desviación estándar de la población) que se obtenga en cada caso. Es imposible determinar dicho número a priori y hay que hacerlo después de observar los primeros resultados experimentales.

En las anteriores ecuaciones correspondientes al *test-t*, está implícita la suposición de que la distribución de la población de las medidas es normal. Dicha suposición es bastante razonable en general.

El nivel de significancia tomado en el estadístico *test-t* es 0.1.

Veamos algunos ejemplos de selección del subconjunto óptimo.

Por ejemplo, supongamos los valores de la capacidad de generalización de  $58.7 \pm 0.8$  con 1 entrada omitida, y  $58.1 \pm 0.7$  con 5 entradas omitidas. Encontramos con la prueba de hipótesis *test-t* que son claramente indistinguibles, por lo que nos quedaríamos con el que tiene más entradas omitidas (el segundo caso).

### 3.7. METODOLOGÍA

En otro caso, con  $58.1 \pm 0.7$  con 5 entradas omitidas, y  $55 \pm 2$  con 7 entradas omitidas, las capacidades de generalización son claramente distinguibles y es mejor la primera, por lo que nos quedaríamos con ese subconjunto.

En definitiva, si las capacidades de generalización son distinguibles, escogemos el subconjunto que proporcione la mayor, y si son indistinguibles, el subconjunto que tenga menor número de entradas (más entradas omitidas).

El proceso será partir del subconjunto de entradas formado por una entrada omitida y compararlo con el de dos entradas omitidas. Nos quedaremos con subconjunto que proporcione mayor eficiencia media, o en caso de que sean indistinguibles con el que posea menor número de entradas (más entradas omitidas). Después, compararemos el subconjunto seleccionado con otro, e iremos repitiendo el proceso hasta haber pasado por todos, siendo el último conjunto seleccionado el que será finalmente el subconjunto óptimo.

Al final del proceso tendremos un subconjunto óptimo de entradas para cada método y problema.

#### ***Comparación de métodos entre sí***

Finalmente, podemos comparar la eficiencia entre los distintos métodos de selección de entradas en un mismo problema, mediante la comparación de la eficiencia media de sus subconjuntos óptimos.

Aplicando de nuevo el estadístico *test-t*, vamos comparando los métodos de dos en dos, concluyendo en cada comparación que será mejor el método que proporcione mayor capacidad de generalización y que en caso de que no sean distinguibles, aquel que contiene menor número de entradas (más entradas omitidas).

De esta forma, comparando los métodos dos a dos en cada problema podemos obtener una tabla en donde se describa si un método funciona mejor, peor o igual que otro en cada problema. Para obtener unas conclusiones más generales sobre la eficiencia relativa de dos métodos, lo que se ha hecho ha sido emplear el criterio simple de que un método es mejor que otro si funciona mejor en un mayor número de problemas.

Con esta metodología, al final se puede obtener una ordenación aproximada de todos los métodos de acuerdo con su eficiencia.

Este proceso es bastante laborioso, iremos obteniendo los resultados progresivamente, es decir compararemos cada método con todos los demás. En el apéndice II de la Tesis Doctoral se incluyen todos estos resultados.



## 3.8. Resultados Experimentales

Como ya se indicó en el apartado de metodología, el primer paso después de escoger los métodos de selección de entradas, fue seleccionar los problemas donde aplicarlos. Se escogieron un total de 15 problemas del repositorio UCI que abarcan una gran variedad de problemas de clasificación. Se normalizaron las entradas y para todos los datos se introdujo una entrada irrelevante de forma aleatoria (la primera entrada).

Para los métodos de ARNMF, entrenamos con el algoritmo Backpropagation 30 redes con diferente inicialización de los pesos y partición de los datos entre conjunto de entrenamiento, cros-validación y test.

La misma operación se realizó para los métodos ARNMF con weight-decay, utilizando el algoritmo Backpropagation con el término de regularización de weight-decay descrito antes.

La arquitectura de red neuronal que entrenamos viene determinada por el problema concreto (capa de entrada y salida). El número de unidades ocultas se escogió en cada problema de forma cuidadosa mediante un proceso de prueba y error. Los valores que se tomaron para los parámetros de entrenamiento fueron los siguientes: intervalo de inicialización de pesos 0.1, momento 0.05, y paso de adaptación 0.1.

Aplicando los métodos a las 30 redes, se obtendrá 30 medidas de relevancia para cada entrada y método. Calculando la media de los 30 valores para cada entrada se obtiene el valor final de la relevancia de cada entrada para cada método y problema.

Para los métodos ACE se obtiene las relevancias de cada entrada, método y problema sin más que aplicar el método sobre los ejemplos de entrenamiento.

Para el método denominado BA, se utilizaron seis valores diferentes para el parámetro  $\beta$ , por lo que para este método tendremos seis resultados diferentes. Los valores que escogimos fueron: 0.5, 0.6, 0.7, 0.8, 0.9 y 1. Al final obtendremos seis resultados diferentes que llamaremos en las tablas de resultados como BA seguido del valor de  $\beta$ .

Como dijimos en la metodología obtendremos para cada problema y método un vector cuya dimensión es igual al número de entradas, donde la primera componente será la relevancia media correspondiente a la primera entrada, la segunda la relevancia media de la segunda entrada y así sucesivamente.

### 3.8. RESULTADOS EXPERIMENTALES

Por ejemplo, para el problema BUPA Liver Disorders (BU) y el método de selección BL2 obtenemos el vector de relevancias medias de la Tabla 1.

**Tabla 1. Vector de relevancias para el problema BU, método BL2.**

Entrada	1	2	3	4	5	6	7
Media	0.048786	0.165080	0.379562	1.355403	1.108591	0.944618	0.199835
Desviación	0.044321	0.102213	0.208929	0.501284	0.394386	0.324958	0.092751

A partir de estos vectores y aplicado el criterio de importancia de entradas dado por cada método, podemos obtener, un nuevo vector cuyas componentes sean los índices de las entradas ordenadas por importancia.

La primera componente sería el índice de la entrada cuya importancia es menor y la componente más alta sería el índice de la entrada de mayor importancia.

Por ejemplo, en el método BL2 decíamos que una entrada era la más importante si el valor de la medida de la relevancia para ella era el mayor. Bastaría con ordenar los valores de las relevancias y guardar en un vector los índices de las entradas que corresponden a esta ordenación, de forma que primero estará el índice de la entrada con menor relevancia, y el último sería el que corresponde a una mayor relevancia.

Por ejemplo, para el problema BU y método BL2 utilizando los valores de la tabla anterior podemos ordenarlos y tendremos los resultados de la Tabla 2.

**Tabla 2. Ordenación de entradas para el problema BU, método BL2.**

Componente del vector	Ordenación de entradas de menor a mayor importancia						
	1 (menor)	2	3	4	5	6	7 (mayor)
Entrada	1	7	2	3	6	5	4

En la Tabla 1, vemos que la entrada 4 tiene un valor de relevancia de 1.355, que resulta el más grande, por lo que la entrada 4 será la más importante. El valor más pequeño de la relevancia vemos que corresponde con la entrada 1 por lo cual esta será la menos importante.

En principio, en este ejemplo, vemos que la entrada menos importante es precisamente la entrada irrelevante que nosotros hemos introducido, con lo cual en principio parece un método que funciona bien para este problema.

Repitiendo este proceso con todos los métodos y con todos los problemas, podemos obtener las diferentes ordenaciones de las entradas. En el apéndice II de la Tesis Doctoral damos, para cada problema todas las ordenaciones para los diferentes métodos, Tablas 11-25.

En cada tabla (problema) aparece para cada método (fila) la ordenación de las entradas según su importancia. Igual que en la Tabla 2, la columna numerada como 1 será la entrada menos importante, y según nos movemos hacia la derecha de la tabla las entradas serán más importantes.

Se puede observar que algunos métodos no han podido aplicarse en ciertos problemas.

El método TEKE no pudo aplicarse en tres problemas ya que se obtenían todos los valores de la relevancia nulos. El método YOU no fue aplicable en 6 de los quince problemas utilizados, debido a problemas de división por cero en los cálculos. En tres de los problemas donde aplicamos el método GD\_DS hemos encontrado que la matriz de transinformación es singular y el método no se pudo aplicar. Hemos detectado que el método RLF no es aplicable en 4 de los 15 problemas. Por ejemplo, no se pudo aplicar en problemas artificiales en los que sólo disponemos de un patrón de entrada por cada clase, como en *DISPLAY1*. Por último, el método denominado FUZ no fue aplicable en 4 de los 15 problemas, ya que se obtenían excepciones cero.

Ahora, a partir de estas ordenaciones, necesitamos obtener los subconjuntos de entradas óptimos en cada caso. Para ello, partimos como explicamos en la metodología de las ordenaciones anteriores y para cada vector obtenemos los diferentes subconjuntos que resultan de ir eliminando sucesivamente una entrada (la menos importante) en este vector.

Por ejemplo, para el caso anterior del problema BU y el método BL2 los subconjuntos posibles serían los que se dan en la Tabla 3.

**Tabla 3. Subconjuntos de entradas para el problema BU, método BL2**

Nº entradas omitidas	Operación realizada	Subconjuntos posibles
1	<i>omito la entrada 1</i>	{7, 2, 3, 6, 5, 4}
2	<i>omito la entrada 1 y 7</i>	{2, 3, 6, 5, 4}
3	<i>omito la entrada 1, 7 y 2</i>	{3, 6, 5, 4}
4	<i>omito la entrada 1, 7, 2 y 3</i>	{6, 5, 4}
5	<i>omito la entrada 1, 7, 2, 3 y 6</i>	{5, 4}
6	<i>omito la entrada 1, 7, 2, 3, 6 y 5</i>	{4}

Para un mismo problema, puedo tener dos subconjuntos iguales (las mismas entradas omitidas) para métodos distintos. En ese caso no necesito hacer dos veces las mismas simulaciones para comprobar la eficiencia, ya que se trata de los mismos subconjuntos (el mismo problema) aunque los métodos sean distintos.

### 3.8. RESULTADOS EXPERIMENTALES

Por ejemplo, para el problema BU, el subconjunto de una entrada omitida para los métodos BL1, BL2, CLO, PRI, DEV, LEE, SAN, TEKA, TEKB, DER2, MAO, es el mismo (la entrada 1). Para el método TEK la entrada omitida no es la misma (es la 7) por lo que el subconjunto será diferente.

Tenemos que decidir de estos subconjuntos cuál es el que proporciona mayor capacidad de generalización con el menor número de entradas, y este será el subconjunto que tomaremos como óptimo para cada método y problema.

Aplicamos la metodología que ya explicamos, para poder comparar los subconjuntos. Primero, tendremos que evaluar sus eficiencias entrenando al menos una red neuronal con cada uno. Lo que se hizo, fue para cada subconjunto de entradas entrenar un mínimo de 10 redes, y luego obtener la media entre las eficiencias de las mismas con un error. Este valor será el que se utilizara para comparar la eficiencia de los diferentes subconjuntos.

En el apéndice II de la Tesis Doctoral se encuentra para cada problema una tabla con las eficiencias de cada método para todos los posibles subconjuntos, Tablas 34-48.

Para cada problema hemos realizado una tabla con las capacidades de generalización medias con su error obtenidas para cada subconjunto de entradas. Cada fila de la tabla representa un método de selección de entradas y cada columna representa uno de los posibles subconjuntos, donde la numeración representa cuántas entradas han sido omitidas del conjunto inicial de entradas.

En la Tabla 4 se da un extracto de la tabla general para el problema AB (Tabla 34 del apéndice II de la Tesis Doctoral).

**Tabla 4. Capacidad de generalización para el problema AB (BL1, TEKE).**

Método	1	2	3	4	5	6	7	8
BL1	51±3	53±2	48±3	53±2	52±2	52±2	55.5±1.7	46±3
TEKE	58.7±0.8	55±2	57.9±1.0	57.5±0.8	58.1±0.7	53.9±1.8	50±3	46±3

La primera columna será la eficiencia con su error para el caso del subconjunto con una entrada omitida, la segunda columna para el caso de dos entradas omitidas y así sucesivamente hasta la última que tendrá ocho entradas omitidas (red con una sola entrada). Recordemos que AB es un problema con ocho entradas más la entrada irrelevante que nosotros añadimos para realizar los experimentos (nueve en total).

A partir de estas tablas y aplicando la prueba estadística de hipótesis *test-t* que explicamos antes, se puede seleccionar para cada método y problema el subconjunto óptimo de entradas.

Por ejemplo, para el problema AB (Tabla 34 del apéndice II de la Tesis Doctoral), podemos ver para el método BL1 (primera fila de la Tabla 4) que la capacidad de generalización para el caso de 7 entradas omitidas es  $55.5 \pm 1.7$ . Dicho valor es mejor y distinguible del resto de eficiencias para dicho método, por lo que el subconjunto de entradas  $\{7,9\}$  (ver Tabla 11 del apéndice II de la Tesis) para el problema AB es el óptimo de acuerdo con los resultados del método BL1.

También, para AB y el método TEKE (segunda fila de la Tabla 4), el subconjunto que parece obtener mejor capacidad de generalización es el de una entrada omitida con un valor  $58.7 \pm 0.8$ . Pero, en realidad el subconjunto de cinco entradas omitidas proporciona una capacidad de generalización de  $58.1 \pm 0.7$  y ambos valores son indistinguibles. Por ello, de entre los dos hemos de seleccionar como óptimo el que utiliza menor número de entradas, es decir el de 5 entradas omitidas, el subconjunto  $\{6,8,3,9\}$  con capacidad de generalización de  $58.1 \pm 0.7$ .

Podemos comparar a partir de las Tablas 34-48, la eficiencia de dos métodos distintos de selección de entradas en el mismo problema mediante la comparación de sus subconjuntos óptimos (el mejor de cada método). De nuevo aplicamos la hipótesis *test-t* sobre las eficiencias, y será mejor un subconjunto si proporciona mayor capacidad de generalización y en el caso de no ser distinguibles aquel que contiene un menor número de entradas.

Compararemos los métodos de dos en dos.

Por ejemplo, tomemos de nuevo el problema AB. Para el método TEKE el subconjunto óptimo de 5 entradas omitidas tiene una capacidad de generalización de  $58.1 \pm 0.7$ . Para el mismo problema y el método BL1 la capacidad de generalización del subconjunto óptimo (con 7 entradas omitidas) es  $55.5 \pm 1.7$ . Los valores de ambos porcentajes son distinguibles por lo que consideraremos que la eficiencia del método TEKE es superior que la BL1 para el problema AB.

De esta forma, comparando los métodos dos a dos en cada problema podemos obtener otra tabla en donde se describa si un método es mejor, peor o igual que los demás en cada problema.

En el apéndice II de la Tesis Doctoral se dan los resultados en la Tabla 57 de todas las comparaciones entre los métodos para todos los problemas.

Así comparemos la eficiencia de un método con la de otro iremos anotando los problemas en los que el primer método es *mejor* que en el segundo, los problemas en los que es *peor* y los problemas en los que es *igual* (no son distinguibles).

### 3.8. RESULTADOS EXPERIMENTALES

Veamos algún ejemplo extraído de la Tabla 57 del apéndice II de la Tesis para el caso de los métodos BA 0.5, BA 0.6 y SCH. En la Tabla 5 (extracto de la 57), tenemos la comparación de la eficiencia proporcionada por tres métodos diferentes: BA 0.5 (BA con el valor de  $\beta=0.5$ ), BA 0.6 (BA con el valor de  $\beta=0.6$ ) y el método SCH.

**Tabla 5. Comparación de eficiencias entre métodos BA 0.5, BA 0.6 y SCH.**

		Método BA 0.5
BA 0.6	Mejor BA 0.5:	HE, VO
	Iguales:	AB, BL, BN, BU, CR, D1, GL, LE, M1, M2, M3, PI, WD
	Peor BA 0.5:	
SCH	Mejor BA 0.5:	BL, BN, CR, D1, GL, HE, M1, M3, PI, VO
	Iguales:	M2
	Peor BA 0.5:	AB, BU, LE, WD

Una forma de obtener una conclusión sobre la eficiencia general relativa de los métodos, consiste en contar el número de problemas en el que funciona mejor un determinado método. Por ejemplo, en la Tabla 5, se presenta la comparación del método BA 0.5 con dos métodos BA 0.6 y SCH. Se puede observar que el método BA 0.5, obtuvo una eficiencia mejor que el de SCH en 10 problemas (BL, BN, CR, D1, GL, HE, M1, M3, PI, VO), la eficiencia fue la misma en 1 problema (M2) y fue mejor para SCH en 4 problemas (AB, BU, LE, WD). Podemos concluir que la eficiencia general de BA 0.5 es superior a la de SCH ya que BA 0.5 funciona mejor en un número bastante superior de problemas.

De igual forma, si comparamos las eficiencias entre los métodos BA 0.5 y BA 0.6, vemos que la eficiencia del primero es ligeramente superior al segundo ya que en dos problemas funciona mejor y en el resto los métodos funcionan igual.

Siguiendo esta metodología, con los resultados de la Tabla 57 del apéndice II de la Tesis, se puede llegar a una ordenación aproximada de todos los métodos de acuerdo con su eficiencia. A continuación mostramos la ordenación que hemos obtenido a partir de dicha tabla.

<p>UTA &gt; GD_DS = TEKA &gt; BL2 &gt; DEV &gt; TEKB = DER2 = MAO &gt; CLO &gt; BL1 =          =BOW = YOU = BA 0.5 &gt; RLF &gt; BA 0.7 = BA 0.9 = BA 1 &gt; BA 0.6 =BA 0.8 =SET&gt;          &gt;PRI = SAN &gt; CIB = TEKE &gt; CHI &gt; TEK &gt; TEKC &gt; FUZ &gt; DER3 &gt; SCH &gt; LEE</p>
--

Hemos distinguido en verde los métodos basados en el análisis del conjunto de entrenamiento (ACE), y en azul los basados en el análisis de la red entrenada (ARNMF).

Con un simple vistazo se puede concluir fácilmente que la variabilidad en las eficiencias de los métodos basados en la red entrenada (ARNMF) es superior.

También se observa, que la mayoría de los métodos ACE se agrupan desde el centro a la parte más baja de la ordenación, por lo que podemos concluir que este tipo de métodos tiene una eficiencia inferior que los mejores métodos de ARNMF.

Por ello, podemos asegurar que los métodos ACE son menos apropiados para seleccionar las entradas de una red neuronal.

Sin embargo, existe una excepción, el método GD\_DS basado en la utilización de la Distancia GD y situado en el segundo lugar de la clasificación.

Si consideramos el coste computacional de los métodos, vemos que los métodos ACE tienen un coste muy inferior a los de los métodos ARNMF, ya que estos últimos requieren el entrenamiento previo de al menos una red con todas las entradas candidatas.

Si sopesados ambos factores, eficiencia y coste computacional, podemos considerar que el método GD\_DS es posiblemente el más interesante y apropiado.

Sin embargo, dicho método presenta un problema, para poder ser aplicado, la matriz de transformación debe ser no singular puesto que tenemos que calcular su inversa.

En nuestros experimentos hemos encontrado matrices singulares y el método no fue aplicable en 3 de los 15 problemas. En dicho caso, la mejor alternativa sería la utilización del método UTA asumiendo un mayor coste computacional.

#### ***Análisis de los métodos ARNMF con y sin WEIGHT-DECAY.***

Por otro lado, en el caso de los métodos de tipo ARNMF también se estudió el efecto de la introducción de weight-decay. En nuestros experimentos, utilizaremos 17 métodos del tipo ARNMF, en los cuales se ha introducido el término de regulación de weight-decay durante el entrenamiento de la red. Se obtuvieron resultados en ocho problemas diferentes: BN, BU, CR, GL, HE, PI, VO y WD.

El primer paso fue introducir en el proceso de entrenamiento de las redes iniciales, Backpropagation con una función de error modificada que incluya el término de regularización, tal y como explicamos en el apartado 3.6.

### 3.8. RESULTADOS EXPERIMENTALES

Se entrenaron 10 redes diferentes (con diferente inicialización de los datos y pesos), obteniendo 10 medidas de relevancia para cada entrada y método y se calculo como medida final la media de las mismas.

El parámetro  $\lambda$  de weight-decay, se determinó por un procedimiento de prueba y error escogiendo el más adecuado en cada caso.

En los resultados experimentales para distinguir entre usar o no weight-decay, denotaremos a los métodos en los que usemos weight-decay con su nombre y la cadena “& WD”.

Nuestro primer objetivo fue analizar si el uso de weight-decay disminuye las diferencias existentes entre las ordenaciones de los diferentes métodos para un cierto problema. La hipótesis parece razonable ya que el objetivo de weight-decay es podar pesos.

En la Tabla 6, contamos primero el número total de subconjuntos de cada base de datos para los 17 métodos. Para ello, multiplicamos el número de entradas de la base de datos menos una por el número de métodos. Dicho resultado aparece en la tercera columna de la Tabla 6 (etiquetado como N° TOTAL). Después, contamos el número de diferentes subconjuntos de cada base de datos en los dos casos, 17 métodos ARNMF y 17 métodos ARNMF & WD. Los resultados se encuentran en la primera y la segunda columnas respectivamente y representan el porcentaje de subconjuntos diferentes con respecto del número total de ellos. En la Tabla 6, hemos sombreado los métodos que proporcionan un menor número de subconjuntos diferentes para cada base de datos.

**Table 6. Porcentaje de subconjuntos de entradas distintos de cada problema.**

MÉTODOS	ARNMF	ARNMF& WD	N° TOTAL
BANDS	49.5%	55.7%	663
BUPA	30.2%	40.6%	96
CREDIT	53.8%	47.9%	240
GLASS	41.2%	33.1%	160
HEART	57.0%	41.2%	221
PIMA	31.6%	32.4%	136
VOTING	58.8%	40.8%	272
WDBC	47.3%	53.9%	510

En la Tabla 6 vemos que el resultado depende de la base de datos. Por ejemplo, para los problemas CR, GL, HE y VO el uso de weight-decay hace que las ordenaciones de los métodos se parezcan más (menos subconjuntos diferentes). Con BN, BU y WD el resultado es el contrario.



Veamos ahora cuáles son las ordenaciones de las entradas para el caso de usar weight-decay. Aplicando la misma metodología que hemos seguido en los métodos ARNMF sin weight-decay, obtendremos para cada problema y método de selección un vector con la ordenación por importancia de las entradas candidatas.

En el apéndice II de la Tesis Doctoral, mostramos en las Tablas 26-33 para cada problema las ordenaciones para los diferentes métodos *ARNMF & WD*.

A partir de estas ordenaciones, obtenemos de nuevo los subconjuntos de entradas y medimos la capacidad de generalización de los mismos.

En el apéndice II de la Tesis Doctoral en las Tablas 49-56, se encuentran para cada problema las eficiencias medias de todos los subconjuntos, para los diferentes métodos ARNMF & WD.

Tenemos pues las eficiencias tanto para los 17 métodos ARNMF, como para los 17 métodos ARNMF & WD. En la Tabla 7 reflejamos los resultados para los 8 problemas estudiados.

Para cada fila de la Tabla 7 tenemos los resultados de un método seguidos de los resultados del mismo método con el uso de weight-decay. Por ejemplo, las dos primeras filas representan el método BL1 y el mismo método con weight-decay BL1&WD respectivamente.

En las columnas, hay ocho problemas diferentes: BN, BU, CR, GL, HE, PI, VO y WD y para cada uno tenemos dos columnas. De dichas columnas la primera (encabezada por %) representa la capacidad de generalización del subconjunto óptimo (porcentaje de correcta clasificación), y la segunda (encabezada por *N*) el número de entradas que contiene dicho subconjunto.

Los subconjuntos óptimos para el caso de weight-decay se obtuvieron de la misma forma que en el resto de métodos: a partir de las Tablas 49-56, y aplicando la prueba estadística de hipótesis *test-t* para determinar cuando eran distinguibles los resultados.

De nuevo, para analizar los resultados de qué métodos son mejores hay que analizar si las eficiencias son o no distinguibles, y en caso de no serlo, seleccionar el de menor número de entradas.

Con los sombreados hemos tratado de mostrar las diferencias entre los resultados de un mismo método con y sin la utilización de weight-decay, el mejor resultado se encuentra en una celda sombreada.

### 3.8. RESULTADOS EXPERIMENTALES

**Tabla 7. Eficiencias de 17 métodos ARNMF con y sin weight-decay.**

<b>MÉTODO</b>	<b>BN</b>		<b>BU</b>		<b>CR</b>		<b>GL</b>	
	<b>%</b>	<b>N</b>	<b>%</b>	<b>N</b>	<b>%</b>	<b>N</b>	<b>%</b>	<b>N</b>
BL1	66.6±1.2	18	69.5±0.2	5	88±0	1	60±3	1
BL1 & WD	68.6±1.0	12	65±2	5	88±0	1	60±3	1
BL2	70.6±0.7	5	67.4±1.6	4	89.4±0.2	8	60±3	1
BL2 & WD	68.6±1.0	12	67±2	6	88±0	1	60±3	1
CLO	66.4±1.3	15	69.5±0.2	5	88±0	1	60±3	1
CLO & WD	68.6±1.0	12	65±2	5	88±0	1	60±3	1
PRI	66.6±1.2	18	69.5±0.2	5	88±0	1	60±3	1
PRI & WD	68.6±1.0	12	65±2	5	88±0	1	60±3	1
DEV	69.4±0.5	6	69.5±0.2	5	88±0	1	60±3	1
DEV & WD	68.6±1.0	12	67.4±1.6	4	88±0	1	60±3	1
LEE	64.5±0.8	22	60.1±1.3	6	76.0±0.4	12	41.9±1.5	6
LEE & WD	71.6±0.7	24	69.5±0.2	5	76.7±0.5	3	52±4	10
TEK	67.9±1.1	10	69.1±0.3	6	88±0	1	56±3	2
TEK & WD	68.2±0.9	15	60.1±1.3	6	88±0	1	60±3	1
SAN	67.4±0.9	21	69.5±0.2	5	88±0	1	60±3	1
SAN & WD	68.6±1.0	12	65±2	5	88±0	1	60±3	1
TEKA	69.4±1.8	4	69.5±0.2	5	89.3±0.2	9	60±3	1
TEKA & WD	72.0±0.9	6	67±2	6	88±0	1	60±3	1
TEKB	69.0±0.6	6	67.4±1.6	4	89.05±0.16	10	56±3	2
TEKB & WD	68.6±1.0	12	67±2	6	88±0	1	60±3	1
DER2	67.2±0.5	6	67.4±1.6	4	88.25±0.08	5	56±3	2
DER2 & WD	69.8±1.1	2	67±2	6	88±0	1	56±3	2
DER3	68.9±0.3	3	63.5±1.7	6	88.0±0.3	14	43.6±1.3	3
DER3 & WD	74.0±1.4	29	69±2	6	88±0	1	49.9±1.0	3
MAO	73.7±0.5	8	69.1±0.3	6	88.0±0.4	14	60±3	1
MAO & WD	70.7±1.5	4	60.6±0.8	4	88±0	2	60±3	1
UTA	73.1±1.0	4	69.5±0.2	5	88.30±0.13	6	60±3	1
UTA & WD	72.53±0.12	2	65±2	5	88±0	1	60±3	1
CIB	73.7±0.5	8	69.1±0.3	6	87.70±0.17	12	54±3	2
CIB & WD	74.6±0.5	6	63.5±1.7	6	87.9±0.4	12	60±3	1
TEKE	---	--	63.1±1.2	5	---	--	54±3	4
TEKE & WD	---	--	68.3±0.4	6	---	--	60±3	1
YOU	73.6±1.0	5	---	--	88.3±0.3	11	---	--
YOU & WD	70.1±0.9	14	---	--	88±0	1	---	--

**Tabla 7. Eficiencias de 17 métodos ARNMF con y sin weight-decay (continuación).**

<b>MÉTODO</b>	<b>HE</b>		<b>PI</b>		<b>VO</b>		<b>WD</b>	
	<b>%</b>	<b>N</b>	<b>%</b>	<b>N</b>	<b>%</b>	<b>N</b>	<b>%</b>	<b>N</b>
BL1	79.1±1.0	8	77.0±0.2	5	93.1±1.6	9	99.63±0.17	12
BL1 & WD	79.1±1.0	8	77.0±0.2	5	95.6±0.5	6	99.63±0.17	12
BL2	81.3±0.8	10	77.0±0.3	3	95.9±0.2	12	97.7±1.8	11
BL2 & WD	79.1±1.0	8	77.0±0.2	5	95.6±0.5	6	99.63±0.17	12
CLO	80±2	6	77.0±0.2	5	94.7±0.4	10	99.63±0.17	12
CLO & WD	79.1±1.0	8	77.0±0.2	5	95.6±0.5	6	99.63±0.17	12
PRI	79.1±1.0	8	77.0±0.2	5	93.1±1.6	9	99.63±0.17	12
PRI & WD	79.1±1.0	8	77.0±0.2	5	95.6±0.5	6	99.63±0.17	12
DEV	81.3±0.8	10	77.0±0.3	3	95.9±0.2	12	99.25±0.13	17
DEV & WD	79.1±1.0	8	77.0±0.2	5	95.6±0.5	6	99.63±0.17	12
LEE	76.9±1.8	11	69.7±0.7	7	92.1±0.4	11	98.4±0.3	21
LEE & WD	72±2	12	70.8±0.9	4	92.2±0.6	12	98.2±0.4	25
TEK	79.9±0.8	5	76.16±0.18	6	94.9±0.3	16	99.50±0.18	19
TEK & WD	80.8±0.3	11	77.0±0.2	5	94±0	3	99.38±0.16	18
SAN	78±3	7	76.76±0.15	6	93.1±1.6	9	99.63±0.17	12
SAN & WD	79.1±1.0	8	77.0±0.2	5	95.6±0.5	6	99.63±0.17	12
TEKA	81.3±0.8	10	77.0±0.3	3	95.9±0.2	12	99.13±0.17	9
TEKA & WD	79.1±1.0	8	77.32±0.18	3	95.6±0.5	6	99.63±0.17	12
TEKB	79.4±1.8	5	77.0±0.3	3	95.0±0.3	9	99.63±0.17	12
TEKB & WD	79.1±1.0	8	77.0±0.2	5	95.6±0.5	6	99.63±0.17	12
DER2	82.9±0.5	5	77.0±0.2	5	95.40±0.16	6	98.94±0.10	10
DER2 & WD	77±3	5	77.0±0.2	5	95.6±0.5	6	98.63±0.16	16
DER3	74.8±0.5	5	76.3±0.3	8	92.3±1.5	13	98.4±0.5	27
DER3 & WD	72±3	10	78.5±0.4	8	93.4±0.5	9	99.31±0.06	26
MAO	79.3±1.1	8	76.16±0.18	6	95.50±0.17	6	99.00±0.10	9
MAO & WD	76.3±0.6	12	75.52±0.10	3	95.2±0.3	7	99.00±0.10	9
UTA	76±3	5	77.32±0.18	3	96±0	5	98.94±0.16	11
UTA & WD	76.4±0.5	2	77.32±0.18	3	95.6±0.5	6	99.38±0.16	23
CIB	79.3±1.1	8	76.16±0.17	5	95.50±0.17	7	99.25±0.13	18
CIB & WD	77±3	5	77.0±0.2	5	95.7±0.3	5	99.25±0.13	18
TEKE	76.2±0.5	7	70.8±0.9	5	93.7±0.2	16	99.50±0.08	29
TEKE & WD	78.7±0.5	11	75.76±0.14	2	94±0	3	98.50±0.19	21
YOU	79.9±0.8	5	76.52±0.12	5	95.50±0.17	7	99.50±0.15	20
YOU & WD	77.1±0.7	6	77.32±0.18	3	95.6±0.3	7	92±3	18

### 3.8. RESULTADOS EXPERIMENTALES

No hemos sombreado los casos en los que las eficiencias son indistinguibles, lo cual supone que las capacidades de generalización son indistinguibles y además el número de entradas es el mismo.

Los criterios para afirmar que una eficiencia es mejor que otra son los mismo de antes, primero se consideran los porcentajes y en el caso de que sean indistinguibles, el número de entradas.

Por ejemplo, para el problema BN en las celdas de BL1, podemos encontrar una eficiencia de  $66.6 \pm 1.2$  con  $N=18$ , y en las celdas de BL1&WD  $68.6 \pm 1.0$  con  $N=12$ . Los porcentajes son distinguibles y la eficiencia con weight-decay es superior, por lo que dicha celda está sombreada.

Otro ejemplo, sería el caso de los resultados del método de TEKA para el problema BN. En la fila TEKA tenemos una eficiencia de  $69.4 \pm 1.8$  con  $N=4$ , y en la fila TEKA&WD de  $72.0 \pm 0.9$  con  $N=6$ . En este caso, los porcentajes son indistinguibles (la diferencia no es significativa), por lo que observando los números de entradas, el de TEKA sin weight-decay es menor y consideramos que su eficiencia ha sido superior, por lo cual lo sombreamos.

A primera vista, en la Tabla 7 no se puede observar una clara mejoría por la utilización de weight-decay en combinación con los métodos de selección de entradas. Los resultados parecen depender en gran medida del método e incluso del problema concreto. Por ejemplo, para el caso de CIB hay una mejora en cinco de un total de ocho problemas y un empeoramiento en uno. En este caso el uso de weight-decay ha aumentado en general la eficiencia.

Sin embargo, para el método DER2 el uso de weight-decay empeoró la situación: en cuatro problemas fue peor y solo en uno mejor, en el resto indistinguible.

Este hecho supone ya un primer motivo en contra de la utilización de weight-decay, porque no hay una mejoría clara de la eficiencia y mediante su uso tendremos la complejidad adicional de determinar el parámetro  $\lambda$ , mediante un procedimiento de prueba y error.

Analizando más detenidamente los resultados, podemos obtener conclusiones más interesantes sobre lo que realmente ocurre.

En la Tabla 8 hemos representado los métodos ordenados en función de su eficiencia. En la primera fila tenemos el método UTA (el más eficiente), y en la última LEE (el menos eficiente). En las columnas se incluye el número de problemas en los que el método de selección de entradas ha funcionado mejor,

igual o peor mediante la incorporación de weight-decay como algoritmo de aprendizaje para el entrenamiento de la red a analizar.

Recordemos que para los métodos de selección de entradas ARNMF teníamos la siguiente ordenación de los mismos según su eficiencia.

**UTA > TEKA > BL2 > DEV > TEKB = DER2 = MAO > CLO > BL1 =**  
**=YOU > PRI = SAN > CIB = TEKE > TEK > DER3 > LEE**

En la Tabla 8 hemos sombreado la celda (o celdas) de cada fila en donde el número de problemas es el mayor.

*Tabla 8. Comparación de eficiencias con y sin WD, número de problemas.*

	Mejor con WD	IGUAL	Peor con WD
UTA	2	2	4
TEKA	2	2	4
BL2	2	1	5
DEV	1	3	4
TEKB	3	0	5
DER2	1	3	4
MAO	1	2	5
CLO	2	4	2
BL1	2	5	1
YOU	3	1	2
PRI	2	5	1
SAN	4	2	2
CIB	5	2	1
TEKE	5	0	1
TEK	3	1	4
DER3	7	0	1
LEE	4	0	4

Podemos ver que para los primeros métodos, UTA, TEKA, BL2, DEV, TEKB, DER2 y MAO, la utilización de weight-decay provoca un deterioro en la eficiencia. En los métodos CLO, BL1 y PRI la influencia de usar weight-decay no es apreciable. Por último, en los métodos SAN, CIB, TEKE y DER3 (los de peor eficiencia) hay una clara mejoría mediante el uso de weight-decay.

A partir de estos resultados, parece que el uso de weight-decay mejora la eficiencia de los peores métodos de selección de entradas, deteriora la eficiencia de los mejores métodos y no afecta al resto, los intermedios. En este sentido, su uso disminuye las diferencias en la eficiencia entre los distintos métodos. Se trata de un resultado bastante curioso e interesante.

### 3.8. RESULTADOS EXPERIMENTALES

La situación puede observarse de forma más clara si obtenemos la ordenación de los métodos de acuerdo con su eficiencia, comparando los métodos dos a dos, igual que hicimos antes. En la Tabla 58 del apéndice de la Tesis Doctoral se dan las comparaciones de cada método ARNMF&WD con todos los métodos ARNMF.

A continuación mostramos la ordenación aproximada que hemos obtenido, comparando los métodos dos a dos, donde hemos distinguido en azul los métodos basados en el análisis de la red entrenada (ARNMF) sin uso de weight-decay y en verde los que utilizan weight-decay.

**UTA > TEKA > BL2 > TEKA&WD > DEV > TEKB = DER2 = MAO =**  
**=UTA&WD > YOU&WD = DEV&WD = SAN&WD > CLO = BL1&WD =**  
**=BL2&WD = CLO&WD = PRI&WD = TEKB&WD > CIB&WD > BL1 =**  
**=YOU > PRI = SAN > DER2&WD = TEK&WD > CIB = TEKE > TEKE&WD =**  
**=DER3&WD = MAO&WD > TEK > LEE&WD > DER3 > LEE**

De dicha ordenación parece claro que la variación de la eficiencia disminuye con el uso de weight-decay. La primera conclusión sería que las eficiencias de los métodos tienden a ser más similares con el uso de weight-decay.

En segundo lugar, podemos ver que para la mayoría de los métodos el uso de weight-decay hace que su eficiencia se desplace hacia la zona media de la ordenación.

En otras palabras, la conclusión es que se deteriora la eficiencia de los mejores métodos y se mejora la de los peores.

### 3.9. Aplicación a un problema real: patología de la voz

Revisando la bibliografía se puede ver que es bien conocido que las posibles alteraciones sobre el aparato fonador causan cambios sobre la señal acústica de voz. Usualmente el diagnóstico de alteraciones en la voz es realizado por un experto: este hace hablar al paciente y analiza la voz.

Esta forma de realizar el diagnóstico tiene algunos inconvenientes, por ejemplo:

- *El diagnóstico es subjetivo, es decir, el propio estado de ánimo del experto podría influir en el diagnóstico final*
- *El diagnóstico a través de un experto suele ser muy lento, los problemas se detectan cuando la enfermedad o patología está ya avanzada.*

Otra forma de realizar el diagnóstico sería realizar un estudio a partir de las características de la señal de voz, de la cual se pueden extraer multitud de parámetros utilizando técnicas de procesado digital de la señal.

Este último tipo de diagnóstico de patologías a partir de la señal de voz presenta numerosas ventajas respecto a otros métodos:

- No es una herramienta invasiva, el paciente solo debe hablar.
- Se puede construir un sistema automático para el diagnóstico.
- El diagnóstico es objetivo porque está basado en el valor de parámetros acústicos.
- Se reduce el coste y tiempo de diagnóstico.
- Puede ser útil para evaluar tratamientos y procesos de rehabilitación, usando la medida actualizada de los parámetros acústicos para observar la evolución.

Los parámetros acústicos definen claramente las características de la voz y serán útiles para determinar si una voz es “normal” o “patológica”. Incluso, se podría distinguir entre patologías, o utilizar la información para la prescripción de tratamientos o rehabilitación de voz.

Sin embargo, si estudiamos la bibliografía vemos que no hay un acuerdo en cuantos, ni en qué parámetros son realmente útiles para definir de forma clara y sencilla la señal de voz. En las referencias [41-51], podemos encontrar diferentes algoritmos para calcular los parámetros acústicos que se usarán para distinguir entre voces patológicas y normales. En las referencias [52] y [53], se aplican redes neuronales para resolver el problema. En la primera, se utiliza una red Multilayer Feedforward y en la segunda se utiliza una red SOM.

### 3.9. APLICACIÓN A UN PROBLEMA REAL: PATOLOGÍA DE LA VOZ

En nuestras investigaciones, como describiremos más adelante, usaremos una red Multilayer Feedforward con una capa oculta y diferentes métodos de selección de entradas para escoger los parámetros apropiados de entrada a la red. A diferencia de las referencias [52] y [53], veremos que nuestros resultados son mejores, ya que escogemos algunos de los métodos de selección de características de entrada, y el número final de parámetros que utilizaremos es muy bajo, obteniendo sin embargo, una eficiencia del 100%.

Durante la realización de nuestras investigaciones en el campo de selección de entradas en redes neuronales, surgió la participación en el proyecto CICYT **TER96-1938-C02-01** titulado: “*Sistemas de Evaluación y Rehabilitación de Problemas de Fonación y/o Audición*”. El proyecto estaba dirigido por Dr. Santiago Aguilera de la Universidad Politécnica de Madrid y en él participaban varias entidades.

La Universidad Jaume I se incorporó al proyecto con el objetivo de tratar de aplicar redes neuronales al problema de diagnóstico de patologías de voz, lo cual no estaba en principio contemplado en el proyecto inicial pero pareció interesante incorporarlo.

Nuestra primera meta, fue tratar de aplicar redes neuronales al problema de distinguir entre voces patológicas y no patológicas. Se trata de un problema de clasificación, para el cual utilizamos redes Multilayer Feedforward.

El primer problema que nos encontramos fue precisamente el de seleccionar los parámetros acústicos que debíamos usar como entradas de nuestra red.

Si para un experto esto ya es un problema complejo, para alguien no experto en el tema puede resultar imposible.

Sin embargo, pensamos que podríamos aplicar las técnicas de selección de entradas que estabamos estudiando en aquellos momentos, con el objetivo de reducir el número de parámetros acústicos necesario para resolver el problema de clasificación que abordábamos.

El procedimiento que seguimos de forma resumida fue:

- *Utilizar redes Multilayer Feedforward para distinguir entre voces patológicas y normales, utilizando todos los parámetros acústicos de los que disponíamos.*
- *Aplicar algunas de las técnicas de selección de entradas, para reducir el número de parámetros y ver si la eficiencia resultante era igual o mejor que la obtenida con todos los parámetros.*



Para abordar el problema disponíamos de una base de datos comercial de la compañía Kay Elemetrics: “Disorders voice Database”, Versión 1.03, Kay Elemetrics Corp, 1994. Esta base, fue originalmente desarrollada y grabada por “The Massachusetts Eye and Ear Infirmary (MEEI) Voice and Speech Laboratories”.

Se trata de una base de datos en CDROM que contiene alrededor de 1400 ejemplos de voces de unos 700 sujetos. Los ejemplos están obtenidos tanto de pacientes con voces normales como con alguna patología.

Para cada ejemplo, la base de datos contiene un total de 33 parámetros acústicos calculados utilizando el “*Multi-Dimensional Voice Program (MDVP)*”.

Los 33 parámetros acústicos son abreviados en la base de datos con los siguientes nombres: “*Fo*”, “*To*”, “*Fhi*”, “*Flo*”, “*STD*”, “*PFR*”, “*Fftr*”, “*Fatr*”, “*Tsam*”, “*Jita*”, “*Jitt*”, “*RAP*”, “*PPQ*”, “*sPPQ*”, “*vFo*”, “*ShdB*”, “*Shim*”, “*APQ*”, “*sAPQ*”, “*vAm*”, “*NHR*”, “*VTI*”, “*SPI*”, “*FTRI*”, “*ATRI*”, “*DVB*”, “*DSH*”, “*DUV*”, “*NVB*”, “*NSH*”, “*NUV*”, “*SEG*” y “*PER*”.

A continuación damos una corta descripción de los mismos.

- Fo*** *Frecuencia fundamental media en la vocalización.*
- To*** *Periodo, en milisegundos, del periodo glotal medio.*
- Fhi*** *Frecuencia fundamental más alta en la vocalización (en Hz).*
- Flo*** *Frecuencia fundamental más baja en la vocalización.*
- STD*** *Desviación estándar de la frecuencia fundamental en la vocalización.*
- PFR*** *Rango de frecuencia fundamental fonatoria en semi-tonos.*
- Fftr*** *Frecuencia de la más intensa baja frecuencia de la componente de modulación de *Fo*.*
- Fatr*** *Frecuencia de la más intensa baja frecuencia de la componente de modulación de amplitud.*
- Tsam*** *Longitud en segundos de un ejemplo de voz (dato) analizado.*
- Jita*** *Jitter absoluto, da una evaluación en microsegundos de la variabilidad de periodo-a-periodo del periodo del pitch.*
- Jitt*** *Porcentaje jitter, da una evaluación de la variabilidad del periodo del pitch en porcentaje. Representa la variación relativa de periodo a periodo (muy a corto plazo).*
- RAP*** *Perturbación media relativa: evaluación de la variabilidad del periodo pitch con factor de suavizado de 3 periodos.*
- PPQ*** *Cociente de la perturbación del periodo del pitch: evaluación en porcentaje de la variabilidad del pitch con factor de suavizado de 5 periodos.*

### 3.9. APLICACIÓN A UN PROBLEMA REAL: PATOLOGÍA DE LA VOZ

- sPPQ** *Cociente de la perturbación del periodo del pitch suavizado, que da una evaluación en porcentaje de la variabilidad a largo plazo del periodo del pitch, con un factor de suavizado de 55 periodos. Se relaciona con la intensidad de la frecuencia de temblor (tremor, en inglés).*
- vFo** *Variación de la frecuencia fundamental que representa la desviación estándar relativa de periodo a periodo, calculada de la frecuencia fundamental. Refleja a muy largo plazo la variación de Fo.*
- ShdB** *Shimmer en dB, da una evaluación de la variabilidad de periodo a periodo de la amplitud pico a pico.*
- Shim** *Porcentaje Shimmer, evalúa en porcentaje la variabilidad de la amplitud de pico a pico. Representa la variabilidad periodo a periodo relativa (a muy corto plazo) de la amplitud pico a pico.*
- APQ** *Cociente de la perturbación de amplitud, da una evaluación en porcentaje de la variabilidad de la amplitud pico a pico, con un factor de suavizado de 11 periodos.*
- sAPQ** *Cociente de la perturbación de amplitud suavizada: evaluación en porcentaje de la variabilidad a largo plazo de la amplitud pico a pico, con un factor de suavizado de 55 periodos, se relaciona con la intensidad de la amplitud de temblor (tremor, en inglés).*
- vAm** *Variación del pico de amplitud, representa la desviación estándar relativa de periodo a periodo, calculado de pico a pico de la amplitud. Refleja a muy largo plazo las variaciones de la amplitud.*
- NHR** *Relación de armónicos-ruido: relación media de la energía de las componentes de los armónicos en el rango de 1,5-4,5 kHz, a las componentes en el rango 70-4500 H. Evaluación general de la presencia de ruido en la vocalización.*
- VTI** *Índice de turbulencia vocal: relación media del espectro de energía de los armónicos de alta frecuencia, al espectro de energía de armónicos en áreas de fonación estables, mide el nivel de energía relativa de ruido de alta frecuencia.*
- SPI** *Índice de fonación suave: relación media de la energía de los armónicos de baja a alta frecuencia. Mide el tipo de espectro en la vocalización.*
- FTRI** *Índice de intensidad de “Fo-tremor” muestra (en porcentaje) la relación de la magnitud de frecuencia de la componente (Fo-tremor) de modulación de la más intensa baja frecuencia, a la magnitud total de frecuencia.*
- ATRI** *Índice de la intensidad de “Amplitude tremor” muestra (en porcentaje) la relación entre la amplitud de la componente (amplitude tremor) de modulación-amplitud de la más intensa baja frecuencia, y la amplitud total.*

- DVB** Grado de parada de voz, muestra (en porcentaje) la relación de la longitud total de las áreas que representan paradas de voz en el tiempo total del ejemplo de voz.
- DSH** Grado de sub-armónicos, es una estimación relativa de los sub-armónicos de las componentes de  $F_0$ .
- DUV** Grado de no-voz: evaluación relativa de las áreas de no-armónicos (donde  $F_0$  no puede ser detectado). Considera como no-voz todas las pausas antes y después de áreas de voz.
- NVB** Número de paradas de voz, muestra las veces que  $F_0$  es interrumpido desde el comienzo, desde la primera parada hasta el final de la última área de voz.
- NSH** Número de segmentos sub-armónicos encontrados durante el análisis.
- NUV** Número de segmentos de no-voz detectados durante el análisis de autocorrelación.
- SEG** Número total de segmentos computados durante el análisis de autocorrelación MDVP.
- PER** Periodo pitch detectado durante la extracción de pitch de periodo a periodo usando MDVP.

También se incluye en la base de datos otra información como la edad (AGE), sexo (SEX), lenguaje nativo (NATLANG), origen racial (ORIGIN) y si el sujeto es fumador o no (SMOKE), etc.

Nosotros hemos utilizado una red Multilayer Feedforward con una capa oculta para implementar la aplicación. Primero, hemos usado todos los parámetros acústicos de la base de datos más la edad y sexo como entradas a la red, es decir, 35 entradas.

El primer paso para entrenar la red fue eliminar aquellos ejemplos en los cuales encontramos “entradas desconocidas” (parámetro cuyo valor se desconoce). Esto se debía a que el software MDVP no había sido capaz de calcular dichos parámetros. La mayor parte de las entradas desconocidas se encuentran en los parámetros “*FTRI*”, “*ATRI*”, “*Ftr*” y “*Fatr*”.

Si descartamos los ejemplos con entradas desconocidas, quedan un total de 70 registros normales y 211 patológicos. El número total de ejemplo será de 281, lo cual representa una reducción alta respecto el número inicial de 1400.

Realmente teníamos dos posibilidades a la hora de podar la base de datos. Una sería quitar ejemplos con valores desconocidos de algún parámetro, y la otra descartar parámetros acústicos que presentan valores desconocidos. De la segunda forma, el número final de ejemplos de entrenamiento sería mayor, sin embargo, la

### *3.9. APLICACIÓN A UN PROBLEMA REAL: PATOLOGÍA DE LA VOZ*

importancia de los parámetros no es conocida a priori, por lo que descartando parámetros podríamos eliminar algunos que fueran importantes en la predicción, por lo que, finalmente, adoptamos la primera solución de descartar ejemplos y conservar todos los parámetros.

En cuanto al algoritmo de entrenamiento a usar, se optó por el descrito en el apartado (3.4) de este capítulo de la Tesis Doctoral. Es un algoritmo con eficiencia similar a Backpropagation y bastante más rápido.

Puesto que no disponíamos de muchos ejemplos para el entrenamiento, 281, usamos la técnica de entrenar 36 redes diferentes para medir la eficiencia. Se realizaron 36 particiones de los datos entre conjunto de entrenamiento y test. Así, habría 8 ejemplos diferentes en cada conjunto de test (273 en cada conjunto de entrenamiento), excepto en el último donde solo habría un ejemplo en el conjunto de test (280 en el de entrenamiento).

Después de entrenar las redes y medir la eficiencia nos enfrentamos al problema de reducir el número de entradas a la red, es decir, de seleccionar los parámetros acústicos de tal forma que sean un número mínimo y suficiente para resolver el problema.

Los métodos de selección de entradas aportan ciertas ventajas, se reduce el tamaño final de la red, se reducen los datos a procesar y es posible incrementar la capacidad de generalización de la red. Como hemos visto a lo largo de este capítulo, hay muchos métodos de selección de entradas y en general la eficiencia final suele depender del método que usemos, y del problema al que se aplique.

En el momento que realizamos esta investigación aun no habíamos completado el estudio de todos los métodos. Lo que se hizo fue escoger 6 de los métodos que se habían estudiado en esas fechas.

Los métodos escogidos fueron los siguientes: BL1, BL2, CLO, PRI, DEV, TEK, todos ellos basados en el análisis de redes entrenadas Multilayer Feedforward (ARNMF) y todos descritos al principio de este capítulo.

El entrenamiento de la red inicial se realizó usando todas las entradas. Se entrenaron 36 redes con diferente inicialización (de datos y de pesos).

Para la aplicación de estos métodos, el conjunto de datos utilizado debe estar normalizado, ya que el rango de variabilidad de las entradas influye en las magnitudes finales de los pesos conectados a ellas.

Para realizar la normalización restamos al valor de una entrada el valor mínimo de la misma, y dividimos el resultado entre la diferencia entre el valor

máximo y el mínimo. Esta operación la realizamos para todas las entradas y todos los datos y al final obtuvimos los datos normalizados en el intervalo [0,1].

A continuación, veremos la metodología utilizada en nuestros experimentos.

La arquitectura de la red neuronal inicial fue: 35 entradas y 2 salidas. El número de unidades ocultas se determinó por procedimientos de prueba y error. Finalmente se escogieron 6 unidades ocultas.

Las redes fueron entrenadas y testeadas obteniendo un porcentaje de correcta clasificación del 100%. Esto quiere decir, que la red es capaz de distinguir perfectamente entre voces patológicas y no patológicas.

**Tabla 9. Ordenación de los parámetros acústicos por importancia.**

		<i>Métodos</i>						
		<i>BL1</i>	<i>BL2</i>	<i>CLO</i>	<i>PRI</i>	<i>DEV</i>	<i>TEK</i>	
Entrada menos importante ↑		"FTRI"	"Shim"	"FTRI"	"FTRI"	"NVB"	"NVB"	
		"PFR"	"ATRI"	"NVB"	"PFR"	"ATRI"	"VTI"	
		"Jitt"	"NUV"	"To"	"Jitt"	"Shim"	"NHR"	
		"To"	"FTRI"	"NUV"	"To"	"RAP"	"ShdB"	
		"NVB"	"NVB"	"Jitt"	"NVB"	"NUV"	"DVB"	
		"NUV"	"APQ"	"PFR"	"NUV"	"APQ"	"SEX"	
		"ShdB"	"Fftr"	"NHR"	"ShdB"	"ShdB"	"NSH"	
		"Fftr"	"RAP"	"SEX"	"Fftr"	"FTRI"	"FTRI"	
		"SEX"	"ShdB"	"DVB"	"SEX"	"DSH"	"RAP"	
		"ATRI"	"To"	"RAP"	"ATRI"	"VTI"	"PPQ"	
		"RAP"	"sPPQ"	"vAm"	"RAP"	"Tsam"	"Tsam"	
		"vAm"	"vAm"	"ShdB"	"vAm"	"vAm"	"sPPQ"	
		"PPQ"	"VTI"	"sAPQ"	"PPQ"	"To"	"NUV"	
		"NHR"	"PPQ"	"PPQ"	"NHR"	"PPQ"	"DSH"	
		"sAPQ"	"Jitt"	"Shim"	"sAPQ"	"NHR"	"Jitt"	
	Entrada más importante ↓		"DSH"	"Tsam"	"STD"	"DSH"	"SEX"	"vFo"
			"DVB"	"vFo"	"APQ"	"DVB"	"sPPQ"	"PFR"
		"DUV"	"DSH"	"DUV"	"DUV"	"Fftr"	"APQ"	
		"Tsam"	"NHR"	"VTI"	"Tsam"	"vFo"	"ATRI"	
		"APQ"	"PFR"	"Tsam"	"APQ"	"DUV"	"Fatr"	
		"STD"	"NSH"	"Fftr"	"STD"	"NSH"	"Fftr"	
		"Shim"	"SEX"	"ATRI"	"Shim"	"Jitt"	"STD"	
		"sPPQ"	"sAPQ"	"DSH"	"sPPQ"	"STD"	"Shim"	
		"VTI"	"DVB"	"sPPQ"	"VTI"	"PFR"	"sAPQ"	
		"vFo"	"STD"	"Fatr"	"vFo"	"sAPQ"	"DUV"	
		"Fatr"	"Fatr"	"NSH"	"Fatr"	"DVB"	"To"	
		"NSH"	"DUV"	"vFo"	"NSH"	"Fatr"	"vAm"	
		"AGE"	"SPI"	"AGE"	"AGE"	"SPI"	"SPI"	
		"SEG"	"AGE"	"SEG"	"SEG"	"AGE"	"AGE"	
		"SPI"	"SEG"	"SPI"	"SPI"	"SEG"	"SEG"	
		"Flo"	"Flo"	"Flo"	"Flo"	"Flo"	"Flo"	
		"Fo"	"Fo"	"Fo"	"Fo"	"Fo"	"Fo"	
		"Fhi"	"Fhi"	"Fhi"	"Fhi"	"Fhi"	"Jita"	
	"Jita"	"Jita"	"Jita"	"Jita"	"Jita"	"Fhi"		
	"PER"	"PER"	"PER"	"PER"	"PER"	"PER"		

### 3.9. APLICACIÓN A UN PROBLEMA REAL: PATOLOGÍA DE LA VOZ

El paso siguiente, fue aplicar los 6 métodos de selección escogidos y obtener las diferentes ordenaciones de la importancia de las entradas.

En la Tabla 9 damos la ordenación de los diferentes parámetros acústicos en orden de importancia para cada uno de los métodos.

A partir de estas ordenaciones se pueden construir diferentes subconjuntos por sucesiva eliminación de una entrada. Por ejemplo, para el método BL1 la primera entrada que se elimina será “FTRI”, la segunda “PFR”. Siguiendo este proceso, el subconjunto de las 5 entradas más importantes para el método BL1 sería el formado por {“Flo”, “Fo”, “Fhi”, “Jita”, “PER”}, y el subconjunto de las dos más importantes para el método TEK {“Fhi”, “PER”}.

Finalmente, se evaluaría la eficiencia de las diferentes redes (entrenando y testeando) usando como nueva capa de entrada los subconjuntos construidos.

En la Tabla 10, se da un extracto de los resultados de la eficiencia para los subconjuntos formados por 6 o menos entradas. El resto de resultados no lo reflejamos aquí porque la eficiencia para todos ellos es del 100%.

**Tabla 10. Eficiencia de los métodos de selección aplicados.**

Nº entradas en el subconjunto final	Métodos					
	BL1	BL2	CLO	PRI	DEV	TEK
6	100%	100%	100%	100%	100%	100%
5	100%	100%	100%	100%	100%	100%
4	100%	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%	100%
2	—	—	—	—	—	100%
1	—	—	—	—	—	—

En la tabla el símbolo “—” significa que la red no ha convergido bien con ese subconjunto, el porcentaje obtenido en el entrenamiento era demasiado bajo por lo que no hemos testeado el funcionamiento de dicha red.

Podemos concluir, que el sexto método, TEK, da mejores resultados para este problema concreto. Con este método podemos escoger un subconjunto de 2 entradas para resolver perfectamente el problema de clasificación entre voces patológicas y no patológicas, sin que se produzca pérdida de la eficiencia en la red.

En otros métodos, la entrada “*Jita*” es más importante que la entrada “*Fhi*”, pero según los resultados del método **TEK** podemos ver que “*Jita*” puede considerarse como innecesaria, pero la entrada “*Fhi*” es necesaria.

Podemos finalmente concluir, que los parámetros acústicos “*Fhi*” y “*PER*” son suficientes para resolver el problema de clasificación entre voces patológicas y no patológicas. La eficiencia (porcentaje de correcta clasificación) con sólo estas dos entradas es del 100%.

Vimos al principio de este apartado que en la referencia [52] también se usaba una red Multilayer Feedforward para resolver este problema. Los resultados que se dan en dicha referencia son de un 85.8% de clasificación correcta. La principal razón es que los parámetros acústicos usados en la red son diferentes de los nuestros.

En la referencia [53], sí que se realiza una selección previa de los parámetros acústicos a usar. Los seleccionados eran: “*APQ*”, “*BVD*”, “*RAM*”, “*SPI*”, “*STD*” y “*vAm*”. La relación de correcta clasificación es del 83.3%. En este caso las razones de la diferencia pueden ser, que tanto la red utilizada (SOM) como los parámetros escogidos, son diferentes. En nuestras ordenaciones por ejemplo, el parámetro “*APQ*” está entre la 6 y 20 entradas menos importantes, “*BVD*” entre la 5 y la 26, *SPI* entre la 28 y 30, “*STD*” entre la 16 y la 25 y finalmente “*vAm*” entre la 11 y 27. En general, la importancia de estos parámetros no es demasiado buena teniendo en cuenta que trabajamos con 35 entradas.

Pensamos pues que nuestros resultados son muy buenos en comparación con otro tipo de técnicas que se han aplicado.

Otro problema que se puede abordar usando la misma metodología es el de diagnóstico médico, es decir clasificar entre diferentes patologías.

Esto es un trabajo futuro que esperamos resolver. Nosotros pensamos que este problema no se podrá resolver bien con la base de datos de “*Kay Elemetrics*” ya que ésta contiene más de 40 patologías diferentes y el número de ejemplo de voces patológicas es muy bajo. El problema podría resolverse de dos formas, o restringimos el número de patologías, de forma que el número de ejemplo de las que seleccionemos sea suficiente para resolver el problema, o simplemente descartamos esta base de datos e intentamos obtener una propia.

La segunda opción es muy costosa en cuanto a tiempo e infraestructura pero pensamos que puede merecer la pena, ya que se dispondría de una herramienta muy interesante tanto para nuestro equipo de investigación como para otros.

### *3.9. APLICACIÓN A UN PROBLEMA REAL: PATOLOGÍA DE LA VOZ*

Además, la resolución del problema de distinción entre diagnósticos, proporcionaría una fuerte herramienta de diagnóstico médico muy interesante para otros campos de investigación.



## 3.10. Conclusiones

En este capítulo hemos presentado una revisión bastante completa de los métodos de selección de entradas basados en el análisis de una red entrenada Multilayer Feedforward (ARNMF), y de los métodos basados en el análisis del conjunto de entrenamiento (ACE).

También hemos propuesto una metodología que permita comparar los métodos de selección de entradas, mediante la obtención de los subconjuntos óptimos de entradas. Hemos aplicado dicha metodología a 26 métodos de selección de entradas, y 15 problemas de clasificación diferentes.

El primer resultado que obtuvimos fue una ordenación de los 26 métodos de acuerdo con su eficiencia como métodos de selección de entradas.

A partir de dicha ordenación, se puede concluir que la variabilidad de la eficiencia de los métodos de tipo ARNMF es superior a la de los métodos ACE. Además, en general, la eficiencia de los últimos puede ser considerada como mediocre con respecto a los mejores métodos del tipo ARNMF.

Sin embargo, hemos encontrado una excepción en el método basado en la Distancia GD. Pensamos que este método puede ser útil y prometedor, porque combina la característica de un bajo coste computacional (que habitualmente tienen los métodos del tipo ACE), con la alta eficiencia de los métodos de tipo ARNMF. Hemos encontrado algunos problemas en los cuales este método no ha sido aplicable ya que la matriz de transformación fue singular, por ello, pensamos que aún se podría introducir alguna mejora en el método para solventar este problema. Esto podría ser un trabajo futuro a desarrollar.

También, en este capítulo, se ha estudiado el efecto de introducir un término de regularización (weight-decay) en el entrenamiento de las redes para los métodos ARNMF. Según la bibliografía se esperaría una mejora de la eficiencia en la tarea de selección de entradas. Hemos aplicado weight-decay en 17 métodos, y sobre 8 problemas y hemos comparado los resultados con los obtenidos para el caso de no aplicar weight-decay.

Los resultados son que el uso de weight-decay mejora la eficiencia de los peores métodos, pero deteriora la eficiencia de los mejores. En dicho sentido, disminuyen las diferencias entre las eficiencias de los métodos de selección de entradas. Por esta razón, no recomendamos la utilización de weight-decay para esta tarea. En lugar de ello, se debería seleccionar uno de los métodos de selección de entradas ARNMF más eficientes, y aplicarlo a una red neuronal entrenada con Backpropagation.

### *3.10. CONCLUSIONES*

También en este capítulo hemos presentado una aplicación a un caso real de determinación de patologías en la voz. El problema a resolver era distinguir entre voces normales o voces patológicas y el número de entradas de la red era muy elevado. Mediante la aplicación de algunos de los métodos vistos en este capítulo, conseguimos reducir a dos el número de entradas de la red necesarias para resolver el problema.

Vemos pues, que este tipo de métodos puede ser muy útil en la resolución de problemas reales, en los que a veces la ayuda de un experto puede no resolver el problema con eficiencia.

Los resultados de las investigaciones realizadas en este capítulo han sido publicados en varios congresos internacionales cuyas referencias se encuentran en la bibliografía y son [31], [32], [33], [34], [35], [54], [55], [56], [57], [58], [59] y [60].

### 3.11. Bibliografía del Capítulo

- [1] Peck, C.C., Dhawan, A.P., "SSME parameter model input selection using genetic algorithms". IEEE Transactions on Aerospace and Electronic Systems, vol. 32, no. 1, January 1996.
- [2] Greene, R.A., "Neural net applications in the cognitive sciences: a case study". Journal of Microcomputer Applications, vol. 16, pp. 197-201, 1993
- [3] Hammitt, A.M., Bartlett, E.B., "Determining functional relationships from trained neural networks". Mathematical Computing Modelling, vol. 22, no. 3, pp. 83-103, 1995.
- [4] Peck, C.C., Dhawan, A.P., "Genetic algorithm based input selection for a neural network function approximator with applications to SSME health monitoring". Proceedings of the 1993 IEEE International Conference on Neural Networks, vol. 22, pp. 1115-1122, 1993.
- [5] Bartlett, E.B., "Self determination of input variable importance using neural networks". Neural, Parallel & Scientific Computations, vol. 2, pp. 103-114. 1994.
- [6] Kwok, T.Y., Yeung, D.Y., "Constructive neural networks: some practical considerations". Proceedings. of the 1994 IEEE International Conference on Neural Networks, vol. 1, pp. 198-203, 1994.
- [7] Prechelt, L., "A quantitative study of experimental evaluations of neural network learning algorithms: current research practice". Neural Networks, vol. 9, no. 3, pp. 457-462, 1996.
- [8] Bishop, C., "Exact calculation of the hessian matrix for the multilayer perceptron". Neural Computation, vol. 4, pp. 494-501, 1992.
- [9] Belue, L.M., Bauer, K.W., "Determining input features for multilayer perceptrons". Neurocomputing, vol. 7, no. 2, pp. 111-121, 1995.
- [10] Tetko, I.V., Villa, A.E.P., Livingstone, D.J., "Neural network studies. 2 Variable selection". Journal of Chemical Information and Computer Sciences, vol. 36, no. 4, pp. 794-803, 1996.
- [11] Cibas, T., Soulié, F.F., Gallinari, P., Raudys, S., "Variable selection with neural networks". Neurocomputing, vol. 12, pp. 223-248, 1996.
- [12] Devena, L., "Automatic selection of the most relevant features to recognize objects". Proceedings of the International Conference on Artificial Neural Networks, vol. 2, pp. 1113-1116, 1994.
- [13] El-Deredy, W., Branston, N.M., "Identification of relevant features in HMR tumor spectra using neural networks". Proceedings of the 4<sup>th</sup> International Conference on Artificial Neural Networks, pp. 454-458, 1995.
- [14] Tetko, I.V., Tanchuk, V.Y., Luik, A.I., "Simple heuristic methods for input parameter estimation in neural networks". Proceedings of the IEEE International Conference on Neural Networks, vol. 1, pp. 376-380, 1994.
- [15] Engelbrecht, AP., Cloete, I., "A sensitivity analysis algorithm for pruning feedforward neural networks". Proceedings of the International Conference on Neural Networks, vol. 2, pp. 1274-1277, 1996.
- [16] Priddy, K.L., Rogers, S.K., Ruck D.W., Tarr G.L., Kabrisky, M., "Bayesian selection of important features for feedforward neural networks". Neurocomputing, vol. 5, no. 2&3, pp. 91-103, 1993.

### 3.11. BIBLIOGRAFÍA DEL CAPÍTULO

- [17] Sano, H., Nada, A., Iwahori, Y., Ishii, N., “A method of analyzing information represented in neural networks”. Proceedings of 1993 International Joint Conference on Neural Networks, pp. 2719-2722, 1993.
- [18] Mao, J., Mohiuddin, K., Jain, A.K., “Parsimonious network design and feature selection through node pruning”. Proceedings of the 12<sup>th</sup> IAPR International Conference on Pattern Recognition, vol. 2, pp. 622-624, 1994.
- [19] Utans, J., Moody, J., et alt, “Input variable selection for neural networks: Application to predicting the U.S. business cycle”. Proceedings of IEEE/IAFE Computational Intelligence for Financial Engineering, pp. 118-122, 1995.
- [20] Lee, H., Mehrotra, K., Mohan, C. Ranka, S., “Selection procedures for redundant inputs in neural networks”. Proceedings of the World Congress on Neural Networks, vol. 1, pp. 300-303, 1993.
- [21] Bowles, A.: Machine learns which features to select. Proceedings of the 5<sup>th</sup> Australian Joint Conference on Artificial Intelligence, pp. 127-132, 1992.
- [22] Younes, B., Fabrice, B., “A neural network based variable selector”. Proceedings of the Artificial Neural Network in Engineering, (ANNIE'95), pp. 425-430, 1995.
- [23] Battiti, R., “Using mutual information for selecting features in supervised neural net learning”. IEEE Transactions on Neural Networks, vol. 5, no. 4, pp. 537-550, 1994.
- [24] Chi, Jabri, “Entropy based feature evaluation and selection technique”. Proceedings of 4<sup>th</sup> Australian Conference on Neural Networks, (ACNN'93), pp. 193-196, 1993.
- [25] Setiono, R., Liu, H., “Improving Backpropagation learning with feature selection”. Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies, vol. 6, no. 2, pp. 129-139, 1996.
- [26] Lorenzo, Hernández, Méndez, “Attribute selection through a measurement based on information theory”.(in Spanish). 7<sup>a</sup> Conferencia. de la Asociación Española para la Inteligencia Artificial, (CAEPIA), pp. 469-478, 1997.
- [27] Kira, K., Rendell, L.A., “The feature selection problem: Traditional methods and a new algorithm”. Proceedings of 10<sup>th</sup> National Conference on Artificial Intelligence, pp. 129-134, 1992.
- [28] Scherf,, “A new approach to feature selection”. Proceedings of the 6<sup>th</sup> Conference on Artificial Intelligence in Medicine, (AIME'97), pp. 181-184, 1997.
- [29] Thawonmas, R., Abe, S., “Feature reduction based on analysis of fuzzy regions”. Proceedings of the 1995 International Conference on Neural Networks, vol. 4, pp. 2130-2133, 1995.
- [30] Watzel, R., Meyer-Bäse, A., Meyer-Bäse, U., Hilberg, H., Scheich, H., “Identification of irrelevant features in phoneme recognition with radial basis classifiers”. Proceedings of 1994 International Symposium on Artificial Neural Networks. pp. 507-512, 1994.
- [31] Fernández, M., Hernández, C., “Input selection by Multilayer Feedforward trained networks”. Proceedings of the 1999 IEEE International Joint Conference on Neural Networks. pp. 1834-1839, 1999.

- [32] Fernández, M., Hernández, C., “Neural networks input selection by using the training set”. Proceeding of the 1999 IEEE International Joint Conference on Neural Networks. pp. 1189-1194, 1999.
- [33] Fernández, M., Hernández, C., “A Comparison among feature selection methods based on trained networks”. Proceedings of the 1999 IEEE Neural Networks for Signal Processing Workshop (NNSP'99), pp. 205-214, 1999.
- [34] Fernández, M., Hernández, C., “Optimal use of a trained neural network for input selection”. Proceedings of the 5<sup>th</sup> International Work Conference on Artificial and Natural Neural Networks (IWANN'99), pp. 506-515, 1999.
- [35] Fernández, M., Hernández, C., “How to select the inputs for a Multilayer Feedforward Network by using the training set”. Proceedings of the 5<sup>th</sup> International Work Conference on Artificial and Natural Neural Networks (IWANN'99), pp. 477-486, 1999.
- [36] Bos, S., Chug, E., “Using weight-decay to optimize the generalization ability of a perceptron”. Proceedings of IEEE International Conference on Neural Networks, vol. 1, pp. 241-246, 1996.
- [37] Merz, C. J., Murphy, P. M., (1998). UCI repository of machine learning databases. Disponible en: <http://www.ics.uci.edu/~mlearn/MLRepository.html>
- [38] Montgomery, D. C., Runger, G. C., *Probabilidad y estadística aplicadas a la Ingeniería*. McGraw-Hill, 1996.
- [39] Heinke, D., Hamker, F. H., “Comparing Neural Networks: a Benchmark on Growing Neural Gas, Growing Cell Structures, and Fuzzy ARTMAP”. IEEE Transactions on Neural Networks, vol. 9, no. 6, pp. 1279-1291, November 1998.
- [40] Xiao-Hu Yu, Guo-An Chen, Shi-Xin Cheng, “Dynamic Learning Rate Optimization of the Backpropagation Algorithm”. IEEE Transactions on Neural Networks, vol. 6. no. 3, pp. 669-677, 1995.
- [41] Banci G., Monini S., Falaschi A., Sairo N., “Vocal fold disorder evaluation by digital speech analysis”. Journal of Phonetics, vol. 14, pp. 495-499, 1986.
- [42] Boyanov B., Hadjitodorov S., Teston B., Doskov D., “Software system for pathological voice analysis”. LARYNX 1997, pp. 139-142, 1997.
- [43] Boyanov B., Hadjitodorov S., Teston B., Doskov D., “Robust hybrid detector for pathological voice analysis”. LARYNX 1997, pp. 55-58, 1997.
- [44] Zhu W., Kikuchi Y., Endo Y., et al., “An integrated acoustic evaluation system of pathologic voice”. Proceedings of the International Conference on Spoken Language Processing, pp. 1983-1986, 1994.
- [45] Boyanov B., Chollet G., “Pathological voice analysis using cepstra, bispectra and group delay functions”. Proceedings of the International Conference on Spoken Language Processing, pp. 1039-1042, 1992.
- [46] Boyanov B., Hadjitodorov S., Ivanov T., “Analysis of voiced speech by means of bispectrum”. Electronics Letters, vol. 7, no. 24, pp. 2267-2268, 1991.
- [47] Kasuya H., Endo Y., Saliu S., “Novel acoustic measurements of jitter and shimmer characteristic from pathologic voice”. Eurospeech 1993, pp. 1973-1976. Berlin 1993.
- [48] Kasuya H., Ogawa Sh., Mashima K., Ebihara S., “Normalized noise energy as an acoustic measure to evaluation pathologic voice”. Journal of Acoustic Society of America, vol. 80, no. 5, pp. 1329-1334, 1986.

### 3.11. BIBLIOGRAFÍA DEL CAPÍTULO

- [49] Yunik M., Boyanov B., “Method for evaluation of the noise-to-harmonic component ratios in pathological and normal voices”. *Acustica*, vol. 70, pp. 89-91, 1990.
- [50] Schoentgen J., “Jitter in discriminating between normal and dysphonic speakers”. *Applied stochastic models and data analysis*, vol. 4, pp. 127-135, 1985.
- [51] Boyanov B., Hadjitodorov S., Baudoin G., “Acoustical analysis of pathological voice”. *Proceedings of the Slovenian German Workshop on Speech Analysis and Recognition*, 1996.
- [52] Wallen E.J., Hansen J.H.L., “A screening test for speech pathology assessment using objective quality measures”, *Proceedings of the International Conference on Spoken Language Processing*, pp. 776-779, 1996.
- [53] Callant, D.E., Kent, R.D., et al., “Self-organizing map for the classification of normal and disordered female voices”. *Journal of Speech, Language, and Hearing Research*, vol. 42, no. 2, pp. 355-366, 1999.
- [54] Godino-Llorente, J.I., Aguilera-Navarro, S., Hernández-Espinosa, C., Fernández-Redondo, M., Gómez-Vilda, P., “On the selection of meaningful speech parameters used by a pathologic/non Pathologic voice register classifier”. *6th European Conference on Speech Communication and Technology (EuroSpeech'99)*, vol. 1, pp. 563-566, 1999.
- [55] Hernández, C., Fernández, M., Gomez, P., Godino, J.I., Aguilera, S., “Diagnosis of Vocal and Voice Disorders by the Speech Signal”. *Proceedings of 2000 International Joint Conference on Neural Networks, (IJCNN'2000)*, vol. 4, pp. 253-258, 2000.
- [56] Fernández, M., Hernández, C., “On the Combination of Weight-Decay and Input Selection Methods”. *Proceedings of 2000 International Joint Conference on Neural Networks, (IJCNN'2000)*, vol. 1, pp. 191-196, 2000.
- [57] Fernández, M., Hernández, C., “Influence of Weight-Decay Training in Input Selection Methods”, *European Symposium on Artificial Neural Networks (ESANN'2000)*, pp.135-140, Brujas (Belgica) 2000.
- [58] Hernández, C., Fernández, M., Gomez, P., “Neural Network based Input Selection and Diagnosis of Pathologic Voices”. *Neural Network World International Scientific Journal*. En prensa.
- [59] Hernández, C., Fernández, M., “On the Effect of Weight-Decay in Input Selection”. *Neural Network World International Scientific Journal*, vol. 10, no. 4, pp. 575-588, Praga, Julio de 2000.
- [60] Fernández, M., Hernández, C., “Analysis of Input Selection Methods for Multilayer Feedforward”. *Neural Network World International Scientific Journal*, vol. 10, no. 3, pp. 389-406, Praga, Julio de 2000.

## 3.12. Apéndice del Capítulo

### Cálculo de la matriz Hesiana.

Llamamos  $a_i$  y  $z_i$  a las entradas y salidas en las neuronas de la capa  $i$  de la red neuronal. Sean  $z_j$  las salidas de las neuronas la capa inferior a la capa  $i$ .

El error  $E$ , se puede calcular como la suma de los errores cometidos en cada patrón de entrada,  $E_p$ .

Si introducimos una perturbación en el error, su expresión se puede calcular a partir del desarrollo de Taylor de segundo orden. Las derivadas de segundo orden de dicho desarrollo son las que constituyen la llamada matriz hesiana.

Vamos a calcular las diferentes derivadas. Supongamos una unidad  $i$  en la misma capa que una unidad  $n$ .

$$\begin{aligned} \frac{\partial^2 E_p}{\partial w_{ij} \partial w_{nl}} &= \frac{\partial a_i}{\partial w_{ij}} \cdot \frac{\partial}{\partial a_i} \cdot \left( \frac{\partial E_p}{\partial w_{nl}} \right) = z_j \cdot \frac{\partial}{\partial a_i} \cdot \left( \frac{\partial E_p}{\partial w_{nl}} \right) = z_j \cdot \frac{\partial}{\partial a_i} (\sigma_n \cdot z_l) = \\ &= z_j \cdot \sigma_n \cdot \frac{\partial z_l}{\partial a_i} + z_j \cdot z_l \cdot \frac{\partial \sigma_n}{\partial a_i} = z_j \cdot \sigma_n \cdot f'(a_l) \cdot g_{li} + z_j \cdot z_l \cdot b_{ni} \end{aligned} \quad (3.110)$$

Donde:

$$\begin{aligned} \sigma_n &= \frac{\partial E_p}{\partial a_n} = \sum_r \frac{\partial E_p}{\partial a_r} \cdot \frac{\partial a_r}{\partial a_n} = \sum_r \sigma_r \cdot \frac{\partial a_r}{\partial z_n} \cdot \frac{\partial z_n}{\partial a_n} = \sum_r \sigma_r \cdot w_{rn} \cdot f'(a_n) \\ g_{li} &= \frac{\partial a_l}{\partial a_i} = \sum_r \frac{\partial a_r}{\partial a_i} \cdot \frac{\partial a_l}{\partial a_r} = \sum_r f'(a_r) \cdot w_{lr} \cdot g_{ri}; \text{ para la capa } i: g_{ii} = 1; g_{li} = 0; \\ b_{ni} &= \frac{\partial \sigma_n}{\partial a_i} = \frac{\partial}{\partial a_i} \left( f'(a_n) \sum_r \sigma_r \cdot w_{rn} \right) = f''(a_n) \cdot g_{ni} \cdot \sum_r \sigma_r \cdot w_{rn} + f'(a_n) \cdot \sum_r w_{rn} \cdot b_{ri} \end{aligned} \quad (3.111)$$

Para la capa de salida  $m$  de la red neuronal, podemos calcular las expresiones anteriores. En la ecuación (3.112) se obtiene:

$$\begin{aligned} \sigma_m &= \frac{\partial E_p}{\partial a_m} = \frac{\partial E_p}{\partial z_m} \cdot \frac{\partial z_m}{\partial a_m} = f'(a_m) \cdot \frac{\partial E_p}{\partial z_m} \\ b_{mi} &= \frac{\partial}{\partial a_i} \left( f'(a_m) \cdot \frac{\partial E_p}{\partial z_m} \right) = \left\{ f''(a_m) \cdot \frac{\partial E_p}{\partial z_m} + (f'(a_m))^2 \cdot \frac{\partial^2 E_p}{\partial z_m^2} \right\} \cdot g_{mi} = H_m \cdot g_{mi} \end{aligned} \quad (3.112)$$

### 3.12. APÉNDICE DEL CAPÍTULO

En la ecuación (3.113) se dan las expresiones de la función de error, la función de activación y sus derivadas.

$$E_p = \frac{1}{2} \cdot \sum_m (z_m - t_m)^2; \quad \frac{\partial E_p}{\partial z_m} = (z_m - t_m); \quad \frac{\partial^2 E_p}{\partial z_m^2} = 1; \quad (3.113)$$

$$f'(a) = f(a) \cdot (1 - f(a)); \quad f''(a) = f(a) \cdot (1 - f(a)) \cdot (1 - 2 \cdot f(a))$$

Sustituyendo en la ecuación (3.112) obtenemos:

$$\sigma_m = (z_m - t_m) \cdot z_m \cdot (1 - z_m) \quad (3.114)$$

$$H_m = f(a) \cdot (1 - f(a)) \cdot \{ (1 - 2 \cdot f(a)) \cdot (z_m - t_m) + f(a) \cdot (1 - f(a)) \}$$

#### Cálculo para una red de tres capas.

Vamos a aplicar todas las ecuaciones anteriores al caso de una red Multilayer Feedforward de tres capas (una capa oculta).

Sean  $k$  y  $k'$  dos unidades de la capa de entrada,  $l$  y  $l'$  dos unidades de la capa oculta y  $m$  y  $m'$  dos unidades de la capa de salida.

Calculamos las derivadas segundas de la función de error para cada caso.

**Caso 1.-** Ambos pesos se encuentran entre la capa de salida y la oculta.

$$\frac{\partial^2 E_p}{\partial w_{ml} \partial w_{m'l'}} = z_l \cdot \sigma_{m'} \cdot f'(a_{l'}) \cdot g_{l'm} + z_l \cdot z_{l'} \cdot b_{m'm} = 0 + z_l \cdot z_{l'} \cdot H_m \cdot \delta_{m'm} \quad (3.115)$$

$$H_m = f''(a_m) \cdot \frac{\partial E_p}{\partial z_m} + (f'(a_m))^2 \cdot \frac{\partial^2 E_p}{\partial z_m^2}$$

Donde la expresión de  $H_m$  se puede calcular según la ecuación (3.114) para el caso de la función de activación sigmoide y el error cuadrático medio.

**Caso 2.-** Ambos pesos se encuentran entre la capa oculta y la de entrada.

$$\frac{\partial^2 E_p}{\partial w_{lk} \partial w_{l'k'}} = z_k \cdot \sigma_{l'} \cdot f'(a_{k'}) \cdot g_{k'l} + z_k \cdot z_{k'} \cdot b_{l'l} = 0 + z_k \cdot z_{k'} \cdot b_{l'l} \quad (3.116)$$

$$b_{l'l} = \frac{\partial \sigma_{l'}}{\partial a_l} = \delta_{l'l} \cdot f''(a_{l'}) \cdot \sum_m \sigma_m \cdot w_{ml'} + f'(a_{l'}) \cdot \frac{\partial}{\partial a_l} \left( \sum_m \sigma_m \cdot w_{ml'} \right)$$



Donde:

$$\begin{aligned} \frac{\partial}{\partial a_l} \left( \sum_m \sigma_m \cdot w_{ml'} \right) &= \sum_m w_{ml'} \cdot \frac{\partial \sigma_m}{\partial a_l} = \sum_m w_{ml'} \cdot \frac{\partial}{\partial a_l} \left( \frac{\partial E_p}{\partial a_m} \right) = \\ &= \sum_m w_{ml'} \cdot \frac{\partial a_m}{\partial a_l} \cdot \frac{\partial}{\partial a_m} \left( \frac{\partial E_p}{\partial a_m} \right) = \sum_m w_{ml'} \cdot f'(a_l) \cdot w_{ml} \cdot H_m \end{aligned} \quad (3.117)$$

La derivada resulta:

$$\frac{\partial^2 E_p}{\partial w_{lk} \partial w_{l'k'}} = z_k \cdot z_{k'} \left\{ \delta_{ll'} \cdot f''(a_{l'}) \cdot \sum_m \sigma_m \cdot w_{ml'} + f'(a_{l'}) \cdot f'(a_l) \cdot \sum_m w_{ml'} \cdot w_{ml} \cdot H_m \right\} \quad (3.118)$$

**Caso 3.-** Cuando un peso se encuentran entre la capa de salida y la oculta y el otro entre la capa oculta y la de entrada.

$$\frac{\partial^2 E_p}{\partial w_{lk} \partial w_{ml'}} = z_k \cdot \sigma_m \cdot f'(a_{l'}) \cdot g_{l'l} + z_k \cdot z_{l'} \cdot b_{ml} = z_k \cdot f'(a_l) \cdot \{ \sigma_m \cdot \delta_{l'l} + z_{l'} \cdot w_{ml} \cdot H_m \} \quad (3.119)$$



# Capítulo 4

## ALGORITMOS DE APRENDIZAJE PARA LA SELECCIÓN DEL NÚMERO DE UNIDADES DE LA CAPA OCULTA DE LA RED DURANTE EL PROCESO DE APRENDIZAJE

- 4.1. *Descripción del Problema*
- 4.2. *Descripción de los diferentes métodos*
  - 4.2.1. *Fase constructiva: determinación de nuevos pesos*
  - 4.2.2. *Fase de podado: elección de la neurona a podar*
- 4.3. *Metodología*
- 4.4. *Desarrollo Experimental*
- 4.5. *Conclusiones*
- 4.6. *Bibliografía del Capítulo*
- 4.7. *Apéndice del Capítulo*

### 4.1. Descripción del Problema

A pesar de los muchos avances conseguidos para que las redes neuronales sean de aplicabilidad en problemas reales, todavía quedan algunas cuestiones que deben contestarse.

Una de las cuestiones abiertas reside en la determinación del tamaño de la red más apropiado para resolver una tarea específica, en concreto, qué número de neuronas debe usarse en la capa oculta para el caso de la arquitectura Multilayer Feedforward. La dificultad en la elección del tamaño reside principalmente en el hecho de que tanto redes grandes (con gran número de unidades ocultas), como pequeñas presentan importantes ventajas.

#### 4.1. DESCRIPCIÓN DEL PROBLEMA

Para redes neuronales de gran tamaño en las que tenemos un gran número de parámetros (uniones entre las neuronas) que determinar, el aprendizaje es más rápido y los mínimos locales son más fáciles de evitar. En particular, en la referencia [1] se presenta un estudio teórico en el cual se probó que cuando el número de unidades ocultas iguala el número de ejemplos de entrenamiento menos uno, se garantiza que la superficie de error de Backpropagation no tiene mínimos locales. También se observa para redes grandes, que la complejidad de los contornos de decisión de la red, aumentan con el tamaño de la misma y así esta poseerá un cierto grado de tolerancia a fallos.

Por otro lado, se ha demostrado tanto teórica, como experimentalmente que las redes con pocos parámetros, poseen mejor capacidad de generalización, lo cual es una característica fundamental en este campo de Redes Neuronales. Otra ventaja es que el conocimiento codificado en una red pequeña, es más fácil de interpretar y ello facilitará la extracción de reglas simples que describan el comportamiento de la red.

En general, una red neuronal demasiado pequeña no podrá aprender el problema bien, y una red neuronal demasiado grande tendrá como consecuencia el fenómeno de sobremodelado (“overfitting”) y poseerá una mala capacidad de generalización.

Esto último, se puede entender fácilmente mediante una analogía al problema de aproximación de curvas mediante polinomios. Consideremos unos datos generados a partir de una función suave en la que se introduce ruido. Normalmente, con un polinomio con pocos coeficientes, será imposible capturar la función a partir de los datos, mientras que con un polinomio con demasiados coeficientes se aproximará tanto la función como el ruido, tendremos una pobre representación de la función. Para un número óptimo de coeficientes, el polinomio nos dará la mejor representación de la función y las mejores predicciones para nuevos datos. Algo similar se puede considerar que ocurre con el número de unidades ocultas en la red Multilayer Feedforward.

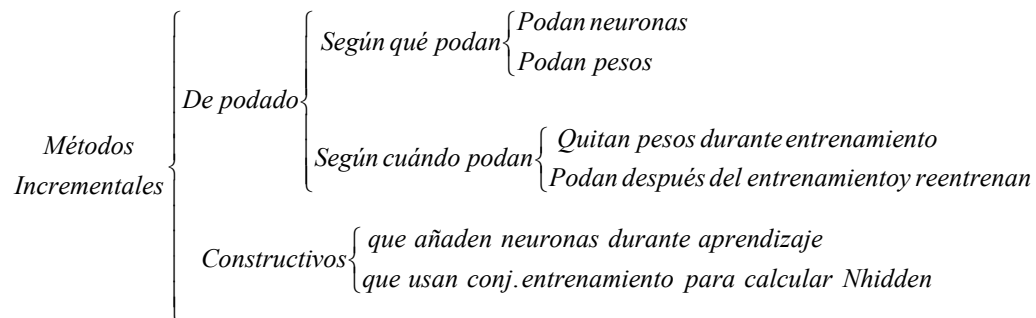
En la práctica, en la mayoría de las ocasiones, la selección del número correcto de unidades ocultas se realiza mediante un proceso de *prueba y error*. Se prueban varias estructuras con diferente número de neuronas ocultas y se selecciona la más idónea desde el punto de vista de la capacidad de generalización. Es decir, se busca la red que ofrece mejor eficiencia, medida ésta en el punto de convergencia de la red, el cual se obtiene utilizando técnicas de cros-validación.

Este procedimiento presenta el serio inconveniente de requerir una gran cantidad de tiempo, puesto que se deben de entrenar varias redes neuronales para

encontrar la arquitectura óptima. Además, se trata de un procedimiento subóptimo, ya que el número de unidades ocultas que se prueba es finito y es posible que no se encuentre la arquitectura más óptima entre las probadas.

Así, para resolver el problema de encontrar el tamaño correcto se han propuesto en la bibliografía varios métodos, dos de ellos de tipo incremental.

Podemos realizar primero una clasificación general de los mismos:



El primero de los métodos incremental se denomina de podado (“pruning”). El funcionamiento es el siguiente: comienza con una red de tamaño suficientemente grande y va eliminando los pesos o neuronas ocultas innecesarias. En cierto modo, se trata de combinar las ventajas de entrenar una red neuronal grande, con aquellas ventajas que presentan las redes pequeñas que se intentan conseguir después de la fase de podado.

Entre los algoritmos de podado se puede establecer varias divisiones.

La primera se hace en función de lo que se poda. Así, tendremos métodos que podan neuronas y métodos que podan pesos de la estructura. Esto último, la poda de pesos, se puede considerar diferente a la poda de unidades, puesto que para eliminar una neurona es necesario que se eliminen todos los pesos de la misma.

Otra clasificación diferente de procedimientos de podado se establece entre métodos que reducen el exceso de pesos durante el propio proceso de aprendizaje, y algoritmos en los que los procesos de entrenamiento y de podado son llevados a cabo en dos fases totalmente distintas. El primero, se realiza normalmente mediante la introducción en la función de error de penaltis o términos que penalizan el exceso de estructura (métodos de regularización). En el segundo, el podado se realiza después del proceso de entrenamiento. Esta alternativa no interfiere con el proceso de aprendizaje pero normalmente requiere un reentrenamiento de la red después del proceso de podado.

#### 4.1. DESCRIPCIÓN DEL PROBLEMA

El segundo método incremental al que nos referíamos, consiste en partir de una red con una estructura pequeña y gradualmente ir añadiendo neuronas hasta que se finaliza el proceso de aprendizaje. A este tipo de algoritmos se les suele denominar algoritmos constructivos.

Otra alternativa para el método anterior son los algoritmos constructivos que parten del conjunto de entrenamiento y a partir de él, determinan el número de unidades ocultas como si se tratase de una inicialización de la red. De hecho, en ellos, a la vez que se determina el número de neuronas ocultas también se suele determinar un valor inicial para los pesos de la red. Normalmente, estos métodos se basan en encontrar hiperplanos de decisión a partir del conjunto de entrenamiento, o usar un conjunto de prototipos extraídos a partir del conjunto de entrenamiento. En este capítulo no consideraremos este tipo de procedimientos al considerarlos más bien como métodos de inicialización de la red. En el capítulo 6 se han descrito y utilizado dos de estos métodos para el proceso de inicialización de pesos de la red.

En la referencia [2], se explica que los algoritmos constructivos que parten de una red pequeña y añaden neuronas poseen una serie de ventajas sobre los procedimientos de podado. Las ventajas son las siguientes:

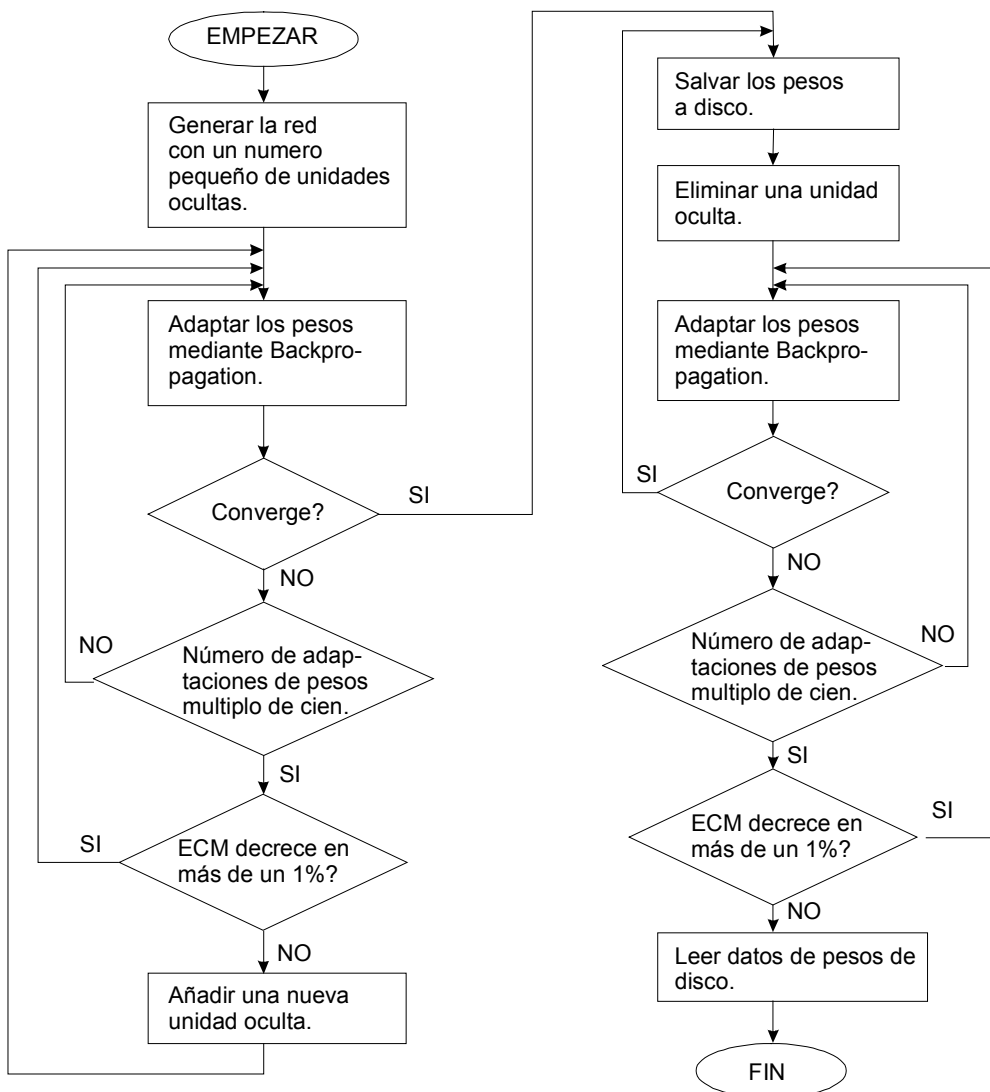
- *Para los algoritmos constructivos es evidente especificar el tamaño de la red inicial: la más pequeña, con sólo una neurona oculta. Sin embargo, para los procedimientos de podado uno no sabe en la práctica como de grande (número de neuronas ocultas) debe ser la red inicial y además el tamaño suele depender del problema a resolver.*
- *Los algoritmos constructivos siempre buscan primero soluciones de redes pequeñas, y por lo tanto son computacionalmente más económicos que los algoritmos de podado.*
- *Es posible encontrar redes de diferentes tamaños capaces de representar bien el problema. Los algoritmos constructivos encontrarán una red de tamaño menor ya que buscan primero redes de tamaño pequeño.*
- *Los métodos de podado normalmente miden el cambio en el error cuando una unidad oculta o peso es eliminado. Sin embargo, tales cambios sólo pueden ser aproximados, y por lo tanto puede que se introduzcan grandes errores, especialmente cuando el número de unidades a podar aumenta.*

Debido a estas razones el presente capítulo se centra en algoritmos constructivos y no en procedimientos de podado. Sin embargo, como veremos

más adelante los procedimientos de podado también son de utilidad dentro del campo de algoritmos constructivos.

El primer algoritmo constructivo propuesto en la bibliografía, fue propuesto por Ash [3]. Sin embargo, existe una versión posterior de Hirose y otros en [4], que con pocas modificaciones ha sido utilizada en la mayoría de los trabajos.

En la figura 1 hemos representado el algoritmo de Hirose. Veamos cuál es su funcionamiento.



**Fig.1. Algoritmo constructivo de Hirose.**

El algoritmo se puede dividir en dos fases, una de crecimiento de la red y otra de podado. Durante la fase de crecimiento, se inicializa en primer lugar la red con un número pequeño de neuronas ocultas. Hirose usó una unidad oculta en

#### 4.1. DESCRIPCIÓN DEL PROBLEMA

todos sus experimentos. Después se entrena dicha arquitectura mediante el algoritmo de Backpropagation y cada cien épocas se chequea el decremento del error cuadrático medio. Si el error ha decrecido más de un uno por cien no se realiza nada y se continúa con el entrenamiento. Por el contrario, si se ha llegado a un mínimo local y el error no decrece más de un uno por cien, se añade una nueva unidad oculta. Los pesos de esta unidad oculta se inicializarán aleatoriamente.

Esta situación se repite hasta que la red neuronal converge. El criterio de convergencia utilizado por Hirose se basa en un umbral especificado para el error cuadrático medio, calculado en el conjunto de entrenamiento. Cuando se sobrepasa el umbral se considera que la red neuronal ha convergido.

Una vez que la red ha convergido, se pasa a la fase de podado. En esta fase, se trata de disminuir el número de unidades ocultas que puede que haya crecido excesivamente en la fase anterior. La fase se inicia guardando los datos iniciales de los pesos en disco y eliminando una neurona oculta.

En sus experimentos, Hirose, elimina la neurona más reciente de las añadidas en la fase de crecimiento. Una vez que se ha eliminado la neurona se reentrena la red mediante Backpropagation.

De nuevo, cada cien épocas, se chequea el error cuadrático medio y si este ha decrecido más de un uno por cien se continua el entrenamiento. Si por el contrario, no ha decrecido más del uno por cien, se chequea la convergencia; si la red ha podido converger sin deterioro del error cuadrático medio la eliminación de la neurona ha tenido éxito y continuamos eliminando otra. Si por el contrario, no ha podido converger sin deterioro del error cuadrático medio, el número válido de neuronas era el que teníamos antes de la eliminación y recuperamos los datos salvados en disco.

Del algoritmo básico de Hirose descrito, ha habido distintas variantes. Citamos aquí algunas que no hemos incluido en nuestro estudio por considerarlas fuera de nuestros objetivos.

En primer lugar, en la referencia [5], se utiliza como algoritmo de entrenamiento en lugar de Backpropagation el llamado RLS (Recursive Least Squares). También, en la referencia [6], se usa un algoritmo diferente para el entrenamiento, el de Quasi-Newton. Obviamente, durante el entrenamiento se puede usar cualquier algoritmo de entrenamiento, sin embargo, no es objeto de esta Tesis Doctoral hacer un estudio comparativo sobre diferentes algoritmos de aprendizaje, nosotros hemos basado nuestra investigación en el uso de Multilayer Feedforward con Backpropagation, motivo por el cual no se han incluido estos algoritmos en nuestro estudio comparativo.



Por otra parte, en la referencia [7], se propone el siguiente procedimiento de entrenamiento, cada vez que se añade una nueva neurona, los pesos de las anteriores se dejan congelados (no se adaptan, en inglés “weight freezing”) y tan sólo se entrenan los pesos de la nueva neurona añadida. En la referencia, se proponen funciones objetivo con las cuales poder adaptar los pesos de la nueva neurona. Este esquema tampoco se ha implementado porque como se reconoce en la propia referencia, es subóptimo con respecto al esquema de Hirose, que adapta los pesos de todas las neuronas. La única razón a favor de este nuevo esquema es que reduce considerablemente la carga computacional puesto que sólo modifica los pesos de la nueva neurona introducida.

## 4.2. Descripción de los diferentes métodos

En primer lugar, tenemos que resaltar que el algoritmo de Hirose tal y como se describió antes, no se puede aplicar a problemas reales como los utilizados en esta Tesis. El problema que presenta el método, reside en el criterio de convergencia que utiliza durante la fase de crecimiento de la red. Se utiliza como criterio, el error cuadrático medio en el conjunto de entrenamiento y se fija un umbral de dicha magnitud para juzgar la convergencia de la red.

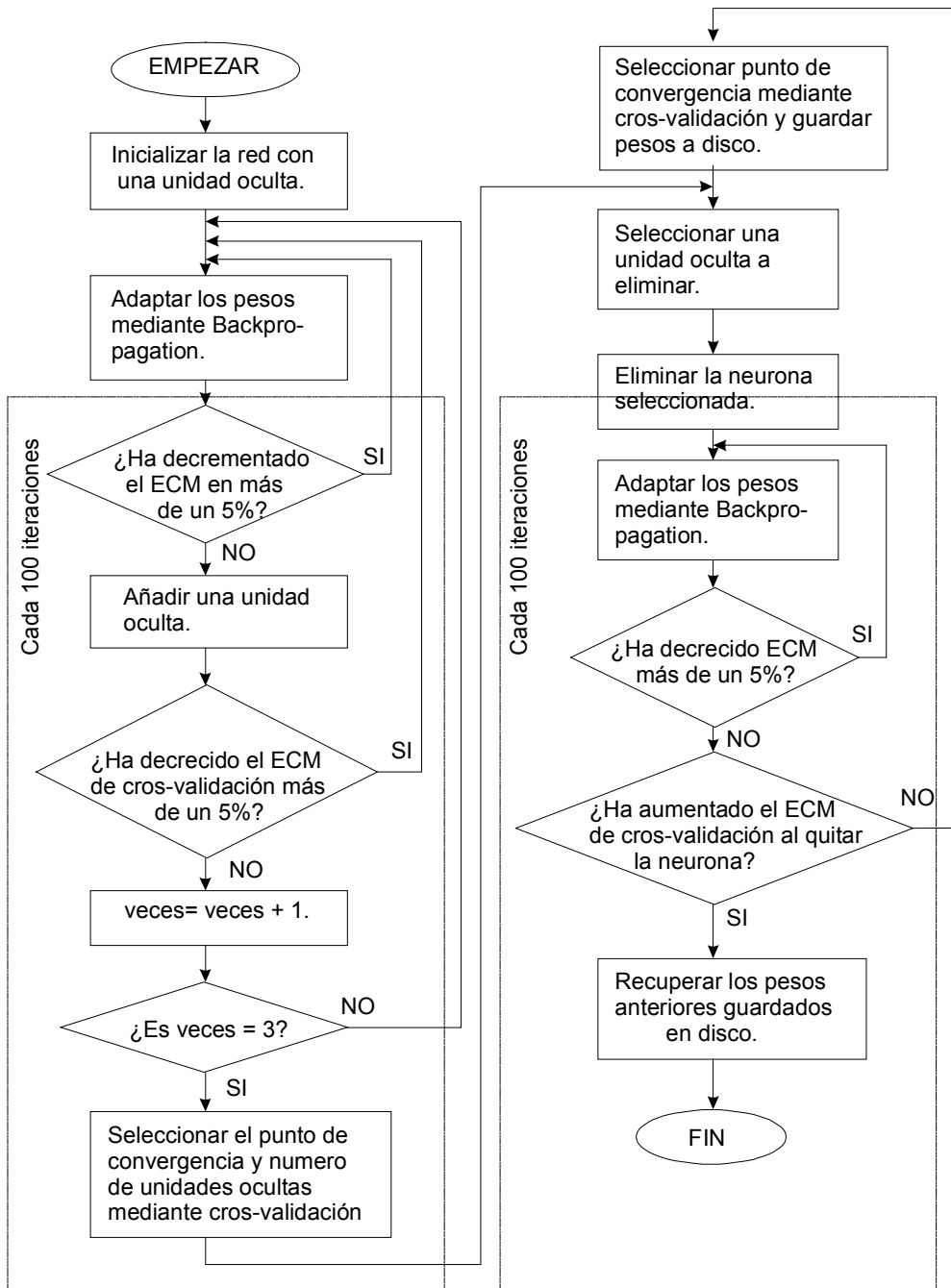
En la práctica, en un problema real en el que por ejemplo puede haber solapamientos entre clases, es imposible determinar cuál es el umbral apropiado para la convergencia. De hecho, Hirose utilizó su algoritmo en problemas artificiales en los que era posible llegar a un error cuadrático medio cero, y por lo tanto, sí que se podía establecer un umbral pequeño.

Para solventar este problema, una idea más apropiada es la que se sugiere en la referencia [8], en donde se propone usar *cross-validation* como criterio de convergencia.

Por ello, nosotros hemos modificado el algoritmo de Hirose para que utilice como criterio de convergencia *cross-validation*. En la figura 2, puede observarse el resultado.

La descripción del algoritmo sería la siguiente. Como antes, tendremos dos fases diferentes, la primera de crecimiento y la segunda de podado. En la fase de crecimiento, se inicializa la red con una sola neurona oculta y se adaptan los pesos mediante *Backpropagation*. Cada cien épocas se chequean ciertas magnitudes. En primer lugar, se comprueba si el error cuadrático medio (ECM) calculado con el conjunto de entrenamiento ha disminuido más de un 5%. Si es así, se continúa el proceso de entrenamiento. Si por el contrario, no ha disminuido, se supone que la red ha convergido con ese número de neuronas y se añade una nueva neurona.

Después, se comprueba si el error cuadrático medio de *cross-validation*, ECM\_cros (que se usa como criterio de convergencia) ha disminuido más de un 5%. Si es así, se continúa el proceso de aprendizaje, la introducción de la nueva neurona ha tenido éxito en el proceso de aumentar la capacidad de generalización. Si por el contrario, no es así, se concede una nueva oportunidad de añadir una nueva neurona para ver si con ella se consigue disminuir el error cuadrático medio de *cross-validation*, este proceso de dar una nueva oportunidad se puede llevar a cabo hasta dos veces, si a la tercera (veces=3) no se ha conseguido decrementar el ECM\_cros se considera que ya se ha superado el tamaño óptimo de la red y que por lo tanto no merece la pena añadir más neuronas. Comienza entonces la fase de podado.



**Fig.2. Algoritmo constructivo de Hirose modificado.**

Antes de la fase de podado, conviene resaltar que se calcula el punto de convergencia de la fase de crecimiento mediante cross-validación, lo cual quiere decir que el número de unidades ocultas final, no es el mismo con el que se finaliza la fase de crecimiento sino aquel valor que da el mínimo error de cross-validación.

## 4.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

En la fase de podado, lo primero que se hace, es seleccionar la unidad oculta a eliminar. Para ello, se utiliza un cierto criterio. Nosotros estudiaremos varios métodos o alternativas. El siguiente paso será eliminar la neurona seleccionada y adaptar los pesos mediante Backpropagation. Cada cien épocas de utilización del algoritmo Backpropagation, de nuevo, se chequean varios valores. En primer lugar, se chequea si el error cuadrático medio calculado con el conjunto de entrenamiento ha decrecido más de un 5%; si es así, la red todavía no ha convergido con el número de neuronas actual y se continúa con el entrenamiento. Si por el contrario, no ha decrecido más de un 5%, se supone que ya ha concluido el entrenamiento después del último podado de una neurona oculta y se pasa a comprobar si el error cuadrático medio de cros-validación ha aumentado, si es así, el último podado de una neurona se considera que no ha tenido éxito y se recuperan los pesos de la red convergida antes del podado. Si por el contrario no ha aumentado el error el proceso de podado ha tenido éxito y se intenta podar una nueva unidad oculta.

Con respecto al algoritmo original de Hirose cabe destacar que se ha modificado para usar en todo momento cros-validación como criterio de convergencia. La principal razón es que ello es una medida de la capacidad de generalización, se espera que el número de neuronas ocultas pueda ser aquel que maximice la capacidad de generalización. De esta forma, el algoritmo es aplicable a problemas reales y se eluden los problemas que presentaba el algoritmo de Hirose original.

Por otra parte, en el nuevo algoritmo aun no hemos especificado cuál es el criterio para seleccionar la neurona a eliminar. Para ello hay varias alternativas y se ha tratado de hallar experimentalmente cuál es la más apropiada.

Por último, tampoco se ha especificado cuales son los pesos de la nueva neurona que se incluye en cada iteración de la fase de crecimiento. Para ello, también disponemos de dos alternativas que se han evaluado experimentalmente.

Tenemos pues dos fases diferentes en nuestro algoritmo: la fase constructiva y la fase de podado. Cada una de estas fases será resuelta mediante alguno de los métodos que describimos a continuación.

Experimentalmente, hemos probado todas las combinaciones posibles entre los métodos de ambas fases.

### 4.2.1. Fase constructiva: determinación de nuevos pesos

Para realizar esta fase disponemos de dos métodos diferentes en la bibliografía.

#### MÉTODO DE HIROSE

Según el algoritmo original de Hirose, los nuevos pesos de las neuronas que se añaden en la fase constructiva se inicializan aleatoriamente dentro de un intervalo pequeño centrado en cero.

#### MÉTODO DE KHORASANI

En las referencias [9] y [10] Khorasani y Weng proponen otra alternativa de inicialización de los pesos de las neuronas que se añaden en la fase constructiva.

Proponen duplicar una neurona ya existente, es decir añadir una nueva neurona oculta cuyos pesos tomen el mismo valor que los de una neurona ya existente en la red.

Para seleccionar la neurona a duplicar se utiliza la medida llamada “rango de fluctuación más alto”, *HFR*, (“Highest Fluctuation Rate”, en inglés) definida por la ecuación (4.1):

$$\begin{aligned}
 HFR &= \text{Max}(\delta W_i \cdot \delta ho_i) \\
 \delta W_i &= |W_i(n) - W_i(n-1)| \\
 \delta ho_i &= |ho_i(n) - ho_i(n-1)|
 \end{aligned}
 \tag{4.1}$$

En ella,  $\delta W_i$  es la variación de los pesos de entrada a la neurona oculta como consecuencia de la última adaptación,  $\delta ho_i$  es la variación en la salida de la neurona como consecuencia de la adaptación de los pesos,  $W_i$  es el vector de pesos para la *i-ésima* neurona en la capa oculta (cuya dimensión coincide con el número de neuronas en la capa de entrada) y  $ho_i$  es la salida de la *i-ésima* neurona oculta.

La idea consiste en duplicar aquella neurona que tiene la máxima variación *HFR*. De esta forma, seleccionamos una neurona que está siendo activamente modificada por parte del algoritmo de aprendizaje.

Para seleccionar la neurona, calcularemos *HFR*, duplicaremos la neurona con mayor valor de este índice y los pesos de la nueva neurona coincidirán con los de la neurona que se duplica.

### 4.2.2. Fase de podado: elección de la neurona a podar

En lo que respecta a criterios para seleccionar la neurona a eliminar en la fase de podado, se han utilizado 13 métodos diferentes. A continuación, se describen con detalle cada uno de estos métodos.

#### MÉTODO 1 (Engelbrecht & Cloete)

En el tercer capítulo, vimos que los autores de la referencia [11], proponían un método que utilizando la medida de la sensibilidad clasificaba las unidades de entrada a la red según su importancia, pudiendo después podar las menos importantes.

Basándose en la misma idea, en [7] se da un método para podar neuronas ocultas. Se define la medida de la sensibilidad en la salida debida a una neurona oculta según:

$$S_{ij} = \frac{\partial o_i}{\partial ho_j} \quad (4.2)$$

En donde  $o_i$  es la salida de la  $i$ -ésima neurona de salida y  $ho_j$  es la salida de la  $j$ -ésima neurona oculta.

Para cada pareja de salida y unidad oculta obtendremos una sensibilidad diferente y esta también cambiará para cada patrón de entrenamiento, por lo que para tener una medida más exacta de la sensibilidad es más adecuado medir la sensibilidad media.

En la referencia, se propone medir la sensibilidad media en el conjunto de entrenamiento (“mean square average sensitivity”, en inglés) mediante la ecuación siguiente:

$$A_{ij} = \sqrt{\frac{\sum_{p=1}^P (S_{ij})^2}{P}} \quad (4.3)$$

En donde  $P$  es el número de patrones del conjunto de entrenamiento.

Con la anterior ecuación, tendremos varias medidas de sensibilidad para cada unidad oculta, tantas como número de salidas en la red. Para obtener una única medida de sensibilidad, se calcula el máximo, es decir, se mide la sensibilidad de una unidad  $j$  en la capa oculta, sobre todas las unidades  $i$  en la capa de salida:

$$S_i = \max_i \{A_{ij}\} \quad (4.4)$$

De esta forma, podaremos aquella neurona cuya sensibilidad  $S_i$  dada por la ecuación (4.4) sea menor.

### **MÉTODO 2 (Gosh & Tumer)**

En la referencia [12], se analizan diferentes métodos para reducir los parámetros de la red neuronal.

En concreto, referente a nuestro tema de estudio, se revisa un método de podado de unidades ocultas, denominado supresión de la actividad de unidades ocultas (“supression of hidden unit activations”, en ingles).

El método consiste en añadir un penalti a la función de error que se minimiza en el entrenamiento de forma que dicho término penalice la activación de las unidades ocultas.

En la referencia no se especifica la función penalti usada, solo sugieren que puede ser una función no lineal tipo sigmoide o logarítmica.

Nosotros en concreto hemos utilizado el siguiente penalti:

$$Penalti = \sum_i ho_i^2 = \sum_i (f(hi_i))^2 \quad (4.5)$$

$$hi_i = \sum_i wi_{ij} \cdot x_i; \quad f(v) = \frac{1}{1 + e^{-v}}$$

En donde  $ho_i$  es la activación (salida) de la  $i$ -ésima neurona oculta. Recordemos, que las funciones de activación en las redes que aquí utilizamos, son de tipo sigmoide.

Al añadir dicho término a la función de error y aplicar técnicas de gradiente descendente para minimizar su valor, se afecta directamente a las ecuaciones de adaptación de los pesos entre la capa de entrada y la oculta.

Así, para dichos pesos habría que sumar al término de Backpropagation la siguiente expresión:

$$\frac{\partial ho_i^2}{\partial wi_{ij}} = 2 \cdot ho_j \cdot ho_j \cdot (1 - ho_j) \cdot x_i \quad (4.6)$$

En donde  $ho_j$  es la salida de la  $j$ -ésima neurona oculta y  $x_i$  es el valor de la  $i$ -ésima entrada.

## 4.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

La modificación de los pesos se haría según la siguiente ecuación:

$$\Delta w_{ij} = -\lambda \cdot \frac{\partial ho_i^2}{\partial w_{ij}} - \beta \cdot \frac{\partial E}{\partial w_{ij}} \quad (4.7)$$

Aquí  $\lambda$  es un parámetro que controla el término de penalti, y  $\beta$  es el paso de adaptación usado en Backpropagation.

En este método, hay que modificar las funciones que se utilizan durante el entrenamiento en la fase de podado.

Una vez completado el entrenamiento en la forma especificada, para seleccionar la neurona a eliminar, se utiliza el criterio del llamado *índice*  $R_i$  que se define de la siguiente forma:

$$R_i = \frac{1}{P} \cdot \sum_{p=1}^P ho_i \quad (4.8)$$

Donde  $P$  es el número de patrones del conjunto de entrenamiento y  $ho_i$  es la salida de la neurona oculta  $i$ -ésima.

Así, la neurona que tiene un índice  $R_i$  menor es aquella que se selecciona y es podada.

### **MÉTODO 3 (Matsunaga “Goodness Factor”)**

En la referencia [13], Hagiwara, revisa un nuevo método de selección de la unidad a podar propuesto por Matsunaga y otros en [22].

El método mide el denominado *Factor de Bondad* (“Goodness Factor”, en inglés). El factor de bondad  $G_i$  para la  $i$ -ésima neurona oculta se define con la siguiente ecuación:

$$G_i = \sum_p \sum_j (wh_{ij} \cdot ho_i)^2 \quad (4.9)$$

En donde, la primera suma es sobre todos los patrones  $p$  del conjunto de entrenamiento,  $wh_{ij}$  es el peso entre la  $i$ -ésima neurona oculta, y la  $j$ -ésima neurona de salida y  $ho_i$  es la salida de la  $i$ -ésima neurona oculta.

Se podará aquella neurona oculta que tenga el menor factor de bondad. La red será entonces reentrenada para repetir el proceso probando a eliminar otra neurona.



Con este factor se trata de tener en cuenta el efecto que una neurona oculta tiene sobre las salidas, por ello se multiplica la salida de la neurona oculta por el peso entre esa neurona y la salida.

**MÉTODO 4 (Hagiwara “Consuming energy”)**

De nuevo en la referencia [13] se propone otro método que mide el llamado factor *Energía Consumida* (“Consuming energy”, en inglés) que serviría para elegir la neurona a podar.

El factor de energía consumida  $E_j$  de la  $j$ -ésima neurona de la capa oculta vendrá dado por la ecuación:

$$E_j = \sum_{p=1}^P \left\{ \sum_{k=1}^{N_{salida}} wh_{kj} \cdot o_k \cdot ho_j + \sum_{i=1}^{N_{input}} wi_{ji} \cdot ho_j \cdot x_i \right\} \quad (4.10)$$

En donde  $P$  es el número de patrones del conjunto de entrenamiento,  $N_{salida}$  y  $N_{input}$  el número de neuronas de salida y de entrada respectivamente,  $wh_{kj}$  es el peso entre la  $k$ -ésima unidad de salida y la  $j$ -ésima unidad oculta,  $o_k$  es la salida de la  $k$ -ésima unidad de salida,  $ho_j$  es la salida de la  $j$ -ésima unidad oculta,  $x_j$  es la salida de la  $i$ -ésima unidad de entrada y por último  $wi_{ji}$  es el peso entre la  $i$ -ésima unidad de entrada y la  $j$ -ésima unidad oculta.

Igual que en el método anterior el factor de energía consumida trata de medir el efecto que tiene una neurona oculta sobre la salida, pero este método es más complejo que el anterior, ya que en él se usan todos los pesos asociados a la neurona oculta.

De nuevo se debe de eliminar aquella neurona que tenga factor de energía consumida menor.

**MÉTODO 5 (Hagiwara “Weights power”)**

De nuevo Hagiwara en la referencia [13], propone un nuevo factor que es una versión simplificada del factor de bondad, el denominado *poder de los pesos* (“weights power”, en inglés).

El factor poder de los pesos de la  $j$ -ésima neurona oculta se define con la ecuación siguiente:

$$W_j = \sum_{k=1}^{N_{salida}} (wh_{kj})^2 + \sum_{i=1}^{N_{input}} (wi_{ji})^2 \quad (4.11)$$

## 4.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

En donde  $wh_{kj}$  es el peso entre la  $k$ -ésima neurona de salida y la  $j$ -ésima neurona oculta, y  $wi_{ji}$  es el peso entre la  $j$ -ésima neurona oculta y la  $i$ -ésima neurona de entrada.

El criterio de podado será eliminar aquella neurona oculta que tenga un menor factor de *poder de los pesos*.

### **MÉTODO 6 (Hagiwara “Badness factor”)**

En la referencia [14], se describe otro método que utiliza el llamado “Badness factor”, como criterio para eliminar neuronas ocultas en el proceso de podado.

Se define el “Badness factor” para la  $j$ -ésima unidad oculta por la ecuación:

$$BAD_j = \sum_{p=1}^P (e_j)^2 = \sum_{p=1}^P \left( \sum_{k=1}^{N_{salida}} wh_{jk} \cdot \delta_k \right) \quad (4.12)$$

El factor de la  $j$ -ésima neurona oculta, no es más que la suma, para todos los ejemplos del conjunto de entrenamiento, del error de Backpropagation (que se utiliza para adaptar los pesos) al cuadrado.

En el primer capítulo se pueden encontrar las ecuaciones de Backpropagation, en concreto  $\delta_k$  viene dada por la ecuación (1.14).

El criterio de podado para este método será el de eliminar aquella neurona oculta que tenga un mayor “Badness factor”.

### **MÉTODO 7 (Hirose)**

Aquí simplemente hemos incluido el método original de Hirose para podar neuronas ocultas, descrito en [4].

En el método de podado descrito por Hirose, las neuronas de la capa oculta se podan en orden inverso a como fueron añadidas, es decir, primero se poda la añadida más recientemente y así sucesivamente.

Ahora no se utiliza una ecuación como criterio de podado, como en el resto de métodos que hemos visto, es eminentemente un método empírico.

En la referencia, se comprobó que esta elección de neuronas a podar daba buenos resultados.

**MÉTODO 8 (Kamimura)**

En las referencias [15] y [16], Kamimura, describe un método basado en la minimización de la entropía de las neuronas ocultas para extraer una estructura más simple de la red.

Para ello propone modificar el algoritmo de aprendizaje, para disminuir simultáneamente la entropía de las neuronas ocultas y la función de error de Backpropagation. Disminuyendo la entropía lo que se consigue es que estén activadas y funcionando el menor número posible de neuronas ocultas.

En la referencia, se define la función entropía para el conjunto de neuronas ocultas de la siguiente forma:

$$H = -\alpha \cdot \sum_{i=1}^M p_i \cdot \log(p_i) \quad (4.13)$$

Donde  $M$  es el número de neuronas ocultas,  $\alpha$  es un parámetro que regula el valor de la entropía y  $p_i$  representa la actividad normalizada de la neurona oculta  $i$ , que se define con la siguiente ecuación:

$$p_i = \frac{ho_i}{\sum_{j=1}^M ho_j} \quad (4.14)$$

Donde  $ho_i$  es la salida de la  $i$ -ésima neurona oculta.

La minimización de la entropía afecta sólo a los pesos entre las unidades de entrada y las unidades de la capa oculta.

Cuando la entropía se minimiza, solo una de las neuronas ocultas se debe activar. Contrariamente, cuando la entropía se maximiza, las neuronas ocultas deben de activarse todas por igual. Por ello, se espera que si la entropía se hace suficientemente pequeña, solo un pequeño número de neuronas ocultas se debería activar. De esta forma, se puede utilizar la función entropía para determinar cuál sería la estructura más simple de la red, es decir, que neuronas ocultas se pueden eliminar.

La ecuación de adaptación de los pesos dada por Backpropagation se verá ahora modificada y su nueva expresión será la siguiente:

$$\Delta w_{ij} = -\alpha \cdot \frac{\partial H}{\partial w_{ij}} - \beta \cdot \frac{\partial E}{\partial w_{ij}} \quad (4.15)$$

## 4.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

Donde  $H$  es la función entropía anterior,  $E$  es el error cuadrático,  $w_{ij}$  son los pesos entre las entradas y las unidades ocultas,  $\beta$  es el paso de adaptación y  $\alpha$  el parámetro que regula la entropía.

La salida de una neurona oculta se puede obtener mediante la función  $f$  de transferencia (sigmoide) aplicada sobre la entrada a la neurona:

$$ho_i = f(hi_i) = f\left(\sum_{j=1}^{N_{input}} w_{ij} \cdot x_j\right) \quad (4.16)$$

Donde  $N_{input}$  es el número de neuronas de entrada a la red.

A partir de la ecuación (4.13) podemos calcular la derivada de la entropía respecto de los pesos aplicando para ello la regla de la cadena. Así resulta:

$$\frac{\partial H}{\partial w_{ij}} = -\sum_{i=1}^M \frac{\partial(p_i \cdot \log(p_i))}{\partial w_{ij}} = -\sum_{i=1}^M \left\{ \frac{\partial(p_i)}{\partial w_{ij}} \cdot \log(p_i) + p_i \cdot \frac{\partial(\log(p_i))}{\partial w_{ij}} \right\} \quad (4.17)$$

Operando se obtiene:

$$\frac{\partial H}{\partial w_{ij}} = -\sum_{i=1}^M \left\{ \frac{\partial p_i}{\partial w_{ij}} \cdot (\log p_i + 1) \right\} \quad (4.18)$$

Falta calcular la variación de  $p_i$  respecto de los pesos. A partir de las ecuaciones (4.14) y (4.16) obtenemos:

$$\frac{\partial p_i}{\partial w_{ij}} = \frac{\partial p_i}{\partial ho_i} \cdot \frac{\partial ho_i}{\partial hi_i} \cdot \frac{\partial hi_i}{\partial w_{ij}} = \frac{\left(\sum_{j=1}^M ho_j\right) - ho_i}{\sum_{j=1}^M ho_j} \cdot f'(hi_i) \cdot x_j \quad (4.19)$$

Podemos expresar la derivada final de la entropía por la ecuación:

$$\frac{\partial H}{\partial w_{ij}} = \Phi_i \cdot x_j \quad (4.20)$$

En donde  $x_j$  es la entrada  $j$ -ésima de la red neuronal y  $\Phi_i$  está dada por la siguiente ecuación:

$$\Phi_i = (\log p_i + 1) \cdot \left\{ \frac{\left(\sum_{j=1}^M ho_j\right) - ho_i}{\sum_{j=1}^M ho_j} \right\} \cdot f'(hi_i) \quad (4.21)$$

En donde  $ho_j$  es la salida de la  $j$ -ésima neurona oculta y  $f'(hi_i)$  es la derivada de la función de transferencia de la unidad oculta.

Por todo lo anterior, el entrenamiento durante el proceso de podado se verá modificado tal y como hemos indicado.

Una vez completado el entrenamiento en la forma especificada, para seleccionar la neurona a eliminar, Kamimura utiliza el llamado *índice*  $R_i$  que se define de la siguiente forma:

$$R_i = \frac{1}{P} \cdot \sum_{p=1}^P ho_i \quad (4.22)$$

Donde  $P$  es el número de patrones del conjunto de entrenamiento y  $ho_i$  es la salida de la neurona oculta  $i$ -ésima.

Se trata en definitiva de la media de las salidas de las neuronas ocultas para el conjunto de entrenamiento.  $R_i$  mide la respuesta de una neurona  $i$  de la capa oculta ante los patrones de entrada. Cuanto menor sea  $R_i$  menor será la respuesta por lo que dicha neurona será la candidata a eliminar.

Finalmente tenemos pues que eliminar aquella neurona cuyo índice  $R$  sea menor.

### **MÉTODO 9 (Kurita & otros)**

En la referencia [17] Kurita propone modificar el proceso de aprendizaje y añadir ruido aleatorio a la entrada de las neuronas ocultas.

Las características del ruido son: posee media cero y distribución uniforme dentro de un intervalo. Nosotros hemos utilizado el intervalo  $[-0.5, 0.5]$  de amplitud 1, tal como recomienda la referencia.

Sea  $hi_j$  la entrada a una neurona oculta, según las ecuaciones de la red que ya vimos en el primer capítulo dicha entrada se puede expresar en función de los pesos y umbrales entre la primera y segunda capa de la red.

$$hi_j = \sum_{i=1}^{Ninput} wi_{ij} \cdot x_i + \theta \quad (4.23)$$

Para hallar la salida de las unidades ocultas bastaría con aplicar la función de transferencia (sigmoide) sobre la entrada a la neurona.

Si se le añade ruido “n” a la ecuación (4.23) nos quedará:

#### 4.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

$$hi_j = \sum_{i=1}^{N_{input}} wi_{ij} \cdot x_i + n + \theta \quad (4.24)$$

A partir de esta ecuación el resto de ecuaciones del entrenamiento se verán modificadas por la presencia del ruido tal y como se describe en la referencia [15].

De esta forma, según sus resultados experimentales, se obtiene una red “esquelética”, es decir, en donde los pesos tienen un valor claro distinto de cero o por el contrario tienen un valor muy próximo a cero.

De nuevo en este método, el proceso de aprendizaje durante la fase de podado se verá modificado.

Para seleccionar la neurona a eliminar después del entrenamiento con ruido, en la referencia, no se propone ningún método en particular.

Nosotros hemos empleado uno de los ya comentados anteriormente, en concreto el método del factor de bondad (“Goodness factor”, en inglés), [22].

##### **MÉTODO 10 (Lim & Ho)**

De nuevo se utiliza una idea presentada en el tercer capítulo para determinar que neurona es menos importante. Ahora se aplica al caso de neuronas ocultas de forma que la que sea menos importante, será la candidata a podar.

En la referencia [18], se propone utilizar el *factor*  $A_j$  para la unidad oculta  $j$ -ésima, definido por la siguiente ecuación:

$$A_j = \sum_{k=1}^{N_{salida}} |wh_{jk}| \quad (4.25)$$

En donde  $wh_{jk}$  es el peso entre la  $j$ -ésima unidad oculta y la  $k$ -ésima unidad de salida, y la suma se realiza sobre todas las unidades de salida.

De acuerdo con este índice, se debe de eliminar aquella unidad oculta que tenga el menor valor del índice.

##### **MÉTODO 11 (Yasui)**

Yasui propone en la referencia [19] un nuevo criterio de podado de neuronas en la capa oculta.

Para ello modifica el proceso de adaptación de los pesos (Backpropagation) entre la capa oculta y la de salida introduciendo dos términos adicionales.

La nueva ecuación para la adaptación de los pesos será:

$$\Delta wh_{ij}(t) = -\eta \cdot \frac{\partial E}{\partial wh_{ij}} + \alpha \cdot \Delta wh_{ij}(t-1) + S + F \quad (4.26)$$

Como vemos si eliminamos los términos  $S$  y  $F$  la ecuación se reduce al algoritmo Backpropagation estándar.

Los valores de  $S$  y  $F$  se pueden calcular, según la referencia, mediante las ecuaciones siguientes.

La ecuación de  $S$  será:

$$S = -\frac{\varepsilon}{m-1} \cdot \text{signo}(wh_{ij}) \cdot \left( -|wh_{ij}| + \sum_{k=1}^m |wh_{ik}| \right) \quad (4.27)$$

Donde  $m$  es el número de neuronas ocultas,  $wh_{ij}$  es el peso entre la  $i$ -ésima neurona de salida y la  $j$ -ésima neurona oculta y  $\varepsilon$  es una constante positiva.

Este penalti es denominado en la referencia término de supresión y trata de minimizar el número de neuronas activas poniendo un penalti a todos los pesos asociados a una neurona de salida.

En cuando al término  $F$  su expresión es la siguiente:

$$F = -\frac{\gamma}{n-1} \cdot \frac{\partial E^2}{\partial wh_{ij}} \cdot \left( -|wh_{ij}| + \sum_{k=1}^n |wh_{kj}| \right) \quad (4.28)$$

Donde  $n$  es el número de unidades de salida y  $\gamma$  es una constante positiva.

A este término se le denomina en la referencia término de facilitación, trata de conseguir que crezcan todos los pesos asociados a una neurona oculta reforzando el aprendizaje de Backpropagation.

Igual que antes, este método modificará el aprendizaje que se realiza durante la fase podado de la red.

En la referencia no se propone ningún método para seleccionar que neurona oculta debemos de eliminar.

Nosotros hemos utilizado para seleccionar la neurona a podar el criterio que describimos anteriormente llamado "Factor de Bondad".

#### 4.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

##### **MÉTODO 12 (Bartlett “sensibilidad”)**

De nuevo se utiliza el concepto de sensibilidad como criterio para la elección de la neurona a podar.

En la referencia [20], Bartlett propone que se deberían eliminar aquellas neuronas no útiles de la red. Para ello, propone utilizar una medida de la importancia de las neuronas ocultas en función de las neuronas de salida.

En concreto, dicha medida consiste en calcular la sensibilidad media de la unidad oculta  $j$  respecto de la unidad de salida  $k$  y respecto de todos los patrones de entrada.

$$S_{kj} = \frac{1}{P} \cdot \sum_{p=1}^P \frac{\partial o_k}{\partial ho_j} \quad (4.29)$$

En donde  $P$  es el número de patrones en el conjunto de entrenamiento.

Para estimar la importancia total de una neurona oculta, basta con sumar sobre todas las neuronas de salida:

$$S_j = \sum_{k=1}^{N_{salida}} S_{kj} \quad (4.30)$$

Se podaría la neurona oculta que tenga un valor de la sensibilidad  $S_j$  menor.

##### **MÉTODO 13 (Bartlett “teoría de la información”)**

En la referencia [21], se proponen el uso de conceptos de teoría de la información para seleccionar la neurona oculta a eliminar.

En concreto, se propone medir la importancia total de una neurona oculta a partir de los valores de la interdependencia de la misma con las salidas y la redundancia de dicha neurona respecto del resto de neuronas de la capa oculta.

Así se define dicho factor como:

$$I(ho_i) = \sum_{k=1}^{N_{salida}} I(o_k, ho_i) - \sum_{m=1; m \neq i}^{N_{hidden}} R(ho_i, ho_m) \quad (4.31)$$

En esta expresión,  $I(o_k, ho_i)$  es la información mutua entre la salida  $k$  y la neurona oculta  $j$ . Dicha magnitud, da idea del grado de dependencia entre ambas neuronas, siendo mayor cuanto mayor sea la dependencia. Por lo tanto, el índice crece cuanto mayor sea la dependencia entre las salidas y la neurona oculta (esta será más importante).



Análogamente,  $R(ho_i, ho_m)$  es la información mutua entre la neurona  $i$  y la neurona  $m$  de la capa oculta. Este término trata de tener en cuenta la redundancia entre las distintas neuronas ocultas, por lo tanto, el índice disminuye cuando mayor sea la redundancia.

El valor de  $I(ho_i)$  determinará el grado de importancia de las neuronas ocultas. Se deberá podar aquella neurona que tenga un valor del índice menor, es decir, la neurona oculta menos importante.

Falta por determinar los valores de los sumandos. Estos puede calcularse a partir de las entropías de la siguiente forma:

$$I(o_k, ho_i) = 2 \cdot \left( \frac{H(ho_i) + H(o_k) - H(ho_i, o_k)}{H(ho_i) + H(o_k)} \right) \quad (4.32)$$

$$R(ho_i, ho_m) = 2 \cdot \left( \frac{H(ho_i) + H(ho_m) - H(ho_i, ho_m)}{H(ho_i) + H(ho_m)} \right)$$

Las entropías se pueden calcular a partir de las probabilidades. Para ello se divide el rango de variabilidad de los valores que pueden tomar las salidas en diferentes *bandas l* de igual longitud.

Así la probabilidad de que la salida  $o_k$  esté en una banda  $g$  se medirá por la ecuación:

$$p_{k,g} = \frac{\text{Número veces que } o_k \text{ cae en la banda } g}{\text{Número de patrones de entrenamiento}} \quad (4.33)$$

Para una unidad oculta, se realiza la misma operación, se divide el rango de variabilidad de los valores que puede tomar en  $l$  bandas de igual longitud y la probabilidad de que  $ho_k$  esté en una banda  $t$  se calculará de forma similar a la ecuación (4.33).

Las entropías pueden calcularse a partir de probabilidades con la expresión:

$$H(o_k) = - \sum_{g=0}^l p_{k,g} \cdot \log_2(p_{k,g}) \quad (4.34)$$

$$H(ho_m) = - \sum_{t=0}^l p_{m,t} \cdot \log_2(p_{m,t})$$

Por otra parte la expresión de la entropía conjunta sería la siguiente:

$$H(ho_i, ho_k) = - \sum_{g=0}^l \sum_{t=0}^l p_{g,t} \cdot \log_2(p_{g,t}) \quad (4.35)$$

Donde  $p_{g,t}$  sería la probabilidad de que  $ho_i$  se encuentre dentro de la banda número  $g$  y  $ho_k$  se encuentre dentro de la banda  $t$ . La suma se extiende para todas las bandas en las que hemos dividido  $ho_i$  y  $ho_k$ .

## 4.3. Metodología

En primer lugar hemos realizado una selección de los problemas sobre los que realizar nuestros experimentos. Hemos escogido un total de 11 problemas o bases de datos del repositorio UCI con datos normalizados en el intervalo [0,1]. Hemos abreviado sus nombres como sigue:

*Balance Scale Weight & Distance (BALANCE), Cylinder Bands (BANDS), BUPA Liver Disorders (BUPA), Credit Approval (CREDITO), Display1 (DISPLAY1), Glass Identification (GLASS), Heart Disease (HEART), Image Segmentation (IMAGEN), The Monk's Problems (MOK1, MOK2), 1984 U.S. Congressional Voting Records (VOTING).*

Como vimos al principio de este capítulo, nuestros experimentos se basarán en el algoritmo de “Hirose modificado” dado en la figura 2.

El algoritmo consta de una fase constructiva, en la cual se añadían neuronas a la red mediante el método de HIROSE o el de KHORASANI y después una fase de podado, en la cual se utiliza alguno de los 13 métodos descritos para eliminar neuronas.

Para la primera fase, para cada problema, partiremos de una red con una sola neurona en la capa oculta y se entrenará dicha red con Backpropagation.

El primer paso que dimos, fue determinar los parámetros que se utilizan en el algoritmo, es decir, el valor de la variable “veces” y del “porcentaje de decremento” en ECM y ECM\_cros.

Para ello, realizamos varias pruebas, escogiendo finalmente los valores de veces=3 y un valor en el decremento para los errores ECM y de cros-validación del 5%.

Después de determinados los parámetros, procedimos a la aplicación de la primera fase, la fase constructiva.

Realizamos dos experimentos diferentes, uno aplicando el procedimiento de HIROSE y otro aplicando el de KHORASANI. El procedimiento seguido fue el mismo para ambos, lo único que cambiaba era la manera de inicializar los pesos de la nueva neurona añadida. Para el método de HIROSE, los pesos de las nuevas neuronas ocultas se inicializan aleatoriamente dentro de un intervalo centrado en cero, y para el método de KHORASANI, se duplica el valor de los pesos de una neurona ya existente (aquella con mayor valor del índice *HFR*).

Como hemos dicho antes, se parte de una red con una sola neurona oculta y se aplica la fase constructiva del algoritmo. Cada 100 épocas de entrenamiento se comprobó si el error ECM había disminuido más de un 5%, en ese caso se continuó el entrenamiento 100 épocas más y se volvió a comprobar. Cuando el ECM no disminuía más de un 5% se intentaba añadir una neurona más. Se comprobaba si el ECM\_cros había decrecido más de un 5%, y en ese caso se continuaba entrenando con una neurona más. Cuando el ECM\_cros decrecía menos de un 5%, aún se daban hasta tres oportunidades nuevas de añadir neuronas y después se daba por concluido el aprendizaje.

Finalizada la fase constructiva, se determina mediante cross-validación el punto de convergencia y el número de neuronas ocultas final de la red (es decir los valores para los cuales ECM\_cros es mínimo).

Como siempre, el resultado final del entrenamiento de la red, va a depender de las condiciones iniciales, por ello, en nuestros experimentos para cada problema, hemos repetido la fase constructiva 30 veces con diferentes particiones de los datos e inicialización de los pesos. Así, hemos obtenido 30 resultados diferentes para cada problema.

Para esos 30 resultados anotaremos:

- *Valor máximo y mínimo del número de neuronas ocultas final.*
- *Valor medio del número de neuronas en los 30 experimentos.*
- *Desviación estándar en el número de neuronas ocultas.*
- *Porcentaje o eficiencia media con su error de las 30 redes.*

El proceso anterior se realizó para el caso del algoritmo constructivo de HIROSE y para el procedimiento constructivo de KHORASANI.

Comienza ahora la fase de podado. En la fase constructiva para cada uno de los dos procedimientos hemos obtenido 30 redes cada una con un determinado número de neuronas ocultas. Para realizar la fase de podado aplicamos a cada una de estas redes la fase de podado del algoritmo.

Durante la fase de podado se selecciona la neurona a podar, se entrena la nueva red sin esa neurona y cada 100 épocas se comprueban los valores de ECM y ECM\_cros. Se repite el algoritmo, eliminando neuronas hasta que se cumplan las condiciones de parada de la fase de podado.

La fase de podado se realizará para cada uno de los 13 métodos de podado estudiados.

### 4.3. METODOLOGÍA

Aplicamos uno de los métodos de podado a las 30 redes obtenidas en la fase constructiva y obtenemos 30 redes finales.

Para esas 30 redes finales anotaremos:

- *Valor medio del número de neuronas ocultas en los 30 experimentos.*
- *Desviación estándar en el número de neuronas ocultas.*
- *Porcentaje o eficiencia media con su error de las 30 redes.*

Esto se repite para cada método y tanto en el caso de los resultados del método constructivo de HIROSE, como para los de KHORASANI.

Tendremos para las 30 redes obtenidas aplicando el procedimiento de HIROSE, 30 redes finales para cada uno de los 13 métodos de podado.

También para las 30 redes obtenidas en la fase constructiva aplicando KHORASANI, tendremos 30 redes finales para cada uno de los 13 métodos de podado.

Para el método 2 no se da en la referencia [12] el valor adecuado para el parámetro  $\lambda$ . Nosotros hemos probado con 5 valores diferentes que hemos denominado:  $a=0.01$ ,  $b=0.001$ ,  $c=0.0001$ ,  $d=0.00001$ ,  $e=0.000001$ . En los resultados experimentales tendremos 5 diferentes resultados según el valor de  $\lambda$  escogido.

En el método 8 el parámetro  $\alpha$  se ha escogido con un valor pequeño, tal como especifica la referencia [15] y [16]. Hemos utilizado 5 valores diferentes que hemos denominado:  $a=0.01$ ,  $b=0.001$ ,  $c=0.0001$ ,  $d=0.00001$ ,  $e=0.000001$ . De nuevo en los resultados experimentales tendremos 5 diferentes resultados según el valor de  $\alpha$  escogido.

En el método 11, para los parámetros  $\varepsilon$  y  $\gamma$  hemos escogido los siguientes valores ( $\varepsilon$ ,  $\gamma$ ) que denominamos:  $a=(0.0001,0.1)$ ,  $b=(0.0001,0.5)$ ,  $c=(0.0001,1)$ ,  $d=(0.0001,3)$ ,  $e=(0.0008,0.1)$ ,  $f=(0.0008,0.5)$ ,  $g=(0.0008,1)$ ,  $h=(0.0008,3)$ . Así obtendremos ocho resultados diferentes para la aplicación de este método.

Como proceso final faltaría comparar cuál de los métodos es más eficiente. Realmente lo que hay que determinar es que procedimiento es mejor en la fase constructiva y para ese, ver que método de podado funciona mejor.

Los criterios utilizados para decidir cuál es mejor son los de siempre: la red que suministre mejor capacidad de generalización y sea la más pequeña. El criterio más restrictivo es la capacidad de generalización y solo en el caso de que

las capacidades que se comparen sean similares se escogerá el método que proporcione una red más pequeña.

Para determinar que red es más pequeña, nosotros hemos estudiado el valor medio del número de neuronas ocultas (que se obtiene a partir de los 30 experimentos).

Todos los resultados se encuentran en el apartado siguiente, primero los correspondientes a la fase constructiva para los dos métodos HIROSE y KHORASANI y todas las bases de datos usadas, y después para cada base de datos se darán los resultados de los 13 métodos de podado, tanto para el caso de partir de las redes obtenidas aplicando HIROSE, como para el caso de partir de las de KHORASANI.

## 4.4. Desarrollo Experimental

En la segunda sección de este capítulo se describió el proceso a seguir para determinar el número de neuronas ocultas apropiado en una red neuronal, para resolver un cierto problema.

Vimos que el procedimiento a seguir sería el del algoritmo de Hirose modificado que se encuentra en la figura 2. El algoritmo consta de dos fases diferentes la primera de crecimiento y la segunda de podado. En la fase de crecimiento inicializamos la red con una sola neurona oculta, y siguiendo el algoritmo se van añadiendo neuronas hasta alcanzar la condición de parada. Después llega la fase de podado, en la cual se determina si la eliminación de neuronas ocultas en la red encontrada en la primera fase no empeora la capacidad de generalización de la misma, y si es así, se eliminan neuronas hasta que la eficiencia se deteriore.

Partiendo de una red con una sola unidad oculta hemos aplicado en primer lugar la fase constructiva del algoritmo.

Nuestros experimentos se han aplicado a 11 problemas del repositorio UCI: BALANCE, BANDS, BUPA, CREDITO, DISPLAY1, GLASS, HEART, IMAGEN, MOK1, MOK2, VOTING.

En las Tablas 1 y 2 se muestran los resultados obtenidos para la fase constructiva, durante la que se añaden neuronas. Los experimentos se realizaron tal como se describió en la metodología.

En la Tablas 1 y 2 se encuentran los valores del número medio de neuronas que se alcanza después de la fase constructiva ("*N<sub>hidden medio</sub>*"), la desviación estándar en el número de neuronas ("*Desviación Estándar en N<sub>hidden</sub>*"), el número máximo y mínimo de neuronas que se alcanzarán en el total de 30 redes entrenadas y por último, el porcentaje en el test con la configuración final de pesos, es decir, la capacidad de generalización.

La Tabla 1, contiene los resultados de la aplicación del método de HIROSE, donde los pesos de la nueva neurona añadida se escogen aleatoriamente.

En la Tabla 2 se muestran los resultados de la aplicación del método de KHORASANI, en donde para añadir una nueva neurona se selecciona una neurona existente y se duplica.

Tabla 1. Resultados de la Fase Constructiva de HIROSE.

	Nhidden Medio	Desviación Estándar en Nhidden	Nhidden Mínimo	Nhidden Máximo	Porcentaje
BALANCE	1.87	1.11	1	4	87.6±0.6
BANDS	1.23	0.68	1	4	69.8±0.9
BUPA	2.57	1.52	1	6	67.0±1.2
CREDITO	1.73	1.41	1	6	85.5±0.5
DISPLAY1	8.37	1.73	5	12	72.3±0.3
GLASS	7.5	3.47	4	22	92.5±0.9
HEART	1.1	0.55	1	4	82.0±0.8
IMAGEN	9.53	3.55	4	18	94.2±0.3
MOK1	4.33	1.27	3	6	95.3±0.9
MOK2	3.2	1.45	1	6	75.0±0.8
VOTING	1.07	0.25	1	2	95.3±0.4

Tabla 2. Resultados de la Fase Constructiva de KHORASANI.

	Nhidden Medio	Desviación Estándar en Nhidden	Nhidden Mínimo	Nhidden Máximo	Porcentaje
BALANCE	2.63	1.81	1	6	87.6±0.7
BANDS	1	0	1	1	70.2±0.9
BUPA	1.73	1.14	1	4	65.4±1.2
CREDITO	1.43	1.04	1	4	85.3±0.5
DISPLAY1	3.5	0.78	2	4	33.9±0.5
GLASS	4.77	1.59	1	7	74.5±1.4
HEART	1.03	0.18	1	2	82.1±0.8
IMAGEN	5.5	0.63	4	7	54.8±0.7
MOK1	3.23	1.50	1	7	76.3±0.7
MOK2	2.3	1.34	1	4	66.2±0.5
VOTING	1.23	0.77	1	5	95.3±0.4

Comparando los resultados de ambas tablas, se observa que para el método de KHORASANI se obtiene un número medio de neuronas ocultas final inferior. Esto se puede observar para todas las bases de datos excepto para BALANCE.

Sin embargo, comparando los resultados de la capacidad de generalización se puede ver que en algunos problemas KHORASANI proporciona un porcentaje claramente inferior e inaceptable en comparación con HIROSE.

Por ejemplo, para el problema DISPLAY1, el método HIROSE obtiene un valor de  $72.3 \pm 0.3$  y en KHORASANI tan sólo un  $33.9 \pm 0.5$ . Lo mismo ocurre con GLASS, IMAGEN, MOK1 y MOK2. Si miramos el número de neuronas en esos problemas, vemos que los valores de "Nhidden medio" para KHORASANI son claramente inferiores a los alcanzados por HIROSE. Por ello, podemos concluir que en algunos problemas KHORASANI llega a un número de neuronas

#### 4.4. DESARROLLO EXPERIMENTAL

demasiado bajo, insuficiente para resolver el problema y ello repercute en una capacidad de generalización inferior.

De las dos alternativas para la fase constructiva, en la cual que se añaden neuronas, la conclusión de acuerdo con nuestros resultados experimentales, será que el método de HIROSE es el más apropiado y debe ser utilizado.

Podemos ahora comparar los resultados del método constructivo de HIROSE con los que se obtienen del procedimiento clásico de seleccionar la arquitectura de la red por prueba y error.

Para ello, en la Tabla 3 se dan los resultados de la arquitectura de red seleccionada mediante el procedimiento de prueba y error. En ella, tenemos el número de neuronas ocultas en la columna encabezada por “*Nhidden*” y el porcentaje en el conjunto de test, es decir, la capacidad de generalización.

**Tabla 3. Resultados de la elección de *Nhidden* por el método clásico de prueba y error.**

	<b>Nhidden</b>	<b>Porcentaje</b>
<b>BALANCE</b>	6	58±9
<b>BANDS</b>	13	68.2±1.3
<b>BUPA</b>	8	68.3±1.6
<b>CREDITO</b>	9	86.6±0.6
<b>DISPLAY1</b>	15	71.9±1.3
<b>GLASS</b>	6	78.9±1.1
<b>HEART</b>	6	80.6±0.9
<b>IMAGEN</b>	13	91.0±0.5
<b>MOK1</b>	5	73.6±1.1
<b>MOK2</b>	14	73.6±1.1
<b>VOTING</b>	13	95.9±0.6

Comparando los resultados de las Tablas 1 y 3, lo primero que llama la atención es que el método constructivo de HIROSE llega a una arquitectura de red con un número de neuronas ocultas claramente inferior al que se obtiene mediante el procedimiento de prueba y error.

Por ejemplo, para el problema BALANCE con el procedimiento constructivo se obtiene una media de 1.87 neuronas ocultas y mediante prueba y error 6, en total unas 4 neuronas más. Lo mismo ocurre con todos los problemas con la excepción de GLASS, en donde el procedimiento constructivo tiene una media de 7.5 neuronas y mediante prueba y error se obtienen 6.



El hecho de que el procedimiento constructivo proporcione redes neuronales con menos unidades ocultas de las que se obtienen mediante prueba y error, no significa nada sino examinamos la capacidad de generalización.

Es perfectamente posible que los resultados constructivos sean más apropiados.

Para entenderlo mejor, debemos comentar cómo se han seleccionado las arquitecturas mediante prueba y error, y ver las serias críticas que existen actualmente en la bibliografía en contra de la utilización de este procedimiento.

El procedimiento de prueba y error consiste en entrenar redes neuronales con distinto número de neuronas ocultas. En concreto, en nuestros experimentos se entrenaron redes con números comprendidos entre 1 y 27 neuronas ocultas. Posteriormente, mediante un conjunto de cros-validación se obtiene la capacidad de generalización de cada una de dichas arquitecturas. Finalmente, se selecciona la arquitectura que proporciona una capacidad de generalización mejor en el conjunto de cros-validación.

Recientemente, se puede encontrar un artículo [23] que proporciona resultados experimentales y conclusiones en contra de la idoneidad del procedimiento de prueba y error. En concreto, de los resultados experimentales se concluye que la capacidad de generalización que se espera de un procedimiento de prueba y error no es mejor que la que se obtendría eligiendo aleatoriamente la arquitectura. Además, se concluye también, que se suele obtener un número de neuronas mayor del normalmente necesario.

Una vez comentado esto, para determinar cuál de los dos algoritmos, el procedimiento de prueba y error o el algoritmo constructivo, selecciona de forma más idónea la arquitectura; pasamos a examinar la capacidad de generalización de ambos métodos.

Comparando los resultados de las Tablas 1 y 3 se puede concluir que el procedimiento constructivo proporciona en general una capacidad de generalización mayor.

Ha sido así para los problemas BALANCE, BANDS, GLASS, HEART, IMAGEN, MOK1 y MOK2. En algunos casos, como en BALANCE, la diferencia es muy importante, el algoritmo constructivo proporciona una generalización media de  $87.6 \pm 0.6$  y el procedimiento de prueba y error tan sólo del  $58 \pm 9$ .

#### 4.4. DESARROLLO EXPERIMENTAL

Por otra parte, hay problemas en los que el procedimiento constructivo obtuvo una capacidad de generalización inferior. Estos son, BUPA, CREDITO y VOTING pero la diferencia entre ambos es mínima de alrededor de un 1%.

Por todo ello, pueden considerarse los resultados del algoritmo constructivo como mejores que los de prueba y error y podemos concluir también que el número de neuronas ocultas seleccionadas mediante el procedimiento constructivo es el más adecuado.

Hasta aquí, hemos examinado los resultados de la fase constructiva del algoritmo, en la que se añaden neuronas a la red, pasemos ahora a observar los resultados de la fase de podado.

De nuevo realizamos los experimentos tal como se describe en la metodología.

En las Tablas 4 y 5 del apéndice del capítulo, tenemos los resultados de la fase de podado. La Tabla 4 corresponde a la fase de podado posterior a la utilización del método de HIROSE en la fase constructiva, y la Tabla 5 se corresponde con la utilización del podado sobre el punto final del método de KHORASANI en la fase constructiva.

En nuestros experimentos, aplicamos simultáneamente todos los métodos y los resultados fueron analizados al final, por lo que hasta ese momento no se pudo determinar cuál de los dos procedimientos en la fase constructiva resultaría mejor. Los resultados de la Tabla 5, se incluyen tan sólo para satisfacer la curiosidad del lector.

En ellos se puede observar que no se ha incluido el problema BANDS en el caso del método de KHORASANI. Esto es debido a que en la fase constructiva todas las redes obtenidas tenían tan solo una unidad oculta, por lo cual no puede haber fase de podado posterior.

Nos centraremos pues, en los resultados de la Tabla 4, ya que como hemos concluido anteriormente el método constructivo de HIROSE es el más apropiado.

En la Tabla 4, se incluyen para las distintas bases de datos, en la columna encabezada por "*Nhidden*", el número de unidades ocultas resultante después de la fase de podado. También se incluye la desviación estándar de dicho número entre paréntesis. Por otra parte, en la columna encabezada como "*POR*" se incluye la capacidad de generalización obtenida después de la fase de podado.

Observando los resultados de la Tabla 4, se puede ver que las diferencias entre las capacidades de generalización resultantes de los distintos procedimientos

de podado son mínimas y prácticamente inexistentes. La mayor diferencia entre el peor y mejor método, desde el punto de vista de la capacidad de generalización, se encuentra en el problema MOK2 y es de 1.6%.

Comparando las Tablas 1 y 4 que nos dan los resultados de la fase constructiva y de podado respectivamente, se puede ver que la capacidad de generalización alcanzada en la fase constructiva es similar a la que se obtiene después de la fase de podado.

Es decir, la reducción adicional de neuronas que se consigue en la fase de podado no repercute en un aumento de la capacidad de generalización. Por lo tanto, la fase de podado debe verse simplemente, como un mecanismo para conseguir un modelo de red neuronal más sencillo, pero sin ninguna repercusión en la capacidad de generalización de la red.

Observando de nuevo la Tabla 4, vemos que sí que hay diferencias entre los métodos en el número medio de neuronas alcanzado después de la fase de podado. En algunos casos, dicha diferencia, llega a ser de una neurona oculta o incluso dos en uno de los casos (MOK1, DISPLAY1, GLASS, IMAGEN).

Comparando los números medios de unidades ocultas podemos finalmente concluir que el mejor método de podado es el denominado “*Método 9*”. Dicho método obtiene los mejores resultados en 9 de las 11 bases de datos.

Por último, y a modo de conclusiones finales, podemos decir que los procedimientos constructivos obtienen un número de neuronas adecuado para resolver el problema con una capacidad de generalización superior a la obtenida por el procedimiento de prueba y error. De entre los procedimientos analizados los mejores han sido el método de HIROSE para la fase constructiva y el denominado “*Método 9*” para la fase de podado.

## 4.5. Conclusiones

A lo largo de este capítulo hemos estudiado los diferentes métodos existentes en la bibliografía para seleccionar el número de neuronas ocultas necesario para la red Multilayer Feedforward. El procedimiento que proponemos es la utilización de un algoritmo similar al de Hirose en el cual existe una fase constructiva inicial, durante la cual se añaden neuronas, y una fase de podado posterior, en la que se elimine el exceso de neuronas añadidas en la primera fase. La diferencia con el algoritmo original de Hirose es la condición de parada del algoritmo: nosotros utilizamos como criterio de convergencia de las redes cross-validación.

Se ha aplicado dicho algoritmo a los métodos constructivos de HIROSE y KHORASANI, llegando a la conclusión de que la capacidad de generalización obtenida en la fase constructiva para el algoritmo de HIROSE es mejor que la de KHORASANI, obteniendo al mismo tiempo un número de neuronas adecuado para resolver el problema.

También hemos estudiado la eficiencia de los procedimientos de prueba y error frente al método de HIROSE, concluyendo igualmente que la capacidad de generalización es superior para HIROSE.

Por último, utilizando como método constructivo HIROSE, durante la fase de podado se observa que la capacidad de generalización obtenida es similar a la de la fase constructiva, con lo cual la reducción de neuronas no repercute en la mejora de la capacidad de generalización, sin embargo si que simplifica la estructura de las redes.

Si se comparan los números medios de unidades ocultas se puede concluir que el mejor método de podado es el denominado "*Método 9*" cuyos autores son Kurita y otros, en combinación con el método constructivo de HIROSE.

## 4.6. Bibliografía del Capítulo

- [1] Yu, X. H., “Can Backpropagation error surface have no local minima”. IEEE Transactions on Neural Networks, vol. 3, pp. 1019-1021, 1992.
- [2] Kwok, T. Y., Yeung, D. Y., “Constructive algorithms for structure learning in Feedforward Neural Networks for regression problems”. IEEE Transactions on Neural Networks, vol. 8, no. 3, pp. 630-645, 1997.
- [3] Ash, T., “Dynamic node creation”. Connection Science, vol. 1, no. 4, pp. 365-375, 1989.
- [4] Hirose, Y., Yamashita, K., Hijiya, S., “Back-propagation algorithm which varies the number of hidden units”. Neural Networks, vol. 4, pp. 61-66, 1991.
- [5] Azimi-Sadjadi, M. R., Sheedvash, S., Trujillo, F. O., “Recursive dynamic node creation in Multilayer Neural Networks”. IEEE Transactions on Neural Networks, vol. 4, no. 2, pp. 242-256, 1993.
- [6] Setiono, R., Hui, L. C. K., “Use of Quasi-Newton method in a Feedforward neural network construction algorithm”. IEEE Transactions on Neural Networks, vol. 6, no. 1, pp. 273-277, 1995.
- [7] Kwok, T.Y., Yeung, D.Y., “Objective functions for training new hidden units in constructive neural networks”. IEEE Transactions on Neural Networks, vol. 8, no. 5, pp. 1131-1148, 1997.
- [8] Ji, C., Psaltis, D., “Network synthesis through data-driven growth and decay”. Neural Networks, vol. 10, no. 6, pp. 1133-1141, 1997.
- [9] Khorasani, K., Weng, W., “Structure adaptation in feed-forward neural networks”. Proceedings of IEEE International Conference on Neural Networks, vol. 3, pp. 1403-1408, 1994.
- [10] Weng, W., Khorasani, K., “An adaptive structure neural networks with application to EEG automatic seizure detection”. Neural Networks, vol. 9, no. 7, pp. 1223-1240, 1996.
- [11] Engelbrecht, A.P., Cloete, I., “A sensitivity analysis algorithm for pruning feedforward neural networks”. Proceedings of the International Conference on Neural Networks, vol 2, pp. 1274-1277, 1996.
- [12] Gosh, J., Tumer, K., “Structural adaptation and generalization in supervised feedforward networks”. Journal of Artificial Neural Networks, vol. 1, no. 4, pp. 431-458, 1994.
- [13] Hagiwara, M., “A simple and effective method for removal of hidden units and weights”. Neurocomputing, vol. 6, pp. 207-218, 1994.
- [14] Hagiwara, M., “Backpropagation with selection-reduction of learning time and elimination of hidden units”. Systems and Computers in Japan, vol. 23, no. 8, pp 46-54, 1992.
- [15] Kamimura, R., “Principal Hidden unit analysis with minimum entropy method”. Proceedings of the International Conference on Artificial Neural Networks, pp. 760-763. 1993.
- [16] Kamimura, R., “Principal Hidden unit analysis generation of simple networks by minimum entropy method”. Proceedings of 1993 International Joint Conference on Neural Networks, pp. 317-320, 1993.
- [17] Kurita, T., H-Asoh, Umeyama, S., Akaho, S., Hosomi, A., “A structural learning by adding independent noises to hidden units”. Proceedings of IEEE International Conference on Neural Networks, vol. 1, pp. 275-278, 1994.

#### 4.6. BIBLIOGRAFÍA DEL CAPÍTULO

- [18] Lim, S.F., Ho, S.B., “Dynamic creation of hidden units with selective pruning in Backpropagation”. World Congress on Neural Networks, vol. 3, pp. 492-497, 1994.
- [19] Yasui, S., “Convergence suppression and divergence facilitation: Minimum and joint use of hidden units by multiple outputs”. Neural Networks, vol. 10, no. 2, pp. 353-367, 1997.
- [20] Bartlett, E. B., “A dynamic node architecture scheme for layered neural networks”. Journal of Artificial Neural Networks, vol. 1, no. 2, pp. 229-245, 1994.
- [21] Bartlett, E. B., “Dynamic node architecture learning: An information theoretic approach”. Neural Networks, vol. 7, no. 1, pp. 129-140, 1994.
- [22] Matsunaga, Y., Nakade, Y, Yamakawa, O., Murase, K., “A Backpropagation algorithm with automatic reduction of association units in Multilayered Neural Network”. Proceedings of the International Joint Conference on Neural Networks, pp. 783-788, 1991.
- [23] Andersen, T., Martinez, T., “Cross Validation and MLP Architecture Selection”. International Joint Conference on Neural Networks (IJCNN'99). 1999.

## 4.7. Apéndice del Capítulo

*Tabla 4. Fase de podado para redes construidas con método de HIROSE.*

	BALANCE		BANDS		BUPA		CREDITO	
	Nhidden	POR	Nhidden	POR	Nhidden	POR	Nhidden	POR
<b>Método 1</b>	1.3 (0.59)	87.9±0.6	1.2 (0.55)	69.8±0.9	1.57 (0.86)	67.0±1.1	1.6 (1.22)	85.5±0.5
<b>Método 2a</b>	1.7 (0.92)	87.6±0.6	1.23 (0.67)	69.8±0.9	2.26 (1.39)	66.9±1.2	1.63 (1.27)	85.6±0.5
<b>Método 2b</b>	1.43 (0.86)	87.6±0.6	1.2 (0.61)	69.8±0.9	2.07 (1.20)	67.3±1.2	1.53 (1.11)	85.6±0.5
<b>Método 2c</b>	1.33 (0.55)	87.7±0.6	1.2 (0.61)	69.8±0.9	1.57 (0.77)	67.0±1.2	1.6 (1.16)	85.6±0.5
<b>Método 2d</b>	1.3 (0.65)	87.7±0.6	1.2 (0.61)	69.8±0.9	1.5 (0.78)	67.0±1.1	1.6 (1.16)	85.5±0.5
<b>Método 2e</b>	1.3 (0.65)	87.7±0.6	1.2 (0.61)	69.8±0.9	1.5 (0.78)	67.0±1.1	1.6 (1.16)	85.5±0.5
<b>Método 3</b>	1.3 (0.59)	87.9±0.6	1.2 (0.55)	69.8±0.9	1.57 (0.86)	67.0±1.1	1.6 (1.22)	85.5±0.5
<b>Método 4</b>	1.23 (0.50)	87.8±0.6	1.17 (0.46)	69.8±0.9	1.63 (0.93)	67.0±1.1	1.6(1.22)	85.5±0.5
<b>Método 5</b>	1.3 (0.59)	87.9±0.6	1.2 (0.55)	69.8±0.9	1.57 (0.86)	67.0±1.1	1.6 (1.22)	85.5±0.5
<b>Método 6</b>	1.6 (0.77)	88.2±0.7	1.23 (0.67)	69.8±0.9	2.5 (1.48)	66.8±1.1	1.7 (1.37)	85.5±0.5
<b>Método 7</b>	1.3 (0.59)	87.9±0.6	1.2 (0.55)	69.8±0.9	1.57 (0.86)	67.1±1.1	1.6 (1.22)	85.5±0.5
<b>Método 8a</b>	1.53 (0.90)	87.6±0.6	1.2 (0.61)	69.8±0.9	1.83 (1.05)	67.1±1.2	1.6 (1.24)	85.5±0.5
<b>Método 8b</b>	1.53 (0.90)	87.6±0.6	1.2 (0.61)	69.8±0.9	1.4 (0.77)	67.1±1.1	1.57 (1.14)	85.5±0.5
<b>Método 8c</b>	1.27 (0.58)	87.7±0.6	1.2 (0.61)	69.8±0.9	1.53 (0.78)	67.1±1.1	1.6 (1.16)	85.5±0.5
<b>Método 8d</b>	1.23 (0.57)	87.6±0.6	1.2 (0.61)	69.8±0.9	1.5 (0.78)	67.0±1.1	1.6 (1.16)	85.5±0.5
<b>Método 8e</b>	1.23 (0.57)	87.6±0.6	1.2 (0.61)	69.8±0.9	1.5 (0.78)	67.0±1.1	1.6 (1.16)	85.5±0.5
<b>Método 9</b>	1.2 (0.48)	87.8±0.7	1.17 (0.46)	69.8±0.9	1.87 (1.01)	67.1±1.1	1.5 (1.01)	85.5±0.5
<b>Método 10</b>	1.3 (0.59)	87.9±0.6	1.2 (0.55)	69.8±0.9	1.57 (0.86)	67.0±1.1	1.6 (1.22)	85.5±0.5
<b>Método 11a</b>	1.83 (1.05)	87.7±0.6	1.17 (0.46)	69.8±0.9	2.23 (1.38)	66.9±1.1	1.7 (1.37)	85.5±0.5
<b>Método 11b</b>	1.8 (1.12)	87.7±0.6	1.2 (0.55)	69.9±0.9	2.37 (1.45)	66.8±1.2	1.7 (1.37)	85.5±0.5
<b>Método 11c</b>	1.7 (1.05)	87.7±0.6	1.2 (0.55)	69.9±0.9	2.33 (1.45)	66.5±1.2	1.63 (1.30)	85.6±0.5
<b>Método 11d</b>	1.73 (1.0)	87.7±0.6	1.23 (0.68)	69.8±0.9	2.43 (1.48)	65.5±1.1	1.6 (1.16)	85.6±0.5
<b>Método 11e</b>	1.87 (1.1)	87.6±0.6	1.13 (0.43)	69.6±0.9	2.27 (1.41)	66.3±1.1	1.73 (1.41)	85.5±0.5
<b>Método 11f</b>	1.87 (1.1)	87.6±0.6	1.23 (0.68)	69.8±0.9	2.37 (1.50)	66.3±1.2	1.73 (1.41)	85.5±0.5
<b>Método 11g</b>	1.87 (1.1)	87.6±0.6	1.23 (0.68)	69.8±0.9	2.4 (1.50)	66.7±1.2	1.7 (1.37)	85.6±0.5
<b>Método 11h</b>	1.83 (1.1)	87.6±0.6	1.23 (0.68)	69.8±0.9	2.37 (1.47)	66.2±1.1	1.67 (1.32)	85.5±0.5
<b>Método 12</b>	1.6 (0.77)	88.2±0.7	1.23 (0.67)	69.8±0.9	2 (1.02)	67.2±1.2	1.67 (1.32)	85.5±0.5
<b>Método 13</b>	1.3 (0.59)	87.9±0.6	1.17 (0.46)	69.8±0.9	1.57 (0.86)	67.0±1.1	1.6 (1.22)	85.5±0.5

4.7. APÉNDICE DEL CAPÍTULO

**Tabla 4. Fase de podado para redes construidas con método de HIROSE. (Continuación).**

	DISPLAY1		GLASS		HEART		IMAGEN	
	Nhidden	POR	Nhidden	POR	Nhidden	POR	Nhidden	POR
Método 1	7.53 (1.59)	72.4±0.3	5.13 (1.70)	92.6±1.1	1.1 (0.55)	82.0±0.8	8.23 (3.53)	94.4±0.3
Método 2a	8.23 (1.73)	72.3±0.3	7.23 (3.61)	92.1±0.9	1.1 (0.55)	82.0±0.8	9.53 (3.55)	94.2±0.3
Método 2b	7.8 (1.74)	72.2±0.3	7.07 (3.63)	92.5±0.9	1.1 (0.55)	82.0±0.8	8.83 (3.67)	94.3±0.3
Método 2c	7.33 (1.56)	72.4±0.3	7 (3.66)	92.1±1.0	1.1 (0.55)	82.0±0.8	8.57 (3.80)	94.3±0.3
Método 2d	7.53 (1.63)	72.3±0.3	7.03 (3.64)	92±1.0	1.1 (0.55)	82.0±0.8	8.43 (3.82)	94.4±0.3
Método 2e	7.53 (1.63)	72.3±0.3	7.03 (3.64)	92±1.0	1.1 (0.55)	82.0±0.8	8.43 (3.82)	94.4±0.3
Método 3	7.5 (1.50)	72.3±0.3	5.2 (1.61)	92.9±1.0	1.1 (0.55)	82.0±0.8	8.37 (3.43)	94.3±0.3
Método 4	7.47 (1.55)	72.3±0.3	5.07 (1.60)	92.3±1.0	1.1 (0.55)	82.0±0.8	8.2 (3.46)	94.3±0.3
Método 5	7.53 (1.53)	72.3±0.3	5.07 (1.60)	92.6±1.1	1.1 (0.55)	82.0±0.8	8.27 (3.49)	94.3±0.3
Método 6	8.37 (1.73)	72.3±0.3	7.4 (3.53)	92.7±0.9	1.1 (0.55)	82.0±0.8	9.5 (3.54)	94.2±0.3
Método 7	7.57 (1.55)	72.3±0.3	5.2 (1.58)	92.5±1.1	1.1 (0.55)	82.0±0.8	8.23 (3.51)	94.3±0.3
Método 8a	8.13 (1.70)	72.3±0.3	7.27 (3.59)	92.2±0.9	1.1 (0.55)	82.0±0.8	9.37 (3.62)	94.2±0.3
Método 8b	7.47 (1.63)	72.3±0.3	6.8 (3.74)	92.1±1.0	1.1 (0.55)	82.0±0.8	8.83 (3.85)	94.2±0.3
Método 8c	7.4 (1.57)	72.3±0.3	7 (3.66)	92.3±0.9	1.1 (0.55)	82.0±0.8	8.2 (3.81)	94.5±0.3
Método 8d	7.47 (1.57)	72.4±0.3	7.03 (3.64)	92±1.0	1.1 (0.55)	82.0±0.8	8.37 (3.76)	94.5±0.3
Método 8e	7.57 (1.61)	72.4±0.3	7.03 (3.64)	92±1.0	1.1 (0.55)	82.0±0.8	8.37 (3.77)	94.5±0.3
Método 9	7.17 (1.49)	72.3±0.3	5.97 (3.41)	92.9±1.0	1.07 (0.37)	82.0±0.8	8 (3.24)	94.7±0.3
Método 10	7.5 (1.50)	72.3±0.3	5.1 (1.67)	92.6±1.1	1.1 (0.55)	82.0±0.8	8.3 (3.47)	94.3±0.3
Método 11a	8.37 (1.73)	72.3±0.3	7.23 (3.60)	92.5±0.9	1.1 (0.55)	82.0±0.8	9.47 (3.53)	94.2±0.3
Método 11b	8.37 (1.73)	72.3±0.3	7.1 (3.49)	92.3±0.9	1.1 (0.55)	82.0±0.8	9.53 (3.55)	94.2±0.3
Método 11c	8.37 (1.73)	72.3±0.3	7.1 (3.45)	92.5±0.9	1.07 (0.37)	82.1±0.8	9.53 (3.55)	94.2±0.3
Método 11d	8.37 (1.73)	72.3±0.3	7.2 (3.42)	92.5±0.9	1.07 (0.37)	82.1±0.8	9.53 (3.55)	94.2±0.3
Método 11e	8.37 (1.73)	72.3±0.3	7.43 (3.5)	92.3±1.0	1.1 (0.55)	82.0±0.8	9.53 (3.55)	94.2±0.3
Método 11f	8.37 (1.73)	72.3±0.3	7.47 (3.47)	92.5±0.9	1.1 (0.55)	82.0±0.8	9.53 (3.55)	94.2±0.3
Método 11g	8.37 (1.73)	72.3±0.3	7.37 (3.40)	92.3±1.0	1.1 (0.55)	82.0±0.8	9.53 (3.55)	94.2±0.3
Método 11h	8.37 (1.73)	72.3±0.3	7.47 (3.47)	92.5±0.9	1.1 (0.55)	82.0±0.8	9.53 (3.55)	94.2±0.3
Método 12	7.73 (1.74)	72.3±0.3	7.43 (3.53)	92.6±0.9	1.1 (0.55)	82.0±0.8	9.43 (3.53)	94.3±0.4
Método 13	7.6 (1.63)	72.3±0.3	6.57 (3.63)	92.3±0.9	1.1 (0.55)	82.0±0.8	8.4 (3.38)	94.3±0.3



**Tabla 4. Fase de podado para redes construidas con método de HIROSE.  
(Continuación).**

	MOK1		MOK2		VOTING	
	Nhidden	POR	Nhidden	POR	Nhidden	POR
<b>Método 1</b>	3.47 (0.97)	95.3±0.9	2.63 (1.22)	74.7±0.9	1 (0)	95.3±0.4
<b>Método 2a</b>	4.27 (1.22)	95.7±0.9	2.93 (1.34)	74.5±0.8	1.03 (0.18)	95.2±0.4
<b>Método 2b</b>	4.13 (1.31)	94.8±1.1	2.87 (1.22)	75.4±0.8	1.03 (0.18)	95.3±0.4
<b>Método 2c</b>	4.27 (1.28)	95.3±0.9	2.77 (1.14)	75.0±0.8	1 (0)	95.3±0.4
<b>Método 2d</b>	4.23 (1.25)	95.3±0.9	2.83 (1.23)	75.3±0.8	1 (0)	95.3±0.4
<b>Método 2e</b>	4.23 (1.25)	95.3±0.9	2.83 (1.23)	75.3±0.8	1 (0)	95.3±0.4
<b>Método 3</b>	3.43 (0.89)	95.3±0.9	2.67 (1.18)	75.1±0.9	1 (0)	95.3±0.4
<b>Método 4</b>	3.3 (0.65)	95.3±0.9	2.67 (1.21)	74.7±0.9	1 (0)	95.3±0.4
<b>Método 5</b>	3.5 (1.04)	95.3±0.9	2.67 (1.21)	74.6±0.9	1 (0)	95.3±0.4
<b>Método 6</b>	4.23 (1.28)	95.3±0.9	2.93 (1.17)	76.1±0.8	1.07 (0.25)	95.3±0.4
<b>Método 7</b>	3.5 (1.04)	95.3±0.9	2.67 (1.21)	74.6±0.9	1 (0)	95.3±0.4
<b>Método 8a</b>	4.33 (1.27)	95.3±0.9	2.83 (1.21)	75.2±0.8	1 (0)	95.3±0.4
<b>Método 8b</b>	4.13 (1.31)	95.3±0.9	2.8 (1.16)	75.2±0.8	1 (0)	95.3±0.4
<b>Método 8c</b>	4.2 (1.24)	95.3±0.9	2.73 (1.11)	75.3±0.8	1 (0)	95.3±0.4
<b>Método 8d</b>	4.23 (1.25)	95.3±0.9	2.83 (1.23)	75.3±0.8	1 (0)	95.3±0.4
<b>Método 8e</b>	4.23 (1.25)	95.3±0.9	2.83 (1.23)	75.3±0.8	1 (0)	95.3±0.4
<b>Método 9</b>	3.3 (0.70)	95.3±0.9	2.57 (0.97)	74.9±0.8	1 (0)	95.3±0.4
<b>Método 10</b>	3.46 (0.97)	95.3±0.9	2.63 (1.22)	74.7±0.9	1 (0)	95.3±0.4
<b>Método 11a</b>	4.23 (1.22)	95.3±0.9	2.93 (1.28)	75.3±0.8	1 (0)	95.3±0.4
<b>Método 11b</b>	4.2 (1.16)	95.3±0.9	3.07 (1.39)	74.6±0.9	1 (0)	95.3±0.4
<b>Método 11c</b>	4.3 (1.24)	95.3±0.9	3.17 (1.49)	74.6±0.9	1.03 (0.18)	95.3±0.4
<b>Método 11d</b>	4.33 (1.27)	95.3±0.9	3.17 (1.49)	74.6±0.9	1.03 (0.18)	95.3±0.4
<b>Método 11e</b>	4.23 (1.17)	95.3±0.9	3.1 (1.34)	75.2±0.8	1.07 (0.25)	95.3±0.4
<b>Método 11f</b>	4.2 (1.19)	95.3±0.9	3.1 (1.42)	74.8±0.9	1.07 (0.25)	95.3±0.4
<b>Método 11g</b>	4.3 (1.24)	95.3±0.9	3.17 (1.49)	74.6±0.9	1.07 (0.25)	95.3±0.4
<b>Método 11h</b>	4.33 (1.27)	95.3±0.9	3.17 (1.49)	74.6±0.9	1.03 (0.18)	95.3±0.4
<b>Método 12</b>	3.7 (0.99)	95.3±0.9	2.77 (1.30)	74.7±0.8	1.03 (0.18)	95.3±0.4
<b>Método 13</b>	3.53 (0.78)	95.3±0.9	2.73 (1.23)	75.1±0.8	1 (0)	95.3±0.4

4.7. APÉNDICE DEL CAPÍTULO

**Tabla 5. Fase de podado para redes construidas con método de KHORASANI.**

	BALANCE		BUPA		CREDITO		DISPLAY1	
	Nhidden	POR	Nhidden	POR	Nhidden	POR	Nhidden	POR
<b>Método 1</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.1±0.6
<b>Método 2a</b>	2.5 (1.78)	87.6±0.6	1.53 (0.97)	65.4±1.2	1.23 (0.68)	85.4±0.6	3.5 (0.78)	33.9±0.5
<b>Método 2b</b>	2.37 (1.67)	87.6±0.6	1.43 (0.82)	65.4±1.2	1.37 (0.89)	85.3±0.5	3.5 (0.78)	33.9±0.5
<b>Método 2c</b>	2.43 (1.72)	87.6±0.6	1.47 (0.90)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.2 (0.85)	34.0±0.6
<b>Método 2d</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.17 (0.75)	34.1±0.6
<b>Método 2e</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.17 (0.75)	34.1±0.6
<b>Método 3</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.1±0.6
<b>Método 4</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.1±0.6
<b>Método 5</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.1±0.6
<b>Método 6</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.1±0.6
<b>Método 7</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.1±0.6
<b>Método 8a</b>	2.43 (1.72)	87.6±0.6	1.53 (1.01)	66.0±1.1	1.33 (0.80)	85.3±0.5	3.5 (0.78)	33.9±0.5
<b>Método 8b</b>	2.43 (1.72)	87.6±0.6	1.4 (0.77)	65.9±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.5±0.7
<b>Método 8c</b>	2.43 (1.72)	87.6±0.6	1.4 (0.77)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.17 (0.75)	34.0±0.6
<b>Método 8d</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.1±0.6
<b>Método 8e</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	85.8±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.1±0.6
<b>Método 9</b>	1.87 (1.07)	88.2±0.6	1.4 (0.86)	65.8±1.2	1.3 (0.70)	85.3±0.5	2.67 (0.47)	59±2
<b>Método 10</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.1±0.6
<b>Método 11a</b>	2.57 (1.81)	87.6±0.6	1.5 (0.90)	65.5±1.2	1.3 (0.70)	85.3±0.5	3.5 (0.78)	33.9±0.5
<b>Método 11b</b>	2.57 (1.81)	87.6±0.6	1.63 (1.07)	65.5±1.2	1.33 (0.80)	85.3±0.5	3.5 (0.78)	33.9±0.5
<b>Método 11c</b>	2.53 (1.79)	87.7±0.6	1.63 (1.07)	65.1±1.2	1.3 (0.79)	85.5±0.5	3.5 (0.78)	33.9±0.5
<b>Método 11d</b>	2.5 (1.74)	87.6±0.6	1.7 (1.09)	65.5±1.2	1.37 (0.89)	85.3±0.5	3.5 (0.78)	33.9±0.5
<b>Método 11e</b>	2.63 (1.81)	87.6±0.6	1.67 (1.09)	65.5±1.2	1.4 (0.97)	85.2±0.5	3.5 (0.78)	33.9±0.5
<b>Método 11f</b>	2.63 (1.81)	87.6±0.6	1.7 (1.09)	65.5±1.2	1.37 (0.89)	85.3±0.5	3.5 (0.78)	33.9±0.5
<b>Método 11g</b>	2.63 (1.81)	87.6±0.6	1.7 (1.09)	65.5±1.2	1.37 (0.89)	85.3±0.5	3.5 (0.78)	33.9±0.5
<b>Método 11h</b>	2.5 (1.74)	87.6±0.6	1.63 (1.07)	65.5±1.2	1.37 (0.89)	85.3±0.5	3.5 (0.78)	33.9±0.5
<b>Método 12</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.1±0.6
<b>Método 13</b>	2.43 (1.72)	87.6±0.6	1.43 (0.82)	65.8±1.1	1.33 (0.84)	85.3±0.5	3.13 (0.73)	34.1±0.6

**Tabla 5. Fase de podado para redes construidas con método de KHORASANI.  
(Continuación).**

	GLASS		HEART		IMAGEN	
	Nhidden	POR	Nhidden	POR	Nhidden	POR
<b>Método 1</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.13 (0.78)	54.2±0.7
<b>Método 2a</b>	4.6 (1.52)	74.5±1.4	1.03 (0.18)	82.1±0.8	5.47 (0.63)	54.8±0.7
<b>Método 2b</b>	4.33 (1.42)	74.6±1.4	1.03 (0.18)	82.1±0.8	5.33 (0.71)	54.8±0.7
<b>Método 2c</b>	4.07 (1.60)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.23 (0.73)	54.4±0.7
<b>Método 2d</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.2 (0.76)	54.2±0.7
<b>Método 2e</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.2 (0.76)	54.2±0.7
<b>Método 3</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.13 (0.78)	54.2±0.7
<b>Método 4</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.13 (0.78)	54.2±0.7
<b>Método 5</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.13 (0.78)	54.2±0.7
<b>Método 6</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.13 (0.78)	54.2±0.7
<b>Método 7</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.13 (0.78)	54.2±0.7
<b>Método 8a</b>	4.6 (1.63)	74.7±1.4	1.03 (0.18)	82.1±0.8	5.07 (0.69)	53.6±0.5
<b>Método 8b</b>	4.23 (1.57)	75.1±1.4	1.03 (0.18)	82.1±0.8	5.07 (0.69)	53.6±0.5
<b>Método 8c</b>	4 (1.60)	75.1±1.4	1.03 (0.18)	82.1±0.8	5.17 (0.83)	54.2±0.7
<b>Método 8d</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.13 (0.78)	54.2±0.7
<b>Método 8e</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.13 (0.78)	54.2±0.7
<b>Método 9</b>	3.2 (1.30)	84.1±1.6	1 (0)	82.1±0.8	4.5 (0.63)	87.2±1.4
<b>Método 10</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.13 (0.78)	54.2±0.7
<b>Método 11a</b>	4.5 (1.57)	74.4±1.42	1.03 (0.18)	82.1±0.8	5.5 (0.63)	54.8±0.7
<b>Método 11b</b>	4.67 (1.58)	74.4±1.5	1.03 (0.18)	82.1±0.8	5.5 (0.63)	54.8±0.7
<b>Método 11c</b>	4.77 (1.59)	74.5±1.4	1.03 (0.18)	82.1±0.8	5.5 (0.63)	54.8±0.7
<b>Método 11d</b>	4.77 (1.59)	74.5±1.4	1.03 (0.18)	82.1±0.8	5.5 (0.63)	54.8±0.7
<b>Método 11e</b>	4.7 (1.51)	74.5±1.4	1.03 (0.18)	82.1±0.8	5.5 (0.63)	54.8±0.7
<b>Método 11f</b>	4.63 (1.61)	74.8±1.5	1.03 (0.18)	82.1±0.8	5.5 (0.63)	54.8±0.7
<b>Método 11g</b>	4.77 (1.59)	74.5±1.4	1.03 (0.18)	82.1±0.8	5.5 (0.63)	54.8±0.7
<b>Método 11h</b>	4.77 (1.59)	74.5±1.4	1.03 (0.18)	82.1±0.8	5.5 (0.63)	54.8±0.7
<b>Método 12</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.13 (0.78)	54.2±0.7
<b>Método 13</b>	4.03 (1.61)	75.0±1.4	1.03 (0.18)	82.1±0.8	5.13 (0.78)	54.2±0.7

4.7. APÉNDICE DEL CAPÍTULO

**Tabla 5. Fase de podado para redes construidas con método de KHORASANI. (Continuación).**

	MOK1		MOK2		VOTING	
	Nhidden	POR	Nhidden	POR	Nhidden	POR
Método 1	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 2a	2.97 (1.56)	76.3±0.7	2.23 (1.30)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 2b	1.97 (1.47)	76.3±0.7	2.07 (1.26)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 2c	1.73 (1.28)	76.3±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 2d	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 2e	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 3	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 4	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 5	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 6	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 7	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 8a	2.2 (1.40)	76.4±0.7	2.17 (1.26)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 8b	2.07 (1.28)	76.3±0.7	2.07 (1.20)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 8c	1.47 (0.68)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 8d	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 8e	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 9	1.73 (1.11)	76.4±0.7	1.6 (0.67)	66.2±0.5	1.13 (0.57)	95.3±0.4
Método 10	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 11a	2.9 (1.42)	76.0±0.7	1.97 (1.16)	65.7±0.7	1.23 (0.77)	95.3±0.4
Método 11b	2.97 (1.50)	76.3±0.7	1.87 (1.14)	66.2±0.5	1.2 (0.76)	95.3±0.4
Método 11c	3.07 (1.53)	75.9±0.8	1.97 (1.27)	66.2±0.5	1.2 (0.76)	95.3±0.4
Método 11d	3.23 (1.50)	76.3±0.7	1.93 (1.23)	66.2±0.5	1.2 (0.76)	95.3±0.4
Método 11e	3.1 (1.49)	76.3±0.7	2.2 (1.30)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 11f	3.07 (1.53)	76.3±0.7	2 (1.26)	65.6±0.8	1.23 (0.77)	95.3±0.4
Método 11g	3.17 (1.49)	75.9±0.8	1.93 (1.23)	65.7±0.8	1.23 (0.77)	95.3±0.4
Método 11h	3.23 (1.50)	76.3±0.7	2.03 (1.33)	65.9±0.6	1.2 (0.76)	95.3±0.4
Método 12	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4
Método 13	1.57 (0.82)	76.4±0.7	1.97 (1.19)	66.2±0.5	1.23 (0.77)	95.3±0.4

# Capítulo 5

## ESTUDIO DE LA INFLUENCIA EN LA CAPACIDAD DE GENERALIZACIÓN DE LA RED DEL NÚMERO DE CAPAS OCULTAS

- 5.1. *Descripción del Problema*
- 5.2. *Arquitectura Multilayer Feedforward con dos capas ocultas*
- 5.3. *Metodología*
- 5.4. *Desarrollo Experimental*
- 5.5. *Conclusiones*
- 5.6. *Bibliografía del Capítulo*

### 5.1. Descripción del Problema

En este capítulo presentaremos una comparación experimental de la capacidad de generalización de las redes neuronales Multilayer Feedforward con una y dos capas ocultas.

Veamos en primer lugar cuál es el interés de esta investigación.

Algunos autores se plantearon si habría alguna ventaja en el uso de más de una capa oculta en la arquitectura de la red. Una de las preguntas más importantes planteadas es si esto afecta a la capacidad de generalización de la red.

En la bibliografía [1] hay un teorema sobre redes neuronales que garantiza que una red perceptrón con una capa oculta, puede resolver cualquier problema, si tiene un número adecuado de neuronas en dicha capa. Por ello, no es en modo alguno necesario utilizar una segunda capa en la estructura de la red.

## 5.1. DESCRIPCIÓN DEL PROBLEMA

Sin embargo, sigue siendo un problema discutido por todos, qué arquitectura se debe usar. Por ejemplo, no se sabe si la arquitectura de dos capas ocultas tendrá mejor capacidad de generalización que la de una capa oculta o si podemos encontrar un modelo simple, es decir, con menor número de pesos teniendo la misma eficiencia mediante el uso de una red de dos capas ocultas.

Por ello, hemos decidido realizar una revisión bibliográfica sobre este tema y ver qué conclusión podemos sacar a partir de las diferentes contribuciones.

En [2], Babri y Tong presentan una investigación sobre cómo afecta a la sensibilidad de salida de la red una perturbación en los pesos para redes Multilayer Feedforward con una sola capa oculta y con dos.

La conclusión de esa investigación, era que durante el estado inicial de aprendizaje, la sensibilidad de la red con dos capas ocultas era mayor que para la de una sola capa y se puede llegar a un aprendizaje inestable.

Para resolver este problema se proponen dos métodos: una apropiada inicialización de los pesos y la modificación de la función de transferencia.

En [3], se muestra que en una red Feedforward de tres capas (entrada, salida y oculta) con  $N-1$  unidades ocultas, se pueden representar exactamente  $N$  relaciones entrada-objetivo y con una red de 4 capas, podemos hacer lo mismo con un pequeño error usando solo  $(N/2)+3$  unidades ocultas. Con estos resultados se podría pensar que es mejor una red de cuatro capas, ya que con menos neuronas y pesos se puede conseguir prácticamente el mismo resultado.

En [4], se realiza una comparación entre perceptrones de una y dos capas en el contexto de aproximación funcional de funciones no lineales, en concreto la correspondencia entre variables de control de automóviles y parámetros de representación. La comparación es llevada a cabo para redes con el mismo número de unidades en la capa oculta. En el artículo se concluye que una red con 4 capas (2 capas ocultas) con más neuronas en la primera capa oculta que en la segunda, tiene mejor eficiencia que la de tres capas (solo una oculta).

En [5] y [6], hemos encontrado dos comparaciones experimentales de la capacidad de generalización de redes neuronales Multilayer Feedforward con una o dos capas ocultas. La comparación se hizo con problemas de clasificación. En [6], los problemas usados para el test tenían datos artificiales de distribuciones de gaussianas. En [6] se concluye que la eficiencia del perceptrón de una y dos capas ocultas es similar. Sin embargo, en las redes con 4 capas (2 capas ocultas) es más probable alcanzar un mínimo local. En la referencia [5], se utilizaron dos problemas para realizar el test, la clasificación entre 12 números chinos y la

clasificación entre 10 números arábigos. En este caso la conclusión fue que la eficiencia de un perceptrón con una capa oculta es mejor que la del perceptron con dos capas ocultas.

Como vemos, los resultados que se obtienen en las diferentes investigaciones son contradictorios. En [6], la eficiencia es similar, en [4] la eficiencia de una red con dos capas ocultas es mejor que la eficiencia de una red de una capa oculta, y finalmente en [5] se obtiene lo contrario (la de una capa mejor que la de dos).

Sin embargo, nosotros pensamos que los resultados experimentales presentados en [4] y [5] son exiguos, ya que es muy difícil obtener buenas conclusiones usando sólo simulaciones en uno o dos problemas.

Finalmente, en todas las referencias las redes de tres y cuatro capas utilizadas tienen la misma complejidad, es decir, el mismo número de pesos o neuronas. Nosotros creemos que ésta no es una buena comparación ya que la tarea del diseñador de la red neuronal es encontrar la mínima red que resuelva bien el problema.

Una apropiada comparación debería comparar la mínima red de una capa oculta, con la mínima red de dos capas ocultas.

Uno de los primeros objetivos de esta investigación será encontrar la “mínima red” de cada caso, con una capa oculta y con dos. Entendemos como red mínima aquella que suministra mayor eficiencia (capacidad de generalización) con una estructura mínima.

Basándonos en esta idea, queremos en esta investigación determinar la capacidad de generalización de una red “mínima” Multilayer Feedforward de tres capas (una capa oculta), en comparación con una red, también mínima, de cuatro capas (dos capas ocultas), y concluir si la introducción de una capa adicional puede reportar alguna ventaja a la hora de resolver un problema.

## 5.2. Arquitectura Multilayer Feedforward con dos capas ocultas

En el primer capítulo, se describió la arquitectura de red neuronal Multilayer Feedforward con una sola capa oculta. En este capítulo necesitamos generalizar las ecuaciones para el caso de utilizar dos capas ocultas en la red. Lo realizaremos de forma breve a lo largo de este apartado.

Llamemos a los vectores de salida de las diferentes capas como sigue.

$$\text{Capa de entrada} \quad \{i_1, i_2, \dots, i_N\} = \{io_1, io_2, \dots, io_N\} = \{a_1, a_2, \dots, a_N\}$$

$$1^{\text{a}} \text{ Capa oculta} \quad \{h1o_1, h1o_2, \dots, h1o_{Nh}\} = \{f(h1i_1), f(h1i_2), \dots, f(h1i_{Nh})\}$$

$$2^{\text{a}} \text{ Capa oculta} \quad \{h2o_1, h2o_2, \dots, h2o_{Nh}\} = \{f(h2i_1), f(h2i_2), \dots, f(h2i_{Nh})\}$$

$$\text{Capa de salida} \quad \{oo_1, oo_2, \dots, oo_{Ns}\} = \{f(oi_1), f(oi_2), \dots, f(oi_{Ns})\}$$

La entrada a una neurona  $j$  de la primera capa oculta vendrá dada por:

$$h1i_j = \sum_{i=1}^N wi_{ij} \cdot a_i + \theta_j \quad (5.1)$$

La entrada a una neurona  $m$  de la segunda capa oculta será:

$$h2i_m = \sum_{j=1}^{Nh1} wh1_{jm} \cdot h1o_j + \xi 1_m \quad (5.2)$$

La entrada a una neurona  $k$  de la capa de salida se determinará por:

$$oi_k = \sum_{m=1}^{Nh2} wh2_{mk} \cdot h2o_m + \xi 2_k \quad (5.3)$$

Las salidas de las diferentes capas se pueden escribir utilizando la función de transferencia sigmoide.

Para la primera capa oculta:

$$h1o_j = f(h1i_j) = \frac{1}{1 + \exp(-h1i_j)} \quad (5.4)$$

Para la segunda capa oculta:

$$h2o_m = f(h2i_m) = \frac{1}{1 + \exp(-h2i_m)} \quad (5.5)$$



Y para la salida:

$$oo_k = \frac{1}{1 + \exp(-oi_k)} \quad (5.6)$$

Las ecuaciones de modificación de los pesos serán:

$$\Delta wh2_{m,k}(t+1) = \eta \cdot \left( -\frac{\partial E_P}{\partial wh2_{m,k}} \right) + \beta \cdot \Delta wh2_{m,k}(t) \quad (5.7)$$

$$\Delta wh1_{j,m}(t+1) = \eta \cdot \left( -\frac{\partial E_P}{\partial wh1_{j,m}} \right) + \beta \cdot \Delta wh1_{j,m}(t)$$

$$\Delta wi_{i,j}(t+1) = \eta \cdot \left( -\frac{\partial E_P}{\partial wi_{i,j}} \right) + \beta \cdot \Delta wi_{i,j}(t)$$

Calculamos ahora las diferentes derivadas.

$$\frac{\partial E_P}{\partial wh2_{m,k}} = \frac{\partial E_P}{\partial oo_{P,k}} \cdot \frac{\partial oo_{P,k}}{\partial oi_{P,k}} \cdot \frac{\partial oi_{P,k}}{\partial wh2_{m,k}} \quad (5.8)$$

El primer y segundo término, igual que en el capítulo 1, se pueden calcular mediante las ecuaciones:

$$\frac{\partial E_P}{\partial oo_{P,k}} = -(t_{P,k} - oo_{P,k}) \quad (5.9)$$

$$\frac{\partial oo_{P,k}}{\partial oi_{P,k}} = oo_{P,k} \cdot (1 - oo_{P,k})$$

El último término será:

$$\frac{\partial oi_{P,k}}{\partial wh2_{m,k}} = h2o_{P,m} \quad (5.10)$$

Sustituyendo:

$$\delta_{P,k} = -(t_{P,k} - oo_{P,k}) \cdot oo_{P,k} \cdot (1 - oo_{P,k}) \quad (5.11)$$

$$\frac{\partial E_P}{\partial wh2_{m,k}} = \delta_{P,k} \cdot h2o_{P,m}$$

Por otra parte, la derivada para los pesos de la otra capa oculta será:

$$\frac{\partial E_P}{\partial wh1_{j,m}} = \sum_k \frac{\partial E_P}{\partial oo_{P,k}} \cdot \frac{\partial oo_{P,k}}{\partial oi_{P,k}} \cdot \frac{\partial oi_{P,k}}{\partial wh1_{j,m}} \quad (5.12)$$

## 5.2. ARQUITECTURA MULTILAYER FEEDFORWARD CON DOS CAPAS OCULTAS

El primer y segundo término, se obtienen de (5.9). El tercer término será:

$$\frac{\partial o_{P,k}}{\partial wh_{j,m}} = \frac{\partial o_{P,k}}{\partial h_{2o_{P,m}}} \cdot \frac{\partial h_{2o_{P,m}}}{\partial h_{2i_{P,m}}} \cdot \frac{\partial h_{2i_{P,m}}}{\partial wh_{j,m}} = wh_{2_{m,k}} \cdot h_{2o_{P,m}} \cdot (1 - h_{2o_{P,m}}) \cdot h_{1o_{P,j}} \quad (5.13)$$

Sustituyendo quedará finalmente:

$$\begin{aligned} \frac{\partial E_P}{\partial wh_{j,m}} &= \sum_k \{ \delta_{P,k} \cdot wh_{2_{m,k}} \cdot h_{2o_{P,m}} \cdot (1 - h_{2o_{P,m}}) \cdot h_{1o_{P,j}} \} \\ \delta h_{1_{P,m}} &= \left( \sum_k \delta_{P,k} \cdot wh_{2_{m,k}} \right) \cdot h_{2o_{P,m}} \cdot (1 - h_{2o_{P,m}}) \\ \frac{\partial E_P}{\partial wh_{j,m}} &= \delta h_{1_{P,m}} \cdot h_{1o_{P,j}} \end{aligned} \quad (5.14)$$

Para la última derivada respecto de los pesos de entrada tendremos:

$$\frac{\partial E_P}{\partial wi_{i,j}} = \sum_k \frac{\partial E_P}{\partial oo_{P,k}} \cdot \frac{\partial oo_{P,k}}{\partial oi_{P,k}} \cdot \frac{\partial oi_{P,k}}{\partial wi_{i,j}} \quad (5.15)$$

La última derivada se calculará mediante la siguiente ecuación:

$$\frac{\partial oi_{P,k}}{\partial wi_{i,j}} = \frac{\partial oi_{P,k}}{\partial h_{2o_{P,m}}} \cdot \frac{\partial h_{2o_{P,m}}}{\partial h_{2i_{P,m}}} \cdot \frac{\partial h_{2i_{P,m}}}{\partial h_{1o_{P,j}}} \cdot \frac{\partial h_{1o_{P,j}}}{\partial h_{1i_{P,j}}} \cdot \frac{\partial h_{1i_{P,j}}}{\partial wi_{i,j}} \quad (5.16)$$

Sustituyendo:

$$\frac{\partial oi_{P,k}}{\partial wi_{i,j}} = wh_{2_{m,k}} \cdot h_{2o_{P,m}} \cdot (1 - h_{2o_{P,m}}) \cdot wh_{1_{j,m}} \cdot h_{1o_{P,j}} \cdot (1 - h_{1o_{P,j}}) \cdot a_{P,i} \quad (5.17)$$

Y queda finalmente:

$$\begin{aligned} \delta h_{2_{P,j}} &= \delta h_{1_{P,m}} \cdot wh_{1_{j,m}} \cdot h_{1o_{P,j}} \cdot (1 - h_{1o_{P,j}}) \\ \frac{\partial E_P}{\partial wi_{i,j}} &= \delta h_{2_{P,j}} \cdot a_{P,i} \end{aligned} \quad (5.18)$$

Las ecuaciones finales de las derivadas serán:

$$\begin{aligned} \frac{\partial E_P}{\partial wh_{2_{m,k}}} &= \delta_{P,k} \cdot h_{2o_{P,k}} \\ \frac{\partial E_P}{\partial wh_{1_{j,m}}} &= \delta h_{1_{P,m}} \cdot h_{1o_{P,j}} \\ \frac{\partial E_P}{\partial wi_{i,j}} &= \delta h_{2_{P,j}} \cdot a_{P,i} \end{aligned} \quad (5.19)$$

## 5.2. Metodología

Se han seleccionado 16 problemas para realizar la comparación. Los problemas se encuentran en el “UCI repository of machine learning databases”, y ya fueron descritos en el primer capítulo.

Los problemas utilizados son: *Abalone (AB)*, *Balance Scale Weight & Distance (BL)*, *Cylinder Bands (BN)*, *BUPA Liver Disorders (BU)*, *Credit Approval (CR)*, *Display1 (D1)*, *Display2 (D2)*, *Glass Identification (GL)*, *Heart Disease (HE)*, *Image Segmentation (IM)*, *The Monk’s Problems (M1, M2)*, *Mushroom (MM)*, *Musk (MK)*, *Pima Indians Diabetes (PI)*, *Voting Records (VO)*.

Para cada problema hemos obtenido el mínimo perceptrón de una capa oculta mediante un proceso de *prueba y error*, y usado cros-validación para hallar tanto el número de unidades óptimo como el punto de convergencia.

También hemos obtenido la red mínima para el caso de dos capas ocultas y hemos utilizado el mismo proceso que antes, utilizando una búsqueda exhaustiva entre las estructuras con  $N$  nodos ocultos en la primera capa, y dos nodos ocultos en la segunda capa y viceversa, es decir dos nodos ocultos en la primera capa oculta y  $N$  nodos en la segunda, donde  $N$  es el número de nodos ocultos mínimo para la red de una sola capa oculta.

En algunos casos, hemos encontrado diferentes candidatos con similar eficiencia (calculada con cros-validación) para la estructura de la red mínima, en estos casos hemos tomado varias redes candidatas.

Para el caso de dos capas ocultas en la red, hemos utilizado la sugerencia de la referencia [2] para acelerar y estabilizar la convergencia, es decir, hemos modificado la función de transferencia para las neuronas de salida usando la ecuación (5.20) con  $\xi=0.02$ .

$$f(x) = \frac{1 - 2 \cdot \xi}{1 + e^{-x}} + \xi \quad (5.20)$$

Los pesos iniciales están uniformemente distribuidos en un intervalo de anchura 0.2 y centrado en -0.5 y +0.5, tal y como se recomienda en [2].

Puesto que la modificación de la función de transferencia es muy pequeña, no hemos utilizado la ecuación (5.20) en el cálculo de la variación del error, solamente hemos aplicado la ecuación al calcular la nueva salida de la red neuronal.

## 5.2. METODOLOGÍA

El algoritmo de entrenamiento utilizado para los dos tipos de redes fue el de Backpropagation. En el primer capítulo, se encuentran las ecuaciones de Multilayer Feedforward con Backpropagation para el caso de una sola capa oculta, y en el apartado 5.2 de este capítulo para la estructura de dos capas ocultas.

Una vez obtenida la estructura mínima, hemos entrenado para cada problema 10 redes, tanto para los casos de una capa oculta como para los de dos capas ocultas. Cada red tenía una inicialización y una partición diferente de datos en los conjuntos de entrenamiento, cros-validación y test. Como siempre hemos promediado estos valores obteniendo el resultado con un error (suficientemente pequeño), logrando al mismo tiempo que los resultados no dependan de las condiciones de inicialización de la red.

En el apartado de resultados experimentales se muestran los resultados de las diferentes redes seleccionadas para los 16 problemas.

### 5.3. Desarrollo Experimental

Como hemos dicho antes mediante un proceso de *prueba y error* se han ido seleccionando las diferentes redes tanto para una capa oculta como para dos.

En el caso de una sola capa oculta, se entrenaron con Backpropagation varias redes con diferente número de unidades en la capa oculta. Utilizando el método de prueba y error, se estimó el punto de convergencia de las redes mediante cross-validación y porcentaje de correcta clasificación de cross-validación. Para cada problema escogimos aquella arquitectura de red que proporcionaba un mayor porcentaje de correcta clasificación de cross-validación.

Una vez seleccionadas las arquitecturas se entrenaron 10 redes de cada, con diferente inicialización de pesos y particiones de datos entre conjunto de entrenamiento, cross-validación y test. De nuevo escogimos el punto de convergencia utilizando cross-validación y realizamos el test promediando los resultados para obtener una eficiencia media y un error. En la Tabla 2 se muestran los resultados obtenidos.

Para encontrar las arquitecturas idóneas para el caso de dos capas ocultas partimos inicialmente de una serie de estructuras variando el número de unidades de las capas ocultas.

Es decir, partimos por ejemplo de una estructura con *Ninput* neuronas en la capa de entrada, *Noutput* neuronas en la capa de salida (estas las fijan los problemas en cuestión), y *Nh1* unidades en la primera capa oculta y *Nh2* unidades en la segunda capa oculta.

Hemos entrenado todas las combinaciones de *Nh1* y *Nh2* con un valor mayor o igual a dos y menor o igual que *Nh* (unidades ocultas), que es el valor obtenido para la red de una sola capa oculta. Los valores de *Nh1* y *Nh2* se encuentran en la Tabla 1.

**Tabla 1. Estructura iniciales usadas para dos capas ocultas.**

Nh1-Nh2	Nh1-Nh2	Nh1-Nh2	Nh1-Nh2	Nh1-Nh2
2-2	3-2	4-2	.....	Nh -2
2-3	3-3	4-3	.....	Nh -3
2-4	3-4	4-4	.....	Nh -4
.....	.....	.....	.....	.....
2-Nh	3-Nh	4-Nh	.....	Nh -Nh

### 5.3. DESARROLLO EXPERIMENTAL

Utilizando también la metodología de *prueba y error*, calculamos la arquitectura adecuada para cada problema. En el caso de que encontráramos diferentes estructuras con similar eficiencia se escogían todas las candidatas.

Después entrenamos 10 redes con Backpropagation utilizando diferente inicialización de pesos y particiones de datos. Medimos el punto de convergencia de dichas redes usando cros-validación y después la eficiencia en el test. Obtuvimos la media de la eficiencia de las 10 redes con un error.

En la Tabla 2 se muestran también los resultados obtenidos para las arquitecturas de dos capas ocultas

En la Tabla 2, tenemos los resultados experimentales obtenidos. En ella hemos indicado para los diferentes problemas la estructura utilizada, el número parámetros libres de la misma (pesos y “thresholds”) y el valor promedio del porcentaje de correcta clasificación con su error, tanto para redes con una sola capa oculta, como para redes con dos capas ocultas.

**Tabla 2. Resultados Experimentales.**

Base de Datos	Una Capa Oculta			Base de Datos	Dos Capas Ocultas		
	Arquitectura	Número de Pesos	Porcentaje Clasificación		Arquitectura	Número de Pesos	Porcentaje Clasificación
AB	8-17-3	207	50 ± 5	AB	8-2-4-3	45	50 ± 3
BL	4-6-3	51	58 ± 9	BL	4-3-3-3	39	74 ± 7
BN	39-13-2	506	68.2 ± 1.3	BN	39-3-2-2	134	69 ± 1
BU	6-8-2	74	68.3 ± 1.6	BU	6-7-4-2	91	63 ± 3
				BU	6-6-7-2	107	61 ± 3
CR	15-9-2	164	86.6 ± 0.6	CR	15-5-3-2	106	81 ± 4
				CR	15-6-2-2	116	84 ± 3
D1	7-15-10	280	71.9 ± 1.3	D1	7-6-6-10	160	54 ± 8
D2	14-21-10	535	70.9 ± 0.8	D2	14-13-10-10	445	62 ± 6
				D2	14-17-7-10	461	69.6 ± 0.7
				D2	14-20-4-10	434	50 ± 9
GL	10-6-6	108	78.9 ± 1.1	GL	10-2-4-6	64	52 ± 6
HE	13-6-2	98	80.6 ± 0.9	HE	13-3-3-2	62	71 ± 4
IM	19-13-7	358	91.0 ± 0.5	IM	19-7-5-7	222	73 ± 2
M1	6-5-2	47	73.6 ± 1.1	M1	6-3-5-2	53	77 ± 5
				M1	6-5-3-2	61	78 ± 4
M2	6-14-2	128	66.3 ± 0.9	M2	6-6-4-2	80	68.1 ± 1.5
MM	22-10-2	252	97.6 ± 0.3	MM	22-2-2-2	58	83 ± 5
MK	166-7-2	1185	93.1 ± 0.3	MK	166-2-5-2	361	89 ± 1.6
				MK	166-4-2-2	684	89.0 ± 1.9
PI	8-15-2	167	95.9 ± 0.6	PI	8-14-4-2	196	73 ± 1
				PI	8-9-9-2	191	74 ± 1
VO	16-13-2	249	94.5 ± 0.2	VO	16-2-4-2	56	94 ± 1

En la Tabla 2, vemos que el número de parámetros libres de la red mínima depende del problema pero es usualmente menor en el caso de redes con dos capas ocultas. Encontramos este resultado en 13 de los 16 problemas, en concreto en: AB, BL, BN, D1, D2, CR, GL, HE, IM, M2, MM, MK y VO. La estructura mínima de dos capas ocultas suele tener una menor complejidad.

Podemos observar también que la capacidad de generalización en redes con una capa oculta es en general mejor que en las redes de dos capas ocultas. Se encuentran mejores resultados para redes con una sola capa oculta en 9 de los 16 problemas (BU, CR, D1, GL, HE, IM, MM, MK y PI), en 5 de los 16 problemas la eficiencia obtenida es similar a la de redes con dos capas ocultas, y sólo en 2 de los 16 problemas encontramos mejores resultados para la red con dos capas ocultas (BL y M2).

También encontramos que la variabilidad de eficiencia en los resultados es mayor en el caso de redes con dos capas ocultas. Por ejemplo, observando el error vemos que este suele ser mayor en las redes con dos capas oculta que en las de una.

Finalmente, en algunos problemas M1, MM y MK ha sido realmente difícil encontrar 10 redes con dos capas ocultas que convergieran, por lo que fue necesario entrenar muchas más (con diferentes condiciones iniciales) hasta tener las diez necesarias para los experimentos. Por ejemplo en MK se entrenaron 30 redes para encontrar 14 que convergieran, y en MM se entrenaron 39 redes para obtener finalmente solo 10. Con esto concluimos que Backpropagation es más ineficiente en el caso de redes con dos capas ocultas.

## 5.4. Conclusiones

Hemos presentado en este capítulo una comparación experimental de la capacidad de generalización de las redes Multilayer Feedforward con una sola capa oculta y con dos capas ocultas.

Hemos usado un total de 16 problemas reales para poder medir la capacidad generalización de ambas arquitecturas.

Realizamos una selección previa de las arquitecturas mínimas de cada tipo para cada problema usando un procedimiento de *prueba y error*.

Se entrenaron 10 redes con una sola capa oculta y con dos, con el objetivo de eliminar la dependencia con las condiciones iniciales en el entrenamiento y encontrar un error en los resultados.

De nuestros resultados, podemos concluir que la capacidad de generalización de una red con una capa oculta es en general mejor que en una red con dos capas ocultas.

Por ello, pensamos que en general, con los algoritmos de entrenamiento usados no es razonable utilizar redes con más de una capa oculta.

Además, hemos observado que para redes con más de una capa oculta la convergencia es bastante mala, hay algunos problemas en los que hemos tenido gran dificultad en encontrar redes convergidas.

Los resultados de nuestra investigación sobre este tema fueron publicados y presentados en dos conferencias internacionales. En [10] y [11] tenemos las referencias de dichas publicaciones.



## 5.5. Bibliografía del Capítulo

- [1] Cybenko, George, “Approximation by superposition of sigmoidal functions”. Mathematics of Control Signals, and Systems, vol. 2, pp. 303-314. 1989.
- [2] Babri H.A., Tong Y., “Deep Feedforward Networks: Application to Pattern Recognition”. Proceedings of the 1996 International Conference on Neural Networks. 1996, vol. 3, pp. 1422-1426.
- [3] Tamura S., Tateishi M., “Capabilities of a Four-Layered Feedforward Neural Network: Four Layers versus Three”. IEEE Trans. on Neural Networks. 1997, vol. 8, no. 2, pp. 251-255.
- [4] Xiao J., Chen Z., Cheng J., “Structure Study of Feedforward Neural Networks for Approximation of Highly Nonlinear Real-valued Functions”. Proceedings of 1996 International Conference on Neural Networks, vol. 1, pp. 258-263, 1996.
- [5] Saratchandran P., “Effect of Hidden Layers on Generalization Properties of Feedforward Neural Networks”. Neural Parallel & Scientific Computations, vol. 1, no. 2, pp. 227-240, 1993.
- [6] Villiers J., Bernard E., “Backpropagation Neural Nets with One and Two hidden Layers”. IEEE Transactions on Neural Networks, vol. 4, no. 1, pp. 136-141, 1992.
- [7] Yu X., Chen G., Cheng S., “Dynamic Learning Rate Optimization of the Backpropagation Algorithm”. IEEE Transactions on Neural Networks, vol. 6, no. 3, pp. 669-677, 1995
- [8] Yamasaki, Masami, “The Lower Bound of the Capacity for a Neural Network with Multiple Hidden Layers”. Proceedings of the World Congress on Neural Networks (WCNN’93), vol. 3, pp. 544-547, 1993.
- [9] Cosnard M., Koiran P., Paugam-Moisy H., “A Step Toward the Frontier between One-Hidden-Layer and Two-Hidden Layer Neural Networks”. Proceedings of 1993 International Conference on Neural Networks, vol. 3, pp. 2292-2295, 1993.
- [10] Fernández, M., Hernández, C., “One or Two Hidden Layers Perceptrons”. European Symposium on Artificial Neural Networks (ESANN’98), pp. 129-133, 1998.
- [11] Fernández, M., Hernández, C., “Generalization capability of one and two hidden layers”. Proceedings of the 1999 IEEE International Joint Conference on Neural Networks (IJCNN’99), pp. 1840-1845, 1999.



# Capítulo 6

## MÉTODOS DE INICIALIZACIÓN DE PESOS PARA REDES MULTILAYER FEEDFORWARD

- 6.1. *Descripción del Problema*
- 6.2. *Descripción de los diferentes métodos*
  - 6.2.1. *Método 1*
  - 6.2.2. *Método 2*
  - 6.2.3. *Método 3*
  - 6.2.4. *Método 4*
  - 6.2.5. *Método 5*
  - 6.2.6. *Método 6*
  - 6.2.7. *Método 7*
  - 6.2.8. *Método 8*
  - 6.2.9. *Método 9*
  - 6.2.10. *Método 10*
  - 6.2.11. *Método 11*
  - 6.2.12. *Método 12*
  - 6.2.13. *Método 13*
- 6.3. *Metodología*
- 6.4. *Desarrollo Experimental*
- 6.5. *Conclusiones*
- 6.6. *Bibliografía del Capítulo*
- 6.7. *Apéndice del Capítulo*

### 6.1. Descripción del Problema

Es bien conocido que el algoritmo de entrenamiento Backpropagation [1] se puede interpretar como la optimización de una función de error con respecto a un conjunto de parámetros: los pesos de la red.

## 6.1. DESCRIPCIÓN DEL PROBLEMA

La principal tarea de este algoritmo es la búsqueda de un mínimo en la función de error. Generalmente este mínimo será local ya que no está garantizado que se pueda encontrar un mínimo global.

El mínimo encontrado después de la convergencia de la red será el que determine la calidad de las soluciones que ésta proporcione.

Si el mínimo local encontrado es cercano a un mínimo global de la función de error, podremos considerar que la eficiencia final de la red será aceptable y consideraremos que el proceso de entrenamiento ha tenido éxito, aunque los resultados no serán del todo óptimos.

Sin embargo, hay otros mínimos locales que proporcionan un entrenamiento pobre del clasificador, es decir, proporcionarán una eficiencia final muy lejana del valor óptimo. Estos son los llamados mínimos falsos y en estos casos el resultado final no es aceptable por lo que consideraremos que la red no ha convergido.

En este punto cabe plantearse la pregunta de qué factores son los que determinan cuál será el mínimo final para una cierta estructura de red neuronal.

Como hemos visto ya en otros capítulos, los principales factores son dos: el punto inicial usado en la búsqueda del mínimo (la particular inicialización de los pesos en el caso de redes neuronales) y las técnicas de optimización utilizadas, es decir, el algoritmo de entrenamiento usado.

En otras palabras, dos algoritmos de entrenamiento diferentes con la misma inicialización pueden encontrar dos mínimos locales diferentes y viceversa, dos diferentes inicializaciones para el mismo algoritmo de entrenamiento pueden dar diferentes mínimos locales.

Como ya hemos visto, dos mínimos diferentes proporcionan eficiencias finales diferentes. Veamos ahora como repercute esto en el caso de redes neuronales y la inicialización de los pesos.

Primero pensemos en cuál será la consecuencia inmediata de una particular inicialización de pesos.

Desde nuestro punto de vista, la inicialización de los pesos influye sobre tres propiedades de la convergencia de la red neuronal: la velocidad de convergencia, la probabilidad de que haya éxito en la convergencia y la capacidad de generalización (eficiencia final de la red).

Respecto a la primera, la velocidad de convergencia, una particular inicialización de pesos puede estar cerca de un determinado mínimo local y puede

llevar a que el mínimo local se alcance de forma más rápida. En definitiva, el número de iteraciones para el proceso de entrenamiento y el tiempo de convergencia puede ser menor en función de la inicialización tomada para los pesos. Esta consecuencia se describe ampliamente en la bibliografía [1], [2], [3], [4] y [5]. La relación de rapidez de convergencia puede ser desde un pequeño orden de magnitud hasta ordenes considerables.

Desde el punto de vista de relación de éxito en la convergencia, como discutimos al principio, una cierta inicialización de pesos puede llevar a que el algoritmo de entrenamiento alcance un mínimo local aceptable o un falso mínimo local. En el primer caso, consideramos que la red ha convergido con éxito y en el segundo que la red no converge. Vemos pues, que la relación final de “convergencias con éxito” con respecto al número total de intentos puede depender de la inicialización tomada para los pesos.

El tercer efecto era la influencia sobre la capacidad de generalización final de la red neuronal. Supongamos dos convergencia con éxito que encuentren dos mínimos locales diferentes. Supongamos que la eficiencia que proporcionan ambos mínimos es también aceptable, pero en general será diferente, por lo cual la capacidad de generalización final de las redes entrenadas con diferentes inicialización de pesos resultará también diferente. De hecho en la bibliografía [6] se muestra claramente esta última consecuencia.

Por todo lo dicho hasta aquí, creemos que “la inicialización de los pesos “ es un aspecto muy importante a la hora de entrenar una red neuronal y que merece un estudio detallado.

En general, la forma clásica de inicialización de pesos para redes Multilayer Feedforward es utilizar una distribución uniforme y aleatoria dentro de un pequeño intervalo centrado en cero, por ejemplo  $[-0.1, 0.1]$ .

Sin embargo, como se puede ver en la bibliografía, se han propuesto otros métodos diferentes de inicialización de pesos para redes Multilayer Feedforward. En este capítulo nos proponemos hacer un estudio detallado de los mismos y estudiar la eficiencia relativa entre ellos, para concluir cuál de ellos resulta más interesante usar.

Algunos autores no proponen métodos concretos pero sí suministran ideas interesantes que pueden ser aplicadas a este problema. Por ejemplo, en la referencia [7], se puede observar que hay una relación entre la anchura y la probabilidad de distribución para la inicialización aleatoria de los pesos y el valor óptimo del paso de adaptación. Por ello, los autores proponen que los pesos de las

## 6.1. DESCRIPCIÓN DEL PROBLEMA

diferentes capas de la red se deberían inicializar con diferentes amplitudes, el intervalo para los pesos entre la entrada y capa oculta debería ser mayor.

En la referencia [6], se investiga la influencia de la inicialización sobre la capacidad de generalización y se concluye que hay un rango óptimo de “a” para una inicialización aleatoria en el intervalo  $[-a, a]$ , que será el que proporcione mejor generalización. Desafortunadamente, los autores no explican cómo escoger el mejor valor de “a” para un cierto problema.

En otros artículos se propone una nueva inicialización de pesos. Hay muchas ideas diferentes y aproximaciones.

Por ejemplo, en [8] y [9], se sigue lo que se llama “aproximación clásica”. Es una inicialización aleatoria en el intervalo  $[-a, a]$  y es la misma para todos los pesos de la red neuronal. El artículo propone fórmulas para calcular el valor apropiado de “a”.

Sin embargo, en otros artículos [2], [10], [11], [12] y [13], las ideas son muy diferentes de la aproximación clásica. En el siguiente apartado revisaremos algunos de estos métodos.

En la referencia [14], se presenta una comparación de diferentes métodos de inicialización de pesos. Sin embargo, las simulaciones se realizan solo para cinco métodos, todos ellos del tipo de “aproximación clásica”. Nosotros pensamos que hay otros métodos que utilizan una aproximación muy diferente a la clásica y sería interesante incluirlos en la comparación.

Nos gustaría también hacer notar que en la mayoría de los artículos los autores no suministran un conjunto de resultados completos en los que se mida la velocidad de convergencia, la capacidad de generalización y la relación de éxito en la convergencia. Los resultados son enfocados en uno de estos tres aspectos, generalmente la velocidad de convergencia. Esto hace que los resultados y conclusiones sean limitados. Nosotros pensamos que se deberían tener en cuenta los tres aspectos para determinar que método es más útil.

En este capítulo, presentaremos un análisis y una comparación empírica de los diferentes métodos de inicialización de pesos que hemos encontrado en la bibliografía.

Incluimos también en la comparación métodos que no siguen la aproximación clásica.

Como resultados obtendremos medidas de los tres aspectos que influyen en la inicialización: velocidad de convergencia, capacidad de generalización y relación de éxito en la convergencia.

Al principio hemos dicho que el mínimo local que obtengamos dependerá de la inicialización y del algoritmo de entrenamiento.

El algoritmo de entrenamiento utilizado fue Backpropagation con función de transferencia en las neuronas de tipo sigmoide o tangente hiperbólica, según el método empleado.

Sin embargo, los resultados que obtuvimos para 7 problemas nos hicieron pensar, que en algunos casos, la velocidad de convergencia y la capacidad de generalización podrían mejorarse utilizando una función de transferencia de tipo tangente hiperbólica.

Por ello, el siguiente paso fue repetir los experimentos utilizando como algoritmo de entrenamiento Backpropagation con función de transferencia tangente hiperbólica. Con esta función, el número de iteraciones en el entrenamiento debía disminuir considerablemente. Ahora, se aumento a doce el número de problemas sobre el que aplicar los métodos.

Esto supuso la modificación de algunos de los métodos y de las bases de datos. Normalizamos los valores de los datos (entradas y objetivos) dentro del intervalo  $[-1,1]$  que es el apropiado para la tangente hiperbólica, y en algunos de los métodos introdujimos pequeñas modificaciones de las cuales incluiremos la explicación al describir dichos métodos.

La función de transferencia tangente hiperbólica y su derivada vienen dadas por las ecuaciones siguientes.

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{2}; \quad f'(x) = \frac{\tanh(2x)}{2 \cdot \tanh(x)}; \quad (6.1)$$

A continuación revisaremos detalladamente cada uno de los métodos que vamos a estudiar.

## 6.2. Descripción de los diferentes métodos

### 6.2.1. Método 1

El primer método es la “aproximación clásica” para inicialización de pesos. Se trata de una distribución uniforme y aleatoria dentro de un pequeño intervalo centrado en cero. En nuestro caso utilizaremos el intervalo  $[-0.05, 0.05]$ .

Hemos incluido este método para tener una referencia de su eficiencia respecto del resto de métodos propuestos.

### 6.2.2. Método 2

En la referencia [16], se propone un método que es una variante de la inicialización clásica.

A partir de las ecuaciones del algoritmo de entrenamiento Backpropagation los autores de la referencia obtienen una condición que se debe satisfacer para que sea posible la convergencia.

Los autores proponen por ello, acotar los valores de los pesos en los límites mínimos dados por el intervalo que proporciona la ecuación (6.2).

$$\sqrt{\frac{\eta}{N_{input}}} < |w_i| \quad (6.2)$$

En la ecuación,  $\eta$  representa el paso de adaptación que se utilizará en el algoritmo de entrenamiento y  $N_{input}$  el número de neuronas en la capa de entrada.

Los pesos se inicializan de forma aleatoria pero de forma que su magnitud satisfaga la ecuación anterior (6.2).

En la referencia no se especifica claramente cómo escoger el valor del límite superior para el intervalo en el cual se inicializan los pesos. Nosotros hemos escogido finalmente para todos los experimentos como límite superior, la suma de 0.1 más el límite inferior (dado por la ecuación anterior).

Por otra parte, el paso de adaptación usado en los experimentos fue de 0.1.

Además, los resultados de la referencia sólo se analizan desde el punto de vista de velocidad de convergencia. Nosotros realizaremos el estudio desde los tres puntos de vista que ya nombramos en la introducción.



El entrenamiento con la función de transferencia tangente hiperbólica no supone ningún cambio en la aplicación del método, pero sí hay que escalar los datos y objetivos de entrenamiento en el intervalo  $[-1,1]$ .

### 6.2.3. Método 3

En la referencia [2], se describe un nuevo método un poco diferentes de la inicialización clásica. En él se propone una nueva estrategia de entrenamiento, llamada “Delta Pre-Training (DPT)”, con la cual se optimizan los valores iniciales de los pesos antes del proceso propio de entrenamiento.

Este método esta basado en lo siguiente. Supongamos que empezamos el entrenamiento inicializando todos los pesos a cero. De esta forma no habría saturación prematura y el error y la derivada de la función de transferencia serían máximos, lo cual resultará en una corrección de los pesos máxima y la convergencia será más rápida.

Sin embargo, en el entrenamiento con Backpropagation no se pueden inicializar los pesos a cero. Por ello, cualquier método que comience con los pesos inicializados a cero y que consiga luego aplicar Backpropagation sobre los pesos obtenidos en el método mejoraría considerablemente la convergencia.

Los autores del método que denominan DPT, intentan, utilizando la adaptación de los pesos en la capa superior y la ecuación de corrección del error en una red Multilayer Feedforward, entrenar una serie de perceptrones en los cuales los pesos estén inicializados a cero.

Para ello, en lugar de utilizar una red Multilayer Feedforward ésta es dividida en simples perceptrones. Por ejemplo, en nuestro caso de redes con una sola capa oculta, la red es dividida en dos perceptrones: el primero formado por la capa de entrada y la capa oculta, y el segundo formado por la capa oculta y la de salida. Una vez divididos habrá que definir cuales serán los nuevos objetivos y entradas respectivas de los nuevos perceptrones.

Después, los pesos de los perceptrones son inicializados a cero y se realiza un preentrenamiento de ambos perceptrones usando la Regla Delta, como algoritmo de entrenamiento

Mediante la Regla Delta no podemos entrenar cada perceptrón hacia la convergencia. Sin embargo, podemos adaptar los pesos a unos valores más apropiados para conseguir después la convergencia de la red Multilayer Feedforward con los pesos inicializados en estos valores.

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

Las ecuaciones que describen este algoritmo de Pre-entrenamiento se encuentran a continuación.

$$\begin{aligned}\Delta W_{ji}(t+1) &= \eta \cdot \delta_{P,j} \cdot I_{P,j} + \alpha \cdot \Delta W_{ji}(t) \\ \delta_{P,j} &= (D_{P,j} - O_{P,j}) \cdot f'(Net_{P,j})\end{aligned}\tag{6.3}$$

Donde se denota por  $D$  la salida deseada (objetivos),  $O$  la salida obtenida,  $I$  los patrones de entrada en el perceptrón,  $\eta$  el paso de adaptación,  $\alpha$  el momento,  $W$  los pesos y  $f'(Net_{P,j})$  la derivada de la función de transferencia aplicada sobre la entrada a una neurona de salida  $j$ .

Para el primer perceptrón (compuesto por la capa de entrada y la capa oculta) las entradas serán las mismas, el conjunto de entrenamiento. Los autores del artículo proponen cuatro formas diferentes de definir los objetivos (salidas en la capa oculta) del perceptrón.

Para el segundo perceptrón sus entradas deberían coincidir con los objetivos del primer perceptrón y sus salidas con los objetivos de la red neuronal inicial, es decir, los correspondientes al conjunto de entrenamiento inicial.

Para calcular los objetivos para el primer perceptron los autores de la referencia describen cuatro propuestas dependiendo del tamaño de la red. Dichas propuestas son para el caso de entrenamiento con función de transferencia sigmoide. En el caso de función de transferencia tangente hiperbólica se sustituirán los elementos binarios 0 y 1, usados al generar los vectores por los valores por  $-1$  y  $1$ , ya que las entradas y objetivos deben estar escaladas en el intervalo  $[-1,1]$ .

### **Propuesta 1**

Será aplicable si  $N_{pattern} < 2^{N_{hidden}}$ .

Donde  $N_{pattern}$  es el número de patrones de entrada y  $N_{hidden}$  es el número de elementos de la capa oculta.

Se construyen vectores de elementos binarios formados por todas las posibilidades de  $N_{hidden}$  bits. Tendremos  $2^{N_{hidden}}$  vectores. Después se asigna secuencialmente vectores diferentes como objetivos para cada una de las entradas. Como se cumple  $N_{pattern} < 2^{N_{hidden}}$  cada entrada tendrá asignada un objetivo diferente.

### **Propuesta 2**

Será aplicable si  $N_{pattern} < 2^{N_{hidden}}$ .

Se construyen vectores de elementos binarios formados por todas las posibilidades de  $N_{\text{hidden}}$  bits. Tendremos  $2^{N_{\text{hidden}}}$  vectores. Después, se asigna aleatoriamente vectores diferentes como objetivos para cada una de las entradas. Como se cumple  $N_{\text{pattern}} < 2^{N_{\text{hidden}}}$ , cada entrada tendrá asignada un objetivo diferente. Esta propuesta difiere de la anterior en la forma aleatoria de generar los objetivos. Según la referencia esta propuesta funciona bien en casos en que la anterior no es apropiada.

### **Propuesta 3**

Será aplicable si  $N_{\text{pattern}} > 2^{N_{\text{hidden}}}$ .

Se construyen unos vectores de la forma que se describió antes, asignando vectores aleatoriamente a cada una de las entradas. Ahora, un vector puede asignarse a varias entradas.

### **Propuesta 4**

Será aplicable si  $N_{\text{pattern}} > 2^{N_{\text{hidden}}}$ .

Inicializando aleatoriamente los pesos de la capa inferior se calculan los valores de las unidades ocultas, presentando los diferentes patrones de entrada a la red y estos son los que se tomarán como objetivos.

En nuestros experimentos, nosotros utilizaremos las propuestas 2 y 3 para definir los objetivos del primer perceptrón. Estos objetivos coincidirán con las entradas que utilizemos para el segundo perceptrón.

Para aplicar el método DPT necesitamos escoger tres parámetros, el paso de adaptación, el momento de la regla Delta y el número de iteraciones que se debe usar en el preentrenamiento.

En el entrenamiento con función sigmoide se escogieron 6 diferentes combinaciones para los parámetros; en el caso de la función tangente hiperbólica añadimos dos casos más con menor número de iteraciones ya que pensamos que con dicha función la convergencia podría ser mucho más rápida. El resultado final fue que para ese par de combinaciones no obtuvimos ninguna ventaja.

El procedimiento completo del entrenamiento Backpropagation con preentrenamiento para definir los valores iniciales de los pesos puede describirse con el siguiente algoritmo:

- 1) *Partición de la red en perceptrones.*
- 2) *Usar propuesta 2 o 3 para definir los objetivos en la capa oculta.*

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

- 3) *Poner los pesos a cero.*
- 4) */\*pre-entrenamiento del perceptron entre la capa de entrada y la oculta.\*/  
Para (numero de iteraciones)  
Escoger un patrón de entrada y ajustar los pesos según la ecuación (6.3);*
- 5) */\*pre-entrenamiento del perceptron entre la capa de oculta y la de salida\*/  
Para (numero de iteraciones)  
Escoger un patrón de entrada y ajustar los pesos según la ecuación (6.3);*
- 6) *Para los pesos obtenidos, aplicar el algoritmo BP a la red completa.*

El estudio experimental de este método es analizado desde el punto de vista de capacidad de generalización y éxito en la convergencia.

Nosotros de nuevo lo estudiaremos además desde el punto de vista de velocidad de convergencia.

### 6.2.4. Método 4

En la referencia [17] se propone una nueva inicialización de los pesos de tal forma que se cumplan las ecuaciones siguientes.

$$\left| \frac{w_o}{w_i} \right| < 1 \quad \text{o} \quad |w_o| < |w_i| \quad (6.4)$$

Donde  $w_o$  es el threshold de una unidad, y  $w_i$  el resto de pesos conectados desde la capa inferior a esa unidad.

Como indica la referencia, estas ecuaciones fueron deducidas por consideraciones geométricas para el caso de entradas normalizadas en el intervalo  $[-1, 1]$ , serán pues aplicables al caso del entrenamiento con función de transferencia tangente hiperbólica.

Supongamos la entrada a una neurona de la capa oculta para el caso de una red con  $N_{input}$  elementos en la entrada. La ecuación que define la entrada a dicha neurona oculta será:

$$\sum_{i=1}^{N_{input}} w_i \cdot x_i + w_o \quad (6.5)$$

Si igualamos este valor a cero se obtendrán las líneas de corte de los ejes  $x_i$  con los puntos  $(-w_o/w_1, 0, \dots, 0)$ ,  $(0, -w_o/w_2, 0, \dots, 0)$ , ...,  $(0, 0, \dots, -w_o/w_{N_{input}})$ . La ecuación (6.5) se puede escribir:

$$\sum_{i=1}^{N_{input}} \frac{x_i}{(-w_o / w_i)} = 1 \quad (6.6)$$

Si las entradas están escaladas en el intervalo  $[-1,1]$ , el resultado anterior genera las líneas de división en el espacio de características. Para que dichas líneas caigan dentro del espacio de variabilidad de los valores de entrada, se propone en la referencia utilizar la ecuación (6.4) para su inicialización.

Nosotros hemos aplicado las mismas consideraciones para deducir la ecuación (6.7) para el caso de entradas normalizadas en el intervalo  $[0,1]$ . Esta ecuación será la empleada en el caso del algoritmo de entrenamiento que utiliza la función de transferencia sigmoide.

$$0 < -\frac{w_o}{w_i} < 1 \quad (6.7)$$

De nuevo, en la referencia no se especifica completamente el proceso de inicialización de variables, sabemos que hay una restricción en el valor de los  $w_o$  pero no se dice cómo encontrar los valores adecuados. Finalmente inicializaremos los pesos  $w_i$  dentro del intervalo  $[-0.05,0.05]$  y después de eso los  $w_o$  serán calculados como el máximo  $w_i$  (teniendo en cuenta que los  $w_i$  están conectados a la misma neurona de la capa superior). Este procedimiento se aplica a todos los pesos, primero a los de la capa de entrada y luego a los de la capa oculta.

### 6.2.5. Método 5

En la referencia [10] se da otro método basado en consideraciones geométricas (las mismas del método anterior). Las líneas límite de las unidades de la capa oculta deben estar en el espacio de variabilidad de los valores de las entradas. Pero en este caso, se exige también la condición de que el espacio de variabilidad de las entradas debe estar en la región lineal de la función de transferencia (sigmoide o tangente hiperbólica) y no en la zona de saturación.

Con esta última condición, las unidades ocultas no estarán saturadas al principio, no tendrán una saturación incorrecta y se permite que los pesos evolucionen mucho más rápidamente hacia su valor final ya que la derivada de la función de transferencia es mayor en la región lineal.

El método está basado en el siguiente algoritmo, el cual se utiliza para calcular los pesos  $w_i$  conectados entre la unidad  $i$  de una capa y las unidades de la capa inferior ( $n$  pesos).

**Algoritmo:**

- 1) Calcular  $b$  mediante la ecuación (6.8), donde  $f$  es la función de transferencia y  $\varepsilon$  un cierto parámetro. En la referencia se utiliza el valor 0.1 para  $\varepsilon$  y nosotros utilizaremos también dicho valor en nuestros experimentos.

$$b = | f^{-1}(1-\varepsilon) - f^{-1}(\varepsilon) | \quad (6.8)$$

- 2) Usando  $n$  y un parámetro  $k$ , se calcula  $\hat{w}$  con la ecuación (6.9) para el caso de función de transferencia sigmoide. El espacio de variabilidad es  $[0,1]$  por lo que el espacio de salida es un espacio  $n$ -dimensional  $(0,1)^n$ .

$$\hat{w} = \frac{b}{k \cdot n} \quad \text{para sigmoide} \quad (6.9)$$

Utilizaremos la ecuación (6.10) para el caso de función de transferencia tangente hiperbólica. Ahora el espacio de variabilidad es en  $[-1,1]$  por lo que el espacio de salida será un espacio  $n$ -dimensional  $(-1,1)^n$ . A partir de las ecuaciones de la referencia se puede deducir en este caso la ecuación (6.10).

$$\hat{w} = \frac{b}{\sqrt{2} \cdot k \cdot n} \quad \text{para tangente hiperbólica} \quad (6.10)$$

- 3) Utilizando un parámetro  $\gamma$ , se genera un número  $a_i$  aleatorio uniformemente distribuido en el rango de la ecuación (6.11).

$$-\gamma \leq a_i \leq \gamma \quad (6.11)$$

- 4) Usando el valor de  $a_i$  se calcula  $w_i$  con la ecuación (6.12). Se repiten los pasos 3 y 4  $n$  veces para calcular los  $n$  diferentes pesos.

$$w_i = \hat{w} \cdot \sqrt{|a_i| + 1} \quad (6.12)$$

- 5) Se calculan los threshold  $w_o$ . Para el caso de función de transferencia sigmoide se calculan con la ecuación (6.13). Si la función de transferencia en la tangente hiperbólica se toman los threshold  $w_o$  como cero.

$$w_o = -0.5 \cdot \sum_{i=1}^n w_i \quad (6.13)$$

En la referencia se estudia el éxito en la convergencia y la eficiencia conseguida en el entrenamiento.

Como siempre nosotros haremos un estudio desde los tres puntos de vista nombrados en la introducción.

### 6.2.6. Método 6

En la referencia [12], se propone asignar diferentes inicializaciones de los pesos dependiendo de la importancia de las entradas.

A mayor importancia de una entrada el valor de inicialización de los pesos relacionados debería ser mayor. En particular, en la referencia sólo se muestran simulaciones y resultados para un problema con 8 entradas, se inicializan las entradas más importantes en  $[0.5,1]$ , las menos en  $[0,0.5]$  y el resto en  $[0,1]$ .

En la referencia, se utiliza un método para definir la importancia de las entradas, sin embargo puesto que nosotros ya hemos realizado un estudio muy detallado referente a este tema, hemos utilizado en nuestra investigación el mejor de los métodos que estudiamos y que se describe en la referencia [18].

Una vez realizada la ordenación de las entradas en función de su importancia, nosotros las clasificamos en tres grupos: las entradas más importantes (primeras de la lista), las entradas menos importantes (últimas de la lista) y el resto de entradas. En el primer y segundo grupo incluimos alrededor de un cuarto del total de entradas (en cada uno) y en el otro alrededor de la mitad.

En nuestros experimentos, los pesos de las entradas más importantes (primer grupo) son inicializados aleatoriamente en el intervalo  $[0.5,1]$ , las menos importantes en  $[0,0.5]$  y el resto en  $[0,1]$ , tal y como se sugiere en [12].

Los resultados de la referencia sólo son analizados desde el punto de vista de velocidad de convergencia y capacidad de generalización. Nosotros incluiremos también el estudio desde el punto de vista de éxito en la convergencia.

### 6.2.7. Método 7

Este método se denomina SCAWI y se propone en las referencias [8] y [20]. En primer lugar, se dan una serie de definiciones que serán necesarias para desarrollar el método.

**Definición 1:** *Una neurona está en estado de saturación cuando la magnitud de sus salidas es mayor que un determinado threshold o umbral.*

**Definición 2:** *Una neurona está en estado “paralizado” cuando está saturada y la magnitud de al menos una salida de error (diferencia entre una salida de la red y sus objetivos) es mayor o igual que un valor predeterminado.*

Recordemos que con Backpropagation, cuanto mayor sea la magnitud de salida de una neurona menor será su derivada y el entrenamiento se hará más dificultoso.

En [8] se utiliza el concepto de “paralyzed neuron percentage” (PNP).

**Definición 3:** *Se define un parámetro llamado “paralyzed neuron percentage (PNP), testeando cuantas veces una neurona está en estado “paralizado”.*

Durante las simulaciones puede calcularse el valor de PNP, para un conjunto de pesos, considerando todo el conjunto de entrenamiento y contando cuantas neuronas (de la capa oculta y de salida) están en estado paralizado y dividiendo este resultado entre el número de patrones.

La expresión para inicializar los pesos será:

$$w_{ji} = \frac{R \cdot a_o}{\sqrt{\theta_j}} \cdot r_{ji}; \quad \theta_j = \sum_{i=1}^{N_j} E[x_i^2] \quad (6.14)$$

Donde  $E[y]$  es la expectación de la variable  $y$ ,  $R$  es un escalar que se puede determinar a partir del valor de PNP y  $r_{ij}$  es un número aleatorio uniformemente distribuido en el rango  $[-1,1]$ .

Además,  $a_o$  es un valor que se escogerá según el tipo de función de transferencia usada,  $N_j$  es el número de entradas a la neurona  $j$ -ésima y  $x_i$  es cada una de dichas entradas.

En las referencias se explica de forma detallada cómo calcular el valor de  $R$  a partir del valor de PNP. Después, hay que encontrar una relación entre PNP y



EES(número de épocas de entrenamiento) que representa el coste del proceso de optimización. Se propone un valor de  $PNP_{opt}=5\%$  ( $R_{opt}$  y  $E[S^2]_{opt}$ ).

A partir de las expresiones de (6.14) se pueden deducir expresiones para la topología de red neuronal con una capa de entrada, una oculta y una de salida.

En (6.15) se da el valor medio de la expectación del valor cuadrático de las entradas.

$$v^2 = \frac{1}{N_{input}} \cdot \sum_{i=1}^{N_{input}} E[I_i^2] \quad (6.15)$$

Los valores de  $\theta$  se pueden calcular a partir de la siguiente ecuación:

$$\theta_{input} = 1 + N_{input} \cdot v^2 ; \quad \theta_{hidden} = 1 + N_{hidden} \cdot E[S^2]; \quad (6.16)$$

Para el caso de función de transferencia de tipo tangente hiperbólica en las referencias se propone el uso de un valor de  $a_0=1.47$ .

En la referencia se determina experimentalmente los valores óptimos de  $PNP$ ,  $R$  y  $E[S^2]$ . Finalmente se escoge  $R_{opt}=0.9$  y  $E[S^2]_{opt}=0.3$ . Sustituyendo en las ecuaciones anteriores se obtienen las ecuaciones para los pesos.

En la ecuación (6.17) se calcula los pesos conectados entre la capa de entrada y la capa oculta.  $N_{input}$  es el número unidades de entradas a la red

$$w_{ij}^{input} = \frac{1.3}{\sqrt{1 + N_{input} \cdot v^2}} \cdot r_{ij} \quad (6.17)$$

En la ecuación (6.18) se calcula los pesos conectados entre la capa oculta y la de salida.  $N_{hidden}$  es el número unidades de la capa oculta.

$$w_{ij}^{hidden} = \frac{1.3}{\sqrt{1 + 0.3 \cdot N_{hidden}}} \cdot r_{ij} \quad (6.18)$$

Estas ecuaciones están deducidas en las referencias para el caso de la red Multilayer Feedforward con función de transferencia tangente hiperbólica. Las entradas de la red deben estar normalizadas, en este caso, en el intervalo  $[-1,1]$ .

Los autores analizan los resultados desde el punto de vista de velocidad de convergencia y la relación de convergencias exitosas.

### 6.2.8. Método 8

En la referencia [13] se describen dos métodos de inicialización de pesos que se denominan: “*Forward Estimation Algorithm*” y “*Recurrent Estimation Algorithm*”.

Los métodos están basados en la idea de que si las condiciones iniciales de la red se escogen cercanas a un mínimo, el proceso de entrenamiento posterior tendrá una convergencia más rápida.

Nosotros denominaremos a los métodos como Método8a y Método8b.

Veamos primero cual es la notación que se usará en la descripción de los métodos.

$N_{input}$	Número de unidades de entrada de la red.
$N_{hidden}$	Número de unidades ocultas de la red.
$N_{salida}$	Número de unidades de salida de la red.
$P$	Número de patrones de entrenamiento.
$X$	Matriz de la entrada correspondiente a todos los patrones P.
$H$	Matriz de salidas de las unidades ocultas para los patrones P.
$D$	Matriz de salidas de la red para todos los patrones P.
$W_i$	Matriz de pesos entre la capa de entrada y la oculta.
$W_h$	Matriz de pesos entre la capa oculta y la de salida.
$B_i$ y $B_h$	Vector de umbrales de la capa oculta y de salida.
$b$	Vector constante $b = \{1 \ 1 \ 1 \ \dots \ 1\}^T$
$y_j^P$	Salida actual de la unidad j de la red al presentar el patrón P
$d_j^P$	Salida deseada de la unidad j de la red al presentar el patrón P

La ecuación (6.19) expresa las salidas de las unidades ocultas a partir de las entradas y pesos.

$$f \left( \begin{bmatrix} X & b \end{bmatrix} \begin{bmatrix} W_i \\ \dots \\ B_i \end{bmatrix} \right) = H \quad (6.19)$$

La ecuación (6.20) proporciona las salidas de la red neuronal.

$$f \left( \begin{bmatrix} H : b \\ \dots \\ Bh \end{bmatrix} \right) = D \quad (6.20)$$

En las ecuaciones,  $f$  representa la función de transferencia tangente hiperbólica:  $f(x)=\tanh(x)$ .

**Método 8a: “Forward Estimation Algorithm”**

Se escogen inicialmente de forma aleatoria los pesos y umbrales entre la capa de entrada y la capa oculta  $W_i$  y  $B_i$ . Con estos valores podemos calcular las salidas de las neuronas de la capa oculta,  $H$ , utilizando la ecuación (6.19).

Para estimar los pesos  $Wh$  se utiliza la ecuación (6.20). A partir de ella se obtiene la siguiente ecuación:

$$\begin{bmatrix} H : b \\ \dots \\ Bh \end{bmatrix} = G \cdot \begin{bmatrix} Wh \\ \dots \\ Bh \end{bmatrix} = f^{-1}(D) = Y \quad (6.21)$$

Los valores de  $f(x)$  están limitados al intervalo  $[-1,1]$  y por ello para poder calcular la inversa hay que limitarse a las cercanías de dicho intervalo.

$$f^{-1}(x) = (\tanh(x))^{-1} \quad (6.22)$$

Se tomará pues, el intervalo de definición de la tangente en  $[-0.99,0.99]$  por lo que los valores de las salidas deseadas (objetivos de la red), deberán de ser modificados para pertenecer a este intervalo.

El valor de los pesos se determina a partir del valor del rango  $\rho$  de las matrices  $G$  y  $Y$ . Hay tres casos distintos.

**Caso a)**

Si  $\rho([G]) = \rho([G \ Y]) = (N_{hidden} + 1)$  Habrá solución única:

$$WH = \begin{bmatrix} Wh \\ \dots \\ Bh \end{bmatrix} = G^{-1} \cdot Y \quad (6.23)$$

Donde  $G^{-1}$  es la matriz pseudo-inversa de  $G$ .

**Caso b)**

Si  $\rho([G]) = \rho([G \ Y]) < (N_{hidden} + 1)$

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

Entonces, existen infinitas soluciones, que resuelvan la ecuación (6.21). No existen métodos que indiquen que solución es mejor por lo que en la referencia han optado por tomar la solución conocida como de “mínima norma” que se muestra en la ecuación (6.24)

$$WH = \begin{bmatrix} Wh \\ \dots \\ Bh \end{bmatrix} = G^T \cdot (G \cdot G^T)^{-1} \cdot Y = G^\# \cdot Y \quad (6.24)$$

Donde  $G^\#$  es la pseudo-inversa de G.

### Caso c)

Si  $\rho([G]) < \rho([G \ Y])$  No hay solución exacta.

Pueden seguirse diferentes estrategias para encontrar una solución aproximada. Los autores de la referencia han optado por la solución llamada de “mínimo error cuadrático”.

$$WH = \begin{bmatrix} Wh \\ \dots \\ Bh \end{bmatrix} = (G^T \cdot G)^{-1} \cdot G^T \cdot Y = G^\# \cdot Y \quad (6.25)$$

Donde  $G^\#$  es la pseudo-inversa de G.

El algoritmo “*Forward Estimation*” queda finalmente como:

- 1) *Inicializar aleatoriamente las matrices  $W_i$  y  $B_i$  en el intervalo  $(-\lambda, \lambda)$ .*
- 2) *Calcular H (salidas de la capa oculta).*
- 3) *Estimar los valores de  $W_h$  y  $B_h$  utilizando una de las ecuaciones de los 3 casos anteriores.*

### **Método8b: “Recurrent Estimation Algorithm”**

Este algoritmo utilizará como primer paso el resultado del anterior.

El objetivo es acercar cada vez más los valores de inicialización de los pesos a un mínimo, minimizando para ello la raíz del error cuadrático medio (RMSE) cuya expresión viene dada por la siguiente ecuación:

$$RMSE = \frac{1}{P} \cdot \sum_{i=1}^P \left( \frac{1}{N_{salida}} \cdot \sum_{j=1}^{N_{salida}} \sqrt{(d_j^i - y_j^i)^2} \right) \quad (6.26)$$

A partir de los resultados del método anterior (Método 8a), se fijan los valores de  $Wh$  y  $Bh$  (en los obtenidos) y se calcula la inicialización para los pesos y umbrales  $Wi$  y  $Bi$ . Para ello se reescribe la ecuación (6.21) como sigue:

$$H \cdot Wh = f^{-1}(D) - b \cdot Bh = Z \quad (6.27)$$

Para cada uno de los tres casos del método anterior calculamos  $H$  según la ecuación (6.28).

$$H = \begin{cases} Z \cdot Wh^{-1} & \text{si } \rho([Wh^T : Z^T]) = \rho(Wh^T) = N_{hidden} \\ Z \cdot (Wh^T \cdot Wh)^{-1} \cdot Wh^T & \text{si } \rho([Wh^T : Z^T]) = \rho(Wh^T) < N_{hidden} \\ Z \cdot Wh^T \cdot (Wh \cdot Wh^T)^{-1} & \text{si } \rho([Wh^T : Z^T]) \neq \rho(Wh^T) \end{cases} \quad (6.28)$$

Si se da la última condición (caso c) se debería parar el proceso porque  $Wi$  y  $Bi$  son los mismos que ya teníamos.

Si se da la primera condición (caso a) hay una única solución para  $H$ . Puesto que  $H$  es la salida de las unidades ocultas, sus valores no deben superar el rango de la función de transferencia, es decir  $[-1,1]$ .

Como suele ocurrir que el número de unidades ocultas es mayor que el número de unidades de salida, la estimación de  $H$  suele ser la de la segunda condición (caso b). En este caso, el valor de  $H$  calculado por la ecuación (6.28) debe ser ajustado teniendo en cuenta el valor anterior,  $H_1$ , dado por la ecuación (6.19). El nuevo valor de  $H$  será:

$$H = H_1 + \lambda \cdot H \quad (6.29)$$

Donde  $\lambda$  es parámetro que toma valores en  $(0,1)$ .

Transformando la ecuación (6.19) podemos obtener los valores de  $Wi$  y  $Bi$  como sigue:

$$\begin{bmatrix} X : b \\ \dots \\ Bi \end{bmatrix} = A \cdot \begin{bmatrix} Wi \\ \dots \\ Bi \end{bmatrix} = f^{-1}(H) = K \quad (6.30)$$

De nuevo tenemos tres casos posibles para la solución según los valores de los rangos de las matrices. Los pesos  $Wi$  y  $Bi$  se pueden calcular por la ecuación (6.31).

$$\begin{bmatrix} Wi \\ \dots \\ Bi \end{bmatrix} = \begin{cases} A^{-1} \cdot K & \text{si } \rho([A : K]) = \rho(A) = (N_{input} + 1) \\ A \cdot (A \cdot A^T)^{-1} \cdot K & \text{si } \rho([A : K]) = \rho(A) < (N_{input} + 1) \\ (A^T \cdot A)^{-1} \cdot A^T \cdot K & \text{si } \rho([A : K]) = \rho(A) \end{cases} \quad (6.31)$$

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

Finalmente el algoritmo tendrá los siguientes pasos:

- 1) *Inicializar aleatoriamente las matrices  $W_i$  y  $B_i$  en el intervalo  $(-\lambda, \lambda)$ .*
- 2) *Calcular  $H$  (salidas de la capa oculta) mediante la ecuación (6.19).*
- 3) *Estimar los valores de  $W_h$  y  $B_h$  mediante (6.23), (6.24) o (6.25) que describían los 3 casos posibles de solución a la ecuación (6.21). Calcular las salidas de la red y con ella el error RMSE. Si el valor del error RMSE es inferior al valor 0.05 (criterio de parada) pasar al paso 7, sino seguir con el paso 4.*
- 4) *Calcular  $H$  utilizando los valores de  $W_h$  y  $B_h$  obtenidos usando la ecuación (6.28).*
  - i) *Si se da el caso c) ir al paso 7 y acabar.*
  - ii) *Si se da el caso b) recalcular  $H$  con la ecuación (6.29).*
- 5) *Ajustar los valores de  $H$  obtenidos al intervalo  $[-0.99, 0.99]$  (cercañas del rango de la función de transferencia).*
- 6) *Utilizando el valor de  $H$  obtenido antes (según el caso a, b o c):*
  - i) *Estimar los valores de  $W_i$  y  $B_i$*
  - ii) *Calcular  $H$  para los nuevos valores de  $W_i$  y  $B_i$ .*
  - iii) *Dado  $W_h$  y  $H$  calcular las salidas de la red.*
  - iv) *A partir de estas, calcular el nuevo RMSE.*

*Si el valor de RMSE no se ha reducido pasar al paso 7 y acabar. Si el nuevo valor del error es menor volver al paso 2.*
- 7) *Acabar el proceso de inicialización de pesos.*

### 6.2.9. Método 9

En la referencia [11] se describe un nuevo método de inicialización de pesos basado en la idea de reducir el tiempo de entrenamiento.

De nuevo, se intenta buscar un mejor punto de inicio para el proceso de aprendizaje, y para ello se aplican conceptos de tipo geométrico modificando de forma simple el modelo de red Multilayer Feedforward.

Supongamos un perceptrón de dos capas en la cual las neuronas de salida son lineales. Se puede expresar dicha salida como:

$$f(\vec{x}) = \sum_{i=1}^n \gamma_i \cdot \sigma(\vec{\alpha}_i \cdot \vec{x} + \beta_i) + \gamma_o \quad (6.32)$$

Donde  $\bar{x}$  será el vector de entrada a la red, y  $\sigma$  es la función de transferencia de las neuronas ocultas. Para aumentar la velocidad de entrenamiento de la red los autores proponen realizar la siguiente modificación geométrica.

$$f(\bar{x}) = \sum_{i=1}^n \gamma_i \cdot \sigma(\bar{d}_i \cdot (\bar{x} - \bar{t}_i)) + \gamma_o \quad (6.33)$$

Donde llaman  $\bar{d}_i$  al “parámetro de dilatación”, equivalente a los antiguos pesos y  $\bar{t}_i$  al “parámetro de translación” cuyo producto por el parámetro de dilatación sería la antigua  $\beta_i$ .

Con la fórmula inicial la derivada parcial respecto de los pesos dependía del vector  $\bar{x}$ , mientras que ahora la derivada respecto del parámetro de dilatación dependerá de  $(\bar{x} - \bar{t}_i)$ .

La posición de los hiperplanos que dividen el espacio de las entradas, como hemos visto en otras ocasiones, viene dado por la ecuación:

$$\bar{a}_i \cdot \bar{x} + \beta_i = 0; \quad \Rightarrow \quad -\frac{\beta_i}{\|\bar{a}_i\|} \quad (6.34)$$

En el método propuesto en [11] la posición de los hiperplanos dependerá de los parámetros  $\bar{t}_i$  y  $\bar{d}_i$ . De hecho, la salida de una neurona dependerá de tres parámetros:

1. *La posición de los “centros de activación” dados por los parámetros  $\bar{t}_i$ .*
2. *La “orientación” de los hiperplanos de decisión dada por  $(\bar{d}_i / \|\bar{d}_i\|)$ .*
3. *La “selectividad” de la función de transferencia dada por  $\|\bar{d}_i\|$ .*

En este método en lugar de dar una inicialización aleatoria se separa el proceso en tres partes, cada una relacionada con uno de los parámetros. Los parámetros son seleccionados aleatoriamente en tres pasos.

El parámetro de translación debe estar dentro de la parte del espacio de entrada en la cual se define la función a aprender. La orientación se definirá dentro de la esfera de radio unidad, y el parámetro de selectividad será escogido de forma que evite la saturación de las neuronas en todos los ejemplos de entrada.

Consideremos el caso de la red Multilayer Feedforward, si los ejemplos de entrenamiento están distribuidos en el intervalo  $[a, b]$ , se tomará para las translaciones el intervalo  $[a, b]$ , para la orientación se escogerá algún valor de la esfera de radio unidad y para la selectividad un valor comprendido en el intervalo

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

$[\lambda_1/(b-a), \lambda_2/(b-a)]$ , donde las  $\lambda_i$  son constantes que dependen de la función de transferencia.

En nuestro caso, las entradas están definidas en el intervalo  $[-1,1]$ , ya que utilizamos la tangente hiperbólica como función de transferencia en el entrenamiento. Por ello los intervalos de variación de los parámetros quedarán:

$$\begin{aligned} \text{translación} \quad & \bar{t}_i \text{ en } [-1, 1] \\ \text{orientación} \quad & \frac{\vec{d}_i}{\|\vec{d}_i\|} \text{ en esfera de radio unidad} \\ \text{selectividad} \quad & \left[ \frac{\lambda_1}{2}, \frac{\lambda_2}{2} \right] \text{ con } \lambda_1 = 2; \lambda_2 = 10 \end{aligned} \quad (6.35)$$

Los valores de  $\lambda_1$  y  $\lambda_2$  se escogen tal y como se recomienda en la referencia.

El algoritmo final para la aplicación de este método puede quedar como:

- 1) *Inicializar aleatoriamente las orientaciones y normalizarlas a 1*
  - i) *Inicializamos aleatoriamente  $\vec{d}_i$ .*
  - ii) *Calculamos  $\|\vec{d}_i\|$ .*
  - iii) *Calcular las orientaciones  $\vec{d}_i/\|\vec{d}_i\|$ .*
- 2) *Inicializar aleatoriamente la selectividad  $\|\vec{d}_i\|$  en el intervalo  $[1,5]$ .*
- 3) *Construir el vector de dilataciones  $\vec{d}_i$  como el producto de las orientaciones y la selectividad calculadas en los pasos 1 y 2.*
- 4) *Construir el vector de translaciones  $\bar{t}_i$  aleatoriamente en el intervalo de valores de los vectores de entrada  $[-1,1]$*
- 5) *Construir la matriz de pesos y umbrales entre la capa de entrada y la oculta.*

$$W_i = \vec{d}_i; \quad (6.36)$$

$$\text{Bias} = \vec{d}_i \cdot \bar{t}_i;$$

- 6) *Inicializar aleatoriamente los pesos entre la capa oculta y la de salida en el intervalo  $[0, 0.05]$  (ceranos a cero), tal y como recomienda la referencia.*



### 6.2.10. Método 10

El método descrito en la referencia [22] utiliza lo que se llaman Prototipos para inicializar los pesos de la red. Los autores hacen mención de un método de cálculo de prototipos llamado RCE, descrito en la referencia [23]. Con este método el número de prototipos que se escoja determinará el número de unidades ocultas de la red a entrenar. En el apéndice del capítulo se encuentran los resultados finales del número de unidades ocultas escogido.

Veamos en primer lugar en que consiste el método de inicialización propuesto y después el método de generación de prototipos.

#### *Método de inicialización*

Supongamos conocido un conjunto de vectores de referencia  $\vec{p}^i$  con  $i=1, \dots, K$ , llamados “Prototipos” y con clasificación conocida.

Estos vectores se utilizarán para dar una regla de decisión que asigne a cada patrón del conjunto de entrenamiento una clase. Una regla podría ser clasificar cada patrón en la clase de vecinos más próximos a los prototipos, definiendo la proximidad mediante una determinada métrica. De esta forma, un vector lejano de los prototipos o cercano a varios de diferentes clases será rechazado.

La regla de decisión puede ser implementada con una red de N entradas, K unidades ocultas y C salidas. Una unidad oculta estará activada si la distancia entre el vector de entrada y un prototipo  $\vec{p}^i$  es menor que un cierto umbral  $\beta_i$ .

Se propone utilizar esta arquitectura como base para inicializar los pesos de una red Multilayer Feedforward.

El método propone realizar una transformación del espacio de entrada de forma que los nuevos vectores estén normalizados, a costa de añadir una nueva dimensión al espacio de entrada (que será la proyección en una hiperesfera de radio unidad).

La transformación que se realiza será la siguiente:

$$p \circ h: \quad \vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} \Rightarrow \vec{x} = \begin{pmatrix} y_1/r \\ \vdots \\ y_{n-1}/r \\ \left(1 - \frac{1}{r^2} \cdot \sum_{i=1}^{n-1} y_i^2\right) \end{pmatrix} \quad (6.37)$$

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

Donde  $r \geq \max(\|\bar{y}\|)$ ,  $\bar{y}$  es un ejemplo del conjunto de entrenamiento inicial y  $N=(n-1)$  es la dimensión del espacio de entrada original.

Los vectores  $y$  representan las entradas originales (conjunto de entrenamiento), y los  $x$  los nuevos vectores de entrada transformados.

Para poder aplicar el método de inicialización suponemos las entradas normalizadas como se indica en la ecuación (6.37), y que disponemos de  $K$  prototipos.

Supongamos la red Multilayer Feedforward que queremos entrenar tal que la capa de entrada y oculta de la red tengan una unidad umbral con activación constante igual a 1. Las matrices de los pesos (conexiones)  $WI$  y  $WH$  entre la capa primera y la segunda, y entre la segunda y tercera respectivamente, se pueden inicializar según las siguientes ecuaciones.

Las conexiones  $WI$  entre las unidades de la capa de entrada (considerando la unidad umbral para  $N+1$ ) y la capa oculta, tendremos:

$$\begin{aligned} \forall i \in \{1..K\} \quad \forall j \in \{1..N\} \quad WI_{i,j} &= \alpha \cdot p_j^i \\ \forall i \in \{1..K\} \quad WI_{i,N+1} &= -\alpha \cdot \beta_i \quad \beta_i < 1 \end{aligned} \quad (6.38)$$

Para las conexiones  $WH$  entre las unidades de la capa oculta (considerando la unidad umbral para  $K+1$ ) y la capa de salida:

$$\begin{aligned} \forall c \in \{1..C\} \quad \forall i \in \{1..K\} \quad WH_{c,i} &= \begin{cases} \gamma > 1 & \text{si } p^c \in (\text{clase } c) \\ 0 & \text{si } p^c \notin (\text{clase } c) \end{cases} \\ \forall c \in \{1..C\} \quad WH_{c,K+1} &= -1 \end{aligned} \quad (6.39)$$

En las ecuaciones (6.38) y (6.39) la  $i$  recorre los diferentes prototipos, la  $j$  las componentes de un vector de entrada de dimensión  $N=(n-1)$  y la  $c$  las diferentes clases.

Con la inicialización anterior de las matrices de pesos, una entrada que cumpla la ecuación (6.40) causará que la  $i$ -ésima unidad oculta tenga un valor de activación cercano a la unidad y mayor que la inversa de  $\gamma$ . Por ello, la unidad de salida que corresponda a la clase del prototipo  $\bar{p}^i$  será también activada.

$$\bar{p}^i \cdot \bar{x} - \beta_i > 0 \quad (6.40)$$

De aquí se puede deducir:

$$\bar{p}^i \cdot \bar{x} > \beta_i \quad (6.41)$$

Los valores de las constantes  $\alpha$ ,  $\beta_i$  y  $\gamma$  deben ser escogidos tal y como se aconseja en la referencia [22], es decir, de forma que se controle el grado de linealidad de la activación. Valores grandes provocan que el entrenamiento posterior de la red con Backpropagation sea lento, por ello hay que escoger unos valores de compromiso.

### ***Generación de prototipos***

Puesto que el número de unidades ocultas debe ser lo menor posible el conjunto de prototipos tenderá a ser menor que el conjunto de entrenamiento.

Los métodos para la generación de prototipos pueden utilizarse para sintetizar la información contenida en el conjunto de aprendizaje en forma de pocos vectores: los prototipos.

El método descrito en la referencia [23], llamado RCE, se utiliza para problemas de clasificación con algoritmos supervisados, como en el caso que nos ocupa en toda nuestra investigación.

Para desarrollar el método, se parte como dijimos antes, de una arquitectura de  $N$  entradas,  $K$  unidades ocultas y  $C$  unidades de salidas.

Las neuronas de entradas  $F_j$  se proyectan a través de las conexiones dadas por una matriz  $A$  en las neuronas prototipo  $G_i$ , las cuales están unidas por las conexiones dadas por la matriz  $B$  con las neuronas de salida o clasificación  $H_c$ .

Cada clase será representada por la activación de una única neurona  $H_c$ . Si una entrada causa la activación final de una cierta  $H_c$ , esta entrada será clasificada en la categoría asociada con dicha neurona.

Para una clase dada puede haber más de un prototipo, correspondientes a diferentes  $G_i$ . Los diferentes prototipos son de la misma clase por lo que causan que se active una cierta  $H_c$ .

Además el conjunto de neuronas  $H$  de salida tiene otra fuente adicional de entradas que viene representada por un bloque  $T$  (supervisor), el cual corrige la respuesta de clasificación de la red.  $T$  cambiará el valor de las conexiones  $B$  entre las unidades  $G$  y  $H$ .

Supondremos que cada prototipo  $G_i$  tiene una “región de influencia” en el espacio de entradas. Supondremos los patrones de entrada normalizados. Aquí se utilizará la normalización vista en (6.37).

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

Una clase de patrones definirá una región o conjunto de regiones en el espacio de entrada, de forma que los patrones quedarán agrupados en categorías según a la clase a la que pertenecen.

Para poder identificar la clase de un patrón la red neuronal tendrá que aprender cuales son las regiones que definen las clases o categorías.

El modelo descrito en la referencia [23] desarrolla por si solo un conjunto de prototipos, cuyas regiones de influencia nos definirán las áreas correspondientes a las diferentes clases en el espacio de patrones de entrada.

En el método desarrollado en [23] y denominado RCE los prototipos finales que se obtienen serán patrones escogidos del conjunto de entrenamiento.

Supongamos los patrones de entrada  $\vec{f}$  normalizados, es decir,  $\vec{f} \cdot \vec{f} = 1$ .

Cada prototipo tiene asociada una conexión dada por:

$$\vec{A}_i = \lambda_i \cdot \vec{p}^i \quad \text{donde} \quad \vec{p}^i \cdot \vec{p}^i = 1 \quad \text{y} \quad \lambda_i > 1 \quad (6.42)$$

Puede ocurrir que la presentación de un prototipo a las neuronas  $H$  no active la neurona adecuada (la correspondiente a la clase correcta). En este caso, se dará una de las siguientes situaciones: el patrón estará mal clasificado o “confusión” (no esta claro a que clase pertenece).

Estas situaciones pueden ser corregidas mediante el bloque T (supervisor) que debe adaptar las conexiones para que la clasificación sea correcta.

La modificación en las conexiones de las neuronas prototipo vendrán gobernadas por dos condiciones:

1. **Nueva clasificación.** Si un patrón de la clase  $c$  es presentado a la red y la neurona  $H_c$  no responde, una nueva neurona  $G_k$  será asociada al patrón  $\vec{f}(c)$  y la conexión entre  $G_k$  y  $F$  se modifica como sigue:

$$A_{kj} \rightarrow P_{kj} = \lambda_o \cdot f_j \quad \lambda_o > 1 \quad (6.43)$$

2. **Confusión.** Si presentamos el patrón  $\vec{f}(c)$  a la red y causa alguna actividad en  $H_w$  con  $w \neq c$ , entonces habrá que modificar el factor hasta que  $\vec{f}(c)$  alcance un umbral, es decir que se cumpla:

$$\lambda_r \rightarrow \lambda_r' \quad \text{tal que} \quad \lambda_r' \cdot \vec{p}^r \cdot \vec{f}(c) = 1 \quad (6.44)$$

Estas dos reglas para adquirir y modificar los prototipos llevan a la red a aprender la distribución espacial de los patrones.

Para una red no entrenada, las neuronas de  $G$  no están conectadas con las de  $H$  por lo que los elementos de la matriz  $B$  serán todos ceros, o un número muy pequeño.

Cuando se presenta un patrón a la red las neuronas  $H$  no responderán por encima de un cierto umbral.

Una neurona  $G$  es asignada a  $\vec{f}(c)$  como un prototipo para esa clase. Simultáneamente la conexión entre  $G$  y  $H_c$  es puesta a 1. Esta entrada representará el primer ejemplo de una clase de patrones. Sea  $c=I$ , si el mismo patrón vuelve a ser presentado al sistema la respuesta de  $G$  debería ser:

$$\lambda_o \cdot \vec{p}^1(c) \cdot \vec{f}(c) = \lambda_o > 1 \quad (6.45)$$

La señal de salida  $\lambda_o$  desde  $G$  debe causar que  $H_c$  se active.

Para otro patrón diferente  $\vec{f}'(c')$  tal que pertenezca a la misma clase que el anterior  $c'=c$  pueden darse dos casos como se indica en la ecuación (6.46). En el caso a) la neurona  $H_c$  debería activarse y el patrón estaría correctamente clasificado, y en el caso b) el patrón está mal clasificado y debe ser asignado un nuevo prototipo  $P^2(c)$  con la unión entre  $G$  y  $H_c$  igual a 1.

$$\begin{aligned} a) \quad & \lambda_o \cdot \vec{p}^1(c) \cdot \vec{f}'(c) > 1 \\ b) \quad & \lambda_o \cdot \vec{p}^1(c) \cdot \vec{f}'(c) < 1 \end{aligned} \quad (6.46)$$

De esta forma una clase puede ser caracterizada por más de un prototipo.

Cuando las clases son diferentes  $c' \neq c$  se generará también un nuevo prototipo para una clase diferente donde el valor de  $\lambda_o$  se reduce a  $\lambda_l$ .

$$\begin{aligned} & \lambda_o \cdot \vec{p}^1(c) \cdot \vec{f}'(c') > 1 \\ & \lambda_o \rightarrow \lambda_l \quad \text{tal que} \quad \lambda_l \cdot \vec{p}^1(c) \cdot \vec{f}'(c') = 1 \end{aligned} \quad (6.47)$$

Aplicando todo lo anterior podemos resumir el algoritmo para la selección de prototipos como sigue:

Partimos del conjunto de entrenamiento: patrones con objetivo conocido. Escogemos un valor inicial para  $\lambda_o > 1$ , y otros para  $\beta$  y  $\gamma$ .

- 1) *Calcular la norma de los patrones de entrada.*
- 2) *Calcular cuál es la máxima norma de las calculadas en el paso 1.*
- 3) *Redefinir los patrones de entrada según la ecuación (6.37), donde  $r$  es el máximo calculado en el paso 2.*

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

- 4) (Para  $i=1$  hasta Nite). Repetir hasta el punto de convergencia de la red
- 5) (Para  $k=1$  hasta  $N^\circ$  patrones) Repetir para el conjunto de patrones
  - i) Si  $k=1$  &  $i=1$  (primer patrón) coger el primer patrón multiplicarlo por  $\lambda_0$  y guardarlo como primer Prototipo.
  - ii) Si no (para el resto de patrones) ( $k \neq 1$ , excepto el primer patrón). Analizar el resultado de la ecuación (6.41).
    - I) Si no se ha activado ninguna salida (la ecuación 6.41 no se cumple) almacenar un nuevo prototipo.
    - II) Si se activa al menos una unidad de salida y el patrón está mal clasificado (su clase no se corresponde con la del prototipo que hace máxima la ecuación 6.41), estamos en "confusión" y hay que modificar el valor de  $\lambda$  usando las ecuaciones de la (6.44) o (6.47).
- 6) Fin del bucle de patrones.
- 7) Fin del bucle de iteraciones.

### 6.2.11. Método 11

En la referencia [24] se presenta un nuevo método de inicialización de pesos, basado en la aproximación de los límites de decisión de las clases de los patrones de entrada.

En Multilayer Feedforward las clases están separadas por hiperplanos. Los autores desarrollan un método de inicialización que impone una serie de restricciones en la orientación y localización de los hiperplanos de la capa intermedia (ecuaciones de sumas de entrada ponderadas). Estas restricciones intentan eliminar los estados físicos de la red que están altamente relacionados con los falsos mínimos locales en la red, mejorando así la convergencia.

En la referencia [25], en la cual se basan los autores del presente método, se utilizan conceptos de agrupamiento (en inglés "clustering") y de "vecinos más próximos" para encontrar las fronteras de decisión entre las clases (en inglés "boundary segment"), de forma que a partir de estas se pueda encontrar una inicialización de los pesos.

Como hemos dicho antes, la red Multilayer Feedforward se conoce como un clasificador de hiperplanos, es decir se puede encontrar una función discriminante que separe las diferentes clases mediante hiperplanos. Las redes son entrenadas

para poder encontrar los parámetros de dichas funciones discriminantes (pesos y umbrales) a partir de los datos de entrenamiento.

Cada nodo en la capa de salida representa una función discriminante que separa entre una clase y todas las demás. Por ejemplo, para la neurona de salida  $O_i$  la función discriminante se puede escribir como:

$$O_i = f_i(x) = g\left(\sum_j w_{ij} \cdot s_j(x) + b_i\right) \quad (6.48)$$

Donde  $s_j(x)$  es la activación en la capa oculta para el nodo  $j$ , y  $g$  es la función de transferencia usada en dicha capa. Entonces:

$$s_j(x) = g\left(\sum_i w_{jk} \cdot x_k + b_j\right) = g(h_j(x)) \quad (6.49)$$

Las funciones  $h_j(x)$  representan las ecuaciones de los hiperplanos para los diferentes valores de  $j$  si igualamos sus valores a cero.

En la referencia [25], se calcula la posición inicial y orientación de las fronteras de las clases, usando agrupamiento (clustering) de los datos y el método de “vecinos más próximos”.

Los datos de entrenamiento son primero clasificados, agrupados por clases (clusters), del resultado se calcula la media “los centros de cada grupo” (centros de cada cluster) que representarán entonces a cada grupo. Se calcula para cada una de las parejas de centros más cercanos los límites que representarán un segmento.

Por ejemplo para dos dimensiones de entrada, la frontera es perpendicular a la bisectriz de la línea que conecta dos centros de grupos de clases diferentes.

Las ecuaciones que representan dichos segmentos serán:

$$h_j(x) = X^T \cdot (P - Q) - \frac{1}{2} \cdot (P^T \cdot P - Q^T \cdot Q) \quad (6.50)$$

Donde  $X$  es un patrón de entrada,  $P$  y  $Q$  son vectores que representan los centros de los grupos que caen a igual distancia del segmento límite.

Para la capa oculta:

$$h_j(x) = \sum_k w_{jk} \cdot x_k + b_j = W_j^T \cdot X + b_j \quad (6.51)$$

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

De aquí se pueden deducir las ecuaciones para calcular los pesos entre la capa de entrada y la oculta, siendo  $\beta$  un escalar que hay que escoger:

$$\begin{aligned}W_j &= \beta \cdot (P - Q) \\b_j &= -\frac{\beta}{2} \cdot (P^T \cdot P - Q^T \cdot Q)\end{aligned}\tag{6.52}$$

Sin embargo el método propuesto en [25] presenta el problema de que el número de centros de grupo (en inglés “cluster centers”) necesario no está determinado a priori, por lo que el tiempo de entrenamiento se puede alargar demasiado si hemos de probar con varios valores.

En [24], los autores proponen una modificación del método que elimina esta desventaja. Ahora, en lugar de determinar los centros de grupo, se buscan los ejemplos de entrenamiento cercanos a la frontera.

El criterio para seleccionar los ejemplos de la frontera se basa en escoger un subconjunto de ejemplos de entrenamiento que no varíe las fronteras de decisión. A dicho subconjunto se le denomina “conjunto consistente” y de él se generarán las fronteras (boundary segment) que se utilizarán para inicializar la red

Veamos el funcionamiento del método. Partimos del conjunto inicial de patrones de entrenamiento.

- 1) Tomando de uno en uno los patrones del conjunto de entrenamiento, se genera un conjunto llamado  $T_{ORG}$  formado por dichos patrones ordenados por clases, es decir, primero tomamos todos los patrones de la primera clase (subconjunto  $T_1$ ), después tomamos todos los patrones de la segunda clase (subconjunto  $T_2$ ), y así sucesivamente hasta completar el conjunto de entrenamiento.  $T_{ORG} = \{T_1, T_2, \dots, T_m\}$ , donde  $m$  es el número de clases.
- 2) En lugar de estimar como en [25] los centros de grupo, se calcula la “Probabilidad a Posteriori” de cada patrón  $X^i$  del conjunto  $T_{ORG}$  usando el método de  $k$ -vecinos más próximos. Este método consiste básicamente en, determinar de los  $k$ -patrones más próximos a  $X^i$  cuantos son de una determinada clase  $j$ , y dividir dicho valor por el número de vecinos total que hemos escogido. Es decir, se calcula la distancia del patrón  $X^i$  a todos los demás patrones  $X^n$  ( $n$  distinto de  $i$ ) nos quedamos con los  $k$  patrones cuya distancia sea menor y vemos cuantos de ellos son de una cierta clase  $j$ . En la siguiente ecuación podemos ver como se calculan estas probabilidades  $P(\omega_j | X^i)$ .



$$P(\omega_j | X^i) = \frac{k_j}{k}; \quad j = 1, \dots, m \quad i = 1, \dots, p \quad (6.53)$$

- 3) El siguiente paso será eliminar del conjunto  $T_{ORG}$  los patrones que estén mal clasificados usando el método de k-vecinos más próximos.
  - i) Para cada patrón  $X^i$  se calcula cuál es la clase más probable según la ecuación (6.53), para todas las clases  $j$  busco cual tiene mayor probabilidad.
  - ii) Si la clase objetivo de  $X^i$  no coincide con la clase de mayor probabilidad, hay que rechazar ese patrón y eliminarlo del conjunto de patrones.  $T = T_{ORG} - \{X^i\}$
  - iii) Se repite el proceso de i) a iii) para todos los patrones  $i$  de  $T_{ORG}$ .
  - iv) Al acabar, se toma  $T_{ORG} = T$ .
- 4) Eliminar los patrones que están en la región de no aceptación. Para ello primero se busca la región  $L(t)$  formada por los ejemplos o patrones que son muy cercanos al límite, donde los métodos de vecinos más próximos tienen alta probabilidad de error.
  - i) Para cada patrón disponemos de las probabilidades de pertenecer a cada una de las clases, ecuación (6.53). Para un patrón  $X^i$  calculo qué clase tiene mayor probabilidad y la llamo *maximo1*. Calculo qué clase tiene el siguiente máximo para la probabilidad y la llamo *maximo2*. Es decir, las dos clases que tienen mayor probabilidad.
  - ii) Si  $(\text{maximo1} - \text{maximo2}) < \text{ACEPTAR}$  se rechaza el patrón  $X^i$ . Es decir, si las probabilidades de dos clases son muy cercanas no está claro a que clase debe pertenecer por lo que se rechaza. *ACEPTAR* es un umbral que se debe fijar.
  - iii) Se toma  $T = T_{ORG} - \{X^i\}$ , se repite (i) y (ii) hasta recorrer todos los patrones de  $T_{ORG}$ .
  - iv) Al acabar, tomar  $T_{ORG} = T$ .
- 5) Ordenar los patrones de  $T_{ORG}$  en cada subconjunto de clases en orden ascendente según el valor de la probabilidad de pertenencia a las clases. Es decir,  $T_{EDT} = \{T_1, T_2, \dots, T_m\}$ , donde en el subconjunto  $T_j$  de patrones que pertenecen a la clase  $j$ , primero estará el patrón que tiene menor probabilidad de estar en esa clase y el último será el patrón con mayor probabilidad.
- 6) Construir el conjunto  $T_{CNN}$  escogiendo el primer patrón de cada uno de los subconjuntos de clases anteriores. Es decir,  $T_{CNN}$  estará formado por

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

m patrones cada uno de una clase (el que tiene menos probabilidad de ser de la clase, las fronteras).

- 7) Utilizar el algoritmo *CNN* modificado para clasificar y seleccionar patrones. Para ellos, redefinir el conjunto  $T_{CNN}$  añadiendo los patrones necesarios.
  - i) Para cada elemento  $X^i$  de  $T_{EDT}$  encontrar la distancia de los elementos de  $T_{CNN}$  a  $X^i$ . Sea  $X^j$  de  $T_{CNN}$  el elemento que dista menos a  $X^i$
  - ii) Si  $X^j$  y  $X^i$ , no son de la misma clase y  $X^j \notin T_{CNN}$  se añade  $X^j$  al conjunto  $T_{CNN}$ . Poner  $T_{CNN} = \{T_{CNN}, X^j\}$
  - iii) Repetir los pasos i) y ii) para todos los elemento de  $T_{EDT}$ .

Después de aplicado el algoritmo *CNN* modificado, tendremos el conjunto  $T_{CNN}$  de patrones que son necesarios para realizar la inicialización, es decir para calcular las fronteras, ecuación (6.50).

Por cada patrón del conjunto  $T_{CNN}$  cabe la posibilidad de que se genere una unidad oculta en la red neuronal. En el apéndice del capítulo, podemos encontrar el número de unidades ocultas obtenido para este método y para las diferentes bases de datos que usaremos.

### **Proceso de inicialización:**

Partimos de los conjuntos  $T_{ORG}$  y  $T_{CNN}$  anteriores.

- 1) Para un patrón  $X^i$  del conjunto  $T_{CNN}$  encontrar el vecino más próximo  $X^j$  de  $T_{ORG}$ : tendremos el  $X^j$  que dista menos de  $X^i$ .
- 2) Para el patrón  $Q=X^i$  de  $T_{CNN}$  y el vecino calculado  $P=X^j$ .
  - i) **Si** es el primer patrón seleccionado de  $T_{CNN}$ , calcular la frontera o contorno (boundary segment), que es, el bisector perpendicular de la pareja  $(P,Q)$ , según la ecuación (6.50). Podemos representar dicho contorno mediante un vector fila:

$$\text{Contorno}_i = [ -\beta \cdot (P^T \cdot P - Q^T \cdot Q) , \beta \cdot (P - Q) ] \quad (6.54)$$

- ii) **Si no** es el primer patrón seleccionado ver si la pareja  $(P,Q)$  está bien clasificada por alguno de los contornos ya calculados, y si no realizar el paso 2.i) y generar un nuevo contorno.

- 3) Repetir los pasos anteriores hasta completar todos los patrones del conjunto  $T_{CNN}$ . Al final tendremos una matriz *CONTORNO* con vectores fila ( $contorno_i$  para todo  $i$ ).
- 4) Seleccionar el número de unidades de la capa oculta de la red Multilayer Feedforward como el número de contornos o fronteras calculadas.
- 5) A partir de las ecuaciones de los contornos, inicializar los pesos entre la capa de entrada y la capa oculta utilizando las ecuaciones (6.51) (6.52), donde  $\beta$  es un parámetro que se debe escoger. La matriz de pesos coincidirá con la matriz *CONTORNO* del paso 3.
- 6) Para calcular los pesos entre la capa oculta y la capa de salida una vez conocidos los de la capa anterior podemos:
  - i) Calcular la salida de la capa oculta a partir de los pesos anteriores, usando la función de transferencia tangente hiperbólica y añadiendo la neurona umbral con salida 1.
  - ii) Se vuelve a calcular los valores de los objetivos de las entradas en el intervalo  $[-0.99, 0.99]$  para poder calcular la inversa de la función de transferencia.
  - iii) Se pueden calcular las entradas a las neuronas de salida calculando la inversa de la función de transferencia (de la tangente hiperbólica)
  - iv) Los pesos se pueden calcular a partir de la pseudoinversa de las salidas de la capa oculta, multiplicadas por las entradas a las neuronas de salida.

En este método se utilizan tres parámetros que hay que escoger: el número *KNN* de vecinos a usar en el método de k-vecinos más próximos, el umbral límite *ACEPTAR* para rechazar patrones que están en la región de no aceptación y  $\beta$ .

### 6.2.12. Método 12

En la referencia [9] se describe un nuevo método de inicialización basado en conceptos geométricos.

Como hemos visto hasta ahora, la existencia de falsos mínimos locales deteriora el funcionamiento de las redes disminuyendo su eficiencia e, incluso, impidiendo la convergencia o haciendo esta muy lenta.

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

En la referencia los autores apuntan algunas de las causas que provocan la aparición de estos falsos mínimos:

- Aquellos nodos ocultos cuyos límites de decisión (hiperplanos de separación de clases) están fuera de la región del espacio de presentación de ejemplos, o patrones a la red.
- Funciones duplicadas de nodos ocultos.
- Nodos que son inactivos en las regiones de los ejemplos y que generan “regiones muertas”.

En la referencia se dice que un método que intente evitar el problema de aparición de mínimo local, debería inicializar los pesos de forma que se cumpla:

1. Los límites de decisión de las unidades ocultas deberían posicionarse en las regiones que ocupan los patrones de entrada.
2. Las orientaciones de los límites de decisión de las unidades ocultas deben ser lo más variadas posibles.
3. Cada parte de la región de ejemplos necesita tener, al menos, una unidad oculta que active los ejemplos que ocupan esta región.

En coordenadas cartesianas en un sistema  $n$  dimensional los hiperplanos de decisión asociados a las unidades de la capa oculta, están matemáticamente descritos por los pesos que llegan a cada nodo (pesos entre la capa de entrada y oculta).

En este método se propone hacer un cambio a coordenadas esféricas. Se define entonces un vector  $\vec{r}$   $n$ -dimensional de longitud  $r$  y  $(n-1)$  ángulos  $(\theta_1, \dots, \theta_{(n-1)})$ , y con ellos se podrá hacer referencia a los hiperplanos perpendiculares a dicho vector posicionados en un punto.

La longitud  $r$  del vector controla la distancia de separación entre los hiperplanos y el centro de las áreas donde se encuentran los datos, y los ángulos controlan las orientaciones de los hiperplanos.

Hay que escoger los ángulos y longitudes de los vectores de forma que los hiperplanos de decisión estén uniformemente distribuidos a través del espacio de características, y que al mismo tiempo se maximicen las diferentes orientaciones entre los diferentes nodos ocultos.

Para un espacio  $n$ -dimensional tendremos para cada unidad oculta un vector de ángulos  $(n-1)$ -dimensional. Para que se cumplan las condiciones anteriores, sus

valores serán diferentes si el número de la neurona oculta es par o impar (difieren en  $\pi$ ). Dada una unidad oculta  $j$ , el vector  $\vec{\theta}^j$  de  $(n-1)$  ángulos será:

$$\theta_i^j = \begin{cases} \left( \frac{\pi}{N_{hidden}} \cdot j \right) & \text{si } j \text{ par, } \forall i = 1, \dots, n-1 \\ \left( \frac{\pi}{N_{hidden}} \cdot j + \pi \right) & \text{si } j \text{ impar, } \forall i = 1, \dots, n-1 \end{cases} \quad (6.55)$$

Donde  $N_{hidden}$  es el número total de unidades ocultas de la red.

Las longitudes de los vectores, parámetros  $r$ , se escogen de forma que cumplan las condiciones anteriores. En la referencia se realiza la siguiente asignación:

$$\begin{aligned} \text{Para } j = 1 \text{ hasta } N_{hidden} \\ r^j = \frac{0.5}{j} \end{aligned} \quad (6.56)$$

El vector que define la orientación del hiperplano  $j$  vendrá dado por los ángulos según la ecuación:

$$w_j = (\cos(\theta_1^j), \text{sen}(\theta_1^j), \dots, \text{sen}(\theta_{(n-1)}^j)) \quad (6.57)$$

Construimos la matriz  $W$  de dimensión  $(N_{hidden} * n)$  con los vectores fila  $w_j$ . Un punto del hiperplano se calcula multiplicando por la longitud del vector  $r$ .

Los umbrales de las conexiones a la capa oculta se pueden calcular con:

$$B = -\sum_{i=1}^n (W \cdot \text{Punto})^T = \sum (W \cdot (r \cdot W))^T \quad (6.58)$$

Donde  $B$  será un vector fila de dimensión  $N_{hidden}$  que representa las conexiones entre la neurona umbral en la capa de entrada y las neuronas ocultas, (las  $b_j$ ).

Podemos calcular la norma para cada vector orientación  $j$  de una unidad oculta, a partir de la ecuación (6.57), como la raíz cuadrada de la suma de los cuadrados de las componentes de dicho vector. El vector normalizado  $\hat{w}_j$  que nos define la orientación del hiperplano (perpendicular a este) será:

$$\hat{w}_j = \frac{w_j}{\|w_j\|} \quad (6.59)$$

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

Construimos de nuevo la matriz  $W$  de dimensión  $(N_{hidden} * n)$  con los vectores fila  $\hat{w}_j$ .

Hasta ahora hemos escogido la posición de los límites de decisión es, decir, donde se cumple:

$$y = \sum_{i=1}^n \omega_{ji} x_i + b_j = 0 \quad (6.60)$$

Queda por escoger la pendiente de la función de transferencia. En la referencia no se especifica la forma de calcularla, por lo que en nuestra investigación hemos optado por utilizar el procedimiento desarrollado en uno de los métodos aquí descrito, el método 9.

En él, se define la sensibilidad como un vector de sensibilidades de dimensión  $(N_{hidden} \times 1)$  escogidas aleatoriamente en el intervalo  $[1,5]$ .

Transformamos el vector de sensibilidades en una matriz  $S$  de dimensión  $(N_{hidden} \times N_{input})$  repitiendo en cada columna el vector de sensibilidades.

Los pesos, podrán calcularse, como el producto elemento a elemento de la matriz de sensibilidades por la matriz  $W$  de orientaciones normalizadas.

$$WI \text{ tal que } WI_{ij} = S_{ij} \cdot W_{ij} \quad \forall i, j \quad (6.61)$$

La matriz de pesos total entre la capa de entrada y oculta será:

$$W_{TOTAL} = [B^T \quad \vdots \quad WI] \quad (6.62)$$

En cuanto a la elección de los pesos entre la capa oculta y la de salida, en la referencia se aconseja inicializarlos a un valor fijo suficientemente pequeño, por ejemplo, igual o inferior a 0.25. En nuestras investigaciones hemos tomado todos los valores de dichos pesos iguales a 0.1.

### 6.2.13. Método 13

En la referencia [26], se describe un nuevo método que utiliza un aprendizaje no supervisado en el proceso de inicialización de pesos.

Los pesos entre la capa de entrada y la capa oculta pueden ser determinados a través de variaciones lineales o no lineales, de las llamadas “reglas de aprendizaje Hebbianas generalizadas” (en inglés “generalized Hebbian learning rules”), también conocidas como Reglas de Oja. En la investigación, se utilizarán

diferentes reglas para evaluar después experimentalmente su eficiencia en la convergencia.

Presentaremos en primer lugar las reglas de aprendizaje Hebbiana generalizadas, que se utilizan en la referencia para inicializar la representación interna de la red (pesos) a través de un proceso de aprendizaje no supervisado.

Consideremos una red Multilayer Feedforward con una capa oculta.

Definimos los siguientes parámetros:

<i>Unidades de entradas a la red</i>	$x_{m,k} \quad m = 1, 2, \dots, N$
<i>Vector de salida de las neuronas de entrada</i>	$\vec{x}_k^T = (1, x_{1,k}, x_{2,k}, \dots, x_{N,k})$
<i>Salidas de las neuronas ocultas</i>	$ho_{j,k} = \rho(hi_{j,k}) = \rho(\vec{v}_j^T \cdot \vec{x}_k)$
<i>Pesos entre capa de entrada y oculta</i>	$\vec{v}_j^T = (v_{j0}, v_{j1}, \dots, v_{jN})$
<i>Vector de salida de las neuronas ocultas</i>	$\vec{h}_k^T = (1, ho_{1,k}, ho_{2,k}, \dots, ho_{Nh,k})$
<i>Salidas de las neuronas de salida</i>	$oo_{i,k} = \sigma(oi_{i,k}) = \rho(\vec{w}_i^T \cdot \vec{h}_k)$
<i>Funciones de activación</i>	$\rho(\cdot)$ y $\sigma(\cdot)$

La función de activación utilizada aquí será la tangente hiperbólica, tanto para la capa oculta, como para la capa de salida.

La representación interna de la red puede ser inicializada actualizando los valores de los pesos  $V$  que conectan las entradas, con las unidades ocultas de acuerdo con la regla Hebbiana generalizada.

### **Primer caso**

Primero vamos a considerar el caso de que las unidades de salida ocultas sean lineales. Sea  $V_{k-1} = [\vec{v}_{1,k-1}, \vec{v}_{2,k-1}, \dots, \vec{v}_{Nh,k-1}]$  la matriz de pesos resultante de la actualización de los pesos con respecto a las entradas  $x_p$  para  $p=k-1$ .

Según la regla de Oja, se puede obtener la matriz  $V$  de pesos con respecto de los vectores de entrada  $\vec{x}_k$  con la ecuación:

$$\begin{aligned} V_k &= V_{k-1} + \alpha \cdot (I - V_{k-1} \cdot V_{k-1}^T) \cdot \vec{x}_k \cdot \vec{x}_k^T \cdot V_{k-1} & \text{si } \vec{h}_k &= V_{k-1}^T \cdot \vec{x}_k \\ &= V_{k-1} + \alpha \cdot (\vec{x}_k - V_{k-1} \cdot \vec{h}_k) \cdot \vec{h}_k^T \end{aligned} \quad (6.63)$$

Según la ecuación anterior, cada vector de pesos vendrá dado por:

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

$$\bar{v}_{i,k} = \bar{v}_{i,k-1} + \alpha \cdot hi_{i,k} \cdot \left( \bar{x}_k - \sum_{m=1}^{Nh} hi_{m,k} \cdot \bar{v}_{m,k-1} \right) \quad (6.64)$$

Se pueden dar varios casos:

- 1) Si  $Nh < N$  significa que los vectores columna de la matriz de pesos  $V$  son ortonormales, por lo que habrá solución estable si  $V^T \cdot V = I$ . Se puede establecer la condición de parada del algoritmo si  $\bar{h}i_k = V^T \cdot \bar{s}_k$ , donde  $\bar{s}_k = V \cdot \bar{h}i_k$ . Después de completado cada ciclo, es decir, los valores de los vectores de pesos han sido todos actualizados con respecto a todos los vectores de entrada, la proximidad de la estimación de  $V$  a una solución estable se medirá a partir del error. El proceso de entrenamiento no supervisado se acaba si el error es lo suficientemente pequeño.  $E_v = \sum_{k=1}^m \left\| \bar{h}i_k - V^T \cdot \bar{s}_k \right\|^2$ .
- 2) Si  $Nh > N$ , la matriz de covarianzas de entrada es definida positiva, entonces habrá solución estable sí y solo sí  $V \cdot V^T = I$ . Esto quiere decir que las filas de  $V$  son ortonormales. Si  $V$  es una solución estable entonces  $\bar{x}_k = V \cdot \bar{h}i_k$ , donde  $\bar{h}i_k = V^T \cdot \bar{x}_k$ . El criterio de parada será que el error sea suficientemente pequeño.  $E_v = \sum_{k=1}^m \left\| \bar{x}_k - V \cdot \bar{h}i_k \right\|^2$ .
- 3) Si  $Nh = N$ , la matriz cuadrada  $V$  es una solución estable si las columnas y filas son ortonormales. Entonces, se puede introducir como medida de parada cualquiera de los criterios de los dos puntos anteriores.

### Segundo caso

La regla de aprendizaje lineal puede ser generalizada para incluir funciones de activación no lineales.

Oja y otros desarrollaron una variante para las reglas no lineales de actualización. Según la regla de Oja se actualiza la ecuación (6.63) a la expresión:

$$V_k = V_{k-1} + \alpha \cdot \left( \bar{x}_k - V_{k-1} \cdot L_d(\bar{h}i_k) \right) \cdot L_h(\bar{h}i_k^T) \quad (6.65)$$

Donde  $L_h(\bar{h}i_k^T) = [L_h(hi_{1,k}), L_h(hi_{2,k}), \dots, L_h(hi_{Nh,k})]$

Claramente la regla lineal puede ser interpretada como un caso especial donde  $L_h(z) = L_d(z) = z$ .

La ecuación para cada vector de pesos puede obtenerse:

$$\bar{v}_{i,k} = \bar{v}_{i,k-1} + \alpha \cdot L_h(hi_{i,k}) \cdot \left( \bar{x}_k - \sum_{m=1}^{Nh} L_d(hi_{m,k}) \cdot \bar{v}_{m,k-1} \right) \quad (6.66)$$



El criterio de parada, se puede establecer teniendo en cuenta el criterio para el caso de la regla de actualización lineal.

- 1) Si  $Nh < N$ , el criterio se basa en la disminución del error  $E_v = \sum_{k=1}^m \left\| \vec{h}i_k - V^T \cdot \vec{s}_k \right\|^2$ , donde  $\vec{s}_k = \sum_{m=1}^{Nh} L_d(hi_{m,k}) \cdot \vec{v}_m$
- 2) Si  $Nh > N$ , la terminación del proceso no supervisado se basa en el error  $E_v = \sum_{k=1}^{Nh} \left\| \vec{x}_k - \sum_{m=1}^{Nh} L_d(hi_{m,k}) \cdot \vec{v}_m \right\|^2$ , donde  $\vec{h}i_k = V^T \cdot \vec{x}_k$ .

### **Proceso de aprendizaje**

La inicialización de los pesos comienza aplicando el proceso de aprendizaje no supervisado (regla de Oja) sobre los pesos del primer nivel (entre la capa de entrada y oculta). El algoritmo aplicado será el siguiente:

- 1) Inicializar aleatoriamente los pesos en un cierto intervalo.
- 2) Seleccionar las funciones  $Lh(\cdot)$  y  $Ld(\cdot)$  y el valor de  $\alpha$ . Poner  $p=1$ .
- 3) **PARA** cada  $k=1,2,\dots,m$  (índice de patrones de entrada)

i) **Calcular**  $\vec{h}i_k = V^T \cdot \vec{x}_k$

ii) **Actualizar**  $\vec{v}_{i,k}$   $i=1,2,\dots,Nh$  usando la ecuación (6.66).

iii) **Calcular**  $\vec{h}i_k = V^T \cdot \vec{x}_k$  y  $\vec{s}_k = \sum_{m=1}^{Nh} L_d(hi_{m,k}) \cdot \vec{v}_m$ ,  $k=1,2,\dots,m$

iv) **Si**  $Nh < N$  **entonces:**  $E_v = \sum_{k=1}^m \left\| \vec{h}i_k - V^T \cdot \vec{s}_k \right\|^2$

**Si**  $Nh > N$  **entonces:**  $E_v = \sum_{k=1}^{Nh} \left\| \vec{x}_k - \sum_{m=1}^{Nh} L_d(hi_{m,k}) \cdot \vec{v}_m \right\|^2$

v) **Si**  $p > 1$  **entonces:**  $E_v^{rel} = (E_v^{old} - E_v) / E_v^{old}$ ;  $E_v^{old} = E_v$

vi) **Si** ( $p=1$ ) **OR** ( $E_v^{rel} > E_v^o$ ) **entonces:**  $p=p+1$  y volver al punto 3.

**Si no ACABAR.**

Como vemos, la condición de parada del algoritmo es que el error relativo sea menor que un cierto umbral que nosotros fijamos. En la referencia, se recomienda un valor pequeño y nosotros hemos tomado 0.01.

Como valor de  $\alpha$  hemos tomado 0.001 tal y como se recomienda en la referencia [26].

## 6.2. DESCRIPCIÓN DE LOS DIFERENTES MÉTODOS

En cuanto a la elección de las funciones  $L_h(\cdot)$  y  $L_d(\cdot)$  hemos considerado tres casos particulares.

**Primer caso:**  $L_h(z)=z$  y  $L_d(z)=\tanh(z)(1-\tanh(z)^2)$  **Método13a**

**Segundo caso:**  $L_h(z)=\tanh(z)$  y  $L_d(z)=z$  **Método13b**

**Tercer caso:**  $L_h(z)=\tanh(z)$  y  $L_d(z)=\tanh(z)(1-\tanh(z)^2)$  **Método13c**

Obteniendo así tres aplicaciones diferentes del método.

Faltan por calcular los pesos entre la capa oculta y la capa de salida. Para ello, la referencia recomienda realizar un entrenamiento supervisado partiendo de los valores de los pesos entre la primera y segunda capa que ya hemos calculado (fijarlos), y realizar un entrenamiento para adaptar los pesos de salida.

Nosotros hemos aplicado Backpropagation y hemos establecido la condición de parada evaluando si el error relativo,  $(E_{old}-Error)/E_{old}$ , es menor que el valor de parada que fijamos antes, siendo  $Error$  el error cuadrático medio y las funciones de activación de la red tangente hiperbólica.

## 6.3. Metodología

El objetivo principal de la investigación desarrollada en este capítulo es evaluar la eficiencia de diferentes métodos de inicialización de pesos, con relación al procedimiento de inicialización clásico.

En el apartado anterior se describieron detalladamente 13 métodos diferentes de inicialización de pesos, en donde el primero de ellos es la inicialización clásica.

Algunos autores a la hora de desarrollar sus métodos han tenido en cuenta qué función de transferencia (sigmoide o tangente hiperbólica) sería utilizada posteriormente para el entrenamiento de la red.

Así, podemos encontrar varios métodos que se enfocaron al uso de la sigmoide y algunos que utilizan exclusivamente la tangente hiperbólica.

En el resto de métodos no se especificó el tipo de función de transferencia y en principio son válidos para cualquiera de las dos, pero adaptando los datos a un intervalo adecuado. Nosotros hemos utilizado para la sigmoide datos normalizados en el intervalo  $[0,1]$ , y para la tangente hiperbólica datos normalizados en  $[-1,1]$ .

Puesto que el tipo de función de transferencia era diferente en los métodos, pensamos que para poder realizar una comparación fiable entre ellos deberíamos ver en qué sentido podría influir esto. Por ello, en nuestras investigaciones antes de enfrentarnos al problema de utilizar todos los métodos realizamos una pequeña prueba.

Partimos de los 6 primeros métodos, pensados para la utilización de la función de transferencia sigmoide y evaluamos su eficiencia con 8 problemas de clasificación.

Para tener una medida relativa de la eficiencia que proporcionaban ambas funciones de transferencia, adaptamos el método clásico (método 1) para utilizar como función de transferencia la tangente hiperbólica. Esto lo realizamos simplemente adaptando los datos de los problemas para que estuviesen normalizados en el intervalo  $[-1,1]$ .

Después comparamos los resultados obtenidos para el método 1 tanto cuando utilizamos la función sigmoide como cuando utilizamos la tangente hiperbólica. Así, nos dimos cuenta que el uso de la función de transferencia tangente hiperbólica, mejoraba en gran medida tanto la capacidad de

### 6.3. METODOLOGÍA

generalización como la velocidad de convergencia, manteniendo en general un valor similar de la probabilidad de éxito en la convergencia.

Por ello, decidimos finalmente utilizar la función de transferencia tangente hiperbólica en los 13 métodos. Para ello, adaptamos los datos de cada problema para que estuvieran normalizados en el intervalo  $[-1,1]$ . Para algunos de los métodos (que usaban la sigmoide) tuvimos también que modificar algunas ecuaciones. En el apartado 6.2 de este capítulo en la descripción de cada método se indicaron los cambios realizados.

Finalmente, la metodología general usada en nuestras investigaciones fue la de aplicar los métodos de inicialización a diferentes problemas de clasificación, obteniendo para cada método y problema un conjunto de pesos. Realizamos diferentes aplicaciones de un método obteniendo diferentes conjuntos de pesos inicializados para dicho método (*Nredes inicializaciones por cada método*). Para cada problema utilizamos también diferentes particiones de datos entre conjunto de entrenamiento, cros-validación y test (*Nredes particiones*). Así para cada problema y método se entrenaron finalmente varias redes (*Nredes*) con las diferentes inicializaciones de pesos y particiones de datos.

Primero, sólo usamos los métodos basados en la sigmoide (con 20 redes), y después, repetimos los experimentos con todos los métodos adaptados al uso de la tangente hiperbólica (con 30 redes).

Al principio de este capítulo, se justificó de qué manera la inicialización de los pesos influía sobre la convergencia de la red neuronal. Las propiedades de convergencia que se ven generalmente afectadas son: la velocidad de convergencia, la probabilidad de que haya éxito en la convergencia y la capacidad de generalización (eficiencia final de la red).

Por ello, nosotros hemos realizado el estudio comparativo desde esos tres puntos de vista.

La probabilidad de éxito en la convergencia para cada problema y método, se medirá por el número de redes que no convergen " $N$ ", con respecto al número de redes entrenadas, *Nredes* (20 o 30 según el caso).

Consideramos que una red no ha convergido cuando su porcentaje de aciertos con el conjunto de cros-validación es inaceptable. Más concretamente, consideramos que la red no ha convergido si su porcentaje de cros-validación es inferior en un 10 o 15% a los valores más altos del porcentaje de cros-validación que se obtienen con las demás redes entrenadas.

La velocidad en la convergencia se mide a través del número de épocas de convergencia. Para determinar dicho valor se ha utilizado cros-validación en el proceso de entrenamiento. La velocidad de convergencia, que denominaremos como “*ITER*”, vendrá dada por la media de los (*Nredes-N*) valores de los puntos de convergencia obtenidos para las redes convergidas en cada problema y método.

La capacidad de generalización se medirá para cada problema y método calculando el porcentaje medio de correcta clasificación en el test para las (*Nredes-N*) redes convergidas y la denominaremos “*POR*” en los resultados experimentales.

Así, como resultado final, hemos medido para las diferentes redes entrenadas para cada problema y método, el porcentaje medio de aciertos obtenido en el test (capacidad de generalización), la iteración media (punto de convergencia) hallada con cros-validación (relacionada con la velocidad en la convergencia), y el número de redes convergidas (relacionado con la probabilidad de éxito en la convergencia).

Para el método 3, escogimos diferentes valores para los parámetros. Tomamos para el momento  $\alpha$  el valor 0.5. El número de iteraciones *Ite* lo escogimos como un múltiplo del número de patrones de entrenamiento tal y como se recomienda en la referencia [2]. Escogimos ocho combinaciones de (*Ite*,  $\eta$ ) que denominamos con las letras: a=(2.5 $N_{\text{pattern},2}$ ), b=(5 $N_{\text{pattern},2}$ ), c=(7.5 $N_{\text{pattern},2}$ ), d=(10 $N_{\text{pattern},2}$ ), e=(2.5 $N_{\text{pattern},4}$ ), f=(5 $N_{\text{pattern},4}$ ), g=(7.5 $N_{\text{pattern},4}$ ), h=(10 $N_{\text{pattern},4}$ ) y obtuvimos ocho resultados diferentes de la aplicación del método 3.

Para utilizar el método 5 es necesario conocer los parámetros  $k$  y  $\gamma$ . Nosotros hemos utilizados varias combinaciones de estos valores las cuales parecen razonables viendo los experimentos realizados en la referencia [10]. Para el parámetro  $k$  escogimos los valores 2, 5, 8 y para el parámetro  $\gamma$  tomamos los valores 0.3, 0.6 y 0.9. De nuevo utilizamos las diferentes combinaciones de los parámetros ( $\gamma$ ,  $k$ ) que llamaremos con las letras: a=(2,0.3), b=(5,0.3), c=(8,0.3), d=(2,0.6), e=(5,0.6), f=(8,0.6), g=(2,0.9), h=(5,0.9) y i=(8,0.9). Obtendremos así nueve resultados diferentes de la aplicación del método 5.

En el método 10 escogimos el punto de convergencia de la red RCE mediante cros-validación. Para ese valor tendremos el número final de prototipos. Aplicamos este método para los diferentes valores de  $\alpha=5, 10, 20$ , de  $\beta_i=0.5, 0.8$ , y de  $\gamma=1, 4$ . Utilizamos las combinaciones ( $\alpha, \beta_i, \gamma$ ) que denominamos como: a=(5,0.5,1), b=(5,0.5,4), c=(5,0.8,1), d=(5,0.8,4), e=(10,0.5,1), f=(10,0.5,4), g=(10,0.8,1), h=(10,0.8,4), i=(20,0.5,1), j=(20,0.5,4), k=(20,0.8,1), l=(20,0.8,4) y tenemos 12 resultados diferentes de la aplicación de este método.

### 6.3. METODOLOGÍA

En el método 11 se utilizan tres parámetros que hay que escoger: el número *KNN* de vecinos a usar en el método de k-vecinos más próximos, el umbral límite *ACCEPTAR* para rechazar patrones que están en la región de no aceptación y  $\beta$ . En nuestra investigación hemos escogido estos parámetros tal y como se sugiere en la referencia [24]. Utilizaremos diferentes combinaciones (*KNN*, *ACCEPTAR*,  $\beta$ ) de los parámetros que denominaremos: a=(3,0.15,0.3) , b=(3,0.15,0.5), c=(3,0.3,0.3), d=(3,0.3,0.5), e=(5,0.15,0.3), f=(5,0.15,0.5), g=(5,0.3,0.3), h=(5,0.3,0.5), i=(8,0.15,0.3), j=(8,0.15,0.5), k=(8,0.3,0.3), l=(8,0.3,0.5). Obtenemos así 12 resultados diferentes de la aplicación del método 11.

Los métodos 10 y 11 seleccionan además el número de unidades ocultas de las redes que se entrenarán. Para el método 10 el valor final del número de unidades ocultas que se obtiene es independiente de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$  utilizados en el método. Sin embargo, para el método 11, el valor final del número de unidades ocultas de la red a entrenar depende de los parámetros *KNN* y *ACCEPTAR* obteniendo diferentes resultados según sus valores.

En el apartado de resultados experimentales mostramos con más detalle cada uno de los pasos seguidos en nuestra investigación.

## 6.4. Desarrollo Experimental

Como hemos comentado antes, inicialmente realizamos una pequeña prueba con 6 métodos de inicialización basados en la función de transferencia sigmoide, aplicandolos a 7 problemas de clasificación: *Glass Identification (GL)*, *Heart Disease (HE)*, *The Monk's Problems (M1, M2, M3)*, *1984 U.S. Congressional Voting Records (VO)*, *Wisconsin Diagnostic Breast Cancer (WD)*.

El siguiente paso fue obtener, a partir de la programación y ejecución de cada uno de los métodos, los conjuntos iniciales de pesos que utilizaríamos para el entrenamiento de las diferentes redes.

Para que los resultados que obtengamos sean independientes de la inicialización (que todavía tiene componentes aleatorias) y de la partición de los datos en entrenamiento, cros-validación y test, hemos realizado para cada problema 20 diferentes particiones aleatorias de los datos y hemos utilizado aquí datos normalizados en el intervalo  $[0,1]$ . Cada aplicación de un método generará un conjunto diferente de pesos inicializados (uno para cada red que entrenemos).

Las particiones de los datos se realizaron de la forma que se indica en la Tabla 1 del apéndice, escogiendo después para cada partición de forma aleatoria los ejemplos que pertenecen a cada grupo.

Para cada método, con cada conjunto de pesos (20 diferentes inicializaciones para un método), y las particiones realizadas para un cierto problema se entrenaron 20 redes. Con cada una de las redes convergidas, se obtuvo el punto de convergencia utilizando cros-validación, y se realizó el test calculando el porcentaje de correcta clasificación, que mide la capacidad de generalización.

Para las redes que habían convergido ( $N_{redes-N}$ ) se calculó la media de los resultados obteniendo los valores de *ITER* (iteración media de convergencia) y *POR* (eficiencia media en el test) con su error.

En la Tabla 2 del apéndice del capítulo, se muestran los resultados obtenidos para los 6 métodos utilizando en el entrenamiento la función de transferencia sigmoide, y para 7 problemas de clasificación.

En dicha tabla, cada fila representa un método y para cada problema hay tres columnas. La primera *POR*, representa el porcentaje medio de correcta clasificación (para las redes convergidas), la segunda *N* mide el número de redes que no han convergido de las 20 redes entrenadas y la tercera *ITER*, representa el valor medio del punto de convergencia de las redes convergidas.

#### 6.4. DESARROLLO EXPERIMENTAL

De la Tabla 2 se obtienen algunos resultados previos.

Se puede observar que no hay diferencias significativas entre los métodos 1 y 2 en ninguno de los tres aspectos que estudiamos.

Para el método 3 se observa que la relación de éxito en la convergencia es muy baja. El método no funciona bien. La razón puede ser que en dicho método hay que escoger una serie de parámetros. Sin embargo, nosotros hemos probado con seis diferentes combinaciones de los mismos en los rangos que se sugieren en la referencia [2]. Podrá ocurrir que el método sea muy sensible a los parámetros y que estos dependan mucho del problema particular.

El método 4 resulta similar al método 1. Sin embargo, se observa que para el problema M2 no funciona, la relación de éxito en la convergencia es del 0%. Pensamos que este resultado no es debido a un error en nuestras simulaciones sino que es debido al problema particular (M2).

El método 5 ha sido evaluado para nueve combinaciones de los parámetros y el resultado obtenido es muy variado. El principal resultado es una alta variabilidad.

Y por último, en la Tabla 2, los resultados del método 6 son similares a los del método 1, aunque para el caso de la velocidad de convergencia se encuentra de nuevo algo de variabilidad, en unos problemas es mejor el método 6, M1, M3 y WD y en otros es peor, GL, HE, M2 y VO.

Para tener una medida relativa de la eficiencia proporcionada por la función de transferencia sigmoide y la función tangente hiperbólica, adaptamos el método clásico (método 1) para el uso de la tangente hiperbólica. Para ello, simplemente adaptamos los datos de los problemas normalizándolos en el intervalo  $[-1,1]$ .

Después entrenamos 20 redes (con 20 diferentes particiones de datos y las inicializaciones de los pesos obtenidas) pero ahora, con los datos normalizados en el intervalo  $[-1,1]$  y utilizando como función de transferencia la tangente hiperbólica.

En la Tabla 3 del apéndice del capítulo se muestran los resultados del método 1 con función de transferencia tangente hiperbólica, obtenidos para cada uno de los 7 problemas de clasificación utilizados.

Comparando los resultados obtenidos para el método 1 con función de transferencia sigmoide (Tabla 2) y con tangente hiperbólica (Tabla 3), podemos observar de que el uso de la función de transferencia tangente hiperbólica mejora en gran medida tanto la capacidad de generalización (mejor en GL, HE, M1, M2 y



M3, muy parecido en el resto), como la velocidad de convergencia, en GL, M1, M2 y M3 (en el resto de problemas parecido), siendo casi iguales los resultados de la probabilidad de éxito en la convergencia.

Por esta razón, decidimos utilizar la tangente hiperbólica en los 13 métodos. Para ello, adaptamos los datos de cada problema para que estuvieran normalizados en el intervalo  $[-1,1]$ . Para algunos de los métodos (que usaban la sigmoide) tuvimos también que modificar algunas ecuaciones. En el apartado 6.2 del capítulo se indicaron los cambios realizados en la descripción de cada método.

Se aplicaron finalmente los 13 métodos a 12 problemas diferentes de clasificación del repositorio UCI.

Las bases de datos usadas fueron: *Balance Scale Weight & Distance (BL)*, *Cylinder Bands (BN)*, *BUPA Liver Disorders (BU)*, *Credit Approval (CR)*, *Display1 (D1)*, *Glass Identification (GL)*, *Heart Disease (HE)*, *Imagen Segmentation (IM)*, *The Monk's Problems (M1, M2)*, *Pima Indians Diabetes (PI)*, *1984 U.S. Congressional Voting Records (VO)*.

Para cada método y para cada problema hemos realizado 30 particiones diferentes de los datos entre entrenamiento, *cross-validation* y test.

Igual que antes, aplicamos los diferentes métodos y obtuvimos los diferentes conjuntos de pesos iniciales para cada uno de ellos.

En la aplicación de los métodos 10 y 11 se obtiene además de la inicialización de pesos, el número de unidades ocultas que se debe utilizar en las redes que entrenaremos.

Los resultados obtenidos referentes al número de unidades ocultas para el método 10 (el que utiliza prototipos), se pueden encontrar en la Tabla 4 del apéndice del capítulo. Allí se muestra, para cada problema, el valor máximo y mínimo del número de unidades ocultas utilizado, el valor medio del número de unidades ocultas y la desviación estándar en el número de unidades ocultas. Se puede observar que el número medio de unidades ocultas es exageradamente grande en algunos problemas, como por ejemplo IM y PI y en general demasiado alto en todos los demás. Pensamos pues, que este método no suministra una buena elección del número de unidades ocultas.

Por otro lado, para el método 11 se obtuvo que el número medio de unidades ocultas dependía de los parámetros *KNN* (número de *k*-vecinos) y *ACEPTAR* (umbral para rechazar patrones). Se obtuvieron diferentes valores dependiendo de la elección particular de ellos. Como dijimos en la metodología

#### 6.4. DESARROLLO EXPERIMENTAL

utilizamos 12 combinaciones diferentes para los parámetros. En la Tabla 5 del apéndice del capítulo se encuentran los valores obtenidos para una de estas combinaciones (combinación denotada como  $a$ ). Hay algunos problemas en los que el número de unidades ocultas es un poco alto, pero en otros casos es aceptable.

Una vez aplicados los diferentes métodos y obtenidos los diferentes conjuntos de pesos iniciales para cada uno de ellos, pasamos a la fase de entrenamiento.

Para cada método, con cada conjunto de pesos inicializados y las particiones realizadas entrenamos finalmente 30 redes. Igual que antes, para aquellas redes que convergían, obtuvimos el punto de convergencia utilizando cros-validación y después realizamos el test calculando el porcentaje de correcta clasificación para cada red.

De nuevo, calculamos la media de los resultados obteniendo los valores de *ITER* (iteración media de convergencia) y *POR* (eficiencia media en el test) para las redes convergidas.

En la Tabla 6 del apéndice del capítulo se muestran los resultados para los 13 métodos (filas) utilizando en el entrenamiento la función de transferencia tangente hiperbólica, y para los 12 problemas de clasificación (columnas). Para cada problema tenemos de nuevo tres columnas: *POR* (valor medio de los porcentajes de correcta clasificación), *N* (número de redes que no convergen), *ITER* (valor medio de la época de convergencia).

Para algunos de los métodos, como dijimos en la metodología, tuvimos que seleccionar algunos parámetros por lo que el resultado final podía depender de los valores escogidos. Por ello, aplicamos cada uno de estos métodos con diferentes valores de los parámetros, obteniendo diferente inicialización de los pesos. Después entrenamos 30 redes para cada una de las inicializaciones de pesos y las 30 particiones de datos y obtuvimos, finalmente, diferentes resultados según el valor utilizado para los parámetros.

Por ejemplo, para el método 3 escogimos un valor del momento  $\alpha$  fijo de 0.5 y ocho parejas de valores de los parámetros ( $l_{te}, \eta$ ). En la Tabla 6 se muestran los resultados del método para las diferentes combinaciones, con el nombre del método seguido de la letra que corresponde a la combinación de los parámetros tal y como describimos en el apartado de metodología.

Comparemos ahora los diferentes métodos respecto al primero, que representaba la inicialización clásica.

En primer lugar se puede observar que los resultados de la Tabla 6 son mejores que los de la Tabla 2 (menos en VO que son parecidos), por lo cual la aplicación de la tangente hiperbólica ha mejorado considerablemente los resultados de todos los problemas y métodos.

De nuevo observamos que el método 3 no funciona ya que el número de redes que no convergen es inaceptable.

El método 2, en general, presenta una capacidad de generalización ligeramente mejor que la del método 1. Por ejemplo, si miramos en la Tabla 6, en los problemas BU, D1, HE, M1, M2, PI y VO, la capacidad de generalización es ligeramente mejor que en el método 1, en BL, BN, CR es muy parecida y en GL, IM ligeramente peor.

El número de redes convergidas para el método 2 es también mayor que en el método 1, y la velocidad de convergencia unas veces es mejor y otras peor, no obtenemos un resultado claro al respecto.

El método 4 proporciona una capacidad de generalización mayor que la del método 1 y los resultados de la probabilidad de convergencia también son mejores. La velocidad de convergencia es similar o ligeramente inferior que en el método 1.

La capacidad de generalización para el método 5 (para los mejores valores de los parámetros) es claramente mejor que la del método 1. La probabilidad de convergencia se puede considerar muy similar, unas veces peor y otras mejor y la velocidad de convergencia es inferior que la obtenida en el método 1.

Los resultados del método 6 son en general peores que los del método 1. El método 6 tiene una capacidad de generalización inferior al método 1 en casi todos los problemas, la probabilidad de convergencia es también ligeramente inferior y la velocidad de convergencia es claramente peor.

En el método 7 se puede observar que para las bases de datos BN, BU, CR, D1, M1 y M2, la capacidad de generalización es mejor que la del método 1. Por el contrario, es peor para los problemas BL, GL, HE y IM e igual para PI y VO.

La probabilidad de convergencia del método 7 es muy similar a la obtenida en el método 1 en todos los problemas, sólo para BN resulta mejor y para BU, GL y IM un poco peor.

#### 6.4. DESARROLLO EXPERIMENTAL

Finalmente la velocidad de convergencia del método 7 presenta una alta variabilidad, es peor que la del método 1 para los problemas BL, BU, GL, IM y M2, y mejor en los problemas BN, CR, D1, M1 y PI e igual en HE y VO.

Podemos concluir que el método 7 es muy similar al método 1 si consideramos todos los aspectos.

El método 8a, denominado *Forward Estimation algorithm* funciona peor que el método 1 en todos los problemas y aspectos considerados.

El método 8b, denominado *Recurrent Estimation algorithm* no fue aplicable en ninguno de los problemas debido a que MATLAB encontró matrices singulares en el cálculo del rango de algunas matrices.

El método 9 también resultó peor que el método 1 en todos los problemas.

El método 10 funciona en general mal. La capacidad de generalización es, en general, peor que la del método 1 en todos los problemas, aunque en algunos casos tiene un valor similar. Se puede observar que existe una alta dependencia de los resultados con los valores escogidos para los parámetros que utiliza el método (12 combinaciones diferentes), y para el problema particular al que se aplica.

Así, por ejemplo, para el problema BN, CR, HE y VO se obtienen valores de la capacidad de generalización del método 10 (para ciertas combinaciones de los valores de los parámetros) ligeramente peores que los que proporciona el método 1. Por ejemplo, para CR los resultados de la capacidad de generalización que denotamos como  $f, h, i, j$  son similares a los del método 1 (aunque peores), y los que denotamos  $g, k$  son casi iguales. En el resto de casos es peor al resultado proporcionado por el método 1.

En otros problemas como GL, IM el valor de la capacidad de generalización obtenida por el método 10 no es aceptable, se podría decir que las redes no han convergido. En estos casos se obtuvo además que el número de redes que no convergieron era alto en comparación con el resultado del método 1.

Para algunos problemas como BN existen combinaciones de los parámetros en el método 10,  $b, e, f, i$ , en los que el éxito en la convergencia es similar al método 1. En estos casos la velocidad de convergencia es mejor que la del método 1 aunque la capacidad de generalización es un poco peor. Algo parecido ocurre con los problemas CR y HE.

En definitiva, los resultados del método 10 presentan alta dependencia con el problema particular y con los valores de los parámetros que escojamos, aunque ninguna de nuestras elecciones parece especialmente mejor.

El método 11 funciona también peor que el método 1 y en algunos aspectos de forma parecida al método 10. Aquí también se escogen ciertos parámetros y hemos probado con 12 combinaciones diferentes de los mismos.

Este método se caracteriza por una baja convergencia y alta variabilidad en la capacidad de generalización y la velocidad de convergencia, en función de la elección de los parámetros. También varían los resultados con los problemas.

Por ejemplo en el problema IM converge solo una red en 7 de los 12 casos y en el resto no converge ninguna red.

Para los problemas BL, BN, CR y HE también se observa una variabilidad ocasionada por la elección de los parámetros.

Por otra parte, también existen problemas en los que la capacidad de generalización que se obtiene es inaceptable y se puede decir que las redes no han convergido, por ejemplo, D1, GL, IM y M1.

Solo en el caso del problema VO los resultados del método 11 son similares a los del método 1, convergen todas las redes y la capacidad de generalización y la velocidad de convergencia son similares en ambos, aunque un poco peores en el método 11.

Los métodos 12 y 13 presentan una capacidad de generalización inaceptable en algunos problemas como D1, GL, IM, M1 y M2. Podemos decir que en estos casos las redes no han convergido.

En el método 12 podemos observar que el comportamiento desde el punto de vista de la capacidad de generalización es similar al del método 10. La capacidad de generalización del método 12 es peor que la del método 1, el número de redes convergidas es similar y la velocidad de convergencia es en general peor.

En algunos problemas como PI el comportamiento del método 12 es similar al del método 1.

En el método 13 la convergencia es en general un poco peor que la del método 1, la capacidad de generalización peor y la velocidad de convergencia peor (salvo en los problemas que indicamos antes en los que no convergía). De nuevo para el problema VO el funcionamiento es parecido al del método 1.

En general los métodos 2, 4 y 5 parecen resultar mejores que el resto de métodos incluyendo la inicialización clásica del método 1. Por ello, pensamos que es interesante su aplicación cuando queramos obtener un buen resultado.

#### 6.4. DESARROLLO EXPERIMENTAL

El resto de métodos o funcionan peor o presentan una alta variabilidad, dependiendo del problema concreto al que se apliquen o incluso, como hemos visto, de la elección de los parámetros.

Podemos escoger pues, uno de los métodos que proporcionan mejores resultados y que son los métodos 2, 4 y 5.

Los resultados muestran que los métodos en los cuales hay que escoger varios parámetros, no son en general adecuados para resolver el problema de inicialización.

Para el método 5 la eficiencia varía según el valor de los parámetros y la elección de los mismos presenta una complejidad adicional.

Por ello, pensamos que el método que puede resultar más útil en casi todos los problemas es el método 4, que presenta la gran ventaja de su sencillez.

## 6.5. Conclusiones

Hemos realizado un análisis de la utilidad de los diferentes métodos de inicialización de pesos frente a la inicialización clásica, es decir, inicialización aleatoria dentro de un cierto intervalo.

El estudio ha sido realizado desde tres puntos de vista diferentes, capacidad de generalización, probabilidad en la convergencia y velocidad de convergencia.

Con los resultados obtenidos podemos concluir que algunos de los métodos presentan ventajas con respecto al método clásico, sobre todo en el aspecto de mejorar la capacidad de generalización de las redes.

Por ello podemos recomendar el uso de alguno de estos métodos para inicializar los pesos de las redes a la hora de resolver un problema concreto de clasificación.

Aunque parece que el método 5 ofrece mejores resultados presenta la dificultad de tener que escoger ciertos parámetros. Dichos parámetros influyen en su eficiencia, aunque la sensibilidad al valor de los parámetros no fue muy alta. Por ello, quizás sea preferible recomendar el uso del método 4 debido a su simplicidad y buenos resultados para todos los problemas estudiados.

Los resultados obtenidos en la investigación realizada en este capítulo han sido publicados en varios congresos internacionales. En la bibliografía mostramos sus referencias completas que son [19], [27] y [28].

## 6.6. Bibliografía del Capítulo

- [1] Rumelhart, D.E., Hinton, G.E., Williams, R.J., “Learning internal representations by error backpropagation”. *Parallel Distributed Processing*. Cambridge MA: MIT Press. Chapter 8, pp. 318-362, 1986.
- [2] Li, G., Alnuweiri, H., Wu, Y., “Acceleration of Backpropagations through Initial Weight Pre-Training with Delta Rule”. Proceedings of the IEEE International Conference on Neural Networks, vol. 1, pp. 580-585, 1993.
- [3] Fahlman, S.E., “An experimental study of learning speed in Back-Propagation networks”. CMU Technical Report, CMU-CS-162, Sept. 1988.
- [4] Jacobs R.A., “Increased rates of convergence through learning rate adaptation”. *Neural Networks*, vol. 1, 1988.
- [5] Rezqui A. et al., “The effect of the slope of the activation function on the backpropagation algorithm”. Proceedings of the International Joint Conference on Neural Network, vol. 1, pp. 707-710, January 1990.
- [6] Atiya, A., Ji, C., “How initial conditions affect generalization performance in large networks”. *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 448-451, 1997.
- [7] Rojas, R., “Optimal weight initialization for neural networks”, Proceedings of the International Conference on Artificial Neural Networks (ICANN'94), vol. 1, pp. 577-580, 1994.
- [8] Drago, G.P., Ridella, S., “Statistically Controlled Activation Weight Initialization (SCAWI)”. *IEEE Transactions on Neural Networks*, vol. 3, no. 4, pp. 627-631, 1992.
- [9] Wessels, L.F.A., Barnard, E., “Avoiding False Local Minima by Proper Initialization of Connections”. *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 899-905, 1992.
- [10] Shimodaira H., “A Weight Value Initialization Method for Improved Learning Performance of the Back Propagation Algorithm in Neural Networks”. Proceedings of the 6th International Conference on Tools with Artificial Intelligence, pp. 672-675, 1994.
- [11] Rossi, F., Gegout, C., “Geometrical Initialization, Parametrization and Control of Multilayer Perceptrons: Application to Function Approximation”, Proceedings of the IEEE International Conference on Neural Networks (ICNN'94), vol. 1, pp. 546-550, 1994.
- [12] Ho-Sub Yoon, Chang-Seok Bae, Byung-Woo Min, “Neural Networks using Modified Initial Connection Strengths by the Importance of Feature Elements”. *International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 458-461, 1995.
- [13] Chih-Liang Chen, Nutter, Roy S., “Improving the Training Speed of Three-layer Feedforward Neural Nets by Optimal Estimation of the Initial Weights”. *International Joint Conference on Neural Networks*, vol. 3, part. 1, pp. 2063-2068, 1991.
- [14] Thimm, G., Fiesler, E., “High-Order and multilayer perceptron initialization”. *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 349-359, 1997.
- [15] Yu X., Chen G., Cheng S.: “Dynamic Learning Rate Optimization of the Backpropagation Algorithm”. *IEEE Transaction on Neural Networks*, vol. 6, no. 3, pp. 669-677, 1995.



- [16] Kim, Y.K., Ra, J.B., “Weight Value Initialization for Improving Training Speed in the Backpropagation Network”. Proceedings of International Joint Conference on Neural Networks, vol. 3, pp. 2396-2401, 1991.
- [17] Palubinskas, G., “Data-driven Weight Initialization of Back-propagation for Pattern Recognition”. Proceedings of the International Conference on Artificial Neural Networks (ICANN'94), vol. 2, pp. 851-854, 1994.
- [18] Utans, J., Moody, J., et al., “Input variable selection for neural networks: Application to predicting the U.S. business cycle”. Proceedings of IEEE/IAFE Computational Intelligence for Financial Engineering, pp. 118-122, 1995.
- [19] Fernández, M., Hernández, C., “Input selection by Multilayer Feedforward trained networks”. Proceedings of the 1999 IEEE International Joint Conference on Neural Networks, pp. 1834-1839, 1999
- [20] Drago, G.P, Ridella S., “SCAWI: an algorithm for weight initialization of a sigmoidal neural network”. Proceedings. of the International Conference on Artificial Neural Networks (ICANN'92), vol. 2, pp. 983-986, 1992.
- [21] Nakayama K., Hirano A., Ido I., “A multilayer neural network with nonlinear inputs and trainable activation functions: structure and simultaneous learning algorithms”. Proceedings of the International Joint Conference on Neural Networks, 1999.
- [22] Denooux, T., Lengellé, R., “Initializing Back Propagation Networks With Prototypes”. Neural Networks, vol. 6, pp. 351-363, 1993.
- [23] Reilly, D.L., Cooper, L.N., Elbaum, C., “A neural Model for Category Learning”. Biological Cybernetics, vol. 45, pp. 35-41, 1982.
- [24] Tarek Kaylani, Sushil Dasgupta, “Weight initialization of MLP classifiers using boundary-preserving patterns”. 1994.
- [25] Smyth, S.G., “Designing Multilayer Perceptrons from Nearest Neighbor Systems”. IEEE Transactions on Neural Networks, vol. 3, no. 2, pp. 329-333. 1992.
- [26] Karayiannis, N.B., “Accelerating the Training of Feedforward Neural Networks Using Generalized Hebbian Rules for Initializing the Internal Representations”. IEEE Transactions on Neural Networks, vol. 7, no. 2, pp. 419-426. March 1996.
- [27] Fernández, M., Hernández, C., “A comparison among weight inicialization methods for Multilayer Feedforward networks”. Proceedings of the 2000 IEEE International Joint Conference on Neural Networks, vol. 4, pp. 543-548, 2000.
- [28] Fernández, M., Hernández, C., “Weight initialization methods for multilayer feedforward”. European Symposium on Artificial Neural Networks (ESANN'2001). En prensa. Bruges (Belgium).

## 6.6. BIBLIOGRAFÍA DEL CAPÍTULO

## 6.7. Apéndice del Capítulo

*Tabla 1. Relación de particiones realizadas con los datos.*

PROBLEMAS	TOTAL DATOS	ENTRENAMIENTO	CROSSVALIDACIÓN	TEST
BL	625	395	105	125
BN	277	177	45	55
BU	345	220	55	70
CR	653	418	105	130
D1	2900	1830	470	600
GL	214	129	35	50
HE	297	190	48	59
IM	2311	1476	370	465
M1	432	282	70	80
M2	432	282	70	80
M3	432	282	70	80
PI	768	488	125	155
VO	435	285	70	80
WD	569	374	90	105

*Tabla 2. Resultados de Inicialización de pesos con Sigmoide.*

	GL			HE			M1		
	POR	N	ITER	POR	N	ITER	POR	N	ITER
Método 1	78.9±1.1	0	1294±134	80.6±0.9	0	50±12	73.6±1.1	0	522±124
Método 2	79.6±1.1	0	1115±133	80.7±1.0	0	49±13	74.9±0.9	0	510±75
Método 3b	-----	20	-----	52.54	19	2182	-----	20	-----
Método 3c	-----	20	-----	-----	20	-----	-----	20	-----
Método 3d	-----	20	-----	54.24	19	3000	-----	20	-----
Método 3f	-----	20	-----	-----	20	-----	-----	20	-----
Método 3g	-----	20	-----	-----	20	-----	-----	20	-----
Método 3h	-----	20	-----	-----	20	-----	-----	20	-----
Método 4	78.8±1.1	0	1192±118	80.6±1.0	0	54±13	73.1±1.1	0	697±130
Método 5a	-----	20	-----	-----	20	-----	-----	20	-----
Método 5b	-----	20	-----	47±17	17	4177±296	50.5±0.9	0	732±100
Método 5c	-----	20	-----	46.6±0.7	3	1880±214	63.1±1.7	0	672±106
Método 5d	-----	20	-----	-----	20	-----	-----	20	-----
Método 5e	-----	20	-----	83.05	19	1080	53.9±1.7	0	1169±156
Método 5f	13±1	18	3246±398	45.9±0.6	3	2219±310	69±2	0	626±134
Método 5g	-----	20	-----	-----	20	-----	-----	20	-----
Método 5h	-----	20	-----	63±9	16	3432±580	59.4±1.6	0	1062±146
Método 5i	3.5±1.3	16	3667±84	50±3	6	2157±344	75.2±1.4	0	668±131
Método 6	77.7±1.1	0	1369±170	80.4±0.9	0	109±30	74±1	0	140±16

6.7. APÉNDICE DEL CAPÍTULO

Tabla 2. Resultados de Inicialización de pesos con Sigmoide (continuación).

	M2			M3		
	POR	N	ITER	POR	N	ITER
Método 1	66.3±0.9	1	806±82	84.1±1.1	0	476±81
Método 2	66.1±0.9	0	899±187	83.9±1.2	0	724±132
Método 3b	----	20	----	53.75	19	1330
Método 3c	----	20	----	----	20	----
Método 3d	----	20	----	53±4	16	324±128
Método 3f	----	20	----	----	20	----
Método 3g	----	20	----	----	20	----
Método 3h	----	20	----	----	20	----
Método 4	----	20	----	83.8±1.1	0	394±106
Método 5a	----	20	----	67.4±1.1	0	374±33
Método 5b	----	20	----	82.2±1.4	0	157±22
Método 5c	61±4	13	2259±248	88.2±1.1	0	400±69
Método 5d	----	20	----	68.4±1.5	0	335±34
Método 5e	----	20	----	83.9±1.4	0	173±39
Método 5f	63±6	12	3730±495	88.5±1.0	0	480±102
Método 5g	----	20	----	69.9±1.1	0	398±60
Método 5h	----	20	----	85.7±1.5	0	346±86
Método 5i	66±8	16	4122±921	88.0±0.8	0	227±56
Método 6	66.9±0.5	0	1490±317	84.4±1.2	0	286±45

	VO			WD		
	POR	N	ITER	POR	N	ITER
Método 1	95.9±0.6	0	43±16	96.9±0.3	0	110±25
Método 2	95.8±0.6	0	22±3	96.8±0.3	0	115±25
Método 3b	----	20	----	72±9	17	168±36
Método 3c	----	20	----	63.8	19	159
Método 3d	----	20	----	61	19	2301
Método 3f	----	20	----	----	20	----
Método 3g	----	20	----	61	19	3
Método 3h	----	20	----	35.24	19	930
Método 4	95.9±0.5	1	27±4	96.9±0.3	0	102±24
Método 5a	----	20	----	----	20	----
Método 5b	----	20	----	----	20	----
Método 5c	----	20	----	64.4±0.5	8	1986±115
Método 5d	----	20	----	----	20	----
Método 5e	----	20	----	----	20	----
Método 5f	----	20	----	64.4±0.5	4	2064±102
Método 5g	----	20	----	----	20	----
Método 5h	----	20	----	----	20	----
Método 5i	70±2	7	2173±96	64.5±0.5	7	2090±102
Método 6	95.4±0.6	0	97±30	96.7±0.3	0	96±19

**Tabla 3. Resultados de Inicialización de pesos del Método 1 con TANH.**

	GL	HE	M1	M2	M3	VO	WD
<b>POR</b>	90.4±1.1	81.2±1.1	97.6±0.8	72.5±1.2	93.9±1.2	94.3±0.5	96.1±0.4
<b>N</b>	0	0	0	0	2	0	0
<b>ITER</b>	111±12	220±80	74±14	500±200	444±135	120±70	472±99

**Tabla 4. Número de unidades ocultas obtenido por el Método 10.**

	Nhidden Medio	Desviación Nhidden	Nhidden Mínimo	Nhidden Maximo
<b>BL</b>	65.6	25.30	28	108
<b>BN</b>	48.1	19.05	19	81
<b>BU</b>	67.2	32.25	27	121
<b>CR</b>	70.1	32.33	26	117
<b>D1</b>	69.3	5.96	60	80
<b>GL</b>	20.3	7.40	10	36
<b>HE</b>	39.7	14.75	18	67
<b>IM</b>	104.7	31.49	65	161
<b>M1</b>	88.1	23.22	35	134
<b>M2</b>	55.3	25.78	28	117
<b>PI</b>	124.8	51.85	44	214
<b>VO</b>	26.5	12.22	12	56

**Tabla 5. Número de unidades ocultas obtenido por el Método 11.**

	Nhidden Medio	Desviación Nhidden	Nhidden Mínimo	Nhidden Maximo
<b>BL</b>	32.6	4.41	25	39
<b>BN</b>	10.5	1.94	7	15
<b>BU</b>	32.3	6.76	22	45
<b>CR</b>	13.5	2.54	9	20
<b>D1</b>	7	0.26	6	8
<b>GL</b>	12	2.63	7	17
<b>HE</b>	10.8	2.25	5	15
<b>IM</b>	39.4	7.48	27	51
<b>M1</b>	51.8	12.46	30	69
<b>M2</b>	35.4	7.19	19	48
<b>PI</b>	29.4	3.00	24	36
<b>VO</b>	13.1	4.37	4	21

6.7. APÉNDICE DEL CAPÍTULO

Tabla 6. Resultados de Inicialización de pesos con TANH.

	BL			BN			BU		
	POR	N	ITER	POR	N	ITER	POR	N	ITER
Método 1	91.8±0.4	1	350±80	66.8±1.3	11	800±300	59.4±1.0	3	1300±400
Método 2	91.0±0.5	0	680±190	66.9±1.6	8	720±180	60.0±0.9	1	1300±300
Método 3a	-----	30	-----	64.8±1.1	19	1±0	58.6±1.4	23	1±0
Método 3b	-----	30	-----	64.8±1.1	19	1±0	58.6±1.4	23	1±0
Método 3c	-----	30	-----	64.8±1.1	19	1±0	58.6±1.4	23	1±0
Método 3d	-----	30	-----	64.8±1.1	19	1±0	58.6±1.4	23	1±0
Método 3e	-----	30	-----	64.8±1.1	19	1±0	58.6±1.4	23	1±0
Método 3f	-----	30	-----	64.8±1.1	19	1±0	58.6±1.4	23	1±0
Método 3g	-----	30	-----	64.8±1.1	19	1±0	58.6±1.4	23	1±0
Método 3h	-----	30	-----	64.8±1.1	19	1±0	58.6±1.4	23	1±0
Método 4	91.1±0.4	0	550±120	68.3±1.7	8	700±200	60.8±1.3	3	2800±600
Método 5a	91.2±0.5	0	620±130	65.5±1.2	5	1600±400	60.8±1.5	0	1400±400
Método 5b	91.4±0.5	0	540±130	67.4±1.4	8	900±300	61.9±1.3	5	1500±500
Método 5c	91.3±0.4	0	1000±200	67.1±1.2	5	600±200	60.4±1.6	3	500±160
Método 5d	91.1±0.5	0	1000±200	68.3±1.2	4	1200±300	61.0±1.3	2	1700±500
Método 5e	91.0±0.4	0	720±160	68.5±1.4	7	1000±300	62.1±1.4	2	1700±400
Método 5f	91.7±0.4	0	900±200	66.4±1.1	6	1100±300	61.4±1.7	3	1100±300
Método 5g	91.5±0.5	0	1200±300	67.0±1.2	6	470±160	62.4±1.1	1	1300±300
Método 5h	91.5±0.4	0	1100±200	67.3±1.1	7	680±180	63.1±1.4	1	1500±400
Método 5i	91.3±0.4	0	800±200	59±5	6	1200±200	60.8±1.4	3	1400±300
Método 6	91.4±0.5	0	800±200	65.3±1.3	5	800±200	61.3±1.3	4	2100±500
Método 7	90.6±0.4	1	1100±200	67.7±1.2	5	600±200	60.9±0.9	4	1600±400
Método 8a	34±3	19	1±0	61±3	18	7±6	52±2	11	1±0
Método 8b	-----	--	-----	-----	--	-----	-----	--	-----
Método 9	80±2	0	460±130	63.6±1.7	6	690±190	54.3±1.5	3	2000±300
Método 10a	88.7±0.5	0	1200±300	65.2±1.0	17	900±400	53.3±1.6	14	900±400
Método 10b	85.6±1.0	0	600±150	63.8±1.3	12	40±20	51±3	24	500±400
Método 10c	82±2	22	1900±600	61±3	27	900±700	56.3±1.6	12	700±400
Método 10d	79.22±1.2	27	1300±50	---	30	---	50.6±1.8	14	400±400
Método 10e	80.8	29	2367	66.0±1.2	14	450±190	54.7±1.3	16	1200±600
Método 10f	84.6±1.0	0	530±80	63.9±1.2	10	300±200	45.7	29	117
Método 10g	82±4	24	2600±900	64±4	27	2900±1200	54.4±1.4	8	1600±500
Método 10h	80.8	29	2367	---	30	---	53.7±1.9	18	190±100
Método 10i	86.5±0.7	1	450±150	64.1±1.3	14	500±200	56.0±0.6	19	3200±800
Método 10j	83.2±1.0	0	490±190	65.6±1.1	17	1100±300	---	30	---
Método 10k	81±3	20	1500±500	63±3	26	2000±800	55.5±1.1	9	700±300
Método 10l	78.4	29	1229	---	30	---	52.9±1.8	17	320120±
Método 11a	50±2	15	11±9	60±2	6	600±200	52±2	11	1±0
Método 11b	59±6	25	3.8±1.0	58±3	5	420±130	52±2	11	1±0
Método 11c	50±2	15	11±10	60±3	6	600±200	52±2	11	1±0
Método 11d	59±6	25	3.8±1.1	58±3	5	420±130	52±2	11	1±0
Método 11e	54±4	20	1±0	64.2±1.6	5	420±140	52±2	11	1±0
Método 11f	51±4	17	2.5±1.1	62.0±1.8	3	1000±200	52±2	11	1±0
Método 11g	48±2	18	7±6	64±2	5	1200±300	52±2	11	1±0
Método 11h	71.2±0.8	28	360±190	61±3	1	600±200	52±2	11	1±0
Método 11i	45±10	25	1±0	58±3	3	700±200	52±2	11	1±0
Método 11j	52±3	18	150±120	60±3	8	800±300	52±2	11	1±0
Método 11k	56±3	20	60±40	64±2	6	590±190	52±2	11	1±0
Método 11l	75±4	24	1600±800	60±3	8	800±200	52±2	11	1±0
Método 12	79±2	0	530±170	57±2	4	1400±300	54.6±1.7	2	2300±500
Método 13a	80±2	0	810±190	58±3	6	1400±300	57.3±1.2	2	1500±400
Método 13b	81±2	0	1200±200	64.4±1.0	2	1100±300	57.3±1.2	4	800±200
Método 13c	76±3	0	900±200	58±3	8	1800±300	56.8±1.3	4	1700±400

Tabla 6. Resultados de Inicialización de pesos con TANH (continuación).

	CR			D1			GL		
	POR	N	ITER	POR	N	ITER	POR	N	ITER
Método 1	85.5±0.5	0	1300±400	69.3±0.7	3	720±140	90.4±1.1	3	111±12
Método 2	85.6±0.5	0	1700±500	71.2±0.6	3	670±120	88.9±1.1	2	150±40
Método 3a	-----	30	-----	-----	30	-----	-----	30	-----
Método 3b	-----	30	-----	-----	30	-----	-----	30	-----
Método 3c	-----	30	-----	-----	30	-----	-----	30	-----
Método 3d	-----	30	-----	-----	30	-----	-----	30	-----
Método 3e	-----	30	-----	-----	30	-----	-----	30	-----
Método 3f	-----	30	-----	-----	30	-----	-----	30	-----
Método 3g	-----	30	-----	-----	30	-----	-----	30	-----
Método 3h	-----	30	-----	-----	30	-----	-----	30	-----
Método 4	85.7±0.5	0	1000±400	71.1±0.5	3	1090±170	90.7±0.9	2	160±40
Método 5a	85.5±0.5	0	1200±300	71.4±0.5	4	870±180	90.2±1.4	2	190±40
Método 5b	86.0±0.5	0	1900±500	70.6±0.6	3	850±160	90.2±1.2	3	121±14
Método 5c	85.5±0.5	0	1400±400	-----	30	-----	89.1±1.2	2	130±20
Método 5d	85.8±0.4	0	1300±400	70.7±0.5	3	790±130	91.9±0.8	5	220±40
Método 5e	85.4±0.4	0	1400±400	69.3±0.6	4	570±120	90.8±0.9	5	91±11
Método 5f	86.1±0.4	1	2500±500	71.2±0.5	3	760±150	87.6±0.9	2	200±60
Método 5g	86.1±0.5	0	1400±400	70.6±0.5	3	650±150	90.0±0.9	2	200±60
Método 5h	-----	30	-----	-----	30	-----	88.2±1.6	5	160±40
Método 5i	85.9±0.5	1	1100±300	71.0±0.5	4	880±140	89.7±1.0	2	139±19
Método 6	86.1±0.5	0	1700±400	64.8±1.5	5	920±170	85.7±1.1	7	300±100
Método 7	86.0±0.5	0	900±300	70.1±0.6	3	590±140	88.2±1.1	5	230±60
Método 8a	51.9±1.5	1	200±200	---	30	---	24	29	2029
Método 8b	-----	--	-----	-----	--	-----	-----	--	-----
Método 9	73±3	0	1300±400	47.8±1.6	0	900±150	47±3	5	600±200
Método 10a	75±2	0	1200±400	51.9±1.4	7	500±150	36±4	11	400±200
Método 10b	78±3	17	1300±800	59.7±1.2	13	270±180	47±3	15	1100±300
Método 10c	79±3	18	2700±800	51±3	24	550±90	40±3	7	650±190
Método 10d	73.0	29	594	64.4±1.8	20	800±300	50±5	18	900±200
Método 10e	76.2±1.8	0	900±300	60.8±1.2	3	630±130	34±4	14	900±300
Método 10f	81.1±1.8	1	600±200	64.0±1.2	3	350±120	48±4	19	1000±300
Método 10g	85.5±1.0	20	1900±600	63.0±1.6	13	1000±200	53±4	6	660±170
Método 10h	83.8	29	12	62.3±1.5	19	700±300	58±4	17	800±300
Método 10i	80.8±1.7	0	1000±400	63.7±1.5	20	1300±300	37±3	15	1700±300
Método 10j	82.5±1.6	1	1200±300	67.9±0.9	4	630±140	33±3	19	1900±400
Método 10k	85.1±0.9	20	1600±600	60.9±1.3	17	13±7	61±3	9	760±120
Método 10l	---	30	---	63.2±1.7	20	560±190	55±4	18	1800±400
Método 11a	77±6	26	400±200	40.9±1.8	0	600±120	38±3	25	1±0
Método 11b	77±3	15	1700±600	40.9±1.8	0	600±120	33±5	23	300±200
Método 11c	77±6	26	400±200	41.8±1.7	0	960±150	38±3	25	1±0
Método 11d	77±3	15	1700±600	40.6±1.8	0	570±120	32±6	24	400±300
Método 11e	71±4	15	2000±700	40±2	0	550±140	33±5	27	1±0
Método 11f	75±3	8	1100±400	40.6±1.9	0	450±110	44±6	27	2500±1000
Método 11g	69±3	0	1500±500	40±2	0	480±120	40±5	27	3620±190
Método 11h	70±3	0	900±300	39.6±1.6	0	660±160	45±8	23	1800±500
Método 11i	74±3	2	1500±400	38±2	0	520±140	33±4	23	900±500
Método 11j	71±3	0	410±150	38.1±1.8	0	500±130	40±7	25	900±300
Método 11k	72±3	0	1200±400	36±2	0	750±150	33±5	28	1700±1300
Método 11l	73±3	0	700±300	39±2	0	470±110	32±3	22	1900±500
Método 12	69±3	0	800±200	45.3±1.8	0	620±100	42±3	3	800±200
Método 13a	67±3	1	1900±400	40.0±1.9	0	1070±150	55±4	2	1100±300
Método 13b	72±3	8	3500±600	43.8±1.9	0	820±140	48±3	2	900±300
Método 13c	71±3	2	1700±400	39±2	0	1200±150	49±4	0	700±200

6.7. APÉNDICE DEL CAPÍTULO

Tabla 6. Resultados de Inicialización de pesos con TANH (continuación).

	HE			IM			M1		
	POR	N	ITER	POR	N	ITER	POR	N	ITER
Método 1	81.2±1.1	4	220±80	72±2	5	710±130	97.6±0.8	2	74±14
Método 2	81.7±1.1	3	320±140	70±2	4	1010±150	98.2±0.8	2	200±100
Método 3a	75.1±1.5	27	700±700	-----	30	-----	-----	30	-----
Método 3b	56±3	28	1±0	-----	30	-----	-----	30	-----
Método 3c	78±4	26	9±8	-----	30	-----	-----	30	-----
Método 3d	72±3	25	1±0	-----	30	-----	-----	30	-----
Método 3e	65±5	24	1±0	-----	30	-----	-----	30	-----
Método 3f	64±8	28	1±0	-----	30	-----	-----	30	-----
Método 3g	78.4±1.9	26	1±0	-----	30	-----	-----	30	-----
Método 3h	70±3	26	1±0	-----	30	-----	-----	30	-----
Método 4	80.6±0.8	4	430±160	74.7±1.9	5	950±170	97.3±1.1	1	68±16
Método 5a	81.0±0.9	3	430±160	70±2	9	1350±180	98.7±0.6	3	56±5
Método 5b	81.3±0.8	2	700±200	76±2	10	1150±180	96.1±1.2	1	130±50
Método 5c	80.3±0.8	3	500±200	69±3	9	1060±170	97.3±1.2	1	150±50
Método 5d	81.1±0.8	2	630±180	67±2	7	1030±180	97.0±0.8	0	220±80
Método 5e	79.8±1.4	3	600±200	73±2	10	1420±140	98.1±0.8	1	80±30
Método 5f	80.7±1.1	3	390±180	75±2	13	970±190	97.4±0.9	1	80±30
Método 5g	80.0±0.9	2	600±200	75±2	7	1100±180	98.3±0.7	3	110±40
Método 5h	81.5±0.9	2	400±140	73.7±1.7	13	1300±200	-----	30	-----
Método 5i	81.7±1.1	5	290±130	72±2	10	1100±180	97.5±0.8	1	170±70
Método 6	80.9±0.8	3	500±200	59±3	11	1040±180	96.6±1.0	1	105±18
Método 7	80.8±1.0	4	200±100	70±2	8	1090±160	99.0±0.7	2	54±13
Método 8a	65±4	23	600±400	---	30	---	50.7±0.8	4	190±130
Método 8b	-----	--	-----	-----	--	-----	-----	--	-----
Método 9	73±2	0	350±140	41±3	4	1080±160	64±3	0	310±9
Método 10a	79.0±1.1	3	90±50	28±3	16	600±200	---	30	---
Método 10b	80.7±1.1	12	500±200	18.3±1.3	12	280±120	---	30	---
Método 10c	76.7±1.9	23	900±500	16.2±1.8	25	580±170	---	30	---
Método 10d	---	30	---	21.3±1.4	19	80±50	---	30	---
Método 10e	78.3±1.2	1	280±150	18.0±1.4	12	160±70	63±12	28	200±200
Método 10f	79.7±1.2	2	800±300	14.8±0.4	17	400±200	---	30	---
Método 10g	76.7±1.7	26	370±170	25±3	25	170±50	---	30	---
Método 10h	---	30	---	30.7±1.7	9	170±50	---	30	---
Método 10i	80.6±0.8	3	340±190	16.4±0.5	13	330±120	65±6	27	800±500
Método 10j	79.4±1.2	5	700±200	14.6±0.2	18	280±190	---	30	---
Método 10k	75.0±1.6	26	2200±1100	20±2	19	300±100	58.8	29	1303
Método 10l	---	30	---	22±2	18	330±100	---	30	---
Método 11a	76.7±1.7	4	370±150	14.4	29	1	69±8	25	2400±600
Método 11b	75.6±1.9	4	530±170	15.7	29	1	70±5	19	2500±300
Método 11c	76.7±1.7	4	370±150	14.4	29	1	69±8	25	2400±600
Método 11d	75.6±1.9	4	530±170	15.7	29	1	70±5	19	2500±300
Método 11e	77.7±1.7	2	330±170	---	30	---	63±5	26	2200±700
Método 11f	78.6±1.5	2	900±300	---	30	---	68±3	11	400±200
Método 11g	75.8±1.8	3	370±160	14.0	29	1	70±7	25	400±200
Método 11h	76±2	3	490±150	---	30	---	68±5	14	530±170
Método 11i	69±3	2	900±300	13.3	29	1	82±17	28	1100±900
Método 11j	68±3	1	370±190	14.6	29	1	67±3	14	550±130
Método 11k	70±2	3	150±80	---	30	---	52.0±1.1	14	900±200
Método 11l	71±2	3	390±170	---	30	---	63±3	4	450±130
Método 12	76.7±1.5	1	260±80	37±4	12	1340±190	61±3	1	250±70
Método 13a	71±3	4	800±200	14.5±0.2	0	40±20	59±2	2	410±120
Método 13b	71±3	3	600±200	27±2	20	830±180	61±3	5	190±40
Método 13c	72±3	4	600±200	13.6±0.3	17	1770±130	60±3	1	320±60



Tabla 6. Resultados de Inicialización de pesos con TANH (continuación).

	M2			PI			VO		
	POR	N	ITER	POR	N	ITER	POR	N	ITER
Método 1	72.5±1.2	4	500±200	75.6±0.5	0	1400±500	94.3±0.5	0	120±70
Método 2	73.8±1.3	1	900±300	76.0±0.5	0	800±400	94.7±0.4	0	80±50
Método 3a	61±3	26	4±3	65.1±0.8	17	1±0	84.9±1.7	19	1±0
Método 3b	62±4	27	300±300	65.2±0.7	14	1±0	82.0±1.5	16	130±130
Método 3c	62±3	27	2±1	65.0±0.8	16	1±0	84±2	19	60±60
Método 3d	64±4	27	2000±2000	64.9±0.8	17	1±0	83±2	17	1±0
Método 3e	54.4±0.6	28	1±0	64.9±0.9	18	1±0	85.3±1.1	19	1±0
Método 3f	53.1±0.6	28	1±0	64.9±0.8	17	1±0	85.4±1.6	16	1±0
Método 3g	59±5	28	1±0	65.1±0.8	17	1±0	82.8±1.7	18	1±0
Método 3h	59±5	28	1±0	65.1±0.8	17	1±0	83.4±1.6	17	1±0
Método 4	71.9±1.0	1	700±300	74.5±0.7	0	1200±500	94.1±0.5	0	120±60
Método 5a	71.4±0.9	6	320±110	74.8±0.7	0	2800±800	95.0±0.4	0	140±70
Método 5b	72.9±1.1	5	700±200	75.6±0.7	0	1800±600	94.7±0.5	0	80±20
Método 5c	73.4±0.7	4	1100±400	75.5±0.5	0	2200±600	94.4±0.5	0	80±40
Método 5d	72.4±0.9	3	800±300	76.2±0.7	0	1200±500	94.7±0.5	0	150±50
Método 5e	73.7±1.0	1	500±200	74.5±0.6	0	1400±500	95.2±0.4	0	80±50
Método 5f	72.6±1.1	1	460±140	74.6±0.7	0	800±300	95.1±0.4	0	200±90
Método 5g	72.9±1.4	4	1000±300	75.1±0.7	0	1600±500	94.4±0.5	0	110±50
Método 5h	72.5±1.0	1	1000±300	74.9±0.6	0	1500±600	94.8±0.5	0	90±30
Método 5i	72.8±0.8	2	1300±400	75.6±0.7	0	700±200	94.8±0.4	1	120±70
Método 6	72.0±1.3	7	600±180	74.8±0.6	0	1600±500	95.0±0.3	0	270±110
Método 7	73.1±1.0	5	1000±400	75.2±0.6	0	900±300	94.5±0.4	0	130±70
Método 8a	54.9±1.5	2	1±0	64.9±0.9	18	1±0	82.0±1.6	14	50±40
Método 8b	----	--	----	----	--	----	----	--	----
Método 9	61±2	1	900±300	70.8±1.5	0	1400±500	89.0±1.9	0	190±80
Método 10a	60±8	28	11±5	67.1	23	4±4	93.2±1.0	6	290±110
Método 10b	65	29	350	---	30	---	91.3±1.8	21	600±300
Método 10c	65±3	28	450±150	68±3	20	2±2	90.1±1.6	20	600±200
Método 10d	---	30	---	66.1±1.0	18	1±0	---	30	---
Método 10e	57±5	26	1400±1300	---	30	---	92.8±0.7	11	320±110
Método 10f	62.5	29	1199	---	30	---	86±5	27	1000±500
Método 10g	---	30	---	61±4	10	8±3	80±4	21	900±200
Método 10h	---	30	---	63±3	23	2±2	---	30	---
Método 10i	49±4	26	1500±1300	---	30	---	90.8±1.2	12	500±200
Método 10j	---	30	---	---	30	---	87±3	26	1400±500
Método 10k	66±2	27	1400±600	71.3±0.3	16	1700±400	70±3	21	1100±200
Método 10l	---	30	---	---	30	---	---	30	---
Método 11a	66.5±0.5	5	1200±500	64.9±0.9	18	1±0	89±2	0	170±70
Método 11b	66.0±1.0	4	1500±400	64.9±0.9	18	1±0	91.4±1.6	0	540±150
Método 11c	66.5±0.5	5	1200±500	64.9±0.9	18	1±0	89±2	0	170±70
Método 11d	66.0±0.9	2	1400±400	64.9±0.9	18	1±0	91.4±1.6	0	540±150
Método 11e	64.5±1.6	10	900±400	64.9±0.9	18	1±0	90.3±1.4	0	390±130
Método 11f	63.1±1.7	1	1500±300	64.9±0.9	18	1±0	90.0±1.6	0	140±40
Método 11g	63.5±1.5	3	600±300	64.9±0.9	18	1±0	84±3	0	180±90
Método 11h	65.5±1.6	1	1500±400	64.9±0.9	18	1±0	86±2	0	230±110
Método 11i	64.2±1.7	7	1100±400	64.9±0.9	18	1±0	87±2	0	90±30
Método 11j	65.2±1.2	8	1900±1500	64.9±0.9	18	1±0	88±2	0	150±60
Método 11k	57±3	11	1400±400	63±2	17	140±140	90.1±1.9	0	160±60
Método 11l	60±2	1	900±300	63±2	16	40±30	86±3	1	250±100
Método 12	---	30	---	74.3±0.7	0	1200±400	76±4	0	270±100
Método 13a	64.0±1.7	0	600±200	68±3	3	2100±600	90.0±1.7	0	220±90
Método 13b	63±2	4	200±60	61±4	7	3000±800	91.8±1.3	0	250±70
Método 13c	63±3	2	1200±400	64±3	2	2500±600	92.2±1.1	0	190±60

*6.7. APÉNDICE DEL CAPÍTULO*

# Capítulo 7

## CONCLUSIONES Y TRABAJOS FUTUROS

### Conclusiones y Trabajos Futuros

Como hemos repetido a lo largo de esta Tesis Doctoral, su objetivo ha sido el de realizar un estudio comparativo sobre diferentes métodos existentes para resolver diversos aspectos referentes al diseño de la arquitectura de red neuronal Multilayer Feedforward, en problemas de clasificación.

En el capítulo primero, se realizó una pequeña introducción sobre diversos conceptos de Redes Neuronales, la motivación de este trabajo y las herramientas de trabajo de las que hemos dispuesto en el desarrollo del mismo.

En el segundo capítulo se hizo una investigación detallada sobre el problema de codificación de entradas desconocidas. Se realizó un estudio comparativo entre 5 métodos de utilización de entradas desconocidas, aplicándolos en 95 problemas de clasificación con diferentes porcentajes de entradas desconocidas (19 sin entradas desconocidas). Se utilizaron entradas desconocidas tanto en el proceso de entrenamiento como en el de test y más tarde se estudió la influencia de los diferentes porcentajes sobre la eficiencia final de los métodos.

El método clásico de codificación que consiste en sustituir las entradas desconocidas por un valor (método 1), presentó una alta variabilidad en la eficiencia obtenida en el test al variar el porcentaje de entradas desconocidas. También vimos que la eficiencia depende en gran medida del problema particular y que se suele encontrar un porcentaje de entradas desconocidas a partir del cual el método deja de funcionar.

## CONCLUSIONES Y TRABAJOS FUTUROS

Los métodos que utilizan aritmética de intervalo resultaron ser los más eficientes. El método de Hernández (método 2) funcionó mejor que el método de Ishibuchi (método 3) para las 3 primeras reglas de clasificación que se utilizaron (que resultan ser las más adecuadas). Se encontraron problemas en los cuales existía un porcentaje de entradas desconocidas en el entrenamiento para el cual el método 3, y en ocasiones también el 2, dejaba de funcionar.

Realizando una comparación entre todos los métodos se observó: que el *método 1* (clásico) funcionaba bien pero con alta variabilidad, el *método 4* (de probabilidades) no funcionaba bien, el *método 5* (de redundancia) funcionaba bien pero tenía una aplicabilidad limitada (solo para problemas con entradas binarias) y que los *métodos 2 y 3* (métodos con aritmética de intervalos), son los que mejor resolvían cualquier problema. Entre estos últimos recomendamos el uso del método de aritmética de intervalos de Hernández (método 2).

En el tercer capítulo, se realizó un estudio sobre la importancia de la selección de la información de entrada a la red. Analizamos para ello 26 métodos existentes de selección de entradas, 19 basados en el análisis de la red neuronal Multilayer Feedforward y 7 basados en el análisis del conjunto de entrenamiento. Como resultado del análisis de los métodos se obtuvo para cada uno de ellos y para cada problema utilizado una ordenación de las entradas a la red según su importancia.

Para comparar la eficiencia de los métodos entre sí, utilizamos una metodología que presentamos en el capítulo 3. En ella, se busca el subconjunto óptimo de entradas para cada problema y método y se comparan las eficiencias obtenidas para estos subconjuntos usando el test estadístico *test-t*.

Como resultado final del tercer capítulo obtuvimos una ordenación de la eficiencia de los métodos.

Se observó en primer lugar que los métodos basados en el análisis de la red Multilayer Feedforward (ARNMF) presentan alta variabilidad en su eficiencia, y de ellos los que funcionan mejor son más eficientes que los métodos basados en el análisis del conjunto de entrenamiento (ACE). Sin embargo, entre estos últimos encontramos una excepción: el método basado en el cálculo de la Distancia GD (denominado por nosotros GD\_DS).

Pensamos que este método puede ser muy prometedor ya que combina la característica de bajo coste computacional con la alta eficiencia de los mejores métodos de ARNMF. Sin embargo, hemos encontrado algunos problemas en los cuales el método GD\_DS no ha sido aplicable ya que la matriz de transformación que utiliza resultó singular. Por ello, pensamos que se podría

introducir alguna mejora en dicho método que solventará dicha situación. Esto podría ser un trabajo de investigación futuro.

También en el capítulo tercero, se estudió el efecto de introducir para el entrenamiento de las redes iniciales de los métodos basados en el análisis de la red Multilayer Feedforward un término de regularización, weight-decay. Según la bibliografía este término debería mejorar la eficiencia en la tarea de selección de entradas.

Aplicamos weight-decay a 17 de los métodos ARNMF estudiados y sobre ocho problemas. El resultado que obtuvimos fue que el empleo de weight-decay mejoraba la eficiencia de los peores métodos y deterioraba la eficiencia de los mejores. En este sentido, disminuirán las diferencias entre los métodos de selección de entradas. Así, al seleccionar uno cualquiera de los métodos ARNMF para resolver un determinado problema obtendríamos una eficiencia más uniforme si utilizamos el término de regularización weight-decay, pero nunca alcanzaríamos la eficiencia proporcionada por los mejores métodos ARNMF sin el uso de weight-decay. Por ello nosotros no recomendamos su uso.

Como ejemplo de algunos de los resultados del capítulo se presentó la aplicación a un caso real de patología de la voz. En él, se consiguió reducir el número de entradas de la red necesario para resolver el problema de clasificar entre voces normales y voces patológicas de 35 a 2. Vemos así que los métodos de selección de entradas pueden ser muy útiles en la resolución de problemas reales.

Como trabajo futuro, se seguirá realizando revisiones bibliográficas periódicas sobre este tema de manera de que en caso de que surgieran nuevos métodos intentaríamos incluirlos en nuestra ordenación.

En el capítulo 4, se mostró un estudio comparativo entre los algoritmos de aprendizaje avanzados, que seleccionan de forma dinámica el número de unidades de la capa oculta de la red durante el proceso de aprendizaje. Vimos que existían dos tipos de procedimientos: constructivos y de podado. Nosotros propusimos el uso de un algoritmo similar al de Hirose, que combinó ambos procedimientos pero en el cual la condición de parada del algoritmo, se basó en el uso de cross-validación como criterio de convergencia de las redes.

Se estudiaron dos procedimientos constructivos y trece de podado y se combinaron entre sí, cada método constructivo se utilizó con cada uno de los trece métodos de podado. El algoritmo propuesto se aplicó a once problemas.

Se concluyó que como procedimiento constructivo, el de HIROSE proporcionaba una capacidad de generalización superior que el de KHORASANI,

## CONCLUSIONES Y TRABAJOS FUTUROS

obteniendo al mismo tiempo un número de neuronas adecuado para resolver el problema. Comparamos también, la eficiencia del método de HIROSE en relación con los procedimientos clásicos de prueba y error, concluyendo igualmente que la capacidad de generalización es superior para HIROSE.

Utilizamos pues, como procedimiento constructivo, el de HIROSE y medimos la capacidad de generalización para los diferentes métodos de podado. Se observó que el resultado obtenido después de la fase de podado era similar al obtenido en el proceso constructivo, con lo cual la reducción de neuronas no repercute en la mejora de la capacidad de generalización, aunque sí simplifica la estructura de la red.

Comparando los resultados que ofrecieron los diferentes métodos de podado concluimos que el método 9 (de Kurita y otros), en combinación con el método constructivo de Hirose ofrece mejores resultados, por lo que recomendamos su uso.

Pensamos, debido a su sencillez y efectividad, que podremos incluir en nuestros trabajos futuros este tipo de metodología para escoger la arquitectura de la capa oculta de las redes que tengamos que utilizar.

En el quinto capítulo se estudio de qué manera influía en la capacidad de generalización de la red el número de capas ocultas seleccionado para la arquitectura a usar.

Se utilizaron 16 problemas para medir la capacidad que ofrecían las arquitecturas de una y dos capas ocultas. Se realizó una selección previa de las arquitecturas mínimas de cada tipo para cada problema.

De nuestros resultados podemos concluir que la capacidad de generalización de una red con una capa oculta, es en general mejor que en una red con dos capas ocultas. Además hemos observado que para redes con más de una capa oculta la convergencia, en algunos problemas, es bastante mala.

Finalmente, en el sexto capítulo, se abordó el problema de inicialización de los pesos de la red previa a su entrenamiento. Esta inicialización influye sobre aspectos como capacidad de generalización, velocidad de convergencia y éxito en la convergencia.

Mediante un estudio comparativo entre los diferentes métodos que se encontraron en la bibliografía, concluimos que la utilización de la función de transferencia tangente hiperbólica da mejores resultados que el uso de la función sigmoide.

De los 13 métodos estudiados, los denominados como métodos 2, 4 y 5 ofrecen mejores resultados que el método clásico de inicialización (método 1).

Para el resto de métodos se encontró una alta variabilidad de los resultados. Los métodos en los que había que escoger algún parámetro resultaron en general poco eficientes.

En algunos problemas hubo métodos que presentaron un bajo porcentaje de éxito en las convergencias de las redes. En estos casos los métodos no son muy adecuados ya que nos interesa que sean aplicables de la manera más general posible.

Como conclusión del capítulo 6, podemos decir, que el método 5 ofrece mejores resultados para todos los problemas. Sin embargo, presenta la desventaja de tener que escoger ciertos parámetros y en alguna de las elecciones su eficiencia es inferior. Por ello nosotros pensamos que sería más interesante el uso del siguiente método en eficiencia, el método 4, ya que ofrece buenos resultados para casi todos los problemas estudiados y presenta la ventaja de su sencillez.

*CONCLUSIONES Y TRABAJOS FUTUROS*



## Capítulo 8

### REFERENCIAS BIBLIOGRÁFICAS

#### Referencias Bibliográficas

En este capítulo se incluyen las referencias bibliográficas, de libros, proceedings y revistas especializadas que han sido consultados asiduamente para la elaboración de la presente Tesis Doctoral y que pueden ser de utilidad a cualquier persona que pretenda realizar un trabajo de investigación sobre redes neuronales.

Están ordenadas alfabéticamente por el nombre del primer autor.

- [1] Ahmad, S., Tresp, V., "Classification with missing and uncertain inputs". Proceedings of 1993 IEEE International Conference on Neural Networks, pp. 1949-1954, San Francisco, California, March 1993.
- [2] Alefeld, G., Herzberger, J., *Introduction to Interval Computations*. Academic Press. New York. 1983.
- [3] Andersen, T., Martinez, T., "Cross Validation and MLP Architecture Selection". International Joint Conference on Neural Networks (IJCNN'99). 1999.
- [4] Anguita D., Ridella S., Rovetta S., "Limiting the Effects of Weight Errors in Feedforward Networks using Interval Arithmetic". Proceedings of the International Conference on Neural Networks, vol. 1, pp. 414-417, 1996.
- [5] Armitage, W.D., Jien-Chung Lo, "Enhancing the robustness of a Feedforward neural network in the presence of missing data". Proceedings of IEEE International Conference on Neural Networks, pp. 836-839, 1994.
- [6] Ash, T., "Dynamic node creation". Connection Science, vol. 1, no. 4, pp. 365-375, 1989.

- [7] Atiya, A., Ji, C., “How initial conditions affect generalization performance in large networks”. *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 448-451, 1997.
- [8] Azimi-Sadjadi, M. R., Sheedvash, S., Trujillo, F. O., “Recursive dynamic node creation in Multilayer Neural Networks”. *IEEE Transactions on Neural Networks*, vol. 4, no. 2, pp. 242-256, 1993.
- [9] Babri H.A., Tong Y., “Deep Feedforward Networks: Application to Pattern Recognition”. *Proceedings of the 1996 International Conference on Neural Networks*. 1996, vol. 3, pp. 1422-1426.
- [10] Banci G., Monini S., Falaschi A., Sairo N., “Vocal fold disorder evaluation by digital speech analysis”. *Journal of Phonetics*, vol. 14, pp. 495-499, 1986.
- [11] Bartlett, E. B., “A dynamic node architecture scheme for layered neural networks”. *Journal of Artificial Neural Networks*, vol. 1, no. 2, pp. 229-245, 1994.
- [12] Bartlett, E. B., “Dynamic node architecture learning: An information theoretic approach”. *Neural Networks*, vol. 7, no. 1, pp. 129-140, 1994.
- [13] Bartlett, E. B., “Self determination of input variable importance using neural networks”. *Neural, Parallel & Scientific Computations*, vol. 2, pp. 103-114. 1994.
- [14] Battiti, R., “Using mutual information for selecting features in supervised neural net learning”. *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537-550, 1994.
- [15] Belue, L.M., Bauer, K.W., “Determining input features for multilayer perceptrons”. *Neurocomputing*, vol. 7, no. 2, pp. 111-121, 1995.
- [16] Bishop, C., “Exact calculation of the hessian matrix for the multilayer perceptron”. *Neural Computation*, vol. 4, pp. 494-501, 1992.
- [17] Bos, S., Chug, E., “Using weight-decay to optimize the generalization ability of a perceptron”. *Proceedings of IEEE International Conference on Neural Networks*, vol. 1, pp. 241-246, 1996.
- [18] Bounds, D. G., Lloyd, P. J., “A multilayer perceptron for the diagnosis of low back pain”. *Proceedings of the IEEE International Conference on Neural Networks*, vol. 2, pp. 481-489, 1988.
- [19] Bowles, A.: Machine learns which features to select. *Proceedings of the 5<sup>th</sup> Australian Joint Conference on Artificial Intelligence*, pp. 127-132, 1992.
- [20] Boyanov B., Chollet G., “Pathological voice analysis using cepstra, bispectra and group delay functions”. *Proceedings of the International Conference on Spoken Language Processing*, pp. 1039-1042, 1992.
- [21] Boyanov B., Hadjitodorov S., Baudoin G., “Acoustical analysis of pathological voice”. *Proceedings of the Slovenian German Workshop on Speech Analysis and Recognition*, 1996.
- [22] Boyanov B., Hadjitodorov S., Ivanov T., “Analysis of voiced speech by means of bispectrum”. *Electronics Letters*, vol. 7, no. 24, pp. 2267-2268, 1991.
- [23] Boyanov B., Hadjitodorov S., Teston B., Doskov D., “Robust hybrid detector for pathological voice analysis”. *LARYNX* 1997, pp. 55-58, 1997.
- [24] Boyanov B., Hadjitodorov S., Teston B., Doskov D., “Software system for pathological voice analysis”. *LARYNX* 1997, pp. 139-142, 1997.

- [25] Callant, D.E., Kent, R.D., et al., "Self-organizing map for the classification of normal and disordered female voices". *Journal of Speech, Language, and Hearing Research*, vol. 42, no. 2, pp. 355-366, 1999.
- [26] Chi, Jabri, "Entropy based feature evaluation and selection technique". *Proceedings of 4<sup>th</sup> Australian Conference on Neural Networks, (ACNN'93)*, pp. 193-196, 1993.
- [27] Chih-Liang Chen, Nutter, Roy S., "Improving the Training Speed of Three-layer Feedforward Neural Nets by Optimal Estimation of the Initial Weights". *International Joint Conference on Neural Networks*, vol. 3, part. 1, pp. 2063-2068, 1991.
- [28] Cibas, T., Soulié, F.F., Gallinari, P., Raudys, S., "Variable selection with neural networks". *Neurocomputing*, vol. 12, pp. 223-248, 1996.
- [29] Cosnard M., Koiran P., Paugam-Moisy H., "A Step Toward the Frontier between One-Hidden-Layer and Two-Hidden Layer Neural Networks". *Proceedings of 1993 International Conference on Neural Networks*, vol. 3, pp. 2292-2295, 1993.
- [30] Cybenko, George, "Approximation by superposition of sigmoidal functions". *Mathematics of Control Signals, and Systems*, vol. 2, pp. 303-314. 1989.
- [31] Deco, G., Obradovic, D., *An information Theoretic Approach to Neural Computing*. Springer-Verlag New York, Inc. 1996.
- [32] Denoeux, T., Lengellé, R., "Initializing Back Propagation Networks With Prototypes". *Neural Networks*, vol. 6, pp. 351-363, 1993.
- [33] Devena, L., "Automatic selection of the most relevant features to recognize objects". *Proceedings of the International Conference on Artificial Neural Networks*, vol. 2, pp. 1113-1116, 1994.
- [34] Devroye, L., Györfi, L., Lugosi, G., *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag New York, Inc. 1996.
- [35] Drago, G.P., Ridella S., "SCAWI: an algorithm for weight initialization of a sigmoidal neural network". *Proceedings. of the International Conference on Artificial Neural Networks (ICANN'92)*, vol. 2, pp. 983-986, 1992.
- [36] Drago, G.P., Ridella, S., "Statistically Controlled Activation Weight Initialization (SCAWI)". *IEEE Transactions on Neural Networks*, vol. 3, no. 4, pp. 627-631, 1992.
- [37] El-Deredy, W., Branston, N.M., "Identification of relevant features in HMR tumor spectra using neural networks". *Proceedings of the 4<sup>th</sup> International Conference on Artificial Neural Networks*, pp. 454-458, 1995.
- [38] Engelbrecht, A.P., Cloete, I., "A sensitivity analysis algorithm for pruning feedforward neural networks". *Proceedings of the International Conference on Neural Networks*, vol 2, pp. 1274-1277, 1996.
- [39] Engelbrecht, AP., Cloete, I., "A sensitivity analysis algorithm for pruning feedforward neural networks". *Proceedings of the International Conference on Neural Networks*, vol. 2, pp. 1274-1277, 1996.
- [40] Fahlman, S.E., "An experimental study of learning speed in Back-Propagation networks". *CMU Technical Report, CMU-CS-162*, Sept. 1988.
- [41] Fernández, M., Hernández, C., "A comparison among weight inicialization methods for Multilayer Feedforward networks". *Proceedings of the 2000*

- IEEE International Joint Conference on Neural Networks (IJCNN'2000), vol. 4, pp. 543-548, 2000.
- [42] Fernández, M., Hernández, C., "Input selection by Multilayer Feedforward trained networks". Proceedings of the 1999 IEEE International Joint Conference on Neural Networks, (IJCNN'99), pp. 1834-1839, 1999.
- [43] Fernández, M., Hernández, C., "Weight initialization methods for multilayer feedforward". European Symposium on Artificial Neural Networks (ESANN'2001). En prensa. Bruges (Belgium).
- [44] Fernández, M., Hernández, C., "A Comparison among feature selection methods based on trained networks". Proceedings of the 1999 IEEE Neural Networks for Signal Processing Workshop (NNSP'99), pp. 205-214, 1999.
- [45] Fernández, M., Hernández, C., "Analysis of Input Selection Methods for Multilayer Feedforward". Neural Network World International Scientific Journal, vol. 10, no. 3, pp. 389-406, Praga, Julio de 2000.
- [46] Fernández, M., Hernández, C., "Generalization capability of one and two hidden layers". Proceedings of the 1999 IEEE International Joint Conference on Neural Networks (IJCNN'99), pp. 1840-1845, 1999.
- [47] Fernández, M., Hernández, C., "How to select the inputs for a Multilayer Feedforward Network by using the training set". Proceedings. of the 5<sup>th</sup> International Work Conference on Artificial and Natural Neural Networks (IWANN'99), pp. 477-486, 1999.
- [48] Fernández, M., Hernández, C., "Influence of Weight-Decay Training in Input Selection Methods", European Symposium on Artificial Neural Networks (ESANN'2000), pp.135-140, Brujas (Bélgica) 2000.
- [49] Fernández, M., Hernández, C., "Neural networks input selection by using the training set". Proceeding of the 1999 IEEE International Joint Conference on Neural Networks, (IJCNN'99), pp. 1189-1194, 1999.
- [50] Fernández, M., Hernández, C., "On the Combination of Weight-Decay and Input Selection Methods". Proceedings of the 2000 International Joint Conference on Neural Networks, (IJCNN'2000), vol. 1, pp. 191-196, 2000.
- [51] Fernández, M., Hernández, C., "One or Two Hidden Layers Perceptrons". European Symposium on Artificial Neural Networks (ESANN'98), pp. 129-133, 1998.
- [52] Fernández, M., Hernández, C., "Optimal use of a trained neural network for input selection". Proceedings of the 5<sup>th</sup> International Work Conference on Artificial and Natural Neural Networks (IWANN'99), pp. 506-515, 1999.
- [53] Fernández, M., Hernández, C.A., "A comparison between two interval arithmetic learning algorithms". Proceedings of the International Joint Conference on Neural Networks (IJCNN'99), pp. 1293-1298, 1999.
- [54] Fernández, M., Hernández, C.A., "An experimental evaluation of Different Methods for Handling Missing Inputs". Proceeding of the International Joint Conference on Neural Networks (IJCNN'98), vol. 2, pp. 1450-1453, 1998.
- [55] Fernández, M., Hernández, C.A., "How to handle unknown Inputs". Proceeding of the International Computer Symposium (ICS' 98), vol. B, pp. 272-276, 1998.

- [56] Fernández, M., Hernández, C.A., "On the error function of interval arithmetic Backpropagation". Proceeding of the European Symposium on Artificial Neural Networks (ESANN' 98), pp. 135-139, 1998.
- [57] Fine, T.L., *Feedforward Neural Network Methodology*. Springer-Verlag New York, Inc. 1999.
- [58] Godino-Llorente, J.I., Aguilera-Navarro, S., Hernández-Espinosa, C., Fernández-Redondo, M., Gómez-Vilda, P., "On the selection of meaningful speech parameters used by a pathologic/non Pathologic voice register classifier". 6th European Conference on Speech Communication and Technology (EuroSpeech'99), vol. 1, pp. 563-566, 1999.
- [59] Golden, R. M., *Mathematical methods for Neural Network Analysis and Design*. Cambridge, MA: MIT Press. 1996.
- [60] Gosh, J., Tumer, K., "Structural adaptation and generalization in supervised feedforward networks". Journal of Artificial Neural Networks, vol. 1, no. 4, pp. 431-458, 1994.
- [61] Greene, R.A., "Neural net applications in the cognitive sciences: a case study". Journal of Microcomputer Applications, vol. 16, pp. 197-201, 1993
- [62] Hagiwara, M., "A simple and effective method for removal of hidden units and weights". Neurocomputing, vol. 6, pp. 207-218, 1994.
- [63] Hagiwara, M., "Backpropagation with selection-reduction of learning time and elimination of hidden units". Systems and Computers in Japan, vol. 23, no. 8, pp 46-54, 1992.
- [64] Hammitt, A.M., Bartlett, E.B., "Determining functional relationships from trained neural networks". Mathematical Computing Modelling, vol. 22, no. 3, pp. 83-103, 1995.
- [65] Harvey, R. M., "A neural network as a potential means for aiding nursing diagnosis", Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care, pp. 875-879, 1990.
- [66] Haung L., Zhang B., "A Novel-Network-Related Approach for Regression Analysis with Interval Model". Proceedings of the International Conference on Neural Networks, vol. 1, pp. 611-616, 1996.
- [67] Haykin, S., *Neural Networks. A comprehensive Foundation*. IEEE Press. Macmillan College Publishing Company, Inc. 1994.
- [68] Hecht-Nielsen, R., *Neurocomputing*, Addison Wesley, 1991.
- [69] Heinke, D., Hamker, F. H., "Comparing Neural Networks: a Benchmark on Growing Neural Gas, Growing Cell Structures, and Fuzzy ARTMAP". IEEE Transactions on Neural Networks, vol. 9, no. 6, pp. 1279-1291, November 1998.
- [70] Hernández C.A., Espí, J., Nakayama, K., Fernández, M., "Interval Aritmetic Backpropagation". Proceedings of the 1993 International Joint Conference on Neural Networks (IJCNN'93), vol. 1, pp. 375-378, 1993.
- [71] Hernández, C., Espí, J., Nakayama, K., "A generalization of Backpropagation to Interval Aritmetic". Proceedings of the World Congress on Neural Networks (WCNN'93), vol. 4, pp. 131-134, 1993.
- [72] Hernández, C., Espí, J., Nakayama, K., "Effect of training data distribution on Backpropagation Generalization". Proceedings of the 1993 IEEE International Symposium on Circuits and Systems (ISCAS'93), pp. 2148-2151, 1993.

## REFERENCIAS BIBLIOGRÁFICAS

- [73] Hernández, C., Fernández, M., “On the Effect of Weight-Decay in Input Selection”. *Neural Network World International Scientific Journal*, vol. 10, no. 4, pp. 575-588, Praga, Julio de 2000.
- [74] Hernández, C., Fernández, M., Gomez, P., “Neural Network based Input Selection and Diagnosis of Pathologic Voices”. *Neural Network World International Scientific Journal*. En prensa.
- [75] Hernández, C., Fernández, M., Gomez, P., Godino, J.I., Aguilera, S., “Diagnosis of Vocal and Voice Disorders by the Speech Signal”. *Proceedings of 2000 International Joint Conference on Neural Networks, (IJCNN'2000)*, vol. 4, pp. 253-258, 2000.
- [76] Hirose, Y., Yamashita, K., Hijiya, S., “Back-propagation algorithm which varies the number of hidden units”. *Neural Networks*, vol. 4, pp. 61-66, 1991.
- [77] Ho-Sub Yoon, Chang-Seok Bae, Byung-Woo Min, “Neural Networks using Modified Initial Connection Strengths by the Importance of Feature Elements”. *International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 458-461, 1995.
- [78] Ishibuchi, H., Miyazaki, A., Kwon, K., Tanaka, H., “Learning from incomplete training data with missing values and medical application”. *Proceedings of 1993 IEEE International Conference on Neural Networks (Nagoya)*, pp. 1871-1874, 1993.
- [79] Ishibuchi, H., Miyazaki, A., Tanaka, H., “Neural-network-based diagnosis systems for incomplete data with missing inputs”. *Proceedings of 1994 IEEE International Conference on Neural Networks*, pp. 3457-3460, 1994.
- [80] Ishibuchi, H., Tanaka, H., “An extension of the BP-Algorithm to Interval Input Vectors -Learning from Numerical Data and Expert’s Knowledge-“. *Proceedings of the 1991 International Joint Conference on Neural Networks*, Singapore, pp. 1588-1593, 1991.
- [81] Jacobs R.A., “Increased rates of convergence through learning rate adaptation”. *Neural Networks*, vol. 1, 1988.
- [82] Ji, C., Psaltis, D., “Network synthesis through data-driven growth and decay”. *Neural Networks*, vol. 10, no. 6, pp. 1133-1141, 1997.
- [83] Kamimura, R., “Principal Hidden unit analysis generation of simple networks by minimum entropy method”. *Proceedings of 1993 International Joint Conference on Neural Networks*, pp. 317-320, 1993.
- [84] Kamimura, R., “Principal Hidden unit analysis with minimum entropy method”. *Proceedings of the International Conference on Artificial Neural Networks*, pp. 760-763. 1993.
- [85] Karayiannis, N.B., “Accelerating the Training of Feedforward Neural Networks Using Generalized Hebbian Rules for Initializing the Internal Representations”. *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 419-426. March 1996.
- [86] Kasuya H., Endo Y., Saliu S., “Novel acoustic measurements of jitter and shimmer characteristic from pathologic voice”. *Eurospeech 1993*, pp. 1973-1976. Berlin 1993.
- [87] Kasuya H., Ogawa Sh., Mashima K., Ebihara S., “Normalized noise energy as an acoustic measure to evaluation pathologic voice”. *Journal of Acoustic Society of America*, vol. 80, no. 5, pp. 1329-1334, 1986.

- [88] Khorasani, K., Weng, W., "Structure adaptation in feed-forward neural networks". Proceedings of IEEE International Conference on Neural Networks, vol. 3, pp. 1403-1408, 1994.
- [89] Kim, Y.K., Ra, J.B., "Weight Value Initialization for Improving Training Speed in the Backpropagation Network". Proceedings of International Joint Conference on Neural Networks, vol. 3, pp. 2396-2401, 1991.
- [90] Kira, K., Rendell, L.A., "The feature selection problem: Traditional methods and a new algorithm". Proceedings of 10<sup>th</sup> National Conference on Artificial Intelligence, pp. 129-134, 1992.
- [91] Kurita, T., H-Asoh, Umeyama, S., Akaho, S., Hosomi, A., "A structural learning by adding independent noises to hidden units". Proceedings of IEEE International Conference on Neural Networks, vol. 1, pp. 275-278, 1994.
- [92] Kwok, T. Y., Yeung, D. Y., "Constructive algorithms for structure learning in Feedforward Neural Networks for regression problems". IEEE Transactions on Neural Networks, vol. 8, no. 3, pp. 630-645, 1997.
- [93] Kwok, T.Y., Yeung, D.Y., "Constructive neural networks: some practical considerations". Proceedings of the 1994 IEEE International Conference on Neural Networks, vol. 1, pp. 198-203, 1994.
- [94] Kwok, T.Y., Yeung, D.Y., "Objective functions for training new hidden units in constructive neural networks". IEEE Transactions on Neural Networks, vol. 8, no. 5, pp. 1131-1148, 1997.
- [95] Lee, H., Mehrotra, K., Mohan, C. Ranka, S., "Selection procedures for redundant inputs in neural networks". Proceedings of the World Congress on Neural Networks, vol. 1, pp. 300-303, 1993.
- [96] Li, G., Alnuweiri, H., Wu, Y., "Acceleration of Backpropagations through Initial Weight Pre-Training with Delta Rule". Proceedings of the IEEE International Conference on Neural Networks, vol. 1, pp. 580-585, 1993.
- [97] Lim, S.F., Ho, S.B., "Dynamic creation of hidden units with selective pruning in Backpropagation". World Congress on Neural Networks, vol. 3, pp. 492-497, 1994.
- [98] Lorenzo, Hernández, Méndez, "Attribute selection through a measurement based on information theory".(in Spanish). 7<sup>a</sup> Conferencia. de la Asociación Española para la Inteligencia Artificial, (CAEPIA), pp. 469-478, 1997.
- [99] Mao, J., Mohiuddin, K., Jain, A.K., "Parsimonious network design and feature selection through node pruning". Proceedings of the 12<sup>th</sup> IAPR International Conference on Pattern Recognition, vol. 2, pp. 622-624, 1994.
- [100] Maren, A., Harston, C., Pap, R., *Handbook of Neural Computing Applications*, Academic Press Inc. 1990.
- [101] Maren, C. Harston, R. Pap. *Handbook of Neural Computing Applications*. Academic Press Inc. 1990.
- [102] Matsunaga, Y., Nakade, Y, Yamakawa, O., Murase, K., "A Backpropagation algorithm with automatic reduction of association units in Multilayered Neural Network". Proceedings of the International Joint Conference on Neural Networks, pp. 783-788, 1991.
- [103] Merz, C. J., Murphy, P. M., (1998). UCI repository of machine learning databases. Available at: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

- [104] Montgomery, D. C., Runger, G. C., *Probabilidad y estadística aplicadas a la Ingeniería*. McGraw-Hill, 1996.
- [105] Müller, B., Reinhardt, J., Strickland, M.T., *Neural Networks. An Introduction*. Springer-Verlag Berlin Heidelberg. 1<sup>st</sup> Ed. 1990-2<sup>nd</sup> Ed. 1995.
- [106] Mulsant, B. H., Servan-Sheriber, E., “A connectionist approach to the diagnosis of dementia”, *Proceedings of the Twelfth Annual Symposium on Computer Application in Medical Care*, pp. 245-250, 1988.
- [107] Nakayama K., Hirano A., Ido I., “A multilayer neural network with nonlinear inputs and trainable activation functions: structure and simultaneous learning algorithms”. *Proceedings of the International Joint Conference on Neural Networks*, 1999.
- [108] Palubinskas, G., “Data-driven Weight Initialization of Back-propagation for Pattern Recognition”. *Proceedings of the International Conference on Artificial Neural Networks (ICANN'94)*, vol. 2, pp. 851-854, 1994.
- [109] Peck, C.C., Dhawan, A.P., “Genetic algorithm based input selection for a neural network function approximator with applications to SSME health monitoring”. *Proceedings of the 1993 IEEE International Conference on Neural Networks*, vol. 22, pp. 1115-1122, 1993.
- [110] Peck, C.C., Dhawan, A.P., “SSME parameter model input selection using genetic algorithms”. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 1, January 1996.
- [111] Prechelt, L., "A quantitative study of experimental evaluations of neural network learning algorithms: current research practice". *Neural Networks*, vol. 9, no. 3, pp. 457-462, 1996.
- [112] Priddy, K.L., Rogers, S.K., Ruck D.W., Tarr G.L., Kabrisky, M., “Bayesian selection of important features for feedforward neural networks”. *Neurocomputing*, vol. 5, no. 2&3, pp. 91-103, 1993.
- [113] Reed, R. D., Marks, R. J., *Neural Smthing. Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA: MIT Press. 1999.
- [114] Reilly, D.L., Cooper, L.N., Elbaum, C., “A neural Model for Category Learning”. *Biological Cybernetics*, vol. 45, pp. 35-41, 1982.
- [115] Rezqui A. et al., “The effect of the slope of the activation function on the backpropagation algorithm”. *Proceedings of the International Joint Conference on Neural Network*, vol. 1, pp. 707-710, January 1990.
- [116] Rojas, R., “Optimal weight initialization for neural networks”, *Proceedings of the International Conference on Artificial Neural Networks (ICANN'94)*, vol. 1, pp. 577-580, 1994.
- [117] Rossi, F., Gegout, C., “Geometrical Initialization, Parametrization and Control of Multilayer Perceptrons: Application to Function Approximation”, *Proceedings of the IEEE International Conference on Neural Networks (ICNN'94)*, vol. 1, pp. 546-550, 1994.
- [118] Rumelhart, D. E., McClelland, J. L., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. vol. I, II y III. Cambridge, MA: MIT Press, 1986, 1987.
- [119] Rumelhart, D.E., Hinton, G.E., Williams, R.J., “Learning internal representations by error backpropagation”. *Parallel Distributed Processing*. Cambridge MA: MIT Press. Chapter 8, pp. 318-362, 1986.



- [120] Sano, H., Nada, A., Iwahori, Y., Ishii, N., "A method of analyzing information represented in neural networks". Proceedings of 1993 International Joint Conference on Neural Networks, pp. 2719-2722, 1993.
- [121] Saratchandran P., "Effect of Hidden Layers on Generalization Properties of Feedforward Neural Networks". Neural Parallel & Scientific Computations, vol. 1, no. 2, pp. 227-240, 1993.
- [122] Scalia, F., Marconi, L., Ridella, S., Arrigo, P., Mansi, C., Mela, G. S., "An example of back propagation: diagnosis of dyspepsia", Proceedings of the First IEEE International Conference on Artificial Neural Networks, pp. 332-335. 1989.
- [123] Schalkoff, R. J., *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons, Inc. 1992.
- [124] Scherf,, "A new approach to feature selection". Proceedings of the 6<sup>th</sup> Conference on Artificial Intelligence in Medicine, (AIME'97), pp. 181-184, 1997.
- [125] Schoentgen J., "Jitter in discriminating between normal and dysphonic speakers". Applied stochastic models and data analysis, vol. 4, pp. 127-135, 1985.
- [126] Setiono, R., Hui, L. C. K., "Use of Quasi-Newton method in a Feedforward neural network construction algorithm". IEEE Transactions on Neural Networks, vol. 6, no. 1, pp. 273-277, 1995.
- [127] Setiono, R., Liu, H., "Improving Backpropagation learning with feature selection". Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies, vol. 6, no. 2, pp. 129-139, 1996.
- [128] Shimodaira H., "A Weight Value Initialization Method for Improved Learning Performance of the Back Propagation Algorithm in Neural Networks". Proceedings of the 6th International Conference on Tools with Artificial Intelligence, pp. 672-675, 1994.
- [129] Smyth, S.G., "Designing Multilayer Perceptrons from Nearest Neighbor Systems". IEEE Transactions on Neural Networks, vol. 3, no. 2, pp. 329-333. 1992.
- [130] Tamura S., Tateishi M., "Capabilities of a Four-Layered Feedforward Neural Network: Four Layers versus Three". IEEE Trans. on Neural Networks. 1997, vol. 8, no. 2, pp. 251-255.
- [131] Tarek Kaylani, Sushil Dasgupta, "Weight initialization of MLP classifiers using boundary-preserving patterns". 1994.
- [132] Tetko, I.V., Tanchuk, V.Y., Luik, A.I., "Simple heuristic methods for input parameter estimation in neural networks". Proceedings of the IEEE International Conference on Neural Networks, vol. 1, pp. 376-380, 1994.
- [133] Tetko, I.V., Villa, A.E.P., Livingstone, D.J., "Neural network studies. 2 Variable selection". Journal of Chemical Information and Computer Sciences, vol. 36, no. 4, pp. 794-803, 1996.
- [134] Thawonmas, R., Abe, S., "Feature reduction based on analysis of fuzzy regions". Proceedings of the 1995 International Conference on Neural Networks, vol. 4, pp. 2130-2133, 1995.

## REFERENCIAS BIBLIOGRÁFICAS

- [135] Thimm, G., Fiesler, E., "High-Order and multilayer perceptron initialization". IEEE Transactions on Neural Networks, vol. 8, no. 2, pp. 349-359, 1997.
- [136] Tresp, V., Neuneier, R., Ahmad, S., "Efficient methods for dealing with missing data in supervised learning". Advances in Neural Information Processing Systems, pp. 689-696, 1995.
- [137] Utans, J., Moody, J., et al., "Input variable selection for neural networks: Application to predicting the U.S. business cycle". Proceedings of IEEE/IAFE Computational Intelligence for Financial Engineering, pp. 118-122, 1995.
- [138] Villiers J., Bernard E., "Backpropagation Neural Nets with One and Two hidden Layers". IEEE Transactions on Neural Networks, vol. 4, no. 1, pp. 136-141, 1992.
- [139] Wallen E.J., Hansen J.H.L., "A screening test for speech pathology assessment using objective quality measures", Proceedings of the International Conference on Spoken Language Processing, pp. 776-779, 1996.
- [140] Watzel, R., Meyer-Bäse, A., Meyer-Bäse, U., Hilberg, H., Scheich, H., "Identification of irrelevant features in phoneme recognition with radial basis classifiers". Proceedings of 1994 International Symposium on Artificial Neural Networks. pp. 507-512, 1994.
- [141] Weng, W., Khorasani, K., "An adaptive structure neural networks with application to EEG automatic seizure detection". Neural Networks, vol. 9, no. 7, pp. 1223-1240, 1996.
- [142] Wessels, L.F.A., Barnard, E., "Avoiding False Local Minima by Proper Initialization of Connections". IEEE Transactions on Neural Networks, vol. 3, no. 6, pp. 899-905, 1992.
- [143] Xiao J., Chen Z., Cheng J., "Structure Study of Feedforward Neural Networks for Approximation of Highly Nonlinear Real-valued Functions". Proceedings of 1996 International Conference on Neural Networks, vol. 1, pp. 258-263, 1996.
- [144] Xiao-Hu Yu, Guo-An Chen, Shi-Xin Cheng, "Dynamic Learning Rate Optimization of the Backpropagation Algorithm". IEEE Transactions on Neural Networks, vol. 6. no. 3, pp. 669-677, 1995.
- [145] Yamasaki, Masami, "The Lower Bound of the Capacity for a Neural Network with Multiple Hidden Layers". Proceedings of the World Congress on Neural Networks (WCNN'93), vol. 3, pp. 544-547, 1993.
- [146] Yasui, S., "Convergence suppression and divergence facilitation: Minimum and joint use of hidden units by multiple outputs". Neural Networks, vol. 10, no. 2, pp. 353-367, 1997.
- [147] Yoh-Han Pao, *Adaptative Pattern Recognition and Neural Networks*. Addison-Wesley Publishing Company, Inc. 1989.
- [148] Younes, B., Fabrice, B., "A neural network based variable selector". Proceedings of the Artificial Neural Network in Engineering, (ANNIE'95), pp. 425-430, 1995.
- [149] Yu X., Chen G., Cheng S., "Dynamic Learning Rate Optimization of the Backpropagation Algorithm". IEEE Transactions on Neural Networks, vol. 6, no. 3, pp. 669-677, 1995

- [150] Yu, X. H., “Can Backpropagation error surface have no local minima”. IEEE Transactions on Neural Networks, vol. 3, pp. 1019-1021, 1992.
- [151] Yunik M., Boyanov B., “Method for evaluation of the noise-to-harmonic component ratios in pathological and normal voices”. Acustica, vol. 70, pp. 89-91, 1990.
- [152] Zhu W., Kikuchi Y., Endo Y., et al., “An integrated acoustic evaluation system of pathologic voice”. Proceedings of the International Conference on Spoken Language Processing, pp. 1983-1986, 1994.



# Apéndices

## APÉNDICES

- I. *Apéndices del Capítulo 2*
  - I.1 *Resultados de los métodos 1, 4 y 5*
  - I.2 *Resultados de los métodos 2 y 3*
  - I.3 *Comparación entre los métodos 2 y 3*
  
- II. *Apéndices del Capítulo 3*
  - II.1 *Ordenación de las entradas para ARNMF y ACE*
  - II.2 *Ordenación de las entradas para ARNMF&WD*
  - II.3 *Eficiencias para los métodos ARNMF y ACE*
  - II.4 *Eficiencias para los métodos ARNMF&WD*
  - II.5 *Comparación entre métodos ARNMF y ACE*
  - II.6 *Comparación entre métodos ARNMF con y sin WD*

*REFERENCIAS BIBLIOGRÁFICAS*

## I. Apéndices del Capítulo 2

### I.1. Resultados de los métodos 1, 4 y 5

*Tabla 5-23. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test para los métodos 1, 4 y 5 y todas las bases de datos.*

		BASE DE DATOS ABALONE					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
ENTRENAMIENTO Met.1	0%	50±5	36.2±0.9	37.0±1.1	38.4±1.6	38.7±1.9	37.8±1.5
	5%	43±3	47±3	46±3	45±3	41±2	39.7±1.9
	10%	43±6	45±4	44±4	44±4	41±4	41±3
	20%	52.2±0.4	49.55±0.07	49.8±0.2	49.6±0.3	47.7±0.4	46.3±0.3
	30%	47±4	43±3	44±3	44±3	42±3	41±3
	40%	47±4	44±3	44±3	44±3	42±3	43±3
Met.4 0%		---	---	---	---	---	---
Met.5 0%		---	---	---	---	---	---

		BASE DE DATOS BALANCE					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
ENTRENAMIENTO Met.1	0%	58±9	57±7	55±6	52±4	52±4	52±3
	5%	65±8	62±7	60±6	55±5	54±4	53±4
	10%	64±9	62±8	60±7	56±5	55±4	55±3
	20%	64±9	62±8	60±6	56±5	55±4	54±4
	30%	65±9	63±8	60±7	56±5	54±4	54±3
	40%	56±9	55±7	54±6	50±5	50±4	48±4
Met.4 0%		---	---	---	---	---	---
Met.5 0%		---	---	---	---	---	---

**Tabla 7**

		<b>BASE DE DATOS BANDS</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	68.2±1.3	58.1±1.1	57.8±0.8	55.1±0.7	54.2±0.6	50.1±0.4
	<b>5%</b>	72.4±0.9	61.0±0.5	61.5±1.5	58.8±0.6	60.4±0.3	56.9±0.5
	<b>10%</b>	72.2±0.9	61.1±0.9	58.5±0.6	58.2±0.6	58.3±0.8	56.0±0.5
	<b>20%</b>	74.5±0.8	48.3±0.6	49.4±0.6	50.3±0.4	48.5±0.5	49.1±0.5
	<b>30%</b>	72.0±1.1	50.9±1.7	50.9±1.1	54.1±0.6	51.6±0.6	52.9±0.6
	<b>40%</b>	67.8±0.5	52.3±0.8	54.9±0.5	54.2±0.6	52.6±0.6	51.0±0.8
<b>Met.4 0%</b>		---	---	---	---	---	---
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 8**

		<b>BASE DE DATOS BUPA</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	68.3±1.6	70.0±1.2	57.8±1.3	55.0±0.7	52.5±0.7	52.6±0.3
	<b>5%</b>	59±2	57.2±0.5	54.5±0.7	53.1±0.4	52.3±0.4	52.6±0.3
	<b>10%</b>	58.0±1.7	54.9±0.3	51.4±0.3	52.93±0.16	51.52±0.16	51.9±0.3
	<b>20%</b>	53.3±0.6	55.3±0.4	51.6±0.4	52.0±0.3	54.17±0.15	51.9±0.4
	<b>30%</b>	54±0	55.9±0	51.4±0	51.6±0	52.8±0	53.17±0.02
	<b>40%</b>	53.0±1.0	54.0±0.7	50.5±0.3	50.0±0.6	53.3±0.2	51.2±0.5
<b>Met.4 0%</b>		---	---	---	---	---	---
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 9**

		<b>BASE DE DATOS CREDITO</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	86.6±0.6	48.1±1.3	49.2±1.2	51.5±1.1	52.1±1.0	51.9±0.7
	<b>5%</b>	80±3	47.3±0.6	49.6±0.5	52.6±0.5	53.5±0.6	52.9±0.3
	<b>10%</b>	86.2±0.4	49.1±1.2	50±1	50.8±0.5	51.4±0.5	52.2±0.4
	<b>20%</b>	83.4±1.1	49.1±0.8	48.8±0.6	50.6±0.6	49.8±0.3	52.3±0.5
	<b>30%</b>	87.8±1.1	46.5±0.9	48.4±0.9	50.2±0.4	60.0±0.3	51.6±0.3
	<b>40%</b>	87.8±0.2	48.7±1.5	48.7±1.0	50.3±0.8	50.4±0.7	51.2±0.6
<b>Met.4 0%</b>		86.6±0.6	48.1±1.3	49.2±1.2	51.52±1.07	52.07±1.04	52.0±0.7
<b>Met.5 0%</b>		---	---	---	---	---	---



**Tabla 10**

		<b>BASE DE DATOS DISPLAY1</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	71.9±1.3	10.13±0.11	9.39±0.10	9.72±0.13	9.76±0.13	9.4±0.2
	<b>5%</b>	67±4	10.07±0.05	9.5±0.3	9.51±0.12	9.7±0.3	9.1±0.2
	<b>10%</b>	73.20±0.05	9.98±0.02	9.10±0.05	9.46±0.13	9.52±0.06	8.87±0.15
	<b>20%</b>	71±2	10.15±0.07	9.4±0.2	9.86±0.11	10.0±0.2	9.35±0.09
	<b>30%</b>	73.65±0.08	10.05±0.01	9.17±0.02	9.72±0.02	9.75±0.09	9.17±0.05
	<b>40%</b>	73.35±0	10.1±0	9.2±0	9.7±0	9.55±0	9.65±0
<b>Met.4 0%</b>		71.9±1.3	10.13±0.11	9.39±0.10	9.72±0.13	9.76±0.13	9.4±0.2
<b>Met.5 0%</b>		71.9±1.3	70.6±1.3	66.6±1.3	59.1±1.1	52.8±1.1	45.1±1.1

**Tabla 11**

		<b>BASE DE DATOS DISPLAY2</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	70.9±0.8	8.62±0.02	9.07±0.05	7.63±0.05	9.97±0.12	8.47±0.09
	<b>5%</b>	71.6±0.2	8.54±0.03	8.64±0.10	7.65±0.07	9.51±0.12	8.27±0.14
	<b>10%</b>	70.7±0.8	8.58±0.03	8.99±0.06	7.71±0.04	9.81±0.07	8.66±0.15
	<b>20%</b>	72.00±0.15	8.60±0.02	9.07±0.02	7.99±0.04	9.39±0.05	8.57±0.07
	<b>30%</b>	71.22±0.15	8.61±0.02	8.77±0.11	8.11±0.06	9.77±0.13	7.99±0.11
	<b>40%</b>	72.17±0.13	8.55±0	8.42±0.08	7.93±0.03	8.93±0.08	7.82±0.13
<b>Met.4 0%</b>		70.9±0.8	8.62±0.02	9.07±0.05	7.63±0.05	9.97±0.12	8.47±0.09
<b>Met.5 0%</b>		70.9±0.8	67.9±0.8	63.3±0.7	59.4±0.7	53.8±0.8	46.0±0.4

**Tabla 12**

		<b>BASE DE DATOS GLASS</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	53±9	45±6	41±5	41±4	39±4	34.8±1.3
	<b>5%</b>	50±5	55±6	50±5	47±4	43±3	38±2
	<b>10%</b>	50±5	57±8	52±7	50±6	47±5	40±3
	<b>20%</b>	47±3	61±6	56±5	53±4	50±4	43±2
	<b>30%</b>	41±3	50±7	47±6	44±4	42±4	39±2
	<b>40%</b>	41±5	49±7	45±5	43±4	40±4	38±2
<b>Met.4 0%</b>		---	---	---	---	---	---
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 13**

		<b>BASE DE DATOS HEART</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	77.6±0.8	76.1±1.0	71.4±0.6	68.3±0.5	65.1±0.5	61.7±0.5
	<b>5%</b>	74.5±0.4	78.3±0.4	71.9±0.3	68.32±0.18	64.48±0.17	62.6±0.2
	<b>10%</b>	81.0±0.8	77.97±0.15	72.3±0.2	68.6±0.2	63.4±0.2	61.1±0.3
	<b>20%</b>	77±5	76±4	71±4	68±3	64±2	63±2
	<b>30%</b>	82.0±0.2	79.4±0.4	74.4±0.3	70.38±0.14	67.07±0.13	64.78±0.08
	<b>40%</b>	74±4	74±4	69±4	66±3	63±2	62±2
<b>Met.4 0%</b>		---	---	---	---	---	---
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 14**

		<b>BASE DE DATOS IMAGEN</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	91.0±0.5	97±3	61±2	53.2±1.0	44.0±0.8	36.9±0.5
	<b>5%</b>	72±1	79±2	71.9±1.4	61.8±1.0	52.1±0.5	44.4±0.5
	<b>10%</b>	81.0±1.4	78±1	72±1	64±1	52.1±1.5	45.3±0.2
	<b>20%</b>	70±4	75±3	71±3	61.4±1.1	56±3	49±3
	<b>30%</b>	70±2	72±3	70±3	64±2	58±3	51±3
	<b>40%</b>	71±5	53±2	53±3	51±2	48±3	42±2
<b>Met.4 0%</b>		91.2±0	64±3	61±2	53±1	44±1	36.9±0.5
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 15**

		<b>BASE DE DATOS MUSHROOM</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	97.6±0.3	54±5	54±4	56±3	58±2	55.3±1.2
	<b>5%</b>	95.7±0.7	49±2	50±2	53.4±1.5	54.8±1.0	55.0±0.7
	<b>10%</b>	91±6	48±4	51±4	54±3	56±2	56±2
	<b>20%</b>	93.6±0.5	43.6±1.8	44.0±1.5	49.2±1.2	51.1±1.2	51.2±1.3
	<b>30%</b>	92.5±1.6	48±3	49±3	53.9±1.9	56.4±1.7	56.7±1.8
	<b>40%</b>	90.0±0.6	62±2	60.8±1.6	59.6±0.9	59.7±0.4	57.7±0.6
<b>Met.4 0%</b>		---	---	---	---	---	---
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 16**

		<b>BASE DE DATOS MOK1</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	88±5	73±2	70.2±1.3	66±1	64±1	60±1
	<b>5%</b>	73±3	81±2	77±2	73±1	69±1	64±1
	<b>10%</b>	72±5	71±5	68±4	65±3	62±3	58±2
	<b>20%</b>	64±1	68±4	66±3	64±3	63±3	60±2
	<b>30%</b>	59.7±1.3	57±1	57±1	58±1	58±1	56.0±0.5
	<b>40%</b>	62±2	67±2	65±2	64.2±1.3	63.2±1.1	60±1
<b>Met.4 0%</b>		71±0	73±2	70.2±1.3	66±1	64±1	60±1
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 17**

		<b>BASE DE DATOS MOK2</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	70.5±1.5	65±3	65±2	67±2	65.6±1.2	66±1
	<b>5%</b>	70±2	62.5±1.5	62.7±1.2	64.7±1.2	65±1	65±1
	<b>10%</b>	66.5±1.0	68.4±0.5	67.3±0.5	68±1	66±1	66±1
	<b>20%</b>	68±0	62±2	63±2	64.5±1.3	65±1	65.7±0.5
	<b>30%</b>	66.7±0.5	65.5±1.3	66±1	67±1	66±1	66.3±0.5
	<b>40%</b>	60.7±1.5	51.4±1.5	54±1	57±1	59±1	61.1±1.2
<b>Met.4 0%</b>		70.5±0	65±3	65±2	67±2	65.7±1.2	66±1
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 18**

		<b>BASE DE DATOS MOK3</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	89±2	71±3	72±3	72±3	72±3	72±2
	<b>5%</b>	81.2±0.4	84.7±1.6	81.9±1.3	75.7±1.3	72.13±1.14	67±1
	<b>10%</b>	82±3	85.6±1.3	84±2	78±2	75±2	70±2
	<b>20%</b>	88±2	91.4±1.1	90±1	85.9±1.5	83±2	78±2
	<b>30%</b>	91.7±1.5	85±3	85±2	83±2	81±2	78±2
	<b>40%</b>	80±6	75±5	75±5	74±5	74±4	72±4
<b>Met.4 0%</b>		91±0	71±3	72±3	72±3	72±3	72±2
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 19**

		<b>BASE DE DATOS MUSK</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	93.1±0.3	85.01±0.04	85.57±0.09	85.61±0.07	84.81±0.10	84.0±0.3
	<b>5%</b>	81.3±1.8	91.24±0.15	88.71±0.19	84.0±0.4	82.3±0.5	79.4±0.7
	<b>10%</b>	79.9±1.3	90.78±0.08	89.05±0.15	84.0±0.4	82.7±0.6	79.2±1.1
	<b>20%</b>	82.3±0.6	89.11±0.18	88.48±0.13	86.09±0.12	85.41±0.16	82.6±0.3
	<b>30%</b>	88.0±0.8	87.2±0.5	87.7±0.4	86.47±0.12	85.76±0.17	83.11±0.16
	<b>40%</b>	88.2±1.0	85.0±0.3	85.6±0.3	85.16±0.12	85.42±0.14	84.81±0.13
<b>Met.4 0%</b>		---	---	---	---	---	---
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 20**

		<b>BASE DE DATOS PIMA</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	74.9±0.3	45.1±1.5	47.9±1.1	52.2±1.0	55.5±0.6	59.7±0.7
	<b>5%</b>	73.1±1.0	71.1±1.5	71.2±1.1	68.8±1.0	68.7±1.1	69.5±0.9
	<b>10%</b>	71.2±1.2	73.5±1.2	73.8±1.0	70.6±0.9	70.9±0.7	70.4±0.6
	<b>20%</b>	68.0±1.5	69.5±1.8	69.5±1.6	67.6±1.2	68.5±0.9	67.3±1.0
	<b>30%</b>	67.7±0.5	60.8±1.4	62.7±1.1	64.5±0.3	66.5±0.4	66.0±0.3
	<b>40%</b>	68.7±1.1	63.2±3.5	65±2	65.2±1.1	66.4±0.8	67.2±0.6
<b>Met.4 0%</b>		74.9±0.3	45.1±1.5	47.9±1.2	52.2±0.9	55.5±0.6	59.7±0.6
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 21**

		<b>BASE DE DATOS THYROIDES</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	94.63±0.05	93.1±1.0	88.1±1.4	83.7±1.4	81.4±0.9	81.7±0.6
	<b>5%</b>	93.644±0.01	94.63±0.02	93.73±0.02	93.61±0.06	94.29±0.11	93.47±0.17
	<b>10%</b>	93.53±0.02	94.61±0.03	93.66±0.06	93.77±0.05	94.53±0.02	93.861±0.01
	<b>20%</b>	93.53±0.03	94.64±0.04	93.76±0.06	93.83±0.06	94.58±0.02	93.889±0.01
	<b>30%</b>	93.46±0.04	94.52±0.03	93.59±0.02	93.64±0.05	94.51±0.03	93.839±0.01
	<b>40%</b>	93.47±0.03	94.54±0.05	93.63±0.05	93.66±0.07	94.49±0.03	93.83±0.04
<b>Met.4 0%</b>		---	---	---	---	---	---
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 22**

		<b>BASE DE DATOS VOTE</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	94.5±0.2	90.4±0.4	88.6±0.3	83.6±0.2	81.6±0.3	76.10±0.19
	<b>5%</b>	94.2±1.1	92.9±0.6	91.2±0.3	87.0±0.5	86.2±0.4	82.7±0.9
	<b>10%</b>	91±5	88±5	88±5	84±4	80±5	80±4
	<b>20%</b>	96.2±0.7	94.2±0.6	93.53±0.16	90.6±0.2	88.65±0.14	86.98±0.17
	<b>30%</b>	95.2±0.3	93.3±0.5	93.8±0.2	90.7±0.2	88.97±0.10	87.4±0.3
	<b>40%</b>	93.3±0.8	91.6±0.4	91.9±0.3	89.7±0.4	87.8±0.3	87.5±0.4
<b>Met.4 0%</b>		---	---	---	---	---	---
<b>Met.5 0%</b>		---	---	---	---	---	---

**Tabla 23**

		<b>BASE DE DATOS WDBC</b>					
		<b>Test 0%</b>	<b>Test 5%</b>	<b>Test 10%</b>	<b>Test 20%</b>	<b>Test 30%</b>	<b>Test 40%</b>
<b>ENTRENAMIENTO Met.1</b>	<b>0%</b>	93±6	46±4	50±3	62.1±1.1	72.0±1.7	73±2
	<b>5%</b>	87.9±1.9	92.8±0.7	90.3±0.5	86.2±0.2	81.1±0.8	74.5±1.2
	<b>10%</b>	81.1±0.3	92.95±0.16	91.78±0.12	87.85±0.11	80.17±0.16	71.78±0.19
	<b>20%</b>	85.4±0.7	76±3	83.7±1.2	89.53±0.18	83.07±0.08	76.4±0.3
	<b>30%</b>	88.02±0.10	47.5±0.3	57.15±0.18	78.67±0.15	84.25±0.06	80.2±0.2
	<b>40%</b>	93.0±0.4	36.6±0.3	41.7±0.6	57.2±1.0	75.6±0.5	80.03±0.11
<b>Met.4 0%</b>		---	---	---	---	---	---
<b>Met.5 0%</b>		---	---	---	---	---	---

## I.2. Resultados de los métodos 2 y 3

**Tabla 24. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Abalone.**

		BASE DE DATOS ABALONE – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	59.4±0.2	57.6±1.4	57.96±0.16	55.64±0.09	53.60±0.16	50.53±0.15
		56.3±0.2	55.64±0.19	55.1±0.2	54.78±0.15	52.88±0.08	51.71±0.11
		55.0±0.6	55.0±0.5	54.3±0.4	54.5±0.5	51.8±0.8	51.2±0.4
		50.836±0.019	50.689±0.011	50.700±0.009	51.322±0.011	50.9±0	50.172±0.006
		48.59±0.13	48.44±0.17	48.93±0.17	49.5±0.1	49.54±0.18	48.92±0.13
5% 10% 20% 30% 40%	POR 2	---	57.6±1.3	57.2±0.2	54.60±0.14	53.08±0.16	51.13±0.15
		---	55.8±0.2	54.19±0.16	54.21±0.12	52.11±0.12	51.38±0.15
		---	54.9±0.5	53.2±0.2	53.3±0.3	50.6±0.2	48.7±0.7
		---	50.156±0.011	49.994±0.006	50.894±0.006	50.761±0.006	50.167±0.017
		---	47.23±0.18	47.1±0.4	46.80±0.11	47.3±0.2	46.6±0.2
5% 10% 20% 30% 40%	POR 3	---	51±3	50.7±0.3	43.0±0.3	37.6±0.5	34.4±0.8
		---	53.4±0.3	49.8±0.3	45.2±0.3	40.2±0.4	36.1±0.7
		---	52.6±0.7	48.8±0.4	43.8±0.9	37.7±1.3	32.8±1.6
		---	49.689±0.011	48.761±0.006	48.661±0.006	46.53±0	44.567±0.017
		---	46.36±0.04	44.861±0.006	41.7±0.4	39.3±0.5	36.2±0.6
5% 10% 20% 30% 40%	POR 4	---	44±7	43.7±0.3	29.9±0.4	19.0±0.4	11.4±0.3
		---	51.02±0.17	44.9±0.3	36.1±0.4	26.2±0.7	18.7±0.6
		---	51.3±1.0	46.3±0.8	39.0±1.8	31±2	25±2
		---	46.789±0.011	43.528±0.006	37.339±0.006	29.961±0.006	24.039±0.006
		---	44.17±0.05	40.39±0.08	33.9±0.6	27.5±0.8	21.5±0.7
5% 10% 20% 30% 40%	POR 5	---	26±4	26.9±0.2	18.0±0.2	10.9±0.3	6.21±0.19
		---	35.04±0.14	30.79±0.10	23.24±0.09	15.52±0.11	10.60±0.08
		---	25±2	22±2	17.2±1.9	11.8±1.6	9.3±1.4
		---	9.889±0.007	7.7±0	4.83±0	2.5±0	1.6±0
		---	7.08±0.02	5.23±0	3.21±0.02	1.517±0.017	0.728±0.006

**Tabla 25. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Balance.**

		BASE DE DATOS BALANCE – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	84±0	82.5±0	80.60±0.05	75.57±0.19	74.0±0.3	71.1±0.6
		82.53±0.13	80.93±0.10	79.98±0.10	74.92±0.14	74.3±0.3	73.6±0.2
		84±0	82.10±0.03	81.43±0.05	75.85±0.16	75.5±0.2	72.0±0.2
		84.3±0.4	82.0±0.4	80.9±0.4	73.90±0.19	74.0±0.3	71.2±0.2
		81.3±0.6	79.3±0.4	77.2±0.4	73.1±0.2	72.9±0.2	70.13±0.14
5% 10% 20% 30% 40%	POR 2	---	81.22±0.08	75.0±0.2	66.75±0.17	56.2±0.3	46.9±0.2
		---	80.48±0.09	78.40±0.09	70.63±0.14	64.38±0.10	54.1±0.3
		---	82.03±0.02	81.32±0.07	75.65±0.18	75.4±0.2	71.6±0.2
		---	82.0±0.3	80.9±0.4	73.9±0.2	73.8±0.3	71.1±0.3
		---	79.3±0.4	77.3±0.4	72.93±0.14	73.4±0.3	70.3±0.2
5% 10% 20% 30% 40%	POR 3	---	80.32±0.08	74.1±0.2	63.13±0.18	51.6±0.3	40.6±0.2
		---	80.45±0.10	78.00±0.09	68.70±0.18	61.73±0.10	50.3±0.2
		---	82.017±0.017	81.28±0.07	75.5±0.2	75.3±0.2	71.6±0.2
		---	82.0±0.4	80.8±0.4	73.8±0.2	73.8±0.3	70.9±0.3
		---	79.1±0.5	77.0±0.4	72.5±0.3	72.4±0.2	69.1±0.4
5% 10% 20% 30% 40%	POR 4	---	75.8±0	64.8±0	50.3±0	37.7±0	26.1±0
		---	73.97±0.07	64.28±0.08	49.53±0.06	37.23±0.04	25.97±0.04
		---	75.383±0.017	65.52±0.06	50.4±0	36.40±0.06	26.80±0.04
		---	75.9±0.3	66.47±0.10	51.12±0.13	38.5±0.3	28.5±0.2
		---	73.1±0.4	63.9±0.5	49.5±0.3	39.1±0.3	27.80±0.13
5% 10% 20% 30% 40%	POR 5	---	72.3±0	60.77±0.13	45.60±0.13	32.33±0.04	22.30±0.06
		---	72.583±0.017	60.383±0.017	45.60±0.06	32.43±0.04	22.18±0.08
		---	72.87±0.13	61.83±0.16	46.63±0.19	33.45±0.14	23.30±0.17
		---	65.1±0.2	54.9±0.4	41.8±0.3	29.10±0.17	21.68±0.19
		---	49.8±0.3	40.8±0.2	33.1±0.2	22.7±0.2	15.3±0.2

**Tabla 26. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Bands.**

		BASE DE DATOS BANDS – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	77.0±0.8	73.3±0.4	73.28±0.19	67.1±0.3	62.8±1.1	52±2
		73.8±0.5	71.6±0.4	69.0±0.3	65.8±0.5	60.1±0.7	50.4±1.0
		70.7±0.3	69.40±0.18	69.73±0.15	67.53±0.10	68.03±0.06	61.60±0.14
		66.67±0	64.6±0	66.47±0.07	64.00±0.13	67.35±0.10	67.33±0.09
		66.67±0	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
5% 10% 20% 30% 40%	POR 2	---	73.3±0.4	73.3±0.2	67.1±0.3	62.8±1.1	52±2
		---	71.6±0.4	68.9±0.3	65.8±0.5	60.1±0.7	50.4±1.0
		---	69.40±0.18	69.73±0.15	67.55±0.11	68.08±0.07	61.60±0.13
		---	64.6±0	66.47±0.07	64.00±0.13	67.35±0.10	67.33±0.09
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
5% 10% 20% 30% 40%	POR 3	---	73.2±0.4	73.3±0.2	67.0±0.3	62.7±1.1	52±2
		---	71.6±0.4	69.0±0.3	65.8±0.5	60.1±0.8	50.3±1.0
		---	69.40±0.18	69.73±0.15	67.53±0.10	68.00±0.06	61.55±0.13
		---	64.6±0	66.47±0.07	64.00±0.13	67.35±0.10	67.33±0.09
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
5% 10% 20% 30% 40%	POR 4	---	57.9±0.4	41.9±0.2	22.18±0.18	11.55±0.19	2.43±0.05
		---	59.3±0.6	45.8±0.3	29.27±0.12	17.73±0.19	6.4±0.2
		---	64.2±0.3	61.7±0.4	47.95±0.16	38.0±0.4	21.1±0.4
		---	64.55±0.03	66.28±0.07	62.9±0.6	64.7±1.6	62±3
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
5% 10% 20% 30% 40%	POR 5	---	57.9±0.4	41.9±0.2	22.20±0.19	11.53±0.19	2.45±0.05
		---	59.3±0.6	45.8±0.3	29.27±0.12	17.73±0.19	6.4±0.2
		---	64.2±0.3	61.7±0.4	47.95±0.16	37.9±0.4	21.1±0.4
		---	64.55±0.03	66.28±0.07	62.9±0.6	64.7±1.6	62±3
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0



**Tabla 27. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Bupa.**

		BASE DE DATOS BUPA – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	58±0	60.95±0.08	55.97±0.06	57.30±0.06	57.817±0.017	56.33±0.10
		55±0	56.5±0	51.817±0.017	52.317±0.017	53.0±0.1	53.05±0.05
		54±0	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		54±0	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		54±0	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
5% 10% 20% 30% 40%	POR 2	---	60.95±0.08	55.97±0.06	57.30±0.06	57.817±0.017	56.33±0.10
		---	56.5±0	51.817±0.017	52.317±0.017	53.0±0.1	53.05±0.05
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
5% 10% 20% 30% 40%	POR 3	---	60.95±0.08	55.97±0.06	57.30±0.06	57.817±0.017	56.33±0.10
		---	56.5±0	51.817±0.017	52.317±0.017	53.0±0.1	53.05±0.05
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
5% 10% 20% 30% 40%	POR 4	---	48.67±0.11	35.82±0.08	22.25±0.09	13.82±0.13	6.73±0.02
		---	53.78±0.09	43.40±0.14	35.6±0.2	26.9±0.2	21.2±0.2
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
5% 10% 20% 30% 40%	POR 5	---	48.67±0.11	35.82±0.08	22.25±0.09	13.80±0.12	6.73±0.02
		---	53.78±0.09	43.43±0.15	35.7±0.2	26.9±0.2	21.3±0.2
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0

**Tabla 28. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Credito.**

		BASE DE DATOS CREDITO – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	85.8±0.2	85.98±0.11	82.99±0.02	82.49±0.11	80.13±0.09	76.1±0.2
		87.9±0.4	87.5±0.2	83.8±0.2	84.02±0.05	80.92±0.04	78.48±0.06
		88±0	87.95±0	85.15±0	84.1±0	81.65±0	78.85±0
		88±0	87.85±0	85.3±0	85.25±0	82.3±0	79.1±0
		88±0	87.35±0	83.3±0	80.95±0	77.25±0	73.25±0
5% 10% 20% 30% 40%	POR 2	---	85.97±0.11	82.98±0.03	82.50±0.11	80.09±0.09	76.1±0.2
		---	87.5±0.2	83.8±0.2	84.02±0.05	80.92±0.05	78.49±0.05
		---	87.95±0	85.15±0	84.1±0	81.65±0	78.85±0
		---	87.85±0	85.3±0	85.25±0	82.3±0	79.1±0
		---	87.35±0	83.3±0	80.95±0	77.25±0	73.25±0
5% 10% 20% 30% 40%	POR 3	---	85.97±0.11	82.97±0.02	82.49±0.11	80.09±0.09	76.1±0.2
		---	87.5±0.2	83.8±0.2	84.02±0.05	80.90±0.05	78.46±0.07
		---	87.95±0	85.15±0	84.1±0	81.65±0	78.85±0
		---	87.85±0	85.3±0	85.25±0	82.3±0	79.1±0
		---	87.35±0	83.3±0	80.95±0	77.25±0	73.25±0
5% 10% 20% 30% 40%	POR 4	---	80.45±0.17	71.28±0.15	58.48±0.11	45.05±0.13	32.08±0.15
		---	83.34±0.19	74.57±0.10	63.23±0.10	50.85±0.06	39.78±0.07
		---	84.45±0	78.2±0	69.9±0	60.15±0	52.4±0
		---	84.45±0	78.2±0	69.9±0	60.15±0	52.4±0
		---	84.45±0	78.2±0	69.9±0	60.15±0	52.4±0
5% 10% 20% 30% 40%	POR 5	---	80.45±0.17	71.28±0.15	58.47±0.11	45.02±0.14	32.07±0.16
		---	83.34±0.19	74.58±0.10	63.24±0.09	50.85±0.06	39.79±0.07
		---	84.45±0	78.2±0	69.9±0	61.15±0	52.4±0
		---	84.45±0	78.2±0	69.9±0	60.15±0	52.4±0
		---	84.45±0	78.2±0	69.9±0	60.15±0	52.4±0

**Tabla 29. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Display1.**

		BASE DE DATOS DISPLAY1 – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	73.61±0.09	71.9±0.2	67.6±0.2	60.53±0.13	54.8±0.2	45.8±0.4
		73.13±0.17	71.4±0.3	67.10±0.17	60.85±0.07	54.96±0.16	46.02±0.15
		69±3	67±2	63±2	56.3±1.7	50.6±1.5	42.2±1.2
		70.2±0.2	66.8±0.3	63.2±0.5	55.8±0.6	49.3±0.6	41.0±0.4
		66±2	62.7±1.6	59.0±1.7	52.6±1.4	46.6±1.3	39.7±0.9
5% 10% 20% 30% 40%	POR 2	---	71.74±0.19	67.6±0.2	59.81±0.15	52.41±0.14	44.9±0.5
		---	71.00±0.15	67.93±0.18	60.6±0.3	53.7±0.2	47.71±0.14
		---	66±2	61±2	53.7±1.8	46.3±1.6	40.2±1.4
		---	66.2±0.2	62.0±0.6	53.2±0.9	46.0±0.8	39.0±1.2
		---	62.1±1.7	57.6±1.7	50.9±1.5	43.2±1.5	39.2±1.1
5% 10% 20% 30% 40%	POR 3	---	69.7±0.3	63.2±0.4	52.1±0.4	42.3±0.7	32.2±0.3
		---	69.15±0.19	64.0±0.2	53.7±0.2	44.6±0.2	35.4±0.3
		---	63±2	56±2	43.5±1.3	32.8±1.1	23.5±0.8
		---	62.7±0.2	55.3±0.6	41.5±0.8	30.3±0.9	20.9±0.9
		---	58.6±1.6	51.1±1.5	39.4±1.2	28.9±1.2	22.0±1.1
5% 10% 20% 30% 40%	POR 4	---	62.9±0.2	49.62±0.17	30.9±0.4	17.2±0.3	8.74±0.16
		---	62.9±0.3	51.08±0.16	33.03±0.16	18.77±0.15	10.13±0.11
		---	58.6±1.9	46.9±1.7	30.2±1.1	17.6±0.5	9.8±0.4
		---	58.7±0.3	47.9±0.5	30.4±0.8	17.9±0.6	10.2±0.4
		---	55.3±1.6	45.3±1.4	29.8±0.6	17.6±0.6	10.6±0.3
5% 10% 20% 30% 40%	POR 5	---	51.2±0.9	40.3±0.6	23.2±0.5	12.1±0.5	6.6±0.3
		---	51.0±0.5	40.8±0.4	23.7±0.4	12.4±0.3	6.54±0.13
		---	22.3±1.4	17.4±1.3	10.3±0.8	5.1±0.4	2.9±0.3
		---	10.4±1.1	8.1±0.9	4.3±0.5	2.4±0.3	1.28±0.14
		---	0±0	0±0	0±0	0±0	0±0

**Tabla 30. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Display2.**

		BASE DE DATOS DISPLAY2 – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	72.13±0.14	68.78±0.12	62.84±0.19	58.91±0.11	53.60±0.16	45.6±0.2
		72.35±0.08	69.23±0.16	62.94±0.08	58.38±0.13	52.85±0.16	45.66±0.11
		70±2	66±2	62.0±1.7	57.2±1.3	51.2±1.2	44.6±1.1
		67.6±1.5	63.7±1.4	59.5±1.2	54.4±0.7	49.9±0.6	41.5±0.7
		62.5±1.7	58.6±5	54.0±1.6	49.8±1.3	44.0±1.3	35.5±0.9
5% 10% 20% 30% 40%	POR 2	---	68.94±0.07	64.2±0.4	59.08±0.11	51.5±0.3	43.7±0.2
		---	69.6±0.2	64.76±0.13	59.85±0.09	54.1±0.3	46.33±0.14
		---	66.2±1.9	62.9±1.3	55.5±1.0	47.9±1.5	41.4±1.0
		---	63.1±1.6	59.9±1.2	51.8±1.0	45.4±0.9	38.3±0.8
		---	58.0±1.3	52.8±1.2	46.3±0.7	40.3±0.7	35.3±0.6
5% 10% 20% 30% 40%	POR 3	---	66.58±0.09	59.2±0.2	51.7±0.2	42.0±0.3	32.3±0.3
		---	67.11±0.14	59.23±0.09	51.95±0.12	43.6±0.3	34.1±0.2
		---	63.5±1.9	56.6±1.5	45.3±1.2	35.6±1.6	25.9±1.3
		---	59.5±1.4	52.3±1.1	40.1±0.9	31.3±1.0	20.9±0.7
		---	54.6±1.5	46.1±1.3	35.3±1.1	26.6±1.1	17.6±0.9
5% 10% 20% 30% 40%	POR 4	---	58.66±0.12	44.66±0.06	25.32±0.19	12.68±0.16	5.35±0.07
		---	60.38±0.13	48.21±0.15	30.27±0.14	17.23±0.19	9.03±0.13
		---	57.9±1.8	46.7±1.3	29.3±0.9	17.6±0.5	10.0±0.4
		---	55.4±1.3	44.3±0.9	27.8±0.4	17.1±0.3	9.53±0.14
		---	51.4±1.5	41.0±1.4	27.4±1.0	17.3±1.0	9.4±0.6
5% 10% 20% 30% 40%	POR 5	---	49.3±0.4	36.1±0.4	19.73±0.14	9.2±0.3	3.81±0.13
		---	42.56±0.15	31.9±0.3	17.63±0.17	9.22±0.14	4.52±0.07
		---	24.7±1.1	19.1±1.0	11.3±0.4	6.9±0.4	3.43±0.19
		---	20±2	14.9±1.5	9.2±0.9	5.3±0.5	2.6±0.3
		---	0±0	0±0	0±0	0±0	0±0

**Tabla 31. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Glass.**

		BASE DE DATOS GLASS – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	81.25±0	78.67±0.18	75.8±0.2	70.2±0.3	65.2±0.2	58.4±1.0
		78.6±0.5	76.6±0.5	73.3±0.5	68.7±0.3	64.2±0.2	57.3±0.9
		72.1±0.3	70.5±0.3	67.9±0.3	64.1±0.3	60.9±0.3	54.1±0.3
		66.1±0.3	65.7±0.3	62.57±0.09	60.85±0.13	58.05±0.14	52.35±0.11
		61.7±0.3	62.1±0.3	59.4±0.3	57.0±0.3	55.0±0.2	50.8±0.2
5% 10% 20% 30% 40%	POR 2	---	78.23±0.06	73.87±0.13	64.7±0.2	58.0±0.3	49.8±0.3
		---	76.1±0.4	71.8±0.5	64.4±0.3	57.98±0.19	50.40±0.11
		---	70.68±0.10	67.8±0.2	61.2±0.7	58.5±0.8	52.8±1.1
		---	66.1±0.3	62.62±0.18	57.0±0.3	56.2±0.2	54.40±0.16
		---	62.3±0.2	58.67±0.19	54.07±0.14	53.52±0.12	52.55±0.10
5% 10% 20% 30% 40%	POR 3	---	76.88±0.06	71.4±0.2	60.27±0.12	51.3±0.3	42.2±0.2
		---	75.2±0.4	69.6±0.4	60.35±0.19	52.30±0.17	43.15±0.08
		---	69.3±0.4	64.9±0.9	56.9±1.8	52±2	44±3
		---	64.1±0.3	58.60±0.14	50.55±0.17	44.72±0.15	37.90±0.11
		---	60.4±0.2	54.70±0.14	47.43±0.11	41.9±0.2	35.87±0.12
5% 10% 20% 30% 40%	POR 4	---	74.90±0.12	66.4±0.3	51.4±0.4	40.5±0.5	30.5±0.7
		---	74.4±0.4	67.6±0.4	56.1±0.2	47.4±0.4	38.3±0.2
		---	68.5±0.3	63.1±0.2	52.8±0.3	46.2±0.5	37.4±0.7
		---	63.6±0.3	57.68±0.12	47.6±0.3	41.7±0.2	33.5±0.5
		---	60.2±0.3	54.20±0.16	46.5±0.4	40.7±0.3	34.4±0.5
5% 10% 20% 30% 40%	POR 5	---	69.50±0.19	61.2±0.2	47.5±0.6	35.6±0.8	26.9±0.8
		---	69.4±0.2	63.27±0.08	51.65±0.09	42.88±0.18	34.7±0.5
		---	61.7±1.1	55.2±1.3	43.7±1.4	35.8±1.6	28.6±1.3
		---	28.7±0.3	25.53±0.18	20.7±0.2	16.12±0.12	15.03±0.13
		---	15.9±0.5	14.9±0.6	13.2±0.5	10.7±0.4	10.7±0.4

**Tabla 32. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Heart.**

		BASE DE DATOS HEART – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	81.5±0.3	82.2±0.3	80.11±0.08	76.23±0.14	74.12±0.10	69.73±0.10
		82.4±0.4	83.2±0.3	82.43±0.14	78.1±0.3	76.08±0.07	73.9±0.2
		81.2±0.2	81.92±0.18	82.53±0.19	78.43±0.10	76.48±0.10	73.3±0.2
		82.2±0.3	84.08±0.16	80.4±0.4	78.35±0.14	75.7±0.3	73.7±0.3
		78.16±0	79.50±0.06	78.03±0.10	75.87±0.11	73.10±0.11	73.05±0.16
5% 10% 20% 30% 40%	POR 2	---	82.1±0.3	80.03±0.07	76.27±0.13	74.18±0.08	69.73±0.10
		---	83.2±0.3	82.43±0.14	78.0±0.3	76.08±0.07	73.85±0.19
		---	81.92±0.18	82.52±0.19	78.43±0.10	76.50±0.11	73.3±0.2
		---	84.10±0.15	80.4±0.4	78.35±0.14	75.7±0.3	73.7±0.3
		---	79.50±0.06	78.03±0.10	75.85±0.11	73.10±0.11	73.07±0.16
5% 10% 20% 30% 40%	POR 3	---	82.1±0.3	80.03±0.07	76.23±0.14	74.10±0.12	69.73±0.10
		---	83.2±0.3	82.43±0.14	78.0±0.3	76.05±0.06	73.8±0.2
		---	81.92±0.18	82.52±0.19	78.42±0.10	76.48±0.10	73.3±0.2
		---	84.08±0.16	80.4±0.4	78.35±0.14	75.7±0.3	73.7±0.3
		---	79.50±0.06	78.03±0.10	75.85±0.11	73.10±0.11	73.05±0.16
5% 10% 20% 30% 40%	POR 4	---	75.4±0.2	64.47±0.11	46.4±0.3	24.18±0.19	16.3±0.2
		---	77.9±0.3	69.28±0.16	54.20±0.10	33.15±0.04	24.37±0.18
		---	77.12±0.14	71.08±0.20	57.33±0.18	41.1±0.3	32.52±0.18
		---	79.65±0.13	70.7±0.2	59.42±0.13	46.7±0.3	37.3±0.3
		---	75.85±0.17	68.07±0.03	57.78±0.10	45.5±0.3	37.9±0.5
5% 10% 20% 30% 40%	POR 5	---	75.4±0.2	64.48±0.11	46.4±0.3	24.18±0.19	16.3±0.2
		---	77.9±0.3	69.27±0.15	54.22±0.10	33.13±0.06	24.38±0.18
		---	77.12±0.14	71.10±0.19	57.33±0.18	41.1±0.3	32.50±0.17
		---	79.65±0.13	70.7±0.2	59.42±0.13	46.7±0.3	37.3±0.3
		---	75.85±0.17	68.07±0.03	57.78±0.10	45.5±0.3	37.9±0.5

**Tabla 33. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Imagen.**

		BASE DE DATOS IMAGEN – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	89.5±0.4	86.0±0.3	81.8±0.3	68.7±0.3	61.3±0.5	51±3
		88.6±0.3	84.8±0.3	81.3±0.4	70.4±0.4	63.6±0.2	53.6±0.3
		72.0±1.2	70.7±0.9	68.1±0.8	59.5±0.6	56.5±0.6	48.6±0.4
		69±3	68±3	66±3	58±2	54.4±1.7	46.8±1.2
		59±2	60±2	58.5±1.8	53.7±1.5	51.8±1.1	45.8±1.0
5% 10% 20% 30% 40%	POR 2	---	85.58±0.18	78.1±0.5	66.5±1.1	53.7±1.5	41±3
		---	85.8±0.3	79.9±0.3	71.1±0.4	62.4±0.7	50.4±0.8
		---	71.9±0.9	68.0±0.7	62.3±0.5	56.9±0.7	49.7±1.1
		---	67±3	63±2	54.7±1.8	46.6±1.4	40.9±1.4
		---	55±2	49.2±1.7	42.0±1.6	34.5±1.3	32.0±1.1
5% 10% 20% 30% 40%	POR 3	---	81.5±0.2	71.2±0.6	52.9±0.9	40.2±1.3	26.7±1.7
		---	81.39±0.18	72.6±0.2	57.7±0.3	46.7±0.4	33.9±0.6
		---	66.9±0.7	59.7±0.6	48.0±0.7	40.4±1.0	29.6±1.0
		---	63±3	55±2	40.4±1.4	30.6±0.9	21.7±0.8
		---	51.9±1.8	43.8±1.6	32.0±1.4	22.7±0.9	15.6±0.6
5% 10% 20% 30% 40%	POR 4	---	71.0±0.3	53.2±0.3	26.6±0.2	11.99±0.13	3.70±0.15
		---	73.9±0.4	57.7±0.4	32.7±0.3	17.63±0.19	6.79±0.13
		---	63.4±0.7	51.9±0.4	33.5±0.3	21.0±0.3	10.2±0.3
		---	59±3	48±2	29.0±1.6	18.0±1.1	8.6±0.6
		---	49.8±1.7	39.2±1.5	24.5±1.4	14.7±1.0	7.6±0.9
5% 10% 20% 30% 40%	POR 5	---	62.6±0.6	45.7±0.5	24.4±0.4	11.73±0.18	4.13±0.13
		---	59.0±0.7	45.6±0.7	26.2±0.5	14.2±0.2	5.70±0.16
		---	43.34±0.12	35.93±0.17	22.8±0.3	14.05±0.19	6.5±0.2
		---	24.1±0.3	19.5±0.2	10.78±0.15	6.85±0.18	2.87±0.09
		---	10.0±1.6	6.4±1.2	3.6±0.9	2.0±0.5	0.8±0.2

**Tabla 34. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Mushroom.**

		BASE DE DATOS MUSHROOM – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
<b>5%</b>	<b>POR 1</b>	99.389±0.011	98.01±0.10	97.31±0.17	93.2±0.2	87.6±0.4	81.1±0.6
		98.67±0.17	97.66±0.11	97.11±0.19	94.2±0.2	90.2±0.2	84.3±0.5
		96.39±0.06	95.59±0.11	94.34±0.09	93.2±0.3	90.3±0.6	86.9±0.8
		89.9±0.9	89.4±0.7	88.7±0.6	85.7±0.5	84.1±0.5	82.2±0.3
		84.61±0.03	84.489±0.019	85.02±0.02	83.233±0.017	83.800±0.009	82.961±0.013
<b>5%</b>	<b>POR 2</b>	---	98.01±0.10	97.31±0.17	93.2±0.2	87.6±0.4	81.1±0.6
		---	97.66±0.11	97.11±0.19	94.2±0.2	90.2±0.2	84.2±0.5
		---	95.59±0.11	94.34±0.09	93.2±0.3	90.3±0.6	86.9±0.8
		---	89.4±0.7	88.7±0.6	85.7±0.5	84.1±0.5	82.2±0.3
		---	84.489±0.019	85.02±0.02	82.233±0.017	83.800±0.009	82.961±0.013
<b>5%</b>	<b>POR 3</b>	---	98.01±0.10	97.31±0.17	93.2±0.2	87.6±0.4	81.1±0.6
		---	97.66±0.11	97.11±0.19	94.2±0.2	90.2±0.2	84.2±0.5
		---	95.59±0.11	94.34±0.09	93.2±0.3	90.3±0.6	86.9±0.8
		---	89.4±0.7	88.7±0.6	85.7±0.5	84.1±0.5	82.2±0.3
		---	84.489±0.019	85.02±0.02	82.233±0.017	83.800±0.009	82.961±0.013
<b>5%</b>	<b>POR 4</b>	---	94.1±0.3	86.7±0.5	64.0±0.8	39.9±0.8	19.4±0.7
		---	93.98±0.10	88.1±0.3	68.6±0.3	46.9±0.3	26.4±0.4
		---	90.4±0.3	84.0±0.6	70.7±0.6	56.6±0.5	42.7±0.6
		---	85.6±0.7	81.7±0.7	70.68±0.19	60.78±0.19	50.8±1.1
		---	80.66±0.03	75.97±0.04	66.667±0.015	58.06±0.04	51.11±0.10
<b>5%</b>	<b>POR 5</b>	---	94.1±0.3	86.7±0.5	64.0±0.8	39.9±0.8	19.4±0.7
		---	93.98±0.10	88.1±0.3	68.6±0.3	46.9±0.3	26.4±0.4
		---	90.4±0.3	84.0±0.6	70.7±0.6	56.6±0.5	42.7±0.6
		---	85.6±0.7	81.7±0.7	70.68±0.19	60.78±0.19	50.8±1.1
		---	80.66±0.03	75.97±0.04	66.667±0.015	58.06±0.04	51.11±0.10



**Tabla 35. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Mok1.**

		BASE DE DATOS MOK1 – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	79±0	79.067±0.017	76.467±0.017	74.167±0.017	70.358±0.008	67.35±0
		69.5±0.4	69.7±0.4	68.4±0.3	66.1±0.4	65.2±0.2	61.8±0.3
		81.0±1.3	80.5±1.1	77.6±1.0	73.7±0.6	69.6±0.7	65.2±0.7
		79±0	79.05±0	76.45±0	74.15±0	70.35±0	67.35±0
		79±0	79.26±0.03	76.74±0.04	74.38±0.14	70.33±0.05	67.45±0.14
5% 10% 20% 30% 40%	POR 2	---	79.067±0.017	76.467±0.017	74.167±0.017	70.358±0.008	67.35±0
		---	69.7±0.4	68.3±0.3	66.1±0.3	65.3±0.2	61.9±0.2
		---	80.5±1.1	77.6±1.0	73.7±0.6	69.6±0.7	65.2±0.6
		---	79.05±0	76.45±0	74.15±0	70.35±0	67.35±0
		---	79.26±0.03	76.73±0.04	74.38±0.14	70.33±0.05	67.45±0.09
5% 10% 20% 30% 40%	POR 3	---	79.067±0.017	76.467±0.017	74.167±0.017	70.358±0.008	67.35±0
		---	69.7±0.4	68.3±0.3	66.0±0.3	65.3±0.2	61.8±0.3
		---	80.5±1.1	77.6±1.0	73.7±0.6	69.5±0.7	65.2±0.7
		---	79.05±0	76.45±0	74.15±0	70.35±0	67.35±0
		---	79.26±0.03	76.74±0.04	74.38±0.14	70.33±0.05	67.45±0.09
5% 10% 20% 30% 40%	POR 4	---	77.3±0.5	71.65±0	64.35±0	54.7±0	46.75±0
		---	66.3±0.3	61.6±0.2	53.3±0.2	45.1±0.2	36.3±0.3
		---	78.0±0.8	72.6±0.7	63.6±0.4	53.8±0.5	44.5±1.2
		---	76.75±0	71.65±0	64.35±0	54.7±0	46.75±0
		---	76.75±0	71.65±0	64.35±0	54.7±0	46.75±0
5% 10% 20% 30% 40%	POR 5	---	77.3±0.5	71.65±0	64.35±0	54.7±0	46.75±0
		---	66.3±0.3	61.6±0.2	53.4±0.2	45.1±0.3	36.3±0.3
		---	78.0±0.8	72.6±0.7	63.6±0.4	53.8±0.5	44.5±1.2
		---	76.75±0	71.65±0	64.35±0	54.7±0	46.75±0
		---	76.75±0	71.65±0	64.35±0	54.7±0	46.75±0

**Tabla 36. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Mok2.**

		BASE DE DATOS MOK2 – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	68±0	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		68±0	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		68±0	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		68±0	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		68±0	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
5% 10% 20% 30% 40%	POR 2	---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
5% 10% 20% 30% 40%	POR 3	---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
5% 10% 20% 30% 40%	POR 4	---	69.25±0	68.23±0.10	68.9±0.3	66.7±0.4	65.4±0.9
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
5% 10% 20% 30% 40%	POR 5	---	69.25±0	68.23±0.10	68.9±0.3	66.7±0.4	65.4±0.9
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0

**Tabla 37. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Mok3.**

		BASE DE DATOS MOK3 – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	81.7±0.4	82.5±0.3	82.2±0.3	79.3±0.3	79.40±0.19	77.72±0.06
		89.7±0.6	89.1±0.4	88.0±0.4	84.9±0.4	83.8±0.2	80.63±0.14
		90.7±0.6	89.7±0.4	88.6±0.5	85.2±0.6	84.0±0.4	80.6±0.4
		79.8±0.5	81.2±0.4	80.5±0.4	77.9±0.3	78.2±0.2	76.8±0.3
		89±0	87.95±0	86.75±0	82.95±0	81.3±0	77.75±0
5% 10% 20% 30% 40%	POR 2	---	82.5±0.3	82.2±0.3	79.3±0.3	79.42±0.18	77.73±0.06
		---	89.1±0.4	88.0±0.4	84.9±0.4	83.8±0.2	80.63±0.14
		---	89.7±0.4	88.6±0.5	85.2±0.6	84.0±0.5	80.6±0.4
		---	81.2±0.4	80.5±0.4	77.9±0.3	78.2±0.2	76.8±0.3
		---	87.95±0	86.75±0	82.95±0	81.3±0	77.75±0
5% 10% 20% 30% 40%	POR 3	---	82.5±0.3	82.2±0.3	79.3±0.3	79.40±0.19	77.73±0.06
		---	89.1±0.4	88.0±0.4	84.9±0.4	83.8±0.2	80.63±0.14
		---	89.7±0.4	88.6±0.5	85.2±0.6	84.0±0.5	80.6±0.4
		---	81.2±0.4	80.5±0.4	77.9±0.3	78.2±0.2	76.8±0.3
		---	87.95±0	86.75±0	82.95±0	81.3±0	77.75±0
5% 10% 20% 30% 40%	POR 4	---	75.9±0.3	69.7±0.2	56.58±0.16	45.00±0.14	35.48±0.15
		---	83.2±0.5	76.9±0.3	64.0±0.5	52.5±0.5	43.3±0.6
		---	84.0±0.5	78.1±0.4	66.2±0.4	55.7±0.6	46.6±0.7
		---	74.7±0.4	69.24±0.19	57.15±0.10	47.0±0.4	38.8±0.8
		---	82.9±0	77.65±0	66.1±0	56±0	48.05±0
5% 10% 20% 30% 40%	POR 5	---	75.9±0.3	69.7±0.2	56.58±0.16	45.00±0.14	35.48±0.15
		---	83.2±0.5	76.9±0.3	63.9±0.5	52.5±0.5	43.3±0.7
		---	84.0±0.5	78.1±0.4	66.2±0.4	55.7±0.6	46.6±0.7
		---	74.7±0.4	69.24±0.19	57.15±0.10	47.0±0.4	38.8±0.8
		---	82.9±0	77.65±0	66.1±0	56±0	48.05±0

**Tabla 38. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Musk.**

		BASE DE DATOS MUSK – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	92.56±0.08	93.44±0.10	92.06±0.13	89.2±0.2	84.9±1.6	78±6
		90.66±0.09	90.99±0.11	90.71±0.14	89.20±0.07	87.4±0.3	83.5±1.7
		85.000±0.017	84.922±0.011	85.322±0.007	86.98±0.09	87.98±0.17	70.3±1.1
		83.73±0	83.73±0	84.13±0	84.25±0.04	84.97±0.14	87.01±0.13
		83.73±0	83.73±0	84.13±0	84.03±0	83.57±0	84.34±0.07
5% 10% 20% 30% 40%	POR 2	---	93.44±0.10	92.06±0.13	89.1±0.2	84.9±1.6	78±6
		---	91.00±0.11	90.72±0.15	89.20±0.07	87.4±0.3	83.5±1.7
		---	84.922±0.011	85.322±0.007	86.98±0.09	87.99±0.17	70.4±1.1
		---	83.73±0	84.13±0	84.25±0	84.97±0.14	87.01±0.13
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.35±0.07
5% 10% 20% 30% 40%	POR 3	---	93.44±0.10	92.06±0.13	89.1±0.2	84.9±1.6	78±6
		---	90.99±0.11	90.71±0.14	89.20±0.07	87.4±0.3	83.4±1.7
		---	84.922±0.011	85.322±0.007	86.98±0.09	87.98±0.17	70.3±1.1
		---	83.73±0	84.13±0	84.25±0	84.97±0.14	87.01±0.13
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.34±0.07
5% 10% 20% 30% 40%	POR 4	---	86.93±0.16	72.8±0.4	34.7±1.2	8.2±0.8	0.96±0.16
		---	87.61±0.08	82.36±0.07	63.4±0.3	32.9±0.9	9.3±0.6
		---	83.96±0.03	83.972±0.006	81.54±0.05	63.7±0.3	26.3±0.6
		---	83.73±0	84.13±0	83.967±0.009	82.278±0.019	70.82±0.15
		---	83.73±0	84.13±0	84.03±0	83.4±0	82.76±0.03
5% 10% 20% 30% 40%	POR 5	---	86.94±0.16	72.8±0.4	34.74±1.19	8.2±0.8	0.96±0.16
		---	87.61±0.08	82.37±0.08	63.4±0.3	32.9±0.9	9.3±0.6
		---	83.96±0.03	83.972±0.006	81.54±0.05	63.7±0.3	26.3±0.6
		---	83.73±0	84.13±0	83.967±0.009	82.278±0.019	70.82±0.14
		---	83.73±0	84.13±0	84.03±0	83.4±0	82.76±0.03

**Tabla 39. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Pima.**

		BASE DE DATOS PIMA – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	71.6±0.6	69.0±0.4	72.1±0.3	70.7±0.4	68.3±1.0	67.8±1.1
		73.6±0	71.083±0.011	72.8±0.4	70.7±0.5	70.4±0.5	68.9±0.7
		65.80±0.14	65.57±0.15	65.85±0.08	65.83±0.03	67.06±0.06	66.68±0.05
		65.2±0	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		65.2±0	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
5% 10% 20% 30% 40%	POR 2	---	69.0±0.4	72.1±0.3	70.7±0.4	68.3±1.0	67.8±1.1
		---	71.083±0.011	72.8±0.4	70.7±0.5	70.4±0.5	68.9±0.7
		---	65.57±0.15	65.85±0.08	65.83±0.03	67.05±0.06	66.68±0.05
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
5% 10% 20% 30% 40%	POR 3	---	69.0±0.4	72.1±0.3	70.7±0.4	68.3±1.0	67.8±1.1
		---	71.083±0.011	72.8±0.4	70.7±0.5	70.4±0.5	68.9±0.7
		---	65.57±0.15	65.85±0.08	65.83±0.03	67.05±0.06	66.68±0.05
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
5% 10% 20% 30% 40%	POR 4	---	61.1±0.5	55.8±0.5	40.7±0.6	34±7	18.1±0.9
		---	66.7±0.2	64.4±0.2	53.3±0.4	44.2±0.6	35.1±0.5
		---	64.06±0.13	62.11±0.05	56.78±0.02	49.62±0.10	39.82±0.12
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
5% 10% 20% 30% 40%	POR 5	---	61.1±0.5	55.8±0.5	40.7±0.6	34±7	18.1±0.9
		---	66.7±0.2	64.4±0.2	53.3±0.4	44.2±0.6	35.1±0.5
		---	64.06±0.13	62.12±0.05	56.78±0.02	49.64±0.10	39.83±0.11
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0

**Tabla 40. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Thyroides.**

		BASE DE DATOS THYROIDES – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	93.87±0	94.528±0.006	93.6±0	93.272±0.006	94.37±0	93.57±0
		93.27±0	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		93.27±0	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		93.27±0	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		93.27±0	94±0	93.3±0	93.03±0	94.23±0	93.47±0
5% 10% 20% 30% 40%	POR 2	---	94.47±0	93.6±0	93.206±0.006	94.37±0	93.53±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
5% 10% 20% 30% 40%	POR 3	---	94.47±0	93.6±0	93.206±0.006	94.37±0	93.53±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
5% 10% 20% 30% 40%	POR 4	---	94.1±0	92.4±0	87.14±0.04	79.96±0.04	66.28±0.10
		---	94±0	93.23±0	92.778±0.011	92.51±0.04	88.34±0.08
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
5% 10% 20% 30% 40%	POR 5	---	93.772±0	91.461±0.006	84.48±0.02	75.64±0.04	61.01±0.04
		---	94±0	93.072±0.006	91.717±0.017	90.04±0.05	83.07±0.07
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0

**Tabla 41. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Vote.**

		BASE DE DATOS VOTE – METODO2					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	95.3±0.2	95.62±0.03	94.80±0.05	92.30±0.16	89.77±0.08	84.8±0.3
		94.7±0.2	94.97±0.11	93.80±0.10	92.73±0.05	91.43±0.15	87.9±0.2
		95.3±0.3	94.73±0.15	93.88±0.05	91.65±0.03	90.88±0.06	89.57±0.13
		91.83±0.17	92.32±0.10	92.07±0.03	90.42±0.13	91.05±0.12	88.62±0.14
		93±0	92±0	92.283±0.017	91.67±0.02	90.217±0.017	91.02±0.07
5% 10% 20% 30% 40%	POR 2	---	95.62±0.03	94.82±0.06	92.32±0.16	89.77±0.08	84.8±0.3
		---	94.97±0.11	93.80±0.10	92.70±0.05	91.40±0.15	87.9±0.2
		---	94.73±0.15	93.88±0.05	91.63±0.02	90.87±0.05	89.55±0.13
		---	92.32±0.10	92.07±0.03	90.43±0.12	91.05±0.12	88.63±0.13
		---	92±0	92.283±0.017	91.67±0.02	90.217±0.017	91.02±0.07
5% 10% 20% 30% 40%	POR 3	---	95.62±0.03	94.80±0.05	92.30±0.16	89.77±0.08	84.8±0.3
		---	94.97±0.11	93.80±0.10	92.70±0.05	91.40±0.15	87.9±0.2
		---	94.73±0.15	93.88±0.05	91.63±0.02	90.87±0.05	89.55±0.13
		---	92.32±0.10	92.07±0.03	90.42±0.13	91.05±0.12	88.62±0.14
		---	92±0	92.283±0.017	91.67±0.02	90.217±0.017	91.02±0.07
5% 10% 20% 30% 40%	POR 4	---	93.27±0.04	87.05±0.12	74.82±0.19	56.7±0.3	39.4±0.3
		---	92.70±0.13	87.75±0.10	77.85±0.07	64.73±0.15	47.48±0.14
		---	93.9±0.2	88.77±0.16	79.83±0.09	70.27±0.10	54.63±0.13
		---	89.90±0.07	86.23±0.06	78.1±0.3	70.18±0.18	57.37±0.19
		---	88.9±0	85.65±0.10	75.0±0.2	64.27±0.09	53.95±0.12
5% 10% 20% 30% 40%	POR 5	---	93.28±0.03	87.07±0.12	74.85±0.18	56.7±0.3	39.4±0.3
		---	92.70±0.13	87.75±0.10	77.85±0.06	64.73±0.15	47.50±0.15
		---	93.9±0.2	88.78±0.16	79.85±0.09	70.25±0.09	54.62±0.13
		---	89.88±0.08	86.23±0.06	78.1±0.3	70.18±0.18	57.37±0.19
		---	88.9±0	85.65±0.10	75.0±0.2	64.27±0.09	53.95±0.12

**Tabla 42. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 2. Base de datos Wdbc.**

		BASE DE DATOS WDBC – METODO2						
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%	
<b>5%</b>	<b>POR 1</b>	99.375±0	98.283±0.017	98.38±0.03	97.70±0.05	95.20±0.07	91.02±0.08	
		<b>10%</b>	96.875±0	97.48±0.05	96.77±0.06	96.57±0.12	94.43±0.12	90.4±0.2
		<b>20%</b>	93.23±0.19	90.85±0.13	90.08±0.16	89.0±0.2	86.7±0.4	82.1±0.4
		<b>30%</b>	89.0±0.3	87.7±0.3	86.9±0.3	87.35±0.16	85.2±0.2	80.2±0.2
		<b>40%</b>	84.3±1.0	83.2±0.6	80.7±0.9	81.0±0.9	79.2±0.7	75.6±0.6
<b>5%</b>	<b>POR 2</b>	---	98.283±0.017	98.38±0.03	97.70±0.05	95.20±0.07	91.02±0.08	
		<b>10%</b>	---	97.45±0.03	96.78±0.07	96.57±0.12	94.45±0.13	90.5±0.2
		<b>20%</b>	---	90.87±0.12	90.10±0.17	89.1±0.2	86.6±0.4	82.2±0.4
		<b>30%</b>	---	87.7±0.3	86.9±0.3	87.32±0.16	85.18±0.17	80.15±0.19
		<b>40%</b>	---	83.2±0.6	80.7±0.9	81.0±0.9	79.2±0.7	75.6±0.6
<b>5%</b>	<b>POR 3</b>	---	98.283±0.017	98.38±0.03	97.70±0.05	95.20±0.07	91.02±0.08	
		<b>10%</b>	---	97.45±0.03	96.77±0.06	96.57±0.12	94.43±0.12	90.4±0.2
		<b>20%</b>	---	90.85±0.13	90.08±0.16	89.0±0.2	86.6±0.4	82.1±0.4
		<b>30%</b>	---	87.7±0.3	86.9±0.3	87.32±0.16	85.15±0.18	80.1±0.2
		<b>40%</b>	---	83.2±0.6	80.7±0.9	81.0±0.9	79.2±0.7	75.6±0.6
<b>5%</b>	<b>POR 4</b>	---	87.30±0.09	72.98±0.13	43.08±0.07	22.45±0.02	11.60±0.04	
		<b>10%</b>	---	88.87±0.04	77.15±0.09	49.67±0.14	27.37±0.14	15.38±0.17
		<b>20%</b>	---	85.15±0.12	79.7±0.4	70.2±0.6	57.5±0.9	47.9±1.3
		<b>30%</b>	---	83.9±0.3	77.9±0.3	65.8±0.9	53.2±1.1	40.1±1.2
		<b>40%</b>	---	78.4±0.8	73.0±1.0	65.7±1.2	57.4±1.1	49.8±1.0
<b>5%</b>	<b>POR 5</b>	---	87.30±0.09	72.98±0.13	43.07±0.06	22.43±0.02	11.60±0.04	
		<b>10%</b>	---	88.87±0.04	77.15±0.09	49.67±0.14	27.33±0.14	15.33±0.16
		<b>20%</b>	---	85.17±0.11	79.7±0.4	70.1±0.6	57.5±0.9	47.9±1.3
		<b>30%</b>	---	83.8±0.3	77.9±0.3	65.7±0.9	53.3±1.1	40.1±1.2
		<b>40%</b>	---	78.4±0.8	73.0±1.0	65.7±1.2	57.4±1.1	49.8±1.0



**Tabla 43. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Abalone.**

		BASE DE DATOS ABALONE – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	55.0±0.3	54.9±0.3	53.74±0.07	53.9±0.3	51.3±0.4	49.7±0.2
		40±4	40±4	40±4	40±4	39±3	38±3
		34.55±0	34.53±0	34.37±0	34.9±0	34.4±0	32.97±0
		34.55±0	34.53±0	34.37±0	34.9±0	34.4±0	32.97±0
		32.95±0	33.17±0	33.07±0	32.9±0	34±0	33.43±0
5% 10% 20% 30% 40%	POR 2	---	55.4±0.2	53.53±0.19	53.40±0.06	50.5±0.2	48.5±0.5
		---	40±3	40±3	40±3	38±3	37±3
		---	34.53±0	34.37±0	34.9±0	34.4±0	32.97±0
		---	34.53±0	34.37±0	34.9±0	34.4±0	32.97±0
		---	33.17±0	33.07±0	32.9±0	34±0	33.43±0
5% 10% 20% 30% 40%	POR 3	---	53.2±0.2	49.34±0.10	45.03±0.05	39.1±0.4	33.8±0.9
		---	40±3	40±3	39±3	37±2	35±2
		---	34.53±0	34.37±0	34.9±0	34.4±0	32.97±0
		---	34.53±0	34.37±0	34.9±0	34.4±0	32.97±0
		---	33.17±0	33.07±0	32.9±0	34±0	33.43±0
5% 10% 20% 30% 40%	POR 4	---	52.1±0.4	47.8±0.2	42.2±0.8	34.7±0.9	28.7±0.8
		---	39±3	38±3	36.9±1.3	33.9±0.9	31.2±1.6
		---	34.53±0	34.37±0	34.9±0	34.4±0	32.97±0
		---	34.53±0	34.37±0	34.9±0	34.4±0	32.97±0
		---	33.17±0	33.07±0	32.9±0	34±0	33.43±0
5% 10% 20% 30% 40%	POR 5	---	29±2	26.7±1.9	20.7±1.5	15.7±1.2	12.0±0.7
		---	4±3	4±2	2.6±1.8	1.6±1.2	1.2±0.9
		---	0±0	0±0	0±0	0±0	0±0
		---	0±0	0±0	0±0	0±0	0±0
		---	0±0	0±0	0±0	0±0	0±0

**Tabla 44. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Balance.**

		BASE DE DATOS BALANCE – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	84±0	82.5±0	80.683±0.017	75.78±0.12	74.47±0.15	72.52±0.13
		82.93±0.17	81.27±0.12	79.95±0.16	74.9±0.3	73.6±0.3	71.1±0.5
		74.8±0.5	72.6±0.3	71.6±0.4	66.17±0.16	66.2±0.4	62.9±0.5
		44±0	45±0	44.5±0	45.2±0	45.4±0	47.4±0
		44±0	45±0	44.5±0	45.2±0	45.4±0	47.4±0
5% 10% 20% 30% 40%	POR 2	---	82.1±0	79.6±0	71.4±0	64.1±0	52.7±0
		---	81.28±0.12	80.1±0.2	74.9±0.2	73.8±0.2	71.6±0.4
		---	72.6±0.4	71.3±0.4	65.9±0.2	64.8±0.3	61.8±0.3
		---	45±0	44.5±0	45.2±0	45.4±0	47.4±0
		---	45±0	44.5±0	45.2±0	45.4±	47.4±0
5% 10% 20% 30% 40%	POR 3	---	82.1±0	79.3±0	69.4±0	61.2±0	49.1±0
		---	81.27±0.12	79.77±0.18	74.2±0.2	72.4±0.2	69.0±0.4
		---	72.1±0.4	70.8±0.4	65.0±0.2	64.2±0.3	60.9±0.3
		---	45±0	44.5±0	45.2±0	45.4±0	47.4±0
		---	45±0	44.5±0	45.2±0	45.4±	47.4±0
5% 10% 20% 30% 40%	POR 4	---	75.8±0	64.8±0	50.3±0	37.7±0	26.1±0
		---	74.17±0.08	64.60±0.11	49.47±0.08	37.18±0.05	25.50±0.08
		---	69.0±0.3	61.2±0.3	49.45±0.11	38.33±0.09	29.95±0.08
		---	45±0	44.5±0	45.2±0	45.4±0	47.4±0
		---	45±0	44.5±0	45.2±0	45.4±	47.4±0
5% 10% 20% 30% 40%	POR 5	---	73.13±0.06	62.02±0.07	47.25±0.05	34.52±0.10	23.92±0.03
		---	72.57±0.02	61.22±0.08	46.62±0.05	33.27±0.08	22.95±0.15
		---	40.5±0.9	33.8±0.8	26.6±0.8	18.4±0.4	14.7±0.5
		---	45±0	44.5±0	45.2±0	45.4±0	47.4±0
		---	0±0	0±0	0±0	0±0	0±0

**Tabla 45. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Bands.**

		BASE DE DATOS BANDS – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	72.8±0.2	69.8±0.2	70.0±0.2	68.3±0.2	70.1±0.3	69.1±0.3
		66.67±0	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		66.67±0	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		66.67±0	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		66.67±0	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
5% 10% 20% 30% 40%	POR 2	---	69.8±0.2	70.0±0.2	68.3±0.2	70.2±0.3	69.1±0.3
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
5% 10% 20% 30% 40%	POR 3	---	69.8±0.2	70.0±0.2	68.3±0.2	70.1±0.3	69.1±0.3
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
5% 10% 20% 30% 40%	POR 4	---	63.3±0.3	57.1±0.4	44.1±0.8	30.9±1.1	17.7±1.3
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
5% 10% 20% 30% 40%	POR 5	---	63.3±0.3	57.1±0.3	44.1±0.8	30.9±1.1	17.7±1.3
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0
		---	64.6±0	66.4±0	63.8±0	67.2±0	67.2±0

**Tabla 46. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Bupa.**

		BASE DE DATOS BUPA – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	54±0	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		54±0	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		54±0	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		54±0	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		54±0	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
5% 10% 20% 30% 40%	POR 2	---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
5% 10% 20% 30% 40%	POR 3	---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
5% 10% 20% 30% 40%	POR 4	---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
5% 10% 20% 30% 40%	POR 5	---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0
		---	55.9±0	51.4±0	51.6±0	52.8±0	53.2±0

**Tabla 47. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Credito.**

		BASE DE DATOS CREDITO – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	84.83±0.17	85.24±0.11	82.68±0.11	83.83±0.19	80.9±0.3	77.7±0.8
		88.2±0.4	86.8±0.2	83.7±0.2	80.83±0.13	76.72±0.05	71.92±0.09
		88±0	87.95±0	85.15±0	84.1±0	81.65±0	78.85±0
		88±0	87.35±0	83.3±0	80.95±0	77.25±0	73.25±0
		55±0	55.55±0	54.9±0	53.9±0	54.05±0	52.2±0
5% 10% 20% 30% 40%	POR 2	---	85.23±0.11	82.69±0.10	83.84±0.19	80.9±0.3	77.7±0.8
		---	86.8±0.2	83.74±0.19	80.83±0.13	76.72±0.05	71.92±0.09
		---	87.95±0	85.15±0	84.1±0	81.65±0	60.15±0
		---	87.35±0	83.3±0	80.95±0	77.25±0	73.25±0
		---	55.55±0	54.9±0	53.9±0	54.05±0	52.2±0
5% 10% 20% 30% 40%	POR 3	---	84.5±0.7	82.68±0.11	83.83±0.19	80.8±0.3	77.7±0.8
		---	86.8±0.2	83.74±0.19	80.83±0.13	76.72±0.05	71.92±0.09
		---	87.95±0	85.15±0	84.1±0	81.65±0	60.15±0
		---	87.35±0	83.3±0	80.95±0	77.25±0	73.25±0
		---	55.55±0	54.9±0	53.9±0	54.05±0	52.2±0
5% 10% 20% 30% 40%	POR 4	---	80.47±0.09	71.95±0.07	61.2±0.4	48.5±0.8	37.5±1.4
		---	82.7±0.2	75.82±0.19	63.96±0.10	52.33±0.16	41.36±0.10
		---	84.45±0	78.2±0	69.7±0	60.15±0	52.4±0
		---	84.45±0	78.2±0	69.9±0	60.15±0	52.4±0
		---	55.55±0	54.9±0	53.9±0	54.05±0	52.2±0
5% 10% 20% 30% 40%	POR 5	---	80.7±0.4	71.96±0.08	61.2±0.4	48.5±0.8	37.5±1.4
		---	82.7±0.2	75.82±0.19	63.96±0.10	52.33±0.16	41.36±0.10
		---	84.45±0	78.2±0	69.9±0	60.15±0	52.4±0
		---	84.45±0	78.2±0	69.9±0	60.15±0	52.4±0
		---	55.55±0	54.9±0	53.9±0	54.05±0	52.2±0

**Tabla 48. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Display1.**

		BASE DE DATOS DISPLAY1 – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	73.40±0.15	71.4±0.5	67.0±0.3	59.75±0.16	53.8±0.4	45.1±0.3
		60±10	57±9	55±9	49±8	43±7	37±5
		9.1±0	9.3±0	8.8±0	8.7±0	9.55±0	9.3±0
		11±0	12.05±0	11±0	10.75±0	11.25±0	11.85±0
		9.1±0	9.35±0	8.8±0	8.7±0	9.55±0	9.3±0
5% 10% 20% 30% 40%	POR 2	---	71.0±0.3	66.7±0.3	58.9±0.5	51.4±0.6	44.2±0.9
		---	57±9	54±9	48±7	40±6	34±5
		---	9.35±0	8.8±0	8.7±0	9.55±0	9.3±0
		---	12.05±0	11±0	10.75±0	11.25±0	11.85±0
		---	9.35±0	8.8±0	8.7±0	9.55±0	9.3±0
5% 10% 20% 30% 40%	POR 3	---	68.6±0.5	61.9±0.4	50.7±0.4	40.7±0.7	31.2±0.9
		---	55±9	49±8	39±6	29±4	22±2
		---	9.35±0	8.8±0	8.7±0	9.55±0	9.3±0
		---	12.05±0	11±0	10.75±0	11.25±0	11.85±0
		---	9.35±0	8.8±0	8.7±0	9.55±0	9.3±0
5% 10% 20% 30% 40%	POR 4	---	63.1±0.5	50.5±0.4	32.6±0.4	18.8±0.4	9.8±0.2
		---	51±8	42±6	29±4	17.8±1.4	11.55±0.19
		---	9.35±0	8.8±0	8.7±0	9.55±0	9.3±0
		---	12.05±0	11±0	10.75±0	11.25±0	11.85±0
		---	9.35±0	8.8±0	8.7±0	9.55±0	9.3±0
5% 10% 20% 30% 40%	POR 5	---	41.9±0.6	33.5±0.7	20.1±0.5	10.2±0.4	6.0±0.3
		---	19±4	15±3	9±2	4.68±1.04	2.8±0.7
		---	0±0	0±0	0±0	0±0	0±0
		---	0±0	0±0	0±0	0±0	0±0
		---	0±0	0±0	0±0	0±0	0±0

**Tabla 49. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Display2.**

		BASE DE DATOS DISPLAY2 – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	72.65±0.07	69.58±0.13	63.29±0.09	59.58±0.09	53.3±0.2	45.42±0.16
		30±11	28±10	27±9	25±9	23±7	20±6
		10.2±0	10±0	10.65±0	11.15±0	10.2±0	10.7±0
		9.8±0	10.05±0	9.1±0	10.65±0	10.3±0	8.65±0
		9.8±0	10.05±0	9.1±0	10.65±0	10.3±0	8.65±0
5% 10% 20% 30% 40%	POR 2	---	69.63±0.10	64.87±0.04	59.95±0.17	51.6±0.2	44.3±0.3
		---	28±10	28±9	26±9	24±7	21±6
		---	10±0	10.65±0	11.15±0	10.2±0	10.7±0
		---	10.05±0	9.1±0	10.65±0	10.3±0	8.65±0
		---	10.05±0	9.1±0	10.65±0	10.3±0	8.65±0
5% 10% 20% 30% 40%	POR 3	---	67.18±0.10	59.25±0.07	50.93±0.16	40.7±0.3	31.4±0.2
		---	27±10	25±8	21±7	18±5	15±3
		---	10±0	10.65±0	11.15±0	10.2±0	10.7±0
		---	10.05±0	9.1±0	10.65±0	10.3±0	8.65±0
		---	10.05±0	9.1±0	10.65±0	10.3±0	8.65±0
5% 10% 20% 30% 40%	POR 4	---	60.55±0.14	48.03±0.14	30.63±0.08	17.35±0.11	8.82±0.05
		---	25±9	22±7	16±4	12.3±1.9	9.5±0.9
		---	10±0	10.65±0	11.15±0	10.2±0	10.7±0
		---	10.05±0	9.1±0	10.65±0	10.3±0	8.65±0
		---	10.05±0	9.1±0	10.65±0	10.3±0	8.65±0
5% 10% 20% 30% 40%	POR 5	---	46.37±0.18	34.4±0.2	19.06±0.12	10.35±0.12	4.78±0.07
		---	7±5	6±4	3±2	1.9±1.4	1.1±0.9
		---	0±0	0±0	0±0	0±0	0±0
		---	0±0	0±0	0±0	0±0	0±0
		---	0±0	0±0	0±0	0±0	0±0

**Tabla 50. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Glass.**

		BASE DE DATOS GLASS – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	79.2±0.7	77.0±0.5	73.9±0.5	69.6±0.4	64.83±0.17	56.98±0.16
		71.6±0.7	70.2±0.7	67.5±0.6	64.0±0.5	61.1±0.4	54.2±0.4
		52±5	51±5	50±5	48±4	47±4	44±3
		28.125±0	28.2±0	28.8±0	29.7±0	29±0	30.7±0
		28.125±0	28.2±0	28.8±0	29.7±0	29±0	30.7±0
5% 10% 20% 30% 40%	POR 2	---	76.5±0.5	71.8±0.5	63.5±0.2	57.4±0.3	49.3±0.5
		---	70.4±0.5	67.4±0.4	61.0±0.2	59.2±1.0	54.5±1.5
		---	52±2	51±5	47±4	47±4	48±4
		---	28.2±0	28.8±0	29.7±0	29±0	30.7±0
		---	28.2±0	28.8±0	29.7±0	29±0	30.7±0
5% 10% 20% 30% 40%	POR 3	---	75.7±0.4	69.9±0.4	60.27±0.10	52.75±0.08	43.33±0.13
		---	68.8±0.8	63.9±0.7	55.3±0.9	49.7±1.2	42.0±1.4
		---	50±5	48±4	41±3	38±2	34.3±1.0
		---	28.2±0	28.8±0	29.7±0	29±0	30.7±0
		---	28.2±0	28.8±0	29.7±0	29±0	30.7±0
5% 10% 20% 30% 40%	POR 4	---	74.8±0.4	68.7±0.5	56.6±0.4	47.1±0.5	38.1±0.4
		---	68.5±0.8	63.2±0.7	54.0±0.8	47.4±0.7	39.2±0.6
		---	50±5	48±4	41±3	38±2	34.2±0.9
		---	28.2±0	28.8±0	29.7±0	29±0	30.7±0
		---	28.2±0	28.8±0	29.7±0	29±0	30.7±0
5% 10% 20% 30% 40%	POR 5	---	69.5±0.3	63.6±0.4	51.8±0.8	42.7±1.3	35.0±1.6
		---	54±5	48±4	39±4	32±3	27±2
		---	18±4	18±4	16±3	13±3	14±3
		---	0±0	0±0	0±0	0±0	0±0
		---	0±0	0±0	0±0	0±0	0±0



**Tabla 51. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Heart.**

		BASE DE DATOS HEART – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	83.9±0	83.97±0.03	82.48±0.06	78.48±0.09	75.57±0.11	70.88±0.14
		82.0±0.5	82.7±0.3	81.9±0.3	78.4±0.2	76.80±0.16	75.33±0.13
		77.4±0.2	76.62±0.16	78.98±0.17	75.52±0.10	71.53±0.19	68.58±0.14
		54.023±0	54.4±0	52.8±0	54.2±0	53.3±0	52.4±0
		45.98±0	45.6±0	47.2±0	45.8±0	46.7±0	47.6±0
5% 10% 20% 30% 40%	POR 2	---	83.97±0.03	82.50±0.07	78.47±0.10	75.53±0.12	70.92±0.15
		---	82.7±0.3	81.9±0.3	78.4±0.2	76.78±0.17	75.32±0.13
		---	76.62±0.16	76.98±0.17	75.52±0.10	71.53±0.19	68.57±0.13
		---	54.4±0	52.8±0	54.2±0	53.3±0	52.4±0
		---	45.6±0	47.2±0	45.8±0	46.7±0	47.6±0
5% 10% 20% 30% 40%	POR 3	---	83.97±0.03	82.48±0.06	78.47±0.10	75.53±0.12	70.88±0.14
		---	82.7±0.3	81.9±0.3	78.4±0.2	76.78±0.17	75.32±0.13
		---	76.62±0.16	76.98±0.17	75.52±0.10	71.53±0.19	68.57±0.13
		---	54.4±0	52.8±0	54.2±0	53.3±0	52.4±0
		---	45.6±0	47.2±0	45.8±0	46.7±0	47.6±0
5% 10% 20% 30% 40%	POR 4	---	79.42±0.09	68.43±0.12	53.1±0.3	31.9±0.2	23.6±0.2
		---	78.4±0.3	70.9±0.4	57.6±0.5	40.9±0.5	30.6±0.4
		---	72.8±0.3	67.4±0.2	57.5±0.3	45.1±0.7	38.0±0.6
		---	54.4±0	52.8±0	54.2±0	53.3±0	52.4±0
		---	45.6±0	47.2±0	45.8±0	46.7±0	47.6±0
5% 10% 20% 30% 40%	POR 5	---	79.42±0.09	70.1±1.8	53.0±0.3	31.9±0.2	23.6±0.2
		---	78.4±0.3	70.9±0.4	57.7±0.5	40.9±0.5	30.6±0.4
		---	72.8±0.3	67.4±0.2	57.5±0.3	45.1±0.7	38.0±0.6
		---	54.4±0	52.8±0	54.2±0	53.3±0	52.4±0
		---	45.6±0	47.2±0	45.8±0	46.7±0	47.6±0

**Tabla 52. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Imagen.**

		BASE DE DATOS IMAGEN – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
<b>5%</b>	<b>POR 1</b>	79.1±1.4	76.5±1.1	71.4±0.9	59.4±0.4	53.7±0.5	43.4±0.4
		40±11	40±10	39±10	35±8	33±7	28±5
		12.5±0	13.4±0	13.45±0	12.55±0	12.65±0	12.6±0
		14.18±0	14.45±0	14.8±0	13.75±0	13.7±0	13±0
		14.5±0	15.85±0	14.5±0	14.75±0	13.6±0	15.55±0
<b>5%</b>	<b>POR 2</b>	---	77.7±1.1	72.6±0.9	63.4±0.7	51.7±0.6	39.8±0.6
		---	40±10	39±9	34±7	31±5	25±3
		---	13.4±0	13.45±0	12.55±0	12.65±0	12.6±0
		---	14.45±0	14.8±0	13.75±0	13.7±0	13±0
		---	15.85±0	14.5±0	14.75±0	13.6±0	15.55±0
<b>5%</b>	<b>POR 3</b>	---	72.3±0.9	63.6±0.8	46.7±0.7	34.5±1.0	23.6±0.8
		---	39±10	36±8	29±6	24±3	16.0±1.7
		---	13.4±0	13.45±0	12.55±0	12.65±0	12.6±0
		---	14.45±0	14.8±0	13.75±0	13.7±0	13±0
		---	15.85±0	14.5±0	14.75±0	13.6±0	15.55±0
<b>5%</b>	<b>POR 4</b>	---	68.4±0.8	55.7±0.5	31.4±0.2	17.2±0.4	7.2±0.3
		---	38±9	35±8	27±5	21±2	12.5±0.7
		---	13.4±0	13.45±0	12.55±0	12.65±0	12.6±0
		---	14.45±0	14.8±0	13.75±0	13.7±0	13±0
		---	15.85±0	14.5±0	14.75±0	13.6±0	15.55±0
<b>5%</b>	<b>POR 5</b>	---	50.8±0.3	41.57±0.14	24.84±0.14	13.48±0.13	6.09±0.06
		---	18±11	15±9	10±6	7±4	2.9±1.8
		---	0±0	0±0	0±0	0±0	0±0
		---	0±0	0±0	0±0	0±0	0±0
		---	0±0	0±0	0±0	0±0	0±0

**Tabla 53. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Mushroom.**

		BASE DE DATOS MUSHROOM – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	98.4±0.2	97.59±0.09	96.4±0.2	93.5±0.3	89.7±0.4	83.3±0.4
		96.1±0.3	95.76±0.16	94.47±0.13	92.98±0.15	89.00±0.15	80.2±0.3
		94.6±1.7	94.0±1.6	93.0±1.3	92.1±1.1	89.1±1.0	85.9±0.6
		62.27±0	61.3±0	62.1±0	60.5±0	61.67±0	63.07±0
		62.27±0	61.3±0	62.1±0	60.5±0	61.67±0	63.07±0
5% 10% 20% 30% 40%	POR 2	---	97.59±0.09	96.4±0.2	93.5±0.3	89.6±0.4	83.3±0.4
		---	95.76±0.16	94.47±0.13	92.97±0.14	88.99±0.15	80.2±0.3
		---	94.0±1.6	93.0±1.3	92.1±1.1	89.1±1.0	85.9±0.6
		---	61.3±0	62.1±0	60.5±0	61.67±0	63.07±0
		---	61.3±0	62.1±0	60.5±0	61.67±0	63.07±0
5% 10% 20% 30% 40%	POR 3	---	97.59±0.09	96.4±0.2	93.5±0.3	89.6±0.4	83.3±0.4
		---	95.76±0.16	94.47±0.13	92.97±0.14	88.99±0.15	80.2±0.3
		---	94.0±1.6	93.0±1.3	92.1±1.1	89.1±1.0	85.9±0.6
		---	61.3±0	62.1±0	60.5±0	61.67±0	63.07±0
		---	61.3±0	62.1±0	60.5±0	61.67±0	63.07±0
5% 10% 20% 30% 40%	POR 4	---	94.1±0.3	86.88±0.19	63.9±0.4	40.6±0.6	21.0±0.8
		---	92.53±0.12	87.27±0.16	70.16±0.09	50.06±0.12	30.8±0.2
		---	89.3±1.7	83.5±1.4	69.4±1.1	54.3±0.6	39.5±0.6
		---	61.3±0	62.1±0	60.5±0	61.67±0	63.07±0
		---	61.3±0	62.1±0	60.5±0	61.67±0	63.07±0
5% 10% 20% 30% 40%	POR 5	---	94.1±0.3	86.88±0.19	63.9±0.4	40.6±0.6	21.0±0.8
		---	92.52±0.11	87.26±0.16	70.14±0.09	50.04±0.12	30.8±0.2
		---	89.3±1.7	83.4±1.4	69.8±1.1	54.3±0.6	39.5±0.6
		---	61.3±0	62.1±0	60.5±0	61.67±0	63.07±0
		---	61.3±0	62.1±0	60.5±0	61.67±0	63.07±0

**Tabla 54. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Mok1.**

		BASE DE DATOS MOK1 – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	79±0	79.05±0	76.45±0	74.15±0	70.35±0	67.35±0
		75.3±1.8	75.5±1.7	73.5±1.5	71.8±1.3	69.1±0.6	66.1±0.7
		49±0	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
		49±0	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
		49±0	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
5% 10% 20% 30% 40%	POR 2	---	79.05±0	76.45±0	74.15±0	70.35±0	67.35±0
		---	75.5±1.7	73.5±1.5	71.8±1.3	69.1±0.6	66.1±0.7
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
5% 10% 20% 30% 40%	POR 3	---	79.05±0	76.45±0	74.15±0	70.35±0	67.35±0
		---	75.5±1.7	73.5±1.5	71.8±1.3	69.1±0.6	66.1±0.7
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
5% 10% 20% 30% 40%	POR 4	---	76.75±0	71.65±0	64.35±0	54.7±0	46.75±0
		---	72.7±1.9	67.8±1.9	60.5±1.8	51.5±1.5	43.0±1.7
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
5% 10% 20% 30% 40%	POR 5	---	76.75±0	71.65±0	64.35±0	54.7±0	46.75±0
		---	72.7±1.9	67.8±1.9	60.5±1.8	51.5±1.5	43.0±1.7
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0
		---	49.35±0	49.1±0	49.8±0	47.45±0	49.35±0

**Tabla 55. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Mok2.**

		BASE DE DATOS MOK2 – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	68±0	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		68±0	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		68±0	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		68±0	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		68±0	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
5% 10% 20% 30% 40%	POR 2	---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
5% 10% 20% 30% 40%	POR 3	---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
5% 10% 20% 30% 40%	POR 4	---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
5% 10% 20% 30% 40%	POR 5	---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0
		---	69.25±0	68.45±0	69.6±0	67.6±0	67.4±0

**Tabla 56. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Mok3.**

		BASE DE DATOS MOK3 – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	79±0	80.55±0	80.15±0	77.5±0	78.2±0	77±0
		89±0	88.5±0	87.45±0	84.05±0	83.35±0	80.25±0
		89.5±1.6	87.4±1.5	85.5±1.4	81.5±1.1	79.2±0.9	75.8±0.6
		54±0	54±0	53.2±0	53.9±0	55.65±0	53.9±0
		54±0	54±0	53.2±0	53.9±0	55.65±0	53.9±0
5% 10% 20% 30% 40%	POR 2	---	80.55±0	80.15±0	77.5±0	78.2±0	77±0
		---	88.5±0	87.45±0	84.05±0	83.35±0	80.25±0
		---	87.4±1.5	85.5±1.4	81.5±1.1	79.2±0.9	75.8±0.6
		---	54±0	53.2±0	53.9±0	55.65±0	53.9±0
		---	54±0	53.2±0	53.9±0	55.65±0	53.9±0
5% 10% 20% 30% 40%	POR 3	---	80.55±0	80.15±0	77.5±0	78.2±0	77±0
		---	88.5±0	87.45±0	84.05±0	83.35±0	80.25±0
		---	87.4±1.5	85.5±1.4	81.5±1.1	79.2±0.9	75.8±0.6
		---	54±0	53.2±0	53.9±0	55.65±0	53.9±0
		---	54±0	53.2±0	53.9±0	55.65±0	53.9±0
5% 10% 20% 30% 40%	POR 4	---	74.05±0	69±0	57.35±0	47.8±0	40.25±0
		---	82.85±0.03	77.57±0.05	66.18±0.05	55.6±0.2	47.3±0.5
		---	83.7±1.2	78.8±0.7	68.4±0.3	59.3±0.3	51.1±0.7
		---	54±0	53.2±0	53.9±0	55.65±0	53.9±0
		---	54±0	53.2±0	53.9±0	55.65±0	53.9±0
5% 10% 20% 30% 40%	POR 5	---	74.05±0	69±0	57.35±0	47.8±0	40.25±0
		---	82.85±0.03	77.57±0.05	66.18±0.05	55.6±0.2	47.3±0.5
		---	83.7±1.2	78.8±0.7	68.4±0.3	59.3±0.3	51.1±0.7
		---	54±0	53.2±0	53.9±0	55.65±0	53.9±0
		---	54±0	53.2±0	53.9±0	55.65±0	53.9±0

**Tabla 57. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Musk.**

		BASE DE DATOS MUSK – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
<b>5%</b>	<b>POR 1</b>	90.556±0.014	91.02±0.03	91.06±0.04	85.42±0.19	71.3±0.6	46.7±1.1
		87.14±0.02	86.433±0.015	86.106±0.018	84.989±0.007	84.211±0.016	84.456±0.016
		83.73±0	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
		83.73±0	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
		83.73±0	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
<b>5%</b>	<b>POR 2</b>	---	91.02±0.03	91.05±0.04	85.44±0.18	71.3±0.6	46.7±1.1
		---	86.433±0.015	86.106±0.018	84.989±0.007	84.211±0.016	84.456±0.016
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
<b>5%</b>	<b>POR 3</b>	---	91.02±0.03	91.05±0.04	85.42±0.19	71.3±0.6	46.7±1.1
		---	86.433±0.015	86.106±0.018	84.989±0.007	84.211±0.016	84.456±0.016
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
<b>5%</b>	<b>POR 4</b>	---	87.02±0.02	81.34±0.16	59.3±0.5	30.8±0.7	9.7±0.4
		---	85.683±0.014	84.506±0.016	79.79±0.08	72.6±0.3	62.6±0.4
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
		---	83.733±0	84.13±0	84.03±0	83.57±0	84.17±0
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.167±0
<b>5%</b>	<b>POR 5</b>	---	87.02±0.02	81.34±0.16	59.3±0.5	30.8±0.7	9.7±0.4
		---	85.683±0.014	84.500±0.019	79.79±0.08	72.7±0.3	62.6±0.4
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0
		---	83.73±0	84.13±0	84.03±0	83.57±0	84.17±0

**Tabla 58. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Pima.**

		BASE DE DATOS PIMA – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	76±0	74.18±0.15	74.84±0.11	73.48±0.03	73.24±0.05	73.300±0.013
		65.2±0	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		65.2±0	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		65.2±0	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		65.2±0	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
5% 10% 20% 30% 40%	POR 2	---	74.18±0.15	74.84±0.11	73.48±0.03	73.24±0.05	73.300±0.013
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
5% 10% 20% 30% 40%	POR 3	---	74.18±0.15	74.84±0.11	73.48±0.03	73.24±0.05	73.300±0.013
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
5% 10% 20% 30% 40%	POR 4	---	69.85±0.13	66.60±0.06	57.96±0.07	48.52±0.05	40.84±0.10
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
5% 10% 20% 30% 40%	POR 5	---	69.83±0.13	66.60±0.06	57.96±0.07	48.53±0.06	40.83±0.10
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0
		---	65.2±0	65.2±0	65.4±0	66.4±0	65.95±0



**Tabla 59. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Thyroides.**

		BASE DE DATOS THYROIDES – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	93.5±0.3	94.18±0.18	93.5±0.2	93.3±0.2	94.32±0.09	93.63±0.16
		93.27±0	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		93.27±0	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		93.27±0	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		93.27±0	94±0	93.3±0	93.03±0	94.23±0	93.47±0
5% 10% 20% 30% 40%	POR 2	---	94.17±0.17	93.5±0.2	93.21±0.18	94.31±0.08	93.54±0.07
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
5% 10% 20% 30% 40%	POR 3	---	94.17±0.17	93.49±0.19	93.21±0.18	94.29±0.06	93.54±0.07
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
5% 10% 20% 30% 40%	POR 4	---	94.13±0.13	93.27±0.03	92.5±0.6	92.7±1.5	91±3
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
5% 10% 20% 30% 40%	POR 5	---	93.89±0.11	93.0±0.3	91.9±1.2	92±2	90±4
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0
		---	94±0	93.3±0	93.03±0	94.23±0	93.47±0

**Tabla 60. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Vote.**

		BASE DE DATOS VOTE – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	95±0	95.25±0.05	94.00±0.05	92.32±0.04	89.75±0.13	84.72±0.14
		94±0	94.483±0.017	93.617±0.017	92.05±0.03	91.9±0.2	88.8±0.2
		94±0	94.283±0.017	93.03±0.02	91.27±0.02	90.12±0.08	86.42±0.11
		92±0	92.43±0.02	91.15±0.04	87.83±0.08	86.40±0.04	81.73±0.06
		93±0	90.6±0	89±0	85.5±0	85.3±0	81±0
5% 10% 20% 30% 40%	POR 2	---	95.25±0.05	94.00±0.05	92.32±0.04	89.73±0.12	84.72±0.14
		---	94.483±0.017	93.617±0.017	92.05±0.03	91.9±0.2	88.8±0.3
		---	94.283±0.017	93.03±0.02	91.27±0.02	90.08±0.09	86.42±0.09
		---	92.43±0.02	91.15±0.04	87.83±0.08	86.40±0.04	81.73±0.07
		---	90.6±0	89±0	85.5±0	85.3±0	81±0
5% 10% 20% 30% 40%	POR 3	---	95.25±0.05	94.00±0.05	92.32±0.04	89.73±0.12	84.72±0.14
		---	94.483±0.017	93.617±0.017	92.05±0.03	91.9±0.2	88.8±0.3
		---	94.283±0.017	93.03±0.02	91.27±0.02	90.07±0.09	86.40±0.10
		---	92.43±0.02	91.15±0.04	87.83±0.08	86.40±0.04	81.72±0.07
		---	90.6±0	89±0	85.5±0	85.3±0	81±0
5% 10% 20% 30% 40%	POR 4	---	93.48±0.06	88.15±0.04	78.13±0.05	62.5±0.3	47.2±0.3
		---	92.48±0.03	88.08±0.03	79.47±0.07	66.25±0.06	51.10±0.06
		---	93±0	89.17±0.03	81.58±0.15	71.45±0.05	56.28±0.15
		---	89.7±0	86.383±0.017	78.00±0.06	70.90±0.05	57.62±0.07
		---	88.1±0	83.8±0	73.1±0	63±0	54.3±0
5% 10% 20% 30% 40%	POR 5	---	93.48±0.06	88.15±0.04	78.13±0.05	62.5±0.3	47.2±0.3
		---	92.48±0.03	88.08±0.03	79.47±0.07	66.25±0.06	51.10±0.06
		---	93±0	89.17±0.03	81.58±0.15	71.45±0.05	56.28±0.15
		---	89.7±0	86.383±0.017	78.00±0.06	70.88±0.06	57.62±0.07
		---	88.1±0	83.8±0	73.1±0	63±0	54.3±0

**Tabla 61. Porcentaje de correcta clasificación para los diferentes porcentajes de entradas desconocidas en el entrenamiento y test en el método 3. Base de datos Wdbc.**

		BASE DE DATOS WDBC – METODO3					
		Test 0%	Test 5%	Test 10%	Test 20%	Test 30%	Test 40%
5% 10% 20% 30% 40%	POR 1	98.13±0.16	97.93±0.10	97.98±0.05	96.58±0.12	93.3±0.3	87.8±0.5
		95.6±0.3	95.92±0.15	95.82±0.09	94.60±0.14	92.1±0.4	86.7±0.4
		91.25±0	88.8±0	87.8±0	87.1±0	84.3±0	79.8±0
		64.375±0	64.6±0	65±0	64.5±0	65.1±0	63.1±0
		64.375±0	64.6±0	65±0	64.5±0	65.1±0	63.1±0
5% 10% 20% 30% 40%	POR 2	---	97.93±0.10	97.98±0.05	96.57±0.12	93.3±0.3	87.8±0.5
		---	95.92±0.15	95.82±0.09	94.60±0.14	92.2±0.4	86.7±0.4
		---	88.8±0	87.8±0	87.1±0	84.3±0	79.8±0
		---	64.6±0	65±0	64.5±0	65.1±0	63.1±0
		---	64.6±0	65±0	64.5±0	65.1±0	63.1±0
5% 10% 20% 30% 40%	POR 3	---	97.93±0.10	97.98±0.05	96.57±0.12	93.3±0.3	87.8±0.5
		---	95.92±0.15	95.82±0.09	94.60±0.14	92.1 ±0.4	86.7±0.4
		---	88.8±0	87.8±0	87.1±0	84.3±0	79.8±0
		---	64.6±0	65±0	64.5±0	65.1±0	63.1±0
		---	64.6±0	65±0	64.5±0	65.1±0	63.1±0
5% 10% 20% 30% 40%	POR 4	---	88.80±0.15	76.4±0.3	46.4±0.4	24.40±0.15	12.55±0.14
		---	90.4±0.2	82.70±0.12	61.20±0.17	39.7±0.3	23.5±0.3
		---	84.8±0	81.2±0	73.7±0	65.4±0	57±0
		---	64.6±0	65±0	64.5±0	65.1±0	63.1±0
		---	64.6±0	65±0	64.5±0	65.1±0	63.1±0
5% 10% 20% 30% 40%	POR 5	---	88.80±0.15	76.4±0.3	46.4±0.4	24.40±0.15	12.55±0.14
		---	90.4±0.2	82.72±0.12	61.20±0.17	39.7±0.3	23.5±0.3
		---	84.8±0	81.2±0	73.7±0	65.4±0	57±0
		---	64.6±0	65±0	64.5±0	65.1±0	63.1±0
		---	64.6±0	65±0	64.5±0	65.1±0	63.1±0

### I.3. Comparación entre los métodos 2 y 3

*Tabla 62. Resultados de la comparación entre los métodos 2 y 3 para igual porcentaje de entradas desconocidas en el entrenamiento y en el test. Criterio de clasificación POR1.*

Base de datos	Medida	POR	Ishibuchi	Hernández	Categoría
ABALONE	POR1	5%	54.9±0.3	57.6±1.4	3
		10%	40±4	55.1±0.2	4
		20%	34.9±0	54.5±0.5	4
		30%	34.4±0	50.9±0	4
		40%	33.43±0	48.92±0.13	4
BALANCE	POR1	5%	82.5±0	82.5±0	1
		10%	79.95±0.16	79.98±0.10	2
		20%	66.17±0.16	75.85±0.16	3
		30%	45.4±0	74.0±0.3	4
		40%	47.4±0	70.13±0.14	4
BANDS	POR1	5%	69.8±0.2	73.3±0.4	3
		10%	66.4±0	69.0±0.3	3
		20%	63.8±0	67.53±0.10	3
		30%	67.2±0	67.35±0.10	2
		40%	67.2±0	67.2±0	1
BUPA	POR1	5%	55.9±0	60.95±0.08	3
		10%	51.4±0	51.817±0.017	2
		20%	51.6±0	51.6±0	1
		30%	52.8±0	52.8±0	1
		40%	53.2±0	53.2±0	1
CREDITO	POR1	5%	85.24±0.11	85.98±0.11	2
		10%	83.7±0.2	83.8±0.2	2
		20%	84.1±0	84.1±0	1
		30%	77.25±0	82.3±0	3
		40%	52.2±0	73.25±0	4
DISPLAY1	POR1	5%	71.4±0.5	71.9±0.2	2
		10%	55±9	67.10±0.17	4
		20%	8.7±0	56.3±1.7	4
		30%	11.25±0	49.3±0.6	4
		40%	9.3±0	39.7±0.9	4
DISPLAY2	POR1	5%	69.58±0.13	68.78±0.12	5
		10%	27±9	62.94±0.08	4
		20%	11.15±0	57.2±1.3	4
		30%	10.3±0	49.9±0.6	4
		40%	8.65±0	35.5±0.	4
GLASS	POR1	5%	77.0±0.5	78.67±0.18	2
		10%	67.5±0.6	73.3±0.5	3
		20%	48±4	64.1±0.3	4
		30%	29±0	58.05±0.14	4
		40%	30.7±0	50.8±0.2	4
HEART	POR1	5%	83.97±0.03	82.2±0.3	5
		10%	81.9±0.3	82.43±0.14	2
		20%	75.52±0.10	78.43±0.10	3
		30%	53.3±0	75.7±0.3	4
		40%	47.6±0	73.05±0.16	4

Tabla 62. (Continuación)

Base de datos	Medida	POR	Ishibuchi	Hernández	Categoría
IMAGEN	POR1	5%	76.5±1.1	86.0±0.3	3
		10%	39±10	81.3±0.4	4
		20%	12.55±0	59.5±0.6	4
		30%	13.7±0	54.4±1.7	4
		40%	15.55±0	45.8±1.0	4
MUSHROOM	POR1	5%	97.59±0.09	98.01±0.10	2
		10%	94.47±0.13	97.11±0.19	3
		20%	92.1±1.1	93.2±0.3	2
		30%	61.67±0	84.1±0.5	4
		40%	63.07±0	82.961±0.013	4
MOK1	POR1	5%	79.05±0	79.067±0.017	2
		10%	73.5±1.5	68.4±0.3	6
		20%	49.8±0	73.7±0.6	4
		30%	47.45±0	70.35±0	4
		40%	49.35±0	67.45±0.09	4
MOK2	POR1	5%	69.25±0	69.25±0	1
		10%	68.45±0	68.45±0	1
		20%	69.6±0	69.6±0	1
		30%	67.6±0	67.6±0	1
		40%	67.4±0	67.4±0	1
MOK3	POR1	5%	80.55±0	82.5±0.3	2
		10%	87.45±0	88.0±0.4	2
		20%	81.5±1.1	85.2±0.6	3
		30%	55.65±0	78.2±0.2	4
		40%	53.9±0	77.75±0	4
MUSK	POR1	5%	91.02±0.03	93.44±0.10	3
		10%	86.106±0.018	90.71±0.14	3
		20%	84.03±0	86.98±0.09	3
		30%	83.57±0	84.97±0.14	2
		40%	84.17±0	84.34±0.07	2
PIMA	POR1	5%	74.18±0.15	69.0±0.4	6
		10%	65.2±0	72.8±0.4	3
		20%	65.4±0	65.83±0.03	2
		30%	66.4±0	66.4±0	1
		40%	65.95±0	65.95±0	1
THYROIDES	POR1	5%	94.18±0.18	94.528±0.006	2
		10%	93.3±0	93.3±0	1
		20%	93.03±0	93.03±0	1
		30%	94.23±0	94.23±0	1
		40%	93.47±0	93.47±0	1
VOTE	POR1	5%	95.25±0.05	95.62±0.03	2
		10%	93.617±0.017	93.80±0.10	2
		20%	91.27±0.02	91.65±0.03	2
		30%	86.40±0.04	91.05±0.12	3
		40%	81±0	91.02±0.07	4
WDBC	POR1	5%	97.93±0.10	98.283±0.017	2
		10%	95.82±0.09	96.77±0.06	2
		20%	87.1±0	89.0±0.2	2
		30%	65.1±0	85.2±0.2	4
		40%	63.1±0	75.6±0.6	4

**Tabla 63. Resultados de la comparación entre los métodos 2 y 3 para igual porcentaje de entradas desconocidas en el entrenamiento y en el test. Criterio de clasificación POR2.**

Base de datos	Medida	POR	Ishibuchi	Hernández	Categoría
<b>ABALONE</b>	POR2	5%	55.4±0.2	57.6±1.3	3
		10%	40±3	54.19±0.16	4
		20%	34.9±0	53.3±0.3	4
		30%	34.4±0	50.761±0.006	4
		40%	33.43±0	46.6±0.2	4
<b>BALANCE</b>	POR2	5%	82.1±0	81.22±0.08	5
		10%	80.1±0.2	78.40±0.09	5
		20%	65.9±0.2	75.65±0.18	3
		30%	45.4±0	73.8±0.3	4
		40%	47.4±0	70.3±0.2	4
<b>BANDS</b>	POR2	5%	69.8±0.2	73.3±0.4	3
		10%	66.4±0	68.9±0.3	3
		20%	63.8±0	67.55±0.11	3
		30%	67.2±0	67.35±0.10	2
		40%	67.2±0	67.2±0	1
<b>BUPA</b>	POR2	5%	55.9±0	60.95±0.08	3
		10%	51.4±0	51.817±0.017	2
		20%	51.6±0	51.6±0	1
		30%	52.8±0	52.8±0	1
		40%	53.2±0	53.2±0	1
<b>CREDITO</b>	POR2	5%	85.23±0.11	85.97±0.11	2
		10%	83.74±0.19	83.8±0.2	2
		20%	84.1±0	84.1±0	1
		30%	77.25±0	82.3±0	3
		40%	52.2±0	73.25±0	4
<b>DISPLAY1</b>	POR2	5%	71.0±0.3	71.74±0.19	2
		10%	54±9	67.93±0.18	4
		20%	8.7±0	53.7±1.8	4
		30%	11.25±0	46.0±0.8	4
		40%	9.3±0	39.2±1.1	4
<b>DISPLAY2</b>	POR2	5%	69.63±0.10	68.94±0.07	5
		10%	28±9	64.76±0.13	4
		20%	11.15±0	55.5±1.0	4
		30%	10.3±0	45.4±0.9	4
		40%	8.65±0	35.3±0.6	4
<b>GLASS</b>	POR2	5%	76.5±0.5	78.23±0.06	2
		10%	67.4±0.4	71.8±0.5	3
		20%	47±4	61.2±0.7	4
		30%	29±0	56.2±0.2	4
		40%	30.7±0	52.55±0.10	4
<b>HEART</b>	POR2	5%	83.97±0.03	82.1±0.3	5
		10%	81.9±0.3	82.43±0.14	2
		20%	75.52±0.10	78.43±0.10	3
		30%	53.3±0	75.7±0.3	4
		40%	47.6±0	73.07±0.16	4

Tabla 63. (Continuación).

Base de datos	Medida	POR	Ishibuchi	Hernández	Categoría
<b>IMAGEN</b>	POR2	5%	77.7±1.1	85.58±0.18	3
		10%	39±9	79.9±0.3	4
		20%	12.55±0	62.3±0.5	4
		30%	13.7±0	46.6±1.4	4
		40%	15.55±0	32.0±1.1	4
<b>MUSHROOM</b>	POR2	5%	97.59±0.09	98.01±0.10	2
		10%	94.47±0.13	97.11±0.19	3
		20%	92.1±1.1	93.2±0.3	2
		30%	61.67±0	84.1±0.5	4
		40%	63.07±0	82.961±0.013	4
<b>MOK1</b>	POR2	5%	79.05±0	79.067±0.017	2
		10%	73.5±1.5	68.3±0.3	6
		20%	49.8±0	73.7±0.6	4
		30%	47.45±0	70.35±0	4
		40%	49.35±0	67.45±0.09	4
<b>MOK2</b>	POR2	5%	69.25±0	69.25±0	1
		10%	68.45±0	68.45±0	1
		20%	69.6±0	69.6±0	1
		30%	67.6±0	67.6±0	1
		40%	67.4±0	67.4±0	1
<b>MOK3</b>	POR2	5%	80.55±0	82.5±0.3	2
		10%	87.45±0	88.0±0.4	2
		20%	81.5±1.1	85.2±0.6	3
		30%	55.65±0	78.2±0.2	4
		40%	53.9±0	77.75±0	4
<b>MUSK</b>	POR2	5%	91.02±0.03	93.44±0.10	3
		10%	86.106±0.018	90.72±0.15	3
		20%	84.03±0	86.98±0.09	3
		30%	83.57±0	84.97±0.14	2
		40%	84.17±0	84.35±0.07	2
<b>PIMA</b>	POR2	5%	74.18±0.15	69.0±0.4	6
		10%	65.2±0	72.8±0.4	3
		20%	65.4±0	65.83±0.03	2
		30%	66.4±0	66.4±0	1
		40%	65.95±0	65.95±0	1
<b>THYROIDES</b>	POR2	5%	94.17±0.17	94.47±0	2
		10%	93.3±0	93.3±0	1
		20%	93.03±0	93.03±0	1
		30%	94.23±0	94.23±0	1
		40%	93.47±0	93.47±0	1
<b>VOTE</b>	POR2	5%	95.25±0.05	95.62±0.03	2
		10%	93.617±0.017	93.80±0.10	2
		20%	91.27±0.02	91.63±0.02	2
		30%	86.40±0.04	91.05±0.12	3
		40%	81±0	91.02±0.07	4
<b>WDBC</b>	POR2	5%	97.93±0.10	98.283±0.017	2
		10%	95.82±0.09	96.78±0.07	2
		20%	87.1±0	89.1±0.2	2
		30%	65.1±0	85.18±0.17	4
		40%	63.1±0	75.6±0.6	4

**Tabla 64. Resultados de la comparación entre los métodos 2 y 3 para igual porcentaje de entradas desconocidas en el entrenamiento y en el test. Criterio de clasificación POR3.**

Base de datos	Medida	POR	Ishibuchi	Hernández	Categoría
<b>ABALONE</b>	POR3	5%	53.2±0.2	51±3	6
		10%	40±3	49.8±0.3	3
		20%	34.9±0	43.8±0.9	3
		30%	34.4±0	46.53±0	4
		40%	33.43±0	36.2±0.6	3
<b>BALANCE</b>	POR3	5%	82.1±0	80.32±0.08	5
		10%	79.77±0.18	78.00±0.09	5
		20%	65.0±0.2	75.5±0.2	4
		30%	45.4±0	73.8±0.3	4
		40%	47.4±0	69.1±0.4	4
<b>BANDS</b>	POR3	5%	69.8±0.2	73.2±0.4	3
		10%	66.4±0	69.0±0.3	3
		20%	63.8±0	67.53±0.10	3
		30%	67.2±0	67.35±0.10	2
		40%	67.2±0	67.2±0	1
<b>BUPA</b>	POR3	5%	55.9±0	60.95±0.08	3
		10%	51.4±0	51.817±0.017	2
		20%	51.6±0	51.6±0	1
		30%	52.8±0	52.8±0	1
		40%	53.2±0	53.2±0	1
<b>CREDITO</b>	POR3	5%	84.5±0.7	85.97±0.11	2
		10%	83.74±0.19	83.8±0.2	2
		20%	84.1±0	84.1±0	1
		30%	77.25±0	82.3±0	3
		40%	52.2±0	73.25±0	4
<b>DISPLAY1</b>	POR3	5%	68.6±0.5	69.7±0.3	2
		10%	49±8	64.0±0.2	4
		20%	8.7±0	43.5±1.3	4
		30%	11.25±0	30.3±0.9	4
		40%	9.3±0	22.0±1.1	4
<b>DISPLAY2</b>	POR3	5%	67.18±0.10	66.58±0.09	5
		10%	25±8	59.23±0.09	4
		20%	11.15±0	45.3±1.2	4
		30%	10.3±0	31.3±1.0	4
		40%	8.65±0	17.6±0.9	3
<b>GLASS</b>	POR3	5%	75.7±0.4	76.88±0.06	2
		10%	63.9±0.7	69.6±0.4	3
		20%	41±3	56.9±1.8	4
		30%	29±0	44.72±0.15	4
		40%	30.7±0	35.87±0.12	3
<b>HEART</b>	POR3	5%	83.97±0.03	82.1±0.3	5
		10%	81.9±0.3	82.43±0.14	2
		20%	75.52±0.10	78.42±0.10	3
		30%	53.3±0	75.7±0.3	4
		40%	47.6±0	73.05±0.16	4



Tabla 64. (Continuación).

Base de datos	Medida	POR	Ishibuchi	Hernández	Categoría
<b>IMAGEN</b>	POR3	5%	72.3±0.9	81.5±0.2	3
		10%	36±8	72.6±0.2	4
		20%	12.55±0	48.0±0.7	4
		30%	13.7±0	30.6±0.9	4
		40%	15.55±0	15.6±0.6	2
<b>MUSHROOM</b>	POR3	5%	97.59±0.09	98.01±0.10	2
		10%	94.47±0.13	97.11±0.19	3
		20%	92.1±1.1	93.2±0.3	2
		30%	61.67±0	84.1±0.5	4
		40%	63.07±0	82.961±0.013	4
<b>MOK1</b>	POR3	5%	79.05±0	79.067±0.017	2
		10%	73.5±1.5	68.3±0.3	6
		20%	49.8±0	73.7±0.6	4
		30%	47.45±0	70.35±0	4
		40%	49.35±0	67.45±0.09	4
<b>MOK2</b>	POR3	5%	69.25±0	69.25±0	1
		10%	68.45±0	68.45±0	1
		20%	69.6±0	69.6±0	1
		30%	67.6±0	67.6±0	1
		40%	67.4±0	67.4±0	1
<b>MOK3</b>	POR3	5%	80.55±0	82.5±0.3	2
		10%	87.45±0	88.0±0.4	2
		20%	81.5±1.1	85.2±0.6	3
		30%	55.65±0	78.2±0.2	4
		40%	53.9±0	77.75±0	4
<b>MUSK</b>	POR3	5%	91.02±0.03	93.44±0.10	3
		10%	86.106±0.018	90.71±0.14	3
		20%	84.03±0	86.98±0.09	3
		30%	83.57±0	84.97±0.14	2
		40%	84.17±0	84.34±0.07	2
<b>PIMA</b>	POR3	5%	74.18±0.15	69.0±0.4	6
		10%	65.2±0	72.8±0.4	3
		20%	65.4±0	65.83±0.03	2
		30%	66.4±0	66.4±0	1
		40%	65.95±0	65.95±0	1
<b>THYROIDES</b>	POR3	5%	94.17±0.17	94.47±0	2
		10%	93.3±0	93.3±0	1
		20%	93.03±0	93.03±0	1
		30%	94.23±0	94.23±0	1
		40%	93.47±0	93.47±0	1
<b>VOTE</b>	POR3	5%	95.25±0.05	95.62±0.03	2
		10%	93.617±0.017	93.80±0.10	2
		20%	91.27±0.02	91.63±0.02	2
		30%	86.40±0.04	91.05±0.12	3
		40%	81±0	91.02±0.07	4
<b>WDBC</b>	POR3	5%	97.93±0.10	98.283±0.017	2
		10%	95.82±0.09	96.77±0.06	2
		20%	87.1±0	89.0±0.2	2
		30%	65.1±0	85.15±0.18	4
		40%	63.1±0	75.6±0.6	4

**Tabla 65. Resultados de la comparación entre los métodos 2 y 3 para igual porcentaje de entradas desconocidas en el entrenamiento y en el test. Criterio de clasificación POR4.**

Base de datos	Medida	POR	Ishibuchi	Hernández	Categoría
<b>ABALONE</b>	POR4	5%	52.1±0.4	44±7	6
		10%	38±3	44.9±0.3	3
		20%	34.9±0	39.0±1.8	3
		30%	34.4±0	29.961±0.006	6
		40%	33.43±0	21.5±0.7	7
<b>BALANCE</b>	POR4	5%	75.8±0	75.8±0	1
		10%	64.60±0.11	64.28±0.08	5
		20%	49.45±0.11	50.4±0	2
		30%	45.4±0	38.5±0.3	6
		40%	47.4±0	27.80±0.13	7
<b>BANDS</b>	POR4	5%	63.3±0.3	57.9±0.4	6
		10%	66.4±0	45.8±0.3	7
		20%	63.8±0	47.95±0.16	7
		30%	67.2±0	64.7±1.6	6
		40%	67.2±0	67.2±0	1
<b>BUPA</b>	POR4	5%	55.9±0	48.67±0.11	6
		10%	51.4±0	43.40±0.14	6
		20%	51.6±0	51.6±0	1
		30%	52.8±0	52.8±0	1
		40%	53.2±0	53.2±0	1
<b>CREDITO</b>	POR4	5%	80.47±0.09	80.45±0.17	5
		10%	75.82±0.19	74.57±0.10	5
		20%	69.7±0	69.9±0	2
		30%	60.15±0	60.15±0	1
		40%	52.2±0	52.4±0	2
<b>DISPLAY1</b>	POR4	5%	63.1±0.5	62.9±0.2	5
		10%	42±6	51.08±0.16	3
		20%	8.7±0	30.2±1.1	4
		30%	11.25±0	17.9±0.6	3
		40%	9.3±0	10.6±0.3	2
<b>DISPLAY2</b>	POR4	5%	60.55±0.14	58.66±0.12	5
		10%	22±7	48.21±0.15	4
		20%	11.15±0	29.3±0.9	4
		30%	10.3±0	17.1±0.3	3
		40%	8.65±0	9.4±0.6	2
<b>GLASS</b>	POR4	5%	74.8±0.4	74.90±0.12	2
		10%	63.2±0.7	67.6±0.4	3
		20%	41±3	52.8±0.3	4
		30%	29±0	41.7±0.2	4
		40%	30.7±0	34.4±0.5	3
<b>HEART</b>	POR4	5%	79.42±0.09	75.4±0.2	6
		10%	70.9±0.4	69.28±0.16	5
		20%	57.5±0.3	57.33±0.18	5
		30%	53.3±0	46.7±0.3	6
		40%	47.6±0	37.9±0.5	6

Tabla 65. (Continuación).

Base de datos	Medida	POR	Ishibuchi	Hernández	Categoría
IMAGEN	POR4	5%	68.4±0.8	71.0±0.3	3
		10%	35±8	57.7±0.4	4
		20%	12.55±0	33.5±0.3	4
		30%	13.7±0	18.0±1.1	3
		40%	15.55±0	7.6±0.9	6
MUSHROOM	POR4	5%	94.1±0.3	94.1±0.3	1
		10%	87.27±0.16	88.1±0.3	2
		20%	69.4±1.1	70.7±0.6	2
		30%	61.67±0	60.78±0.19	5
		40%	63.07±0	51.11±0.10	7
MOK1	POR4	5%	76.75±0	77.3±0.5	2
		10%	67.8±1.9	61.6±0.2	6
		20%	49.8±0	63.6±0.4	4
		30%	47.45±0	54.7±0	3
		40%	49.35±0	46.75±0	6
MOK2	POR4	5%	69.25±0	69.25±0	1
		10%	68.45±0	68.45±0	1
		20%	69.6±0	69.6±0	1
		30%	67.6±0	67.6±0	1
		40%	67.4±0	67.4±0	1
MOK3	POR4	5%	74.05±0	75.9±0.3	2
		10%	77.57±0.05	76.9±0.3	5
		20%	68.4±0.3	66.2±0.4	6
		30%	55.65±0	47.0±0.4	6
		40%	53.9±0	48.05±0	6
MUSK	POR4	5%	87.02±0.02	86.93±0.16	5
		10%	84.506±0.016	82.36±0.07	6
		20%	84.03±0	81.54±0.05	6
		30%	83.57±0	82.278±0.019	5
		40%	84.17±0	82.76±0.03	5
PIMA	POR4	5%	69.85±0.13	61.1±0.5	6
		10%	65.2±0	64.4±0.2	5
		20%	65.4±0	56.78±0.02	6
		30%	66.4±0	66.4±0	1
		40%	65.95±0	65.95±0	1
THYROIDES	POR4	5%	94.13±0.13	94.1±0	5
		10%	93.3±0	93.23±0	5
		20%	93.03±0	93.03±0	1
		30%	94.23±0	94.23±0	1
		40%	93.47±0	93.47±0	1
VOTE	POR4	5%	93.48±0.06	93.27±0.04	5
		10%	88.08±0.03	87.75±0.10	5
		20%	81.58±0.15	79.83±0.09	5
		30%	70.90±0.05	70.18±0.18	5
		40%	54.3±0	53.95±0.12	5
WDBC	POR4	5%	88.80±0.15	87.30±0.09	5
		10%	82.70±0.12	77.15±0.09	6
		20%	73.7±0	70.2±0.6	6
		30%	65.1±0	53.2±1.1	7
		40%	63.1±0	49.8±1.0	7

**Tabla 66. Resultados de la comparación entre los métodos 2 y 3 para igual porcentaje de entradas desconocidas en el entrenamiento y en el test. Criterio de clasificación POR5.**

Base de datos	Medida	POR	Ishibuchi	Hernández	Categoría
<b>ABALONE</b>	POR5	5%	29±2	26±4	6
		10%	4±2	30.79±0.10	4
		20%	0±0	17.2±1.9	4
		30%	0±0	2.5±0	3
		40%	0±0	0.728±0.006	2
<b>BALANCE</b>	POR5	5%	73.13±0.06	72.3±0	5
		10%	61.22±0.08	60.383±0.017	5
		20%	26.6±0.8	46.63±0.19	4
		30%	45.4±0	29.10±0.17	7
		40%	0±0	15.3±0.2	4
<b>BANDS</b>	POR5	5%	63.3±0.3	57.9±0.4	6
		10%	66.4±0	45.8±0.3	7
		20%	63.8±0	47.95±0.16	7
		30%	67.2±0	64.7±1.6	6
		40%	67.2±0	67.2±0	1
<b>BUPA</b>	POR5	5%	55.9±0	48.67±0.11	6
		10%	51.4±0	43.43±0.15	6
		20%	51.6±0	51.6±0	1
		30%	52.8±0	52.8±0	1
		40%	53.2±0	53.2±0	1
<b>CREDITO</b>	POR5	5%	80.7±0.4	80.45±0.17	5
		10%	75.82±0.19	74.58±0.10	5
		20%	69.9±0	69.9±0	1
		30%	60.15±0	60.15±0	1
		40%	52.2±0	52.4±0	2
<b>DISPLAY1</b>	POR5	5%	41.9±0.6	51.2±0.9	3
		10%	15±3	40.8±0.4	4
		20%	0±0	10.3±0.8	4
		30%	0±0	2.4±0.3	3
		40%	0±0	0±0	1
<b>DISPLAY2</b>	POR5	5%	46.37±0.18	49.3±0.4	3
		10%	6±4	31.9±0.3	4
		20%	0±0	11.3±0.4	4
		30%	0±0	5.3±0.5	3
		40%	0±0	0±0	1
<b>GLASS</b>	POR5	5%	69.5±0.3	69.50±0.19	1
		10%	48±4	63.27±0.08	4
		20%	16±3	43.7±1.4	4
		30%	0±0	16.12±0.12	4
		40%	0±0	10.7±0.4	4
<b>HEART</b>	POR5	5%	79.42±0.09	75.4±0.2	6
		10%	70.9±0.4	69.27±0.15	5
		20%	57.5±0.3	57.33±0.18	5
		30%	53.3±0	46.7±0.3	6
		40%	47.6±0	37.9±0.5	6

Tabla 66. (Continuación).

Base de datos	Medida	POR	Ishibuchi	Hernández	Categoría
IMAGEN	POR5	5%	50.8±0.3	62.6±0.6	4
		10%	15±9	45.6±0.7	4
		20%	0±0	22.8±0.3	4
		30%	0±0	6.85±0.18	3
		40%	0±0	0.8±0.2	2
MUSHROOM	POR5	5%	94.1±0.3	94.1±0.3	1
		10%	87.26±0.16	88.1±0.3	2
		20%	69.8±1.1	70.7±0.6	2
		30%	61.67±0	60.78±0.19	5
		40%	63.07±0	51.11±0.10	7
MOK1	POR5	5%	76.75±0	77.3±0.5	2
		10%	67.8±1.9	61.6±0.2	6
		20%	49.8±0	63.6±0.4	4
		30%	47.45±0	54.7±0	3
		40%	49.35±0	46.75±0	6
MOK2	POR5	5%	69.25±0	69.25±0	1
		10%	68.45±0	68.45±0	1
		20%	69.6±0	69.6±0	1
		30%	67.6±0	67.6±0	1
		40%	67.4±0	67.4±0	1
MOK3	POR5	5%	74.05±0	75.9±0.3	2
		10%	77.57±0.05	76.9±0.3	5
		20%	68.4±0.3	66.2±0.4	6
		30%	55.65±0	47.0±0.4	6
		40%	53.9±0	48.05±0	6
MUSK	POR5	5%	87.02±0.02	86.94±0.16	5
		10%	84.500±0.019	82.37±0.08	6
		20%	84.03±0	81.54±0.05	6
		30%	83.57±0	82.278±0.019	5
		40%	84.17±0	82.76±0.03	5
PIMA	POR5	5%	69.83±0.13	61.1±0.5	6
		10%	65.2±0	64.4±0.2	6
		20%	65.4±0	56.78±0.02	6
		30%	66.4±0	66.4±0	1
		40%	65.95±0	65.95±0	1
THYROIDES	POR5	5%	93.89±0.11	93.772±0	5
		10%	93.3±0	93.072±0.006	5
		20%	93.03±0	93.03±0	1
		30%	94.23±0	94.23±0	1
		40%	93.47±0	93.47±0	1
VOTE	POR5	5%	93.48±0.06	93.28±0.03	5
		10%	88.08±0.03	87.75±0.10	5
		20%	81.58±0.15	79.85±0.09	5
		30%	70.88±0.06	70.18±0.18	5
		40%	54.3±0	53.95±0.12	5
WDBC	POR5	5%	88.80±0.15	87.30±0.09	5
		10%	82.72±0.12	77.15±0.09	6
		20%	73.7±0	70.1±0.6	6
		30%	65.1±0	53.3±1.1	7
		40%	63.1±0	49.8±1.0	7

*I. APÉNDICES DEL CAPÍTULO 2*

## II. Apéndices del Capítulo 3

### II.1. Ordenaciones de las entradas para ARNMF y ACE

*Tabla 11. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema AB.*

#### BASE DE DATOS AB

Método	Ordenación de las entradas por importancia								
	Menor importancia				Mayor importancia				
	1	2	3	4	5	6	7	8	9
BL2	1	5	8	7	3	6	4	9	7
TEKA	1	5	8	7	3	6	4	9	2
CIB	5	1	8	6	7	9	3	4	2
TEKE	2	1	7	4	5	6	8	3	9
TEKC	5	6	8	3	9	7	4	2	1
DEV	1	5	8	7	3	6	4	9	2
DER3	2	4	3	5	7	1	9	6	8
TEKB	1	5	8	7	3	6	4	2	9
TEK	5	8	7	6	1	9	3	2	4
BL1	1	5	3	8	2	6	4	7	9
CLO	1	5	2	8	3	6	4	7	9
PRI	1	5	3	8	2	6	4	7	9
SAN	1	5	8	3	6	7	4	2	9
DER2	5	8	1	3	6	4	7	9	2
MAO	1	5	8	6	7	9	3	4	2
UTA	1	5	8	3	6	4	2	7	9
LEE	9	2	4	6	8	3	5	1	7
BOW	5	7	8	9	2	6	1	3	4
YOU	División por cero								
BA 0.5	6	3	4	8	7	5	1	2	9
BA 0.6	6	3	4	8	7	5	1	2	9
BA 0.7	6	4	3	8	7	5	1	2	9
BA 0.8	6	4	3	8	7	5	1	2	9
BA 0.9	6	4	3	8	7	5	2	1	9
BA 1	6	4	3	8	7	5	2	1	9
CHI	1	2	7	5	3	8	6	4	9
SET	1	7	3	6	4	8	9	2	5
GD DS	1	2	7	8	5	6	9	3	4
RLF	Da cero								
SCH	2	4	6	9	3	8	7	5	1
FUZ	2	3	5	4	8	7	9	1	6

**Tabla 12. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema BL.**

**BASE DE DATOS BL**

Método	Ordenación de las entradas por importancia				
	Menor importancia			Mayor importancia	
	1	2	3	4	5
BL2	4	2	3	5	1
TEKA	4	2	1	5	3
CIB	1	4	2	3	5
TEKE	5	3	1	2	4
TEKC	3	4	5	1	2
DEV	1	4	2	3	5
DER3	3	2	1	4	5
TEKB	3	1	4	2	5
TEK	3	4	1	2	5
BL1	1	4	3	2	5
CLO	1	4	2	3	5
PRI	1	4	3	5	2
SAN	4	5	3	2	1
DER2	5	1	4	3	2
MAO	1	4	5	2	3
UTA	1	2	4	3	5
LEE	5	3	2	4	1
BOW	1	5	3	2	4
YOU	Divisiones por cero				
BA 0.5	1	4	3	5	2
BA 0.6	1	4	3	5	2
BA 0.7	1	4	3	5	2
BA 0.8	1	4	3	5	2
BA 0.9	1	4	3	5	2
BA 1	1	4	3	5	2
CHI	1	4	3	5	2
SET	1	4	3	5	2
GD DS	1	4	3	5	2
RLF	Problemas, da cero				
SCH	2	5	3	4	1
FUZ	5	2	3	1	4



Tabla 13. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema BN.

**BASE DE DATOS BN (parte 1)**

Ordenación de las entradas por importancia																				
	Menor importancia										Mayor importancia									
Método	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
BL2	13	7	10	36	29	27	33	14	5	24	26	2	20	21	31	3	22	18	9	32
TEKA	13	7	10	29	27	36	33	5	14	24	26	2	20	21	31	22	18	3	9	28
CIB	13	34	33	36	29	10	7	9	20	27	12	5	24	2	21	31	11	26	18	14
TEKE	Todos cero																			
TEKC	13	34	9	33	36	25	24	28	29	11	27	20	5	12	22	4	39	26	30	32
DEV	13	7	10	36	29	27	33	14	5	15	24	21	3	18	20	32	26	2	31	17
DER3	24	7	19	33	20	9	35	18	5	12	14	3	26	15	38	10	37	4	36	6
TEKB	13	7	29	36	10	27	5	33	14	2	24	26	21	20	18	15	22	3	31	12
TEK	13	9	34	33	29	36	7	10	21	27	2	5	20	24	26	28	31	12	18	22
BL1	13	2	29	7	27	10	33	36	5	21	18	20	14	11	24	26	12	15	3	22
CLO	13	2	7	10	29	27	36	33	21	5	18	20	14	24	11	15	26	12	31	3
PRI	13	2	29	7	27	10	33	36	5	21	18	20	14	11	24	26	12	15	3	22
SAN	13	7	29	10	36	27	2	33	21	5	14	18	24	20	15	26	11	31	3	12
DER2	13	7	10	36	29	27	2	33	18	5	21	11	14	24	20	15	26	32	31	22
MAO	29	36	13	7	10	33	5	34	20	27	9	12	24	21	2	31	11	18	3	26
UTA	29	7	10	13	36	33	27	24	26	5	2	34	14	18	21	32	31	20	9	22
LEE	19	23	6	16	4	1	30	38	17	37	25	35	15	28	8	39	11	12	22	3
BOW	13	29	31	33	12	3	26	7	10	34	36	21	20	24	19	2	27	32	6	28
YOU	13	33	34	9	10	7	24	29	20	12	36	5	26	21	2	31	11	28	27	18
BA 0.5	3	23	1	31	22	28	37	30	29	5	17	26	39	35	27	32	20	36	12	25
BA 0.6	3	23	1	31	22	28	37	30	29	5	17	26	39	35	27	32	20	36	12	25
BA 0.7	3	23	1	31	22	28	37	30	29	5	17	26	39	35	27	32	20	36	12	25
BA 0.8	3	23	1	31	22	28	37	30	29	5	17	26	39	35	27	32	20	36	12	25
BA 0.9	3	23	1	31	22	28	37	30	29	5	17	26	39	35	27	32	20	36	25	12
BA 1	3	23	1	31	22	28	37	30	29	5	17	26	39	35	27	32	20	36	25	12
CHI	10	7	19	13	9	21	34	15	14	8	40	20	6	11	16	24	2	38	5	29
SET	7	10	19	21	15	24	20	29	5	2	28	16	6	1	14	35	36	27	11	31
GD DS	Matriz Singular, no aplicable																			
RLF	18	35	7	9	10	13	34	1	36	33	26	38	15	22	28	30	32	24	17	25
SCH	40	16	6	12	15	11	37	5	21	20	30	32	14	39	23	31	25	1	26	29
FUZ	Excepciones cero, no aplicable																			

Tabla 13. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema BN. (Continuación).

**BASE DE DATOS BN (parte 2)**

Ordenación de las entradas por importancia																				
	Menor importancia										Mayor importancia									
Método	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
BL2	39	15	28	34	12	17	40	19	8	1	38	25	4	37	11	35	6	16	23	30
TEKA	34	12	32	15	39	19	38	40	8	25	17	1	4	35	37	11	6	16	23	30
CIB	3	28	32	39	15	25	22	37	4	17	1	38	40	16	19	23	6	8	35	30
TEKE	Todos cero																			
TEKC	16	3	37	18	14	2	19	35	23	7	38	15	10	21	8	1	31	17	40	6
DEV	22	16	40	39	9	6	12	1	34	28	8	11	37	38	19	4	25	35	30	23
DER3	13	27	11	8	16	1	32	40	17	22	23	34	30	28	39	21	2	25	31	29
TEKB	32	9	39	34	28	40	11	17	1	8	38	25	19	37	6	16	4	35	23	30
TEK	25	3	14	37	32	4	39	11	15	1	6	17	38	16	23	40	19	8	35	30
BL1	31	32	9	39	6	40	17	16	28	34	1	19	8	38	37	25	4	35	30	23
CLO	22	32	39	40	17	6	9	16	28	34	1	19	37	8	38	25	4	35	23	30
PRI	31	32	9	39	6	40	17	16	28	34	1	19	8	38	37	25	4	35	30	23
SAN	22	32	17	39	40	6	16	9	28	34	1	19	8	38	37	4	25	35	30	23
DER2	12	3	34	17	16	28	39	9	40	1	38	6	25	37	19	4	8	35	30	23
MAO	14	28	39	32	25	22	15	37	17	4	1	38	40	16	19	23	6	8	35	30
UTA	3	39	40	11	15	12	28	25	8	17	37	35	38	1	30	16	6	4	23	19
LEE	9	40	18	34	26	20	2	14	5	31	27	24	33	36	7	10	13	32	21	29
BOW	9	15	5	17	37	38	11	25	16	39	22	8	14	1	4	40	18	23	35	30
YOU	25	3	22	14	37	19	15	38	32	4	23	39	17	8	1	16	6	40	35	30
BA 0.5	19	2	24	6	16	33	11	38	18	8	14	15	21	40	34	13	9	10	7	4
BA 0.6	19	2	24	6	16	33	11	38	18	8	14	15	21	40	34	13	9	10	7	4
BA 0.7	19	2	24	6	16	33	11	38	18	8	14	15	21	40	34	13	9	10	7	4
BA 0.8	19	2	24	6	16	33	18	11	38	8	14	15	21	40	34	13	9	10	7	4
BA 0.9	19	2	24	6	16	33	18	11	38	8	14	15	21	40	34	13	9	10	7	4
BA 1	19	2	24	6	16	33	18	11	38	8	14	15	21	40	34	13	9	10	7	4
CHI	35	12	28	36	27	33	17	37	31	1	22	25	30	18	32	26	3	39	23	4
SET	22	30	3	26	25	32	37	23	17	38	4	33	8	39	40	12	9	13	34	18
GD DS	Matriz Singular, no aplicable																			
RLF	27	37	2	14	20	31	6	12	3	23	5	29	39	8	19	21	4	11	16	40
SCH	24	19	28	3	27	4	35	7	10	38	36	17	34	13	9	33	2	22	18	8
FUZ	Excepciones cero, no aplicable																			

*Tabla 14. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema BU.*

**BASE DE DATOS BU**

Método	Ordenación de las entradas por importancia						
	Menor importancia			Mayor importancia			
	1	2	3	4	5	6	7
BL2	1	7	2	3	6	5	4
TEKA	1	2	7	3	6	4	5
CIB	7	1	6	2	3	4	5
TEKE	6	1	4	7	5	3	2
TEKC	6	4	7	5	3	2	1
DEV	1	2	7	3	6	5	4
DER3	6	5	3	2	7	1	4
TEKB	1	7	2	3	6	5	4
TEK	7	1	6	2	3	4	5
BL1	1	2	7	3	6	5	4
CLO	1	2	7	3	6	5	4
PRI	1	2	7	3	6	5	4
SAN	1	2	7	3	6	5	4
DER2	1	7	2	3	6	5	4
MAO	7	1	2	6	3	4	5
UTA	1	2	7	3	6	5	4
LEE	4	5	6	2	1	7	3
BOW	7	4	1	6	5	3	2
YOU	División por cero						
BA 0.5	1	5	3	6	2	4	7
BA 0.6	1	5	3	6	2	4	7
BA 0.7	1	5	3	6	2	4	7
BA 0.8	1	5	3	4	2	6	7
BA 0.9	1	5	3	6	4	2	7
BA 1	1	5	3	6	4	2	7
CHI	2	5	6	3	4	1	7
SET	3	2	1	5	6	4	7
GD_DS	2	1	3	7	5	4	6
RLF	3	2	6	7	4	1	5
SCH	7	3	1	4	6	2	5
FUZ	2	7	5	4	1	3	6

Tabla 15. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema CR.

**BASE DE DATOS CR**

		Ordenación de las entradas por importancia															
		Menor importancia								Mayor importancia							
Método		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BL2		9	7	2	1	4	3	12	13	14	5	6	8	15	16	11	10
TEKA		9	7	2	1	4	13	3	14	12	5	6	8	15	16	11	10
CIB		16	12	9	4	15	14	6	5	8	3	2	7	1	10	13	11
TEKE		Todos cero															
TEKC		16	12	8	15	14	5	6	4	10	9	3	13	7	1	11	2
DEV		9	7	2	1	4	3	12	14	13	5	6	8	15	16	11	10
DER3		16	8	1	10	5	7	9	3	2	4	6	15	13	11	12	14
TEKB		9	2	7	4	12	3	1	8	5	6	14	13	15	11	16	10
TEK		16	12	9	14	4	5	6	8	15	3	2	7	1	13	11	10
BL1		2	7	9	1	14	13	5	6	4	3	8	12	11	15	16	10
CLO		2	9	7	1	14	13	5	6	4	3	8	12	11	15	16	10
PRI		2	7	9	1	14	13	5	6	4	3	8	12	11	15	16	10
SAN		2	7	9	1	13	14	4	5	6	3	8	12	11	15	16	10
DER2		9	6	5	14	1	7	2	4	15	13	11	8	12	3	16	10
MAO		9	12	16	4	14	15	6	5	8	3	2	7	1	10	13	11
UTA		9	12	2	7	1	4	14	3	5	6	16	15	8	13	11	10
LEE		10	11	13	15	5	6	8	16	4	1	14	2	3	7	12	9
BOW		12	9	3	16	5	6	4	11	2	13	14	8	15	10	7	1
YOU		14	9	16	5	6	15	2	12	7	8	4	1	3	13	11	10
BA 0.5		3	1	7	4	6	9	11	15	8	14	13	5	2	16	12	10
BA 0.6		3	1	7	4	6	9	11	15	8	14	13	5	2	16	12	10
BA 0.7		3	1	7	4	6	9	11	15	8	14	13	5	2	16	12	10
BA 0.8		3	1	4	8	6	9	12	15	14	16	2	11	13	5	7	10
BA 0.9		3	1	7	4	6	12	9	8	16	14	13	5	2	11	15	10
BA 1		3	1	7	4	6	12	9	8	16	11	13	14	5	2	15	10
CHI		2	13	14	6	5	16	1	8	3	4	15	7	11	9	12	10
SET		2	13	1	14	3	4	8	15	7	5	6	9	16	11	12	10
GD DS		Matriz singular, no aplicable															
RLF		16	12	9	13	15	4	11	14	2	5	6	3	1	8	7	10
SCH		10	11	9	4	3	12	6	5	16	13	1	15	7	2	8	14
FUZ		2	10	11	13	14	5	6	7	4	12	15	8	9	16	1	3

**Tabla 16. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema D1.**

**BASE DE DATOS D1**

Método	Ordenación de las entradas por importancia							
	Menor importancia				Mayor importancia			
	1	2	3	4	5	6	7	8
BL2	1	7	8	2	3	6	5	4
TEKA	1	7	8	2	3	6	5	4
CIB	1	8	2	3	6	7	5	4
TEKE	4	7	5	6	1	2	8	3
TEKC	6	1	4	7	8	2	3	5
DEV	1	7	8	2	3	6	5	4
DER3	5	1	2	4	3	8	7	6
TEKB	1	7	8	2	6	3	5	4
TEK	1	6	8	3	7	2	5	4
BL1	1	2	7	8	4	5	3	6
CLO	1	8	2	6	3	4	5	7
PRI	1	2	8	7	4	5	3	6
SAN	1	7	8	2	5	4	6	3
DER2	1	7	2	8	3	4	6	5
MAO	1	8	2	3	7	6	5	4
UTA	1	7	8	2	3	5	6	4
LEE	6	4	3	5	2	8	7	1
BOW	1	4	7	6	5	3	8	2
YOU	Divisiones por cero							
BA 0.5	1	7	2	4	8	3	5	6
BA 0.6	1	7	2	8	4	3	5	6
BA 0.7	1	7	2	8	4	3	5	6
BA 0.8	1	7	2	8	4	3	5	6
BA 0.9	1	7	2	8	4	3	5	6
BA 1	1	7	2	8	4	3	5	6
CHI	1	7	2	4	8	5	3	6
SET	1	7	2	3	8	4	6	5
GD_DS	7	1	2	4	3	5	8	6
RLF	Problemas, da cero							
SCH	5	6	8	4	3	2	7	1
FUZ	2	3	4	5	6	7	1	8

Tabla 17. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema GL.

**BASE DE DATOS GL**

Método	Ordenación de las entradas por importancia										
	Menor importancia						Mayor importancia				
	1	2	3	4	5	6	7	8	9	10	11
BL2	1	8	11	4	7	3	9	10	6	5	2
TEKA	8	11	1	7	4	3	9	10	6	5	2
CIB	8	11	10	1	4	3	7	9	6	5	2
TEKE	1	3	8	4	6	7	9	5	10	2	11
TEKC	10	8	11	3	9	6	7	2	1	4	5
DEV	1	8	4	11	7	3	10	9	5	6	2
DER3	2	3	6	5	10	4	1	11	8	9	7
TEKB	8	11	10	3	9	6	4	1	7	2	5
TEK	8	11	10	3	9	6	4	1	7	2	5
BL1	1	8	11	7	4	3	9	6	10	5	2
CLO	1	8	11	4	7	3	9	10	5	6	2
PRI	8	1	11	4	7	3	9	6	10	5	2
SAN	1	8	11	4	7	3	9	10	6	5	2
DER2	8	1	11	4	7	10	9	6	3	2	5
MAO	11	8	1	10	7	4	9	3	6	5	2
UTA	8	1	11	3	4	9	7	10	6	5	2
LEE	2	5	6	10	4	9	7	8	11	3	1
BOW	11	4	1	8	3	6	7	5	9	10	2
YOU	Divisiones por cero										
BA 0.5	1	6	3	4	7	9	5	11	8	10	2
BA 0.6	1	6	3	4	7	9	5	11	8	10	2
BA 0.7	1	6	3	4	7	5	9	11	8	10	2
BA 0.8	1	6	3	4	5	7	9	11	8	10	2
BA 0.9	6	1	3	4	5	7	9	11	8	10	2
BA 1	6	1	3	4	5	7	9	11	8	10	2
CHI	11	8	10	1	7	3	4	9	6	5	2
SET	1	11	7	4	3	8	9	6	5	2	10
GD_DS	2	8	11	1	7	4	10	3	9	6	5
RLF	Problemas, da cero										
SCH	2	5	6	10	4	7	11	8	9	3	1
FUZ	Excepciones cero, no aplicable										

**Tabla 18. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema HE.**

**BASE DE DATOS HE**

Método	Ordenación de las entradas por importancia													
	Menor importancia							Mayor importancia						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
BL2	5	1	6	7	8	2	11	12	4	3	9	13	10	14
TEKA	5	1	7	6	8	2	11	12	4	3	9	14	10	13
CIB	5	6	11	1	7	8	13	14	12	2	10	3	9	4
TEKE	10	13	12	6	5	11	9	1	2	8	3	4	14	7
TEKC	11	6	12	10	7	1	2	9	8	14	4	3	13	5
DEV	5	1	6	7	8	2	11	4	12	3	9	13	10	14
DER3	13	2	12	1	3	4	6	5	7	14	8	11	9	10
TEKB	5	1	6	8	7	2	11	12	3	4	14	10	9	13
TEK	7	11	5	6	12	1	8	10	2	13	14	3	9	4
BL1	1	5	8	2	7	6	3	14	12	10	4	11	9	13
CLO	5	1	8	7	2	6	3	12	4	11	10	14	9	13
PRI	1	5	8	2	7	6	3	14	12	10	4	11	9	13
SAN	5	1	8	7	2	6	3	4	12	11	10	14	9	13
DER2	5	6	2	7	8	1	11	12	3	14	4	13	9	10
MAO	5	6	11	1	7	8	14	12	13	2	10	3	4	9
UTA	5	1	6	2	11	8	12	7	9	4	10	14	3	13
LEE	13	3	10	14	4	7	9	8	12	11	2	6	1	5
BOW	5	7	12	11	6	9	1	2	10	4	8	14	3	13
YOU	6	7	5	11	8	1	12	2	10	14	3	4	13	9
BA 0.5	1	9	2	5	6	12	14	8	7	10	3	13	4	11
BA 0.6	1	9	2	5	6	12	14	8	4	7	3	13	10	11
BA 0.7	1	9	2	5	6	12	14	4	8	7	3	13	10	11
BA 0.8	1	9	2	5	6	12	14	4	8	13	7	3	10	11
BA 0.9	1	9	2	5	6	12	14	4	8	13	7	3	10	11
BA 1	1	9	2	5	6	12	14	4	8	13	3	7	10	11
CHI	7	8	6	3	5	1	2	12	10	14	4	9	13	11
SET	7	6	5	1	8	2	9	3	11	12	4	13	14	10
GD_DS	7	8	3	6	13	2	5	1	9	11	12	4	10	14
RLF	6	5	2	1	9	11	7	10	12	4	14	8	13	3
SCH	14	10	13	4	3	12	11	9	8	2	5	7	6	1
FUZ	3	7	8	10	14	4	13	12	5	2	11	9	1	6

**Tabla 19. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema LE.**

**BASE DE DATOS LE**

Ordenación de las entradas por importancia																							
Método	Menor importancia											Mayor importancia											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
BL2	18	1	17	19	10	2	7	22	23	20	4	3	6	14	13	8	9	16	15	12	21	5	11
TEKA	18	1	17	19	10	2	7	22	23	4	20	3	6	14	8	16	15	13	9	12	21	5	11
CIB	18	1	17	19	2	10	7	4	22	8	16	23	15	21	20	12	5	6	14	3	13	9	11
TEKE	Todos cero																						
TEKC	18	12	21	6	1	4	11	19	2	3	10	23	8	22	13	14	9	7	17	16	15	20	5
DEV	18	1	17	19	10	2	7	22	14	23	20	16	13	15	4	3	8	6	9	21	5	12	11
DER3	7	6	23	4	11	17	9	8	21	12	14	13	2	18	1	16	10	22	19	5	3	15	20
TEKB	18	1	17	19	10	2	7	22	4	23	15	16	20	14	8	13	6	3	5	9	21	11	12
TEK	18	1	17	19	10	2	8	4	22	21	23	12	7	16	15	20	3	6	5	14	13	11	9
BL1	18	1	17	19	2	7	10	4	22	23	16	15	20	5	8	14	13	6	3	9	11	21	12
CLO	18	1	17	19	2	10	7	22	23	4	16	15	20	8	14	5	13	6	3	9	11	21	12
PRI	18	1	17	19	2	7	10	4	22	23	16	15	20	5	8	14	13	6	3	9	11	21	12
SAN	18	1	17	19	10	2	7	22	23	4	16	15	20	8	14	13	6	3	5	9	21	11	12
DER2	18	17	1	19	2	10	7	22	4	23	5	16	15	20	6	14	8	21	13	3	12	9	11
MAO	18	1	17	19	10	2	7	4	22	16	8	23	15	21	20	5	6	12	14	3	13	9	11
UTA	7	17	18	1	19	22	2	10	15	4	16	23	20	14	13	8	21	5	3	6	12	9	11
LEE	11	9	12	6	3	5	8	21	20	13	14	23	16	4	10	15	19	17	18	1	7	2	22
BOW	18	1	2	17	10	23	4	7	5	20	19	15	22	14	8	13	6	16	3	9	11	21	12
YOU	18	17	10	19	1	2	7	22	8	20	3	4	6	16	23	21	15	5	12	14	11	13	9
BA 0.5	20	14	15	12	16	13	23	4	22	10	5	9	2	8	19	3	11	1	7	18	17	21	6
BA 0.6	20	14	15	16	12	13	23	4	22	10	5	9	11	3	2	19	8	1	7	18	17	21	6
BA 0.7	20	14	15	16	12	13	23	4	22	21	10	5	9	3	2	11	19	8	1	7	17	18	6
BA 0.8	20	14	15	16	12	13	23	4	22	21	10	5	9	3	11	2	19	8	1	7	17	18	6
BA 0.9	20	14	15	16	12	13	23	4	21	22	10	11	5	3	9	2	19	8	1	7	17	18	6
BA 1	20	14	15	16	12	13	23	4	21	22	10	11	5	3	9	2	19	8	1	7	17	18	6
CHI	17	18	7	3	1	19	2	9	8	12	22	23	5	4	10	16	11	15	14	13	20	21	6
SET	17	1	3	2	19	22	23	9	12	10	4	8	18	7	5	16	15	11	14	21	13	20	6
GD DS	Matriz singular																						
RLF	17	18	7	2	1	10	15	16	19	22	9	13	12	23	14	4	11	20	21	8	5	6	3
SCH	11	5	15	16	21	9	23	13	14	12	8	4	20	3	19	7	6	18	17	2	22	1	10
FUZ	Excepciones cero, no aplicable																						



**Tabla 20. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema M1.**

**BASE DE DATOS M1**

Método	Ordenación de las entradas por importancia						
	Menor importancia			Mayor importancia			
	1	2	3	4	5	6	7
BL2	4	1	7	5	3	2	6
TEKA	4	7	1	3	2	5	6
CIB	4	7	1	5	3	2	6
TEKE	3	5	1	7	4	2	6
TEKC	3	2	4	6	1	7	5
DEV	4	1	7	5	3	2	6
DER3	6	5	3	1	2	4	7
TEKB	4	7	5	1	3	2	6
TEK	4	7	5	1	3	2	6
BL1	4	7	3	2	5	1	6
CLO	4	7	5	1	3	2	6
PRI	4	7	3	2	5	1	6
SAN	4	7	1	3	2	5	6
DER2	4	7	5	2	3	1	6
MAO	7	4	1	5	3	2	6
UTA	7	4	5	1	3	2	6
LEE	6	3	2	1	5	4	7
BOW	4	7	5	1	6	2	3
YOU	4	7	5	3	1	2	6
BA 0.5	1	3	2	7	4	5	6
BA 0.6	1	3	2	7	4	5	6
BA 0.7	1	3	2	7	4	5	6
BA 0.8	1	3	2	7	4	5	6
BA 0.9	1	3	2	7	4	5	6
BA 1	1	3	2	7	4	5	6
CHI	4	3	2	7	5	1	6
SET	4	3	2	7	5	1	6
GD_DS	1	2	3	5	4	7	6
RLF	5	4	7	6	1	3	2
SCH	6	2	5	1	4	3	7
FUZ	2	3	4	5	7	1	6

**Tabla 21. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema M2.**

**BASE DE DATOS M2**

Método	Ordenación de las entradas por importancia						
	Menor importancia				Mayor importancia		
	1	2	3	4	5	6	7
BL2	1	6	4	7	5	2	3
TEKA	1	6	4	7	5	2	3
CIB	1	6	5	2	3	4	7
TEKE	1	3	6	4	7	5	2
TEKC	6	4	7	5	1	3	2
DEV	1	6	5	3	2	4	7
DER3	4	7	1	3	2	5	6
TEKB	1	6	4	5	2	3	7
TEK	1	6	2	5	3	4	7
BL1	1	6	4	5	2	3	7
CLO	1	6	5	4	3	2	7
PRI	1	6	4	5	2	3	7
SAN	1	6	4	5	3	2	7
DER2	6	4	5	7	3	2	1
MAO	1	6	5	2	3	4	7
UTA	1	6	5	2	3	4	7
LEE	7	4	3	2	5	6	1
BOW	1	2	6	5	3	4	7
YOU	6	3	5	1	2	7	4
BA 0.5	6	3	2	4	5	7	1
BA 0.6	6	3	2	4	5	7	1
BA 0.7	6	3	2	4	5	7	1
BA 0.8	6	3	2	4	5	7	1
BA 0.9	6	3	2	4	5	7	1
BA 1	6	3	2	4	5	7	1
CHI	4	7	6	2	3	5	1
SET	4	7	6	2	3	5	1
GD_DS	1	6	2	3	5	4	7
RLF	1	4	6	7	5	3	2
SCH	7	4	3	1	2	6	5
FUZ	3	4	5	6	7	1	2

**Tabla 22. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema M3.**

**BASE DE DATOS M3**

Ordenación de las entradas por importancia							
Método	Menor importancia				Mayor importancia		
	1	2	3	4	5	6	7
BL2	4	2	7	1	5	6	3
TEKA	4	2	7	1	5	6	3
CIB	4	2	7	5	1	6	3
TEKE	5	7	2	4	1	6	3
TEKC	5	6	4	7	1	2	3
DEV	2	4	7	1	5	6	3
DER3	2	6	3	7	5	4	1
TEKB	1	7	5	2	4	6	3
TEK	1	7	5	2	4	3	6
BL1	4	2	5	7	1	3	6
CLO	4	2	7	5	1	3	6
PRI	4	2	5	7	1	3	6
SAN	4	2	7	5	1	3	6
DER2	2	7	4	1	5	3	6
MAO	4	2	7	5	1	6	3
UTA	4	1	5	2	7	6	3
LEE	6	3	1	2	4	7	5
BOW	4	2	7	1	5	6	3
YOU	Divisiones por cero						
BA 0.5	1	7	2	4	5	3	6
BA 0.6	1	7	2	4	5	3	6
BA 0.7	1	7	2	4	5	3	6
BA 0.8	1	7	2	4	5	3	6
BA 0.9	1	7	2	4	5	3	6
BA 1	1	7	2	4	5	3	6
CHI	7	4	2	5	1	3	6
SET	7	4	2	5	1	3	6
GD DS	1	2	5	4	7	3	6
RLF	5	2	4	7	1	6	3
SCH	3	6	1	5	2	4	7
FUZ	2	3	4	5	7	1	6

*Tabla 23. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema PI.*

**BASE DE DATOS PI**

Método	Ordenación de las entradas por importancia								
	Menor importancia				Mayor importancia				
	1	2	3	4	5	6	7	8	9
BL2	5	6	1	8	4	2	9	7	3
TEKA	5	6	1	8	4	2	9	7	3
CIB	6	5	8	1	2	9	4	7	3
TEKE	3	4	7	2	8	9	5	6	1
TEKC	6	8	2	9	7	4	3	5	1
DEV	5	6	8	1	4	2	9	3	7
DER3	4	2	9	3	8	7	6	1	5
TEKB	6	5	1	8	4	2	9	3	7
TEK	6	5	8	2	9	1	4	7	3
BL1	1	6	5	9	8	2	4	3	7
CLO	1	6	5	9	8	2	4	3	7
PRI	1	6	5	9	8	2	4	3	7
SAN	1	6	5	8	9	2	4	3	7
DER2	1	6	9	5	8	2	4	3	7
MAO	6	5	8	2	1	9	4	7	3
UTA	6	1	5	8	9	4	2	7	3
LEE	3	7	8	5	4	9	6	2	1
BOW	6	9	5	8	2	4	1	7	3
YOU	6	5	8	9	2	1	4	7	3
BA 0.5	1	9	5	4	6	8	2	7	3
BA 0.6	1	9	5	4	6	8	2	7	3
BA 0.7	1	9	5	4	6	8	2	7	3
BA 0.8	1	9	5	4	6	8	2	7	3
BA 0.9	1	9	5	4	6	8	2	7	3
BA 1	1	9	5	4	6	8	2	7	3
CHI	8	1	4	5	2	6	9	7	3
SET	1	8	4	2	5	6	9	7	3
GD_DS	1	8	4	7	3	2	9	5	6
RLF	6	8	5	2	9	4	1	7	3
SCH	3	2	9	6	4	7	5	8	1
FUZ	2	4	5	9	6	7	3	1	8

Tabla 24. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema VO.

**BASE DE DATOS VO**

Método	Ordenación de las entradas por importancia																
	Menor importancia								Mayor importancia								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
BL2	1	2	17	11	3	14	9	7	12	6	8	15	16	10	13	4	5
TEKA	1	11	2	17	3	14	9	6	16	8	7	15	12	13	4	10	5
CIB	1	2	17	11	16	8	9	13	6	14	10	15	3	12	4	7	5
TEKE	4	12	16	10	3	6	15	2	9	17	1	5	13	11	8	7	14
TEKC	11	8	1	12	5	2	4	15	6	13	9	17	16	10	14	3	7
DEV	17	1	2	11	3	14	12	10	9	6	7	13	16	8	15	4	5
DER3	4	10	8	2	5	7	16	12	11	9	6	14	3	13	15	1	17
TEKB	2	11	1	17	14	3	7	6	16	9	12	15	8	13	10	4	5
TEK	2	1	11	12	16	14	3	13	17	6	9	8	10	15	7	4	5
BL1	1	6	2	14	17	11	9	15	8	16	3	13	7	4	10	12	5
CLO	1	2	6	9	14	11	15	8	17	16	3	13	7	10	4	12	5
PRI	1	6	2	14	17	11	9	15	8	16	3	13	7	4	10	12	5
SAN	1	2	11	17	6	14	9	15	16	8	3	7	13	10	12	4	5
DER2	1	2	6	17	11	14	16	9	15	8	13	3	7	12	10	4	5
MAO	2	1	17	9	11	8	16	6	13	14	15	10	12	3	4	7	5
UTA	1	2	17	6	9	15	11	14	16	8	7	13	3	12	10	4	5
LEE	5	4	10	12	7	3	13	16	8	11	14	15	9	6	17	1	2
BOW	8	6	11	9	2	16	15	17	13	1	7	3	14	10	4	12	5
YOU	17	2	9	11	8	6	15	1	14	13	7	3	16	12	4	10	5
BA 0.5	9	6	8	15	10	14	7	1	2	16	17	3	11	12	4	13	5
BA 0.6	6	9	8	15	4	7	14	1	10	2	16	3	17	12	11	13	5
BA 0.7	6	9	8	15	4	14	7	1	10	2	16	17	12	3	11	13	5
BA 0.8	6	9	8	15	4	14	7	10	1	2	16	17	12	3	11	13	5
BA 0.9	6	9	15	8	4	14	10	7	1	2	16	17	12	3	11	13	5
BA 1	6	9	15	8	4	13	14	7	1	2	16	17	3	11	10	12	5
CHI	3	11	1	12	17	2	7	16	8	14	15	10	9	13	6	4	5
SET	3	11	1	12	17	2	7	16	14	8	10	15	9	13	6	4	5
GD DS	1	3	11	12	17	2	16	7	14	8	15	13	10	4	9	6	5
RLF	1	10	13	17	16	7	9	8	6	3	14	2	15	11	4	12	5
SCH	5	4	13	6	9	10	8	14	15	16	2	7	17	12	1	11	3
FUZ	Excepciones cero, no aplicable																

Tabla 25. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema WD.

**BASE DE DATOS WD (parte 1)**

Ordenación de las entradas por importancia																
Método	Menor importancia								Mayor importancia							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BL2	31	18	27	7	2	19	4	14	5	21	16	20	3	15	13	17
TEKA	31	27	7	18	16	20	2	10	4	19	6	13	14	5	1	3
CIB	31	16	18	20	15	14	27	21	10	7	13	12	6	19	1	2
TEKE	29	17	9	8	28	19	2	6	4	15	27	5	24	18	31	14
TEKC	15	21	14	12	19	18	16	20	13	25	17	8	31	9	7	28
DEV	18	31	2	4	7	27	14	19	3	15	30	5	21	17	12	23
DER3	7	1	11	18	2	4	5	10	29	22	24	27	14	12	31	19
TEKB	31	16	20	10	6	27	7	1	18	2	13	4	19	3	5	14
TEK	31	18	16	20	15	14	21	27	13	7	19	12	10	6	5	2
BL1	31	1	16	10	20	6	7	27	2	4	13	18	19	5	3	14
CLO	31	1	16	20	27	10	6	2	7	4	18	19	13	5	3	14
PRI	31	1	16	10	20	6	7	27	2	4	13	18	19	5	3	14
SAN	31	1	16	20	10	27	6	7	2	4	18	13	19	3	5	14
DER2	31	16	20	18	15	6	7	10	27	2	19	1	4	13	14	5
MAO	27	31	18	16	20	7	14	15	10	21	2	4	6	13	12	19
UTA	16	20	27	18	31	1	4	2	6	7	10	19	5	13	21	15
LEE	23	26	11	30	8	29	22	9	21	12	25	28	17	24	3	13
BOW	6	18	20	19	16	13	7	10	31	27	4	2	21	17	11	5
YOU	18	31	6	19	17	21	20	16	7	13	27	10	15	14	1	2
BA 0.5	4	22	8	24	7	2	9	28	27	17	5	11	1	26	10	19
BA 0.6	4	24	22	8	7	2	9	28	27	17	5	11	1	26	10	19
BA 0.7	4	24	22	8	7	2	9	28	27	17	5	11	1	26	10	19
BA 0.8	4	24	22	8	7	2	9	28	27	5	17	11	1	26	10	19
BA 0.9	4	24	22	8	7	2	9	28	27	5	17	11	26	1	10	19
BA 1	4	24	22	8	7	2	9	28	27	5	17	11	26	1	10	19
CHI	13	16	1	21	20	21	31	6	10	30	18	17	26	19	3	23
SET	1	13	16	11	20	11	6	31	10	26	30	17	23	3	19	18
GD DS	1	16	13	20	30	10	6	26	11	31	21	23	3	19	17	18
RLF	18	31	21	11	17	15	16	19	14	7	20	1	12	27	10	6
SCH	29	8	24	9	22	2	4	25	28	27	5	7	26	23	30	31
FUZ	31	1	20	7	3	8	15	6	2	14	5	27	4	28	9	11

Tabla 25. Ordenación de las entradas según su importancia para los métodos ARNMF y ACE. Problema WD. (Continuación).

**BASE DE DATOS WD (parte 2)**

Método	Ordenación de las entradas por importancia														
	Menor importancia							Mayor importancia							
	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
BL2	10	28	6	30	12	24	25	26	22	8	11	23	29	9	1
TEKA	21	15	17	28	30	12	24	26	11	25	8	22	29	23	9
CIB	4	5	17	28	25	3	8	11	30	24	9	22	26	29	23
TEKE	22	12	7	25	16	20	10	13	21	3	23	11	30	1	26
TEKC	27	5	11	30	22	24	6	26	23	4	10	1	29	3	2
DEV	28	16	20	24	26	13	22	10	25	6	8	29	9	11	1
DER3	3	15	30	23	25	26	16	20	28	13	6	21	17	9	8
TEKB	21	15	28	17	30	12	24	11	26	8	25	22	29	23	9
TEK	4	17	28	1	25	3	8	30	11	24	9	22	26	23	29
BL1	28	15	21	17	30	12	24	11	26	8	25	22	29	23	9
CLO	28	21	15	17	30	24	12	11	8	26	25	29	22	9	23
PRI	28	15	21	17	30	12	24	11	26	8	25	22	29	23	9
SAN	28	21	15	17	30	24	12	11	26	8	25	29	22	23	9
DER2	21	12	3	25	17	24	28	8	11	22	9	30	29	26	23
MAO	5	1	28	17	3	25	8	30	11	24	9	22	26	29	23
UTA	14	24	3	17	28	25	12	22	8	9	29	11	30	26	23
LEE	5	19	14	15	20	1	2	4	18	10	16	31	7	6	27
BOW	3	1	15	30	14	28	24	26	25	29	8	22	12	9	23
YOU	4	11	12	5	3	28	30	25	24	8	26	22	23	9	29
BA 0.5	23	14	25	31	6	3	20	12	21	13	30	18	16	15	29
BA 0.6	23	14	25	31	6	3	20	12	21	30	13	18	16	15	29
BA 0.7	23	14	25	31	6	3	20	12	21	30	13	18	16	15	29
BA 0.8	23	14	25	31	3	6	12	20	21	30	13	18	16	15	29
BA 0.9	23	14	25	31	3	6	12	20	21	30	13	18	16	15	29
BA 1	23	14	25	31	3	6	12	20	21	30	13	18	16	15	29
CHI	12	14	27	7	15	5	28	2	4	8	9	22	25	24	29
SET	27	7	12	2	28	14	4	5	29	8	9	22	24	25	15
GD_DS	27	7	28	29	8	9	12	14	15	2	4	5	24	22	25
RLF	30	28	13	8	5	3	4	2	26	25	24	22	9	23	29
SCH	12	3	14	15	19	10	17	6	11	20	18	21	16	13	1
FUZ	16	30	10	19	13	22	26	23	25	12	21	17	29	18	24

## II.2. Ordenaciones de las entradas para ARNMF&WD

Tabla 26. Ordenación de las entradas para ARNMF&WD. Problema BN.

### BASE DE DATOS BN (parte 1)

		Ordenación de las entradas por importancia																			
		Menor importancia										Mayor importancia									
Método		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
BL2		13	5	22	27	12	10	7	14	33	20	25	28	29	36	2	34	15	9	24	18
TEKA		13	5	22	27	12	20	33	14	7	10	25	28	36	29	2	15	34	9	24	18
CIB		13	34	9	33	22	5	12	27	20	10	25	7	28	14	29	36	2	24	11	18
TEKE		Todos cero																			
DEV		13	22	12	14	5	27	10	7	33	25	20	28	11	15	36	29	2	34	24	9
DER3		13	23	19	31	37	5	38	29	2	3	8	6	24	28	26	25	21	10	7	27
TEKB		13	5	22	27	10	14	7	12	33	25	28	20	36	29	2	15	34	9	24	18
TEK		13	9	34	33	5	22	12	27	25	29	28	10	20	7	36	14	2	24	11	21
BL1		14	13	22	5	12	27	10	7	20	33	25	28	29	36	2	15	34	11	9	24
CLO		14	22	5	13	12	27	10	7	20	33	25	28	29	36	2	15	34	11	9	24
PRI		14	13	22	5	12	27	10	7	20	33	25	28	29	36	2	15	34	11	9	24
SAN		14	13	22	5	12	27	10	20	7	33	25	28	29	36	2	15	34	9	11	24
DER2		22	14	12	13	5	10	7	27	28	20	25	33	29	2	34	26	24	9	36	15
MAO		18	32	11	15	36	29	5	33	35	12	13	14	22	27	20	9	17	34	2	10
UTA		27	14	20	7	10	13	12	5	33	22	2	29	25	24	36	26	28	9	34	18
LEE		23	19	8	37	6	38	40	31	3	30	1	16	17	35	39	21	26	24	4	11
YOU		19	23	21	38	14	26	8	31	22	2	25	27	9	7	34	13	10	28	3	24

### BASE DE DATOS BN (parte 2)

		Ordenación de las entradas por importancia																			
		Menor importancia										Mayor importancia									
Método		21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
BL2		11	26	35	32	21	17	3	8	39	38	40	31	4	6	1	30	16	37	19	23
TEKA		11	26	32	35	17	21	3	39	8	40	4	38	31	1	6	30	16	37	19	23
CIB		26	15	32	21	17	4	31	39	35	3	37	38	1	23	16	8	6	30	40	19
TEKE		Todos cero																			
DEV		21	18	26	32	17	35	3	8	6	38	39	31	4	40	16	37	1	30	19	23
DER3		22	14	12	11	33	15	20	36	34	18	9	17	32	16	4	39	1	40	30	35
TEKB		11	26	21	32	35	17	3	8	39	38	40	31	4	6	1	16	30	37	19	23
TEK		18	26	15	32	4	17	37	3	31	39	35	6	38	1	16	23	8	30	40	19
BL1		18	26	21	35	32	17	3	8	39	38	40	6	31	4	1	16	30	37	19	23
CLO		18	26	21	35	32	17	3	8	39	38	40	6	31	4	1	16	30	37	19	23
PRI		18	26	21	35	32	17	3	8	39	38	40	6	31	4	1	16	30	37	19	23
SAN		18	26	21	35	32	17	3	8	39	38	40	31	6	4	1	16	30	37	19	23
DER2		21	11	18	32	38	3	35	8	17	6	31	37	39	19	4	40	23	1	16	30
MAO		7	25	39	21	28	24	16	40	1	26	4	30	31	3	38	37	6	8	23	19
UTA		32	11	21	15	35	8	38	31	17	3	39	40	37	30	4	1	23	16	6	19
LEE		28	18	29	34	10	7	15	25	27	32	36	33	5	9	13	14	22	2	12	20
YOU		29	33	20	37	5	12	6	36	18	32	15	4	35	11	39	17	1	40	30	16



Tabla 27. Ordenación de las entradas para ARNMF&WD. Problema BU.

**BASE DE DATOS BU**

Ordenación de las entradas por importancia							
Método	Menor importancia				Mayor importancia		
	1	2	3	4	5	6	7
BL2	1	4	7	5	6	2	3
TEKA	1	4	7	5	6	2	3
CIB	6	4	1	5	7	3	2
TEKE	2	6	4	5	1	3	7
DEV	1	7	2	4	3	6	5
DER3	3	4	1	7	2	6	5
TEKB	1	4	7	5	6	2	3
TEK	4	7	6	1	5	3	2
BL1	1	7	4	5	6	2	3
CLO	1	7	4	5	6	2	3
PRI	1	7	4	5	6	2	3
SAN	1	7	4	5	6	2	3
DER2	1	4	7	2	5	3	6
MAO	5	1	7	4	6	2	3
UTA	1	7	4	2	5	6	3
LEE	1	2	3	4	5	6	7
YOU	4	7	1	3	2	5	6

Tabla 28. Ordenación de las entradas para ARNMF&WD. Problema CR.

**BASE DE DATOS CR**

Ordenación de las entradas por importancia																
Método	Menor importancia								Mayor importancia							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BL2	7	2	1	9	13	14	12	15	5	6	16	4	3	8	11	10
TEKA	7	2	1	9	14	13	12	15	6	5	4	16	3	11	8	10
CIB	16	2	12	7	9	15	1	14	4	5	6	8	3	13	10	11
TEKE	Todos cero															
DEV	7	2	1	9	13	14	12	15	5	6	16	4	3	8	11	10
DER3	3	11	14	13	16	6	2	1	7	9	12	8	4	15	5	10
TEKB	7	2	1	9	13	14	12	15	6	5	16	4	3	8	11	10
TEK	16	12	9	2	14	7	15	5	6	4	1	8	13	3	11	10
BL1	7	2	1	9	14	13	12	15	5	6	16	4	3	8	11	10
CLO	7	2	1	9	14	13	12	15	5	6	16	4	3	8	11	10
PRI	7	2	1	9	14	13	12	15	5	6	16	4	3	8	11	10
SAN	7	2	1	9	14	13	12	15	5	6	16	4	3	8	11	10
DER2	7	2	1	15	5	6	9	14	16	13	12	8	4	3	11	10
MAO	5	6	8	2	14	15	9	7	12	16	1	4	11	13	10	3
UTA	2	9	7	1	12	14	16	15	5	6	4	3	13	8	11	10
LEE	10	11	5	15	6	13	8	3	4	2	14	16	7	12	1	9
YOU	1	7	16	15	14	9	2	13	12	4	8	5	6	3	11	10

Tabla 29. Ordenación de las entradas para ARNMF&WD. Problema GL.

**BASE DE DATOS GL**

Ordenación de las entradas por importancia											
	Menor importancia						Mayor importancia				
Método	1	2	3	4	5	6	7	8	9	10	11
BL2	8	7	1	11	4	9	3	10	6	5	2
TEKA	8	7	1	11	9	4	3	10	6	5	2
CIB	8	11	7	1	10	4	9	3	6	5	2
TEKE	11	8	7	9	4	10	3	6	5	1	2
DEV	8	7	1	11	4	9	3	10	6	5	2
DER3	2	6	10	4	8	7	1	11	9	3	5
TEKB	8	7	1	11	9	4	3	10	6	5	2
TEK	8	11	10	7	1	9	3	4	6	5	2
BL1	8	7	1	11	4	9	3	10	6	5	2
CLO	8	7	1	11	4	9	3	10	6	5	2
PRI	8	7	1	11	4	9	3	10	6	5	2
SAN	8	7	1	11	4	9	3	6	10	5	2
DER2	8	7	4	11	1	9	10	6	3	2	5
MAO	8	7	1	11	4	9	10	3	6	5	2
UTA	1	8	7	11	4	9	3	10	6	5	2
LEE	5	2	6	3	9	10	4	1	8	7	11
YOU	División por cero										

Tabla 30. Ordenación de las entradas para ARNMF&WD. Problema HE.

**BASE DE DATOS HE**

Ordenación de las entradas por importancia														
	Menor importancia							Mayor importancia						
Método	1	2	3	4	5	6	7	8	9	10	11	12	13	14
BL2	1	8	6	2	5	7	12	11	10	4	3	14	9	13
TEKA	1	8	2	6	5	7	12	11	10	4	3	14	9	13
CIB	1	6	8	2	11	5	7	12	10	13	14	4	3	9
TEKE	14	12	5	11	1	2	6	8	13	10	7	3	9	4
DEV	1	6	8	2	5	7	12	11	10	4	3	14	9	13
DER3	13	3	4	11	14	12	10	2	1	6	8	5	7	9
TEKB	1	8	6	2	5	7	12	11	10	4	3	14	9	13
TEK	1	6	7	8	2	5	11	12	10	14	13	3	4	9
BL1	8	1	2	6	5	7	12	11	10	4	3	14	9	13
CLO	8	1	2	6	5	7	12	11	10	4	3	14	9	13
PRI	8	1	2	6	5	7	12	11	10	4	3	14	9	13
SAN	1	8	6	2	5	7	12	11	10	4	3	14	9	13
DER2	1	8	6	2	5	11	12	7	10	4	3	14	9	13
MAO	14	3	4	12	11	6	1	8	10	13	2	5	7	9
UTA	1	2	8	5	6	11	12	7	9	4	10	3	14	13
LEE	13	14	4	3	10	7	9	11	12	1	6	8	5	2
YOU	4	3	6	12	11	1	10	7	8	14	2	5	13	9

Tabla 31. Ordenación de las entradas para ARNMF&WD. Problema PI.

**BASE DE DATOS PI**

Método	Ordenación de las entradas por importancia								
	Menor importancia					Mayor importancia			
	1	2	3	4	5	6	7	8	9
BL2	6	1	5	9	8	2	4	7	3
TEKA	6	5	1	9	8	4	2	7	3
CIB	6	1	5	9	8	2	4	7	3
TEKE	8	5	2	9	6	1	4	7	3
DEV	1	6	9	5	8	2	4	7	3
DER3	7	3	8	2	1	9	5	6	4
TEKB	1	6	5	9	8	2	4	7	3
TEK	6	5	9	1	8	2	4	7	3
BL1	1	6	5	9	8	2	4	7	3
CLO	1	6	5	9	8	2	4	7	3
PRI	1	6	5	9	8	2	4	7	3
SAN	1	6	5	9	8	2	4	7	3
DER2	6	1	5	9	8	2	4	7	3
MAO	2	9	8	1	6	5	7	3	4
UTA	6	5	1	9	8	4	2	7	3
LEE	3	7	2	4	5	6	8	9	1
YOU	4	5	6	9	1	8	2	7	3

Tabla 32. Ordenación de las entradas para ARNMF&WD. Problema VO.

**BASE DE DATOS VO**

Método	Ordenación de las entradas por importancia																
	Menor importancia									Mayor importancia							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
BL2	14	15	9	6	11	16	2	1	17	7	8	3	13	10	4	12	5
TEKA	14	15	9	6	2	11	16	1	7	8	17	3	13	10	4	12	5
CIB	14	15	6	16	11	9	1	2	13	3	17	7	8	10	12	4	5
TEKE	4	8	1	14	16	6	15	9	11	2	7	3	17	13	10	12	5
DEV	14	15	6	9	11	16	2	1	17	7	8	3	13	10	4	12	5
DER3	9	5	2	8	17	13	11	14	6	15	16	1	3	10	7	4	12
TEKB	14	15	6	9	11	16	2	1	17	7	8	3	13	10	4	12	5
TEK	14	15	6	16	11	2	9	1	13	3	8	7	17	10	12	4	5
BL1	14	15	6	9	11	16	2	1	17	7	8	3	13	10	4	12	5
CLO	14	15	6	9	11	16	2	1	17	7	8	3	13	10	4	12	5
PRI	14	15	6	9	11	16	2	1	17	7	8	3	13	10	4	12	5
SAN	14	15	6	9	11	16	2	1	17	7	8	3	13	10	4	12	5
DER2	14	15	6	9	11	2	16	17	8	1	7	13	3	10	4	12	5
MAO	17	13	8	6	11	2	9	14	15	16	5	1	3	7	10	4	12
UTA	6	14	15	16	1	11	2	9	17	8	7	13	3	10	4	12	5
LEE	5	10	12	4	3	1	7	16	2	8	9	11	14	15	17	6	13
YOU	8	17	9	2	15	11	14	6	13	7	3	1	16	12	10	5	4

Tabla 33. Ordenación de las entradas para ARNMF&WD. Problema WD.

**BASE DE DATOS WD (parte 1)**

		Ordenación de las entradas por importancia															
		Menor importancia								Mayor importancia							
Método		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BL2		7	6	31	10	1	16	20	27	13	19	2	4	18	5	21	14
TEKA		7	6	31	10	1	16	20	27	13	19	2	4	18	5	21	14
CIB		7	16	31	15	20	18	13	14	6	27	21	10	12	19	1	2
TEKE		30	18	7	2	4	25	5	22	15	27	6	31	14	16	24	12
DEV		31	2	27	16	20	4	6	7	10	13	5	1	18	28	19	14
DER3		23	26	1	30	9	25	22	29	12	3	24	8	15	10	14	28
TEKB		7	6	31	10	1	16	20	27	13	19	2	4	18	5	21	14
TEK		7	31	16	18	20	15	13	6	27	21	14	10	19	1	12	2
BL1		7	6	31	10	1	16	20	27	13	19	2	4	18	5	21	14
CLO		7	6	31	10	1	16	20	27	13	19	2	4	18	5	21	14
PRI		7	6	31	10	1	16	20	27	13	19	2	4	18	5	21	14
SAN		7	6	31	10	1	16	20	27	13	19	2	4	18	5	21	14
DER2		7	31	15	20	16	6	10	1	2	4	14	27	12	13	5	19
MAO		7	31	16	27	20	18	6	10	13	1	15	19	14	21	2	4
UTA		16	6	20	7	18	19	27	31	13	10	1	2	4	21	5	15
LEE		9	23	29	22	25	8	26	24	12	30	11	3	28	17	15	14
YOU		23	26	29	22	30	9	3	24	8	25	28	4	1	2	5	12

**BASE DE DATOS WD (parte 2)**

		Ordenación de las entradas por importancia															
		Menor importancia								Mayor importancia							
Método		17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
BL2		28	15	3	17	30	12	24	26	11	8	25	22	29	9	23	
TEKA		28	15	3	17	30	12	26	24	11	8	25	22	29	9	23	
CIB		4	5	17	28	25	8	3	11	30	24	22	9	26	29	23	
TEKE		19	21	10	13	20	17	28	9	1	8	3	11	29	26	23	
DEV		15	3	21	24	8	12	17	25	22	29	30	11	26	9	23	
DER3		5	27	4	31	2	16	20	6	13	18	19	21	17	11	7	
TEKB		28	15	3	17	30	12	24	26	11	8	25	22	29	9	23	
TEK		4	5	17	28	25	8	3	30	11	24	22	9	26	29	23	
BL1		28	15	3	17	30	12	24	26	11	8	25	22	29	9	23	
CLO		28	15	3	17	30	12	24	26	11	8	25	22	29	9	23	
PRI		28	15	3	17	30	12	24	26	11	8	25	22	29	9	23	
SAN		28	15	3	17	30	12	24	26	11	8	25	22	29	9	23	
DER2		25	18	3	24	28	21	8	22	17	9	30	26	11	29	23	
MAO		5	12	28	17	25	3	8	30	24	11	9	22	26	29	23	
UTA		14	28	17	3	24	12	25	22	11	8	30	29	9	26	23	
LEE		5	21	4	2	18	1	27	10	13	16	20	19	31	7	6	
YOU		10	14	27	31	15	7	18	16	20	13	6	21	19	17	11	

### II.3. Eficiencias para los métodos ARNMF y ACE

Tabla 34. Eficiencias para los métodos ARNMF y ACE.  
Problema AB.

#### BASE DE DATOS AB

Método	1	2	3	4	5	6	7	8
BL2	51±3	53±2	48±3	51.3±1.8	53±2	50±3	55.3±0.5	50.1±1.7
TEKA	51±3	53±2	48±3	51.3±1.8	53±2	50±3	55.3±0.5	50.0±1.7
CIB	52±3	53±2	48±3	52.7±1.8	53±2	50±3	49±2	50.0±1.7
TEKE	58.7±0.8	55±2	57.9±1.0	57.5±0.8	58.1±0.7	53.9±1.8	50±3	46±3
TEKC	52±3	53±2	54±2	55±3	50±2	50±3	55.3±0.5	50.0±1.7
DEV	51±3	53±2	50±3	51.3±1.8	53±2	50±3	55.3±0.5	50.1±1.7
DER3	58.7±0.8	53±2	52±3	54±2	54±2	52±2	51±2	52.0±1.9
TEKB	51±3	53±2	48±3	51.3±1.8	53±2	50±3	55.3±0.5	46±3
TEK	52±3	54±2	51±2	51±3	53±2	50±3	49±2	51.0±1.8
BL1	51±3	53±2	48±3	53±2	52±2	52±2	55.5±1.7	46±3
CLO	51±3	53±2	60.1±0.4	57±3	52±2	52±2	55.5±1.7	46±3
PRI	51±3	53±2	50±3	53±2	52±2	52±2	55.5±1.7	46±3
SAN	51±3	53±2	48±3	53±2	53.7±1.5	50±3	55.3±0.5	46±3
DER2	52±3	53±2	52±3	54±2	52±2	57.4±0.8	55.3±0.5	50.1±1.7
MAO	51±3	53±2	48±3	57.2±0.7	53±2	50±3	49±2	51.0±1.8
UTA	51±3	53±2	48±3	53±2	52±2	57.4±0.8	55.5±1.7	46±3
LEE	48±2	50±2	54±3	49±2	46±2	45±3	44±2	46±2
BOW	52±3	50±2	51±2	50.9±1.9	49±3	47±3	49±3	51.0±1.8
YOU	----	----	----	----	----	----	----	----
BA 0.5	51±2	56.7±1.3	48±3	52±2	54±2	55.4±0.3	55.4±0.4	46±3
BA 0.6	51±2	56.7±1.3	48±3	52±2	54±2	55.4±0.3	55.4±0.4	46±3
BA 0.7	51±2	53±2	48±3	52±2	54±2	55.4±0.3	55.4±0.4	46±3
BA 0.8	51±2	53±2	48±3	52±2	54±2	55.4±0.3	55.4±0.4	46±3
BA 0.9	51±2	53±2	48±3	52±2	53±2	55.4±0.3	45±3	46±3
BA 1	51±2	53±2	48±3	52±2	54±2	55.4±0.3	45±3	46±3
CHI	51±3	55±2	57.9±1.0	57.6±0.7	50.0±1.9	55±2	52±2	46±3
SET	51±3	48±2	49±2	53±2	52±2	52±3	51.2±1.9	41±3
GD_DS	51±3	55±2	57.9±1.0	51±2	53±3	58.9±0.6	49±3	51.0±1.8
RLF	----	----	----	----	----	----	----	----
SCH	58.7±0.8	53±2	58.7±0.8	48±2	49±3	51±2	51±2	44±3
FUZ	58.7±0.8	57.2±0.9	54±2	54±2	54.7±1.9	51±2	46±2	44±3

**Tabla 35. Eficiencias para los métodos ARNMF y ACE.  
Problema BL.**

**BASE DE DATOS BL**

<b>Método</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>BL2</b>	48±2	49±3	48±2	47.8±0.5
<b>TEKA</b>	48±2	49±3	49±2	53±3
<b>CIB</b>	49±2	54±3	49±2	53±3
<b>TEKE</b>	53±3	53±3	50±2	55±2
<b>TEKC</b>	50±3	49±2	50±2	54±2
<b>DEV</b>	49±2	54±3	49±2	53±3
<b>DER3</b>	50±3	58±3	55±3	53±3
<b>TEKB</b>	50±3	45.9±1.6	51±3	53±3
<b>TEK</b>	50±3	49±2	51±3	53±3
<b>BL1</b>	49±2	54±3	51±3	53±3
<b>CLO</b>	49±2	54±3	49±2	53±3
<b>PRI</b>	49±2	54±3	51±3	53±3
<b>SAN</b>	48±2	54±3	50±2	47.8±0.5
<b>DER2</b>	53±3	50±3	53±3	54±2
<b>MAO</b>	49±2	54±3	53±3	53±3
<b>UTA</b>	49±3	56±3	49±2	53±3
<b>LEE</b>	53±3	53±3	54±2	47.8±0.5
<b>BOW</b>	49±3	50±3	50±2	55±3
<b>YOU</b>	----	----	----	----
<b>BA 0.5</b>	49±3	54±3	51±3	54±2
<b>BA 0.6</b>	49±3	54±3	51±3	54±2
<b>BA 0.7</b>	49±3	54±3	51±3	54±2
<b>BA 0.8</b>	49±3	54±3	51±3	54±2
<b>BA 0.9</b>	49±3	54±3	51±3	54±2
<b>BA 1</b>	49±3	54±3	51±3	54±2
<b>CHI</b>	49±3	54±3	51±3	54±2
<b>SET</b>	49±3	54±3	51±3	54±2
<b>GD_DS</b>	49±3	54±3	51±3	54±2
<b>RLF</b>	----	----	----	----
<b>SCH</b>	47±3	56±3	54±3	47.8±0.5
<b>FUZ</b>	53±3	56±3	54±3	55±3

**Tabla 36. Eficiencias para los métodos ARNMF y ACE.  
Problema BN.**

**BASE DE DATOS BN (parte 1)**

Método	1	2	3	4	5	6	7	8
BL2	65.1±1.0	62.9±0.3	64.6±0.6	63.8±1.1	65.1±1.1	64.8±0.8	63.8±1.1	64.1±0.7
TEKA	65.1±1.0	62.9±0.3	64.6±0.6	63.6±1.3	64.6±0.4	64.8±0.8	63.8±1.1	64.1±0.5
CIB	65.1±1.0	66.4±0.9	64.9±0.8	66.9±0.9	64.5±0.5	64.1±0.7	64.0±0.7	66.8±0.9
TEKE	----	----	----	----	----	----	----	----
TEKC	65.1±1.0	66.4±0.8	65.0±0.6	66.0±0.8	66.6±1.0	67.0±0.9	67.5±0.9	63.3±0.7
DEV	65.1±1.0	62.9±0.3	64.6±0.6	63.8±1.1	65.1±1.1	64.8±0.8	63.8±1.1	64.1±0.7
DER3	64.1±0.7	64.0±0.8	64.0±1.0	62.5±0.7	65.4±0.6	66.0±0.9	66.0±0.6	64.5±0.7
TEKB	65.1±1.0	62.9±0.3	63.1±0.8	65.6±0.7	65.1±1.1	64.8±0.8	63.4±1.2	64.1±0.5
TEK	65.1±1.0	64.4±0.9	65.1±0.6	66.0±0.8	66.1±1.0	64.5±1.1	64.8±0.6	66.8±0.9
BL1	65.1±1.0	63.9±0.5	64.9±0.4	65.6±0.6	65.2±0.9	64.5±1.0	64.0±0.8	63.3±1.2
CLO	65.1±1.0	63.9±0.5	65.5±0.6	63.3±0.6	64.7±0.7	64.5±1.0	65.2±0.6	63.3±1.2
PRI	65.1±1.0	63.9±0.5	64.9±0.4	65.6±0.6	65.2±0.9	64.5±1.0	64.0±0.8	63.3±1.2
SAN	65.1±1.0	62.9±0.3	63.1±0.8	63.6±1.3	65.0±1.1	64.8±0.8	65.2±0.6	63.3±1.2
DER2	65.1±1.0	62.9±0.3	64.6±0.6	63.8±1.1	65.1±1.1	64.8±0.8	62.3±0.4	63.3±1.2
MAO	64.9±1.1	64.4±0.7	64.0±0.6	65.6±0.7	65.0±1.1	64.8±0.5	66.2±1.2	64.6±0.6
UTA	64.9±1.1	64.3±0.7	65.7±0.7	63.6±1.3	65.1±1.1	64.8±0.5	63.8±1.1	64.5±0.8
LEE	63.2±0.6	71.7±0.9	70±0.7	71.5±0.3	68.5±0.9	68.4±0.8	70.9±1.1	70.8±0.9
BOW	65.1±1.0	64.8±0.8	64.6±1.1	65.1±0.8	64.4±0.7	64.5±0.3	63.9±0.6	63.3±0.5
YOU	65.1±1.0	65.6±0.8	64.9±0.8	66.0±0.8	66.3±1.0	64.4±0.8	66.3±0.7	65.3±0.4
BA 0.5	65.5±0.7	72.9±0.4	72.0±0.8	74.1±0.5	72.1±0.9	72.4±0.8	73.8±1.0	74.7±1.1
BA 0.6	65.5±0.7	72.9±0.4	72.0±0.8	74.1±0.5	72.1±0.9	72.4±0.8	73.8±1.0	74.7±1.1
BA 0.7	65.5±0.7	72.9±0.4	72.0±0.8	74.1±0.5	72.1±0.9	72.4±0.8	73.8±1.0	74.7±1.1
BA 0.8	65.5±0.7	72.9±0.4	72.0±0.8	74.1±0.5	72.1±0.9	72.4±0.8	73.8±1.0	74.7±1.1
BA 0.9	65.5±0.7	72.9±0.4	72.0±0.8	74.1±0.5	72.1±0.9	72.4±0.8	73.8±1.0	74.7±1.1
BA 1	65.5±0.7	72.9±0.4	72.0±0.8	74.1±0.5	72.1±0.9	72.4±0.8	73.8±1.0	74.7±1.1
CHI	63.7±0.3	66.2±1.0	65.3±0.9	62.1±1.1	64.0±0.8	65.3±0.9	64.3±0.7	66.4±0.6
SET	64.9±1.0	66.2±1.0	65.3±0.9	62.5±0.7	65.2±1.0	65.2±1.0	66.4±1.4	65.4±0.8
GD_DS	----	----	----	----	----	----	----	----
RLF	63.9±1.0	64.0±0.7	64.4±0.7	63.6±0.7	64.0±0.8	63.7±0.6	63.7±0.4	67.9±1.6
SCH	65.7±0.9	66.3±0.7	66.2±0.8	65.4±0.4	66.0±0.6	64.9±0.7	63.0±1.2	62.8±0.6
FUZ	----	----	----	----	----	----	----	----

**Tabla 36. Eficiencias para los métodos ARNMF y ACE.  
Problema BN. (Continuación).**

**BASE DE DATOS BN (parte 2)**

Método	9	10	11	12	13	14	15	16
BL2	65.1±0.8	65.3±0.9	68.0±1.2	65.4±1.0	64.7±1.0	64.3±0.7	65.7±0.4	66.0±0.8
TEKA	65.1±0.8	65.3±0.9	68.0±1.2	65.4±1.0	64.7±1.0	64.3±0.7	65.7±0.4	66.0±0.4
CIB	65.1±0.5	65.9±0.7	66.0±0.6	64.7±1.0	67.1±0.9	66.2±0.9	65.3±0.7	66.1±0.7
TEKE	----	----	----	----	----	----	----	----
TEKC	66.4±1.2	64.0±0.7	65.5±0.7	66.7±0.7	66.3±0.7	66.2±0.5	66.8±0.9	66.9±0.7
DEV	65.1±0.8	67.4±1.0	67.0±0.9	65.7±0.6	67.1±0.8	64.9±1.2	65.5±0.8	67.0±1.2
DER3	65.5±1.1	62.8±1.1	67.7±1.1	65.4±1.1	64.6±1.1	67.1±1.0	67.0±1.0	63.3±0.8
TEKB	65.1±0.8	68.0±0.6	66.1±1.0	65.4±1.0	64.1±0.7	64.2±0.7	63.9±0.7	67.2±0.8
TEK	66.6±0.7	64.6±0.8	63.3±0.6	63.6±0.5	65.6±1.1	65.1±0.8	64.1±1.1	65.5±0.9
BL1	65.6±1.0	66.0±1.2	62.1±0.6	63.1±0.3	64.7±0.7	66.6±0.5	65.4±1.1	66.8±1.0
CLO	62.5±0.7	66.0±1.2	62.1±0.6	63.1±0.3	64.7±0.7	64.4±0.9	65.4±1.1	65.6±1.5
PRI	65.6±1.0	66.0±1.2	62.1±0.6	63.1±0.3	64.7±0.7	66.6±0.5	65.4±1.1	66.8±1.0
SAN	62.5±0.7	66.0±1.2	64.7±0.4	64.3±0.6	64.4±1.0	64.4±0.9	65.2±0.9	67.2±0.8
DER2	61.8±0.7	63.6±0.9	62.1±0.6	64.9±0.4	67.0±0.5	65.5±0.8	65.4±1.1	65.6±1.5
MAO	65.4±0.6	66.0±0.8	64.1±0.7	64.7±1.0	67.1±0.9	64.3±0.8	65.3±0.7	66.1±0.7
UTA	64.6±0.7	62.6±0.5	66.0±1.1	61.3±0.7	66.9±1.1	64.4±0.8	63.1±0.6	64.4±0.4
LEE	68.7±0.8	67.0±0.8	68.7±1.1	67.1±1.3	68.3±0.8	68.2±0.5	67.8±1.4	69.2±1.4
BOW	65.6±0.7	64.3±0.6	64.3±0.6	66.3±1.2	64.3±0.7	64.9±0.6	63.1±0.3	62.4±0.5
YOU	66.2±0.8	67.4±1.2	66.0±0.9	64.8±0.9	67.6±1.2	65.5±0.9	65.2±0.4	65.2±0.5
BA 0.5	73.3±0.9	74.0±0.8	71.4±0.9	76.2±0.9	76.7±0.5	75.9±0.9	75.5±1.0	76.6±1.2
BA 0.6	73.3±0.9	74.0±0.8	71.4±0.9	76.2±0.9	76.7±0.5	75.9±0.9	75.5±1.0	76.6±1.2
BA 0.7	73.3±0.9	74.0±0.8	71.4±0.9	76.2±0.9	76.7±0.5	75.9±0.9	75.5±1.0	76.6±1.2
BA 0.8	73.3±0.9	74.0±0.8	71.4±0.9	76.2±0.9	76.7±0.5	75.9±0.9	75.5±1.0	76.6±1.2
BA 0.9	73.3±0.9	74.0±0.8	71.4±0.9	76.2±0.9	76.7±0.5	75.9±0.9	75.5±1.0	76.6±1.2
BA 1	73.3±0.9	74.0±0.8	71.4±0.9	76.2±0.9	76.7±0.5	75.9±0.9	75.5±1.0	76.6±1.2
CHI	65.5±1.1	67.6±1.4	66.2±0.7	66.7±1.0	70.5±1.2	69.9±0.7	66.9±0.4	66.9±0.6
SET	66.1±0.6	66.4±0.6	65.2±0.7	65.1±0.7	64.5±0.5	63.3±0.4	64.1±0.7	66.8±0.6
GD_DS	----	----	----	----	----	----	----	----
RLF	68.9±1.1	67.2±0.9	67.5±1.0	69.8±1.5	67.7±1.6	67.6±1.1	65.3±1.3	68.7±0.7
SCH	64.1±0.6	63.7±0.7	60.3±1.4	60.3±0.9	61.3±0.9	62.0±1.0	63.2±1.1	60.3±1.3
FUZ	----	----	----	----	----	----	----	----



**Tabla 36. Eficiencias para los métodos ARNMF y ACE.  
Problema BN. (Continuación).**

**BASE DE DATOS BN (parte 3)**

Método	17	18	19	20	21	22	23	24
BL2	67.1±0.5	67.0±0.6	66.6±0.4	65.6±0.5	66.0±0.8	67.9±0.9	66.6±0.6	63.6±1.0
TEKA	66.3±0.9	67.0±0.6	66.6±0.4	65.5±0.9	64.5±0.7	65.1±0.4	65.1±0.5	65.9±1.2
CIB	66.8±0.5	66.0±0.6	65.7±0.8	65.6±0.4	66.2±0.8	64.5±0.4	64.1±0.4	61.3±0.5
TEKE	----	----	----	----	----	----	----	----
TEKC	66.3±1.0	68.7±0.8	67.5±0.7	68.2±0.7	65.9±0.6	64.4±0.7	66.2±0.8	65.2±0.2
DEV	68.3±1.0	67.1±0.5	66.3±1.1	67.1±0.8	67.1±0.8	62.8±0.6	62.8±0.4	61.6±0.5
DER3	66.7±1.0	67.0±0.7	68.3±0.8	65.2±0.7	63.9±1.1	61.1±1.1	59.5±0.5	61.3±0.7
TEKB	66.3±0.8	66.7±1.0	66.9±1.0	65.2±1.0	67.6±1.2	66.2±1.2	63.6±1.1	64.4±0.7
TEK	64.9±0.8	65.1±0.5	63.4±0.9	64.7±0.9	65.1±0.5	65.1±0.5	65.1±0.4	65.5±0.6
BL1	65.9±0.4	65.9±0.8	66.6±0.7	67.1±0.8	65.5±0.9	66.6±1.2	65.3±0.9	64.4±1.2
CLO	66.2±1.1	65.9±0.8	64.5±0.9	65.5±0.8	65.5±0.9	66.6±1.2	65.2±0.9	63.8±1.0
PRI	65.9±0.4	65.9±0.8	66.6±0.7	67.1±0.8	65.5±0.9	66.6±1.2	65.3±0.9	64.4±1.2
SAN	66.2±1.1	65.3±1.1	67.4±0.9	65.5±0.8	65.5±0.9	66.6±1.2	64.0±1.2	66.3±1.4
DER2	66.2±1.1	67.1±0.8	66.6±1.3	65.7±1.1	66.7±1.2	66.6±1.2	64.3±0.7	64.3±1.2
MAO	66.8±0.5	65.1±0.4	65.5±0.3	64.1±0.6	66.2±0.8	64.5±0.4	62.4±0.5	61.3±0.5
UTA	65.2±0.6	64.5±0.4	66.1±0.5	66.7±0.6	67.9±0.4	66.8±0.7	68.3±1.2	67.7±0.7
LEE	69.4±1.1	64.5±0.8	64.4±0.7	63.6±0.8	63.2±0.6	61.8±0.3	59.8±0.8	61.0±1.1
BOW	63.4±0.6	60.6±0.6	60.7±0.2	61.4±0.7	61.4±0.3	59.3±1.2	59.9±1.4	57.8±0.4
YOU	66.7±0.6	66.1±1.3	67.4±0.7	65.1±0.6	66.9±0.9	66.8±0.6	65.6±0.5	66.2±0.3
BA 0.5	76.1±0.8	75.4±1.4	74.1±0.9	73.6±1.2	73.4±1.0	70.2±1.0	70.7±1.1	69.0±1.6
BA 0.6	76.1±0.8	75.4±1.4	74.1±0.9	73.6±1.2	73.4±1.0	70.2±1.0	70.7±1.1	69.0±1.6
BA 0.7	76.1±0.8	75.4±1.4	74.1±0.9	73.6±1.2	73.4±1.0	70.2±1.0	70.7±1.1	69.0±1.6
BA 0.8	76.1±0.8	75.4±1.4	74.1±0.9	73.6±1.2	73.4±1.0	70.2±1.0	70.7±1.1	69.0±1.6
BA 0.9	76.1±0.8	75.4±1.4	76.8±0.6	73.6±1.2	73.4±1.0	70.2±1.0	70.7±1.1	69.0±1.6±
BA 1	76.1±0.8	75.4±1.4	76.8±0.6	73.6±1.2	73.4±1.0	70.2±1.0	70.7±1.1	69.0±1.6
CHI	66.3±1.2	68.7±0.9	68.2±0.5	68.5±0.5	69.7±0.9	68.2±0.4	67.5±0.3	67.8±0.5
SET	68.2±1.1	64.9±0.7	65.3±0.7	65.2±0.9	63.8±1.3	66.1±0.8	63.3±0.9	62.5±0.9
GD_DS	----	----	----	----	----	----	----	----
RLF	68.9±0.6	70.6±0.7	69.7±0.9	69.7±0.5	67.9±1.0	72.8±0.8	72.3±0.8	70.5±0.5
SCH	61.6±1.8	67.6±1.4	67.8±0.7	65.6±1.1	68.2±1.0	66.8±0.9	67.6±1.1	70.5±0.7
FUZ	----	----	----	----	----	----	----	----

**Tabla 36. Eficiencias para los métodos ARNMF y ACE.  
Problema BN. (Continuación).**

**BASE DE DATOS BN (parte 4)**

Método	25	26	27	28	29	30	31	32
BL2	65.3±1.2	62.8±0.8	61.5±1.0	59.3±1.1	63.9±0.8	63.4±0.5	63.2±0.8	62.0±0.4
TEKA	65.3±1.2	62.4±0.9	64.3±0.7	63.0±0.5	64.0±0.5	62.9±0.7	63.8±1.0	62.0±0.4
CIB	62.9±0.5	65.9±1.2	66.1±1.0	64.9±0.5	67.7±0.7	64.4±1.1	71.5±0.9	73.7±0.5
TEKE	----	----	----	----	----	----	----	----
TEKC	65.3±1.0	63.6±0.6	62.5±0.8	63.3±0.9	65.6±0.9	64.9±0.7	66.2±0.7	66.7±0.8
DEV	62.8±0.6	63.4±0.8	63.8±0.6	63.7±0.7	63.7±0.5	63.8±0.8	67.5±1.0	65.4±0.9
DER3	61.3±0.8	61.1±0.6	62.6±0.8	61.5±1.0	60.1±1.3	60.6±0.6	63.4±0.5	63.8±0.6
TEKB	65.3±1.2	63.3±0.5	64.4±1.0	61.4±1.3	62.6±1.1	67.4±1.2	68.2±0.9	69.0±0.7
TEK	65.5±0.3	68.5±0.8	68.0±0.7	66.1±0.6	67.7±0.7	67.9±1.1	64.4±0.7	67.0±0.8
BL1	60.5±0.8	59.4±1.3	60.2±1.0	61.3±0.9	61.8±0.4	62.2±0.6	63.2±0.3	61.6±0.7
CLO	66.4±1.3	59.2±0.2	60.2±1.0	61.3±0.9	61.8±0.4	62.2±0.6	63.2±0.3	61.6±0.7
PRI	60.5±0.8	59.4±1.3	60.2±1.0	61.3±0.9	61.8±0.4	62.2±0.6	63.2±0.3	61.6±0.7
SAN	66.4±1.3	59.2±0.2	60.5±0.4	61.3±0.9	61.8±0.4	62.2±0.6	63.2±0.3	61.6±0.7
DER2	62.9±0.6	64.8±0.8	62.3±0.5	62.8±0.8	61.1±0.4	63.6±0.8	66.8±0.7	64.8±0.4
MAO	63.0±0.4	64.7±0.8	66.1±1.0	64.9±0.5	65.2±0.7	64.4±1.1	71.5±0.9	73.7±0.5
UTA	65.6±0.9	65.4±1.4	63.4±0.7	66.8±1.0	68.6±0.7	67.7±0.8	66.4±0.5	66.0±0.6
LEE	60.1±0.9	61.1±0.6	64.3±1.0	61.4±1.1	62.5±1.2	59.7±1.0	64.3±0.6	67.0±0.4
BOW	59.8±0.8	63.4±0.8	63.4±0.5	63.8±1.0	65.9±0.7	64.7±0.8	64.7±0.6	66.2±0.4
YOU	65.6±0.7	61.7±0.7	61.0±0.4	65.6±0.6	64.8±0.8	63.6±0.7	69.0±0.7	70.3±0.4
BA 0.5	67.4±0.8	67.1±0.5	64.5±1.2	66.7±1.1	67.9±0.8	68.0±0.9	68.4±1.2	70.3±0.4
BA 0.6	67.4±0.8	67.1±0.5	64.5±1.2	66.7±1.1	67.9±0.8	68.0±0.9	68.4±1.2	70.3±0.4
BA 0.7	67.4±0.8	67.1±0.5	64.5±1.2	66.7±1.1	67.9±0.8	68.0±0.9	68.4±1.2	70.3±0.4
BA 0.8	67.4±0.8	67.1±0.5	67.1±0.6	65.2±1.4	67.9±0.8	68.0±0.9	68.4±1.2	70.3±0.4
BA 0.9	67.4±0.8	67.1±0.5	67.1±0.6	65.2±1.4	67.9±0.8	68.0±0.9	68.4±1.2	70.3±0.4
BA 1	67.4±0.8	67.1±0.5	67.1±0.6	65.2±1.4	67.9±0.8	68.0±0.9	68.4±1.2	70.3±0.4
CHI	68.6±0.8	67.1±0.6	69.3±0.7	70.7±0.7	67.1±0.5	67.2±0.6	69.1±0.4	68.9±0.5
SET	63.4±0.6	63.2±1.0	67.7±0.7	68.4±0.8	63±3	68.5±0.4	67.7±0.4	66.9±0.5
GD_DS	----	----	----	----	----	----	----	----
RLF	70.9±0.7	73.1±0.7	69.7±0.7	68.5±0.7	67.0±0.7	70.9±0.8	70.3±1.2	69.1±0.7
SCH	67.8±0.8	66.8±1.1	67.4±0.9	67.1±1.1	67.9±0.7	65.7±0.6	66.4±0.9	66.8±0.4
FUZ	----	----	----	----	----	----	----	----

**Tabla 36. Eficiencias para los métodos ARNMF y ACE.  
Problema BN. (Continuación).**

**BASE DE DATOS BN (parte 5)**

Método	33	34	35	36	37	38	39
BL2	62.1±1.2	69.7±0.8	70.6±0.7	69.4±1.8	61.5±0.9	61.3±0.8	64.6±0.5
TEKA	62.1±1.2	63.2±1.1	69.8±1.3	69.4±1.8	61.5±0.9	61.3±0.8	64.6±0.5
CIB	71.0±0.8	67.8±0.5	61.4±1.4	62.2±1.0	60.5±0.3	61.1±0.7	64.6±0.5
TEKE	----	----	----	----	----	----	----
TEKC	65.3±0.4	69.5±0.8	69.4±0.6	75.3±0.9	71.8±0.6	66.7±0	66.7±0
DEV	64.6±0.7	69.4±0.5	65.9±0.2	64.1±1.2	65.3±0.5	61.3±0.8	64.4±0.6
DER3	63.8±1.1	64.9±0.6	66.9±1.0	68.0±0.4	68.9±0.3	68.4±0.4	66.7±0
TEKB	62.9±0.9	69.0±0.6	62.3±1.1	64.9±0.3	65.3±0.5	61.3±0.8	64.6±0.5
TEK	67.1±0.8	66.4±0.2	64.6±0.6	65.5±0.5	60.5±0.3	60.5±0.4	64.6±0.5
BL1	63.8±1.3	62.0±0.7	65.9±0.2	64.9±0.3	65.3±0.5	61.3±0.8	64.4±0.6
CLO	59.7±0.7	63.8±0.3	65.9±0.2	64.9±0.3	65.3±0.5	61.3±0.8	64.6±0.5
PRI	63.8±1.3	62.0±0.7	65.9±0.2	64.9±0.3	65.3±0.5	61.3±0.8	64.4±0.6
SAN	63.8±1.3	62.0±0.7	65.9±0.2	64.1±1.2	65.3±0.5	61.3±0.8	64.4±0.6
DER2	64.8±0.6	67.2±0.5	62.5±0.4	60.8±0.8	65.3±0.5	61.3±0.8	64.4±0.6
MAO	71.0±0.8	67.8±0.5	61.4±1.4	62.2±1.0	60.5±0.3	60.5±0.4	64.6±0.5
UTA	70.5±0.6	73.1±0.9	66.3±0.6	73.1±1.0	61.8±1.1	65.5±1.0	66.7±0
LEE	67.5±0.5	67.8±0.6	67.7±0.7	68.3±0.7	67.7±0.6	66.9±0.2	66.7±0
BOW	64.9±0.5	66.0±0.5	66.3±0.5	64.5±0.7	65.3±0.5	60.5±0.4	64.6±0.5
YOU	69.5±0.5	70.9±1.1	73.6±1.0	70.2±0.6	62.1±0.7	60.5±0.4	64.6±0.5
BA 0.5	70.3±0.8	70.1±0.8	70.9±0.6	69.9±0.7	71.1±0.6	71.3±0.8	70±0.9
BA 0.6	70.3±0.8	70.1±0.8	70.9±0.6	69.9±0.7	71.1±0.6	71.3±0.8	70±0.9
BA 0.7	70.3±0.8	70.1±0.8	70.9±0.6	69.9±0.7	71.1±0.6	71.3±0.8	70±0.9
BA 0.8	70.3±0.4	70.1±0.8	70.9±0.6	69.9±0.7	71.1±0.6	71.3±0.8	70±0.9
BA 0.9	70.3±0.8	70.1±0.8	70.9±0.6	69.9±0.7	71.1±0.6	71.3±0.8	70±0.9
BA 1	70.3±0.8	70.1±0.8	70.9±0.6	69.9±0.7	71.1±0.6	71.3±0.8	70±0.9
CHI	67.5±0.6	66.6±0.3	67.0±0.6	66.9±0.3	63.3±1.2	60.1±0.9	67.1±0.5
SET	67.2±0.8	66.7±0	68.6±0.3	66.7±0	66.4±0.2	66.55±0.11	66.4±0.4
GD DS	----	----	----	----	----	----	----
RLF	70.7±0.6	68.7±0.7	66.2±1.0	68.2±0.9	67.4±0.5	66.7±0	66.7±0
SCH	66.2±0.3	66.7±0	66.9±0.2	66.3±0.2	65.6±0.3	66.3±0.2	66.7±0
FUZ	----	----	----	----	----	----	----

**Tabla 37. Eficiencias para los métodos ARNMF y ACE.  
Problema BU.**

**BASE DE DATOS BU**

Método	1	2	3	4	5	6
BL2	67±2	65±2	67.4±1.6	64±2	60.8±1.2	52.7±0.7
TEKA	67±2	69.5±0.2	67.4±1.6	64±2	60.8±1.2	55.5±0.5
CIB	69.1±0.3	65±2	62.6±1.2	59.5±1.5	60.8±1.2	55.5±0.5
TEKE	63.5±1.7	63.1±1.2	55.0±0.5	53.9±0.5	50.40±0.16	54±0
TEKC	63.5±1.7	54.9±0.5	54.5±0.5	50.6±0.4	55.3±0.7	53.6±0.2
DEV	67±2	69.5±0.2	67.4±1.6	64±2	60.8±1.2	52.7±0.7
DER3	63.5±1.7	52.3±1.2	53.8±1.0	53.5±1.3	50±0.9	52.7±0.7
TEKB	67±2	65±2	67.4±1.6	64±2	60.8±1.2	52.7±0.7
TEK	69.1±0.3	65±2	62.6±1.2	59.5±1.5	60.8±1.2	55.5±0.5
BL1	67±2	69.5±0.2	67.4±1.6	64±2	60.8±1.2	52.7±0.7
CLO	67±2	69.5±0.2	67.4±1.6	64±2	60.8±1.2	52.7±0.7
PRI	67±2	69.5±0.2	67.4±1.6	64±2	60.8±1.2	52.7±0.7
SAN	67±2	69.5±0.2	67.4±1.6	64±2	60.8±1.2	52.7±0.7
DER2	67±2	65±2	67.4±1.6	64±2	60.8±1.2	52.7±0.7
MAO	69.1±0.3	65±2	67.4±1.6	59.5±1.5	60.8±1.2	55.5±0.5
UTA	67±2	69.5±0.2	67.4±1.6	64±2	60.8±1.2	52.7±0.7
LEE	60.1±1.3	54.1±0.6	52.0±0.5	51.4±0.8	51.3±0.5	50.5±0.6
BOW	69.1±0.3	56.3±0.8	55.7±0.6	53.9±0.5	50.40±0.16	54±0
YOU	----	----	----	----	----	----
BA 0.5	67±2	60.4±1.3	60.4±1.5	53.2±0.6	53.1±0.9	54.3±0.4
BA 0.6	67±2	60.4±1.3	60.4±1.5	53.2±0.6	53.1±0.9	54.3±0.4
BA 0.7	67±2	60.4±1.3	60.4±1.5	53.2±0.6	53.1±0.9	54.3±0.4
BA 0.8	67±2	60.4±1.3	60.4±1.5	57.3±0.7	57.0±0.8	54.3±0.4
BA 0.9	67±2	60.4±1.3	60.4±1.5	57.3±0.7	53.80±0.13	54.3±0.4
BA 1	67±2	60.4±1.3	60.4±1.5	57.3±0.7	53.1±0.9	54.3±0.4
CHI	68.3±0.4	57.5±1.4	51.8±0.9	53.5±1.3	55.5±0.9	54.3±0.4
SET	69±2	70.0±0.7	66±2	60.6±1.3	53.1±0.9	54.3±0.4
GD_DS	68.3±0.4	69.5±0.2	66±2	64±2	62.5±0.8	57.6±0.8
RLF	69±2	70.0±0.7	64.3±0.6	64.3±0.8	54.9±0.4	57.6±0.8
SCH	69.1±0.3	68.7±0.5	67.1±1.5	57.9±1.4	54.1±0.3	55.5±0.5
FUZ	68.3±0.4	68.8±0.7	58.7±1.2	53.6±1.0	53.5±0.6	55.5±0.5

**Tabla 38. Eficiencias para los métodos ARNMF y ACE.  
Problema CR.**

**BASE DE DATOS CR (parte 1)**

Método	1	2	3	4	5	6	7	8
BL2	82±3	83±3	87.5±0.3	88.2±0.3	81±3	84±2	82.0±1.8	89.4±0.2
TEKA	82±3	83±3	87.5±0.3	88.2±0.3	81±3	88.80±0.13	89.3±0.2	88.3±0.4
CIB	83±3	86.4±1.7	83±3	87.70±0.17	82±2	82±3	87.2±0.3	77±3
TEKE	----	----	----	----	----	----	----	----
TEKC	83±3	86.4±1.7	82±2	81±2	78±3	83±2	85.4±0.3	82±3
DEV	82±3	83±3	87.5±0.3	88.2±0.3	82±3	84±3	82.0±1.8	85±2
DER3	83±3	88.0±0.3	85.8±1.6	79.2±0.2	78.7±0.5	77±3	77.5±0.3	76.8±0.2
TEKB	82±3	85±3	87.5±0.3	80±3	84.6±1.8	89.05±0.16	82.0±1.8	88.7±0.2
TEK	83±3	86.4±1.7	82±3	82±2	85.9±1.7	86.7±1.7	82±2	82±3
BL1	86.2±1.7	81±3	87.5±0.3	88.2±0.3	85±2	86.5±1.7	86.3±1.9	83±2
CLO	86.2±1.7	85±3	87.5±0.3	88.2±0.3	85±2	86.5±1.7	86.3±1.9	83±2
PRI	86.2±1.7	81±3	87.5±0.3	88.2±0.3	85±2	86.5±1.7	86.3±1.9	83±2
SAN	86.2±1.7	81±3	87.5±0.3	88.2±0.3	83±3	86.5±1.7	83±3	83±3
DER2	82±3	82±3	84.0±1.8	87.5±0.3	84±2	86.4±1.7	83±2	86.1±1.6
MAO	82±3	88.0±0.4	83±3	87.70±0.17	85.9±1.7	82±3	87.2±0.3	77±3
UTA	82±3	88.0±0.4	87.9±0.3	87.5±0.4	85±2	83±2	87.8±0.4	85±2
LEE	77±2	73.5±1.8	71±3	76.0±0.4	72±2	74±2	76.0±1.1	74±2
BOW	81±2	88.0±0.4	84±2	84±2	85.2±1.9	84±3	82±2	88.6±0.2
YOU	79±2	78±3	88.4±0.4	88.2±0.2	88.3±0.3	80±2	86.6±0.4	86.4±0.2
BA 0.5	83±2	88.7±0.3	81±2	87.4±1.4	85±2	88.5±0.2	83±2	83±3
BA 0.6	83±2	88.7±0.3	81±2	87.4±1.4	85±2	88.5±0.2	83±2	83±3
BA 0.7	83±2	88.7±0.3	81±2	87.4±1.4	85±2	88.5±0.2	83±2	83±3
BA 0.8	83±2	88.7±0.3	81±2	91±3	84±2	87.8±0.5	82±2	82±2
BA 0.9	83±2	88.7±0.3	81±2	81±3	84±2	86.6±1.7	82±2	82±3
BA 1	83±2	88.7±0.3	81±2	81±3	84±2	86.7±1.7	82±2	82±3
CHI	86.2±1.7	83±2	88.1±0.3	86.3±1.6	88.3±0.4	83±2	82±3	87.75±0.15
SET	86.2±1.7	83±2	88.8±0.5	88.7±0.2	86.7±1.7	86.7±1.7	84±2	88.55±0.05
GD_DS	----	----	----	----	----	----	----	----
RLF	83±3	86.4±1.7	82±3	83±2	81±3	82±2	85±2	85.6±1.5
SCH	77±2	73.4±1.8	72.7±1.8	75.1±0.7	68±2	62.3±1.0	62.6±1.5	64.7±1.3
FUZ	86.2±1.7	77±2	70±3	77.5±0.4	76.5±0.9	76.9±0.8	73.8±1.9	77.9±1.2

**Tabla 38. Eficiencias para los métodos ARNMF y ACE.  
Problema CR. (Continuación).**

**BASE DE DATOS CR (parte 2)**

Método	9	10	11	12	13	14	15
BL2	85±2	84±2	84±2	82±2	82±3	85.8±1.5	88±0
TEKA	85±2	84±2	84±2	82±2	82±3	85.8±1.5	88±0
CIB	84.7±0.2	86.2±0.4	83.1±1.8	86.9±0.4	81±3	76.5±0	72±3
TEKE	----	----	----	----	----	----	----
TEKC	76±2	74.9±0.4	76.45±0.05	74±2	76.5±0	72±2	55±0
DEV	83±3	84±2	84±2	82±2	82±3	85.8±1.5	88±0
DER3	74±2	71±3	75.5±0.2	72±3	75.7±0.2	74.65±0.11	55±0
TEKB	88.3±0.2	81±3	87.30±0.13	82±2	82±3	85±2	88±0
TEK	84.1±1.1	80±2	86.0±0.4	83±3	81±3	85.8±1.5	88±0
BL1	84±3	84±2	87.4±0.3	82±2	83±2	85±2	88±0
CLO	84±3	84±2	87.4±0.3	82±2	83±2	85±2	88±0
PRI	84±3	84±2	87.4±0.3	82±2	83±2	85±2	88±0
SAN	84±3	84±3	87.4±0.3	82±2	83±2	85±2	88±0
DER2	83±2	83±2	88.25±0.08	88±0	88±0	85±2	88±0
MAO	84.1±1.1	80±2	86.0±0.4	83±3	81±3	76.5±0	72±3
UTA	83±2	88.30±0.13	86.2±1.7	85±2	81±3	85.8±1.5	88±0
LEE	75.2±0.8	71±2	75±2	76.0±1.3	72±2	72±3	68.3±0.5
BOW	88.60±0.19	81±3	79±3	83±2	88±0	54.7±0.4	53.0±0.5
YOU	81±2	86.2±0.4	82±3	83±2	81±3	85.8±1.5	88±0
BA 0.5	84.5±1.9	86.0±1.6	83±2	85±2	88±0	84±2	88±0
BA 0.6	84.5±1.9	86.0±1.6	83±2	85±2	88±0	84±2	88±0
BA 0.7	84.5±1.9	86.0±1.6	83±2	85±2	88±0	84±2	88±0
BA 0.8	83±2	83±3	83±3	81±2	88±0	85±2	88±0
BA 0.9	83±3	78±3	86.9±1.5	87.5±0.3	87.40±0.16	81±3	88±0
BA 1	83±3	77±2	87.15±0.07	81±2	88±0	81±3	88±0
CHI	82±3	84±3	79±3	84±2	83±3	84±2	88±0
SET	84±2	86.8±1.7	83±2	85±2	88±0	84±2	88±0
GD_DS	----	----	----	----	----	----	----
RLF	88.35±0.10	85±2	88.05±0.05	80±2	83.7±1.4	85±2	88±0
SCH	61.0±1.2	61.0±1.4	59.6±1.3	58.4±0.8	55±2	56±2	55±0
FUZ	76±2	69.6±1.7	68.4±1.6	65±2	62.6±1.0	59.7±0.7	61.1±0.7

**Tabla 39. Eficiencias para los métodos ARNMF y ACE.  
Problema D1.**

**BASE DE DATOS D1**

Método	1	2	3	4	5	6	7
BL2	70.2±1.7	70.7±0.5	68.4±0.4	54.3±0.4	40.2±1.6	23.4±0.2	16.2±0.8
TEKA	70.2±1.7	70.7±0.5	68.4±0.4	54.3±0.4	40.2±1.6	23.4±0.2	16.2±0.8
CIB	70.2±1.7	68.4±1.4	55.86±0.15	44±3	40.2±1.6	23.4±0.2	16.2±0.8
TEKE	61.6±0.4	60.97±0.10	54.00±0.06	37±2	38±2	32.5±0.3	19.2±0.2
TEKC	59.8±0.2	56.0±1.9	52.9±0.2	45.4±0.2	44.39±0.04	32.05±0	16.3±0
DEV	70.2±1.7	70.7±0.5	68.4±0.4	54.3±0.4	40.2±1.6	23.4±0.2	16.2±0.8
DER3	65.4±0.2	65.2±1.0	55.4±0.9	46.33±0.12	32.17±0.11	24.97±0.08	17.64±0.07
TEKB	70.2±1.7	70.8±0.5	68.4±0.4	54.3±0.4	36.81±0.18	23.4±0.2	16.2±0.8
TEK	70.2±1.7	56.0±1.9	54±2	39.8±1.8	38.5±0.2	23.4±0.2	16.2±0.8
BL1	70.2±1.7	60±2	60.0±0.6	54.3±0.4	42.0±0.5	32.85±0	17.64±0.07
CLO	70.2±1.7	68.4±1.4	55.88±0.15	41.9±1.4	30.0±0.2	22.3±1.3	17.6±0
PRI	70.2±1.7	60±2	55.88±0.15	54.3±0.4	42.9±0.5	32.85±0	17.64±0.07
SAN	70.2±1.7	70.8±0.5	68.4±0.4	54.3±0.4	44.11±0.14	32.85±0	19.2±0.2
DER2	70.2±1.7	70.7±0.5	60.0±0.6	54.3±0.4	40.2±1.6	30.44±0.05	16.3±0
MAO	70.2±1.7	68.4±1.4	55.88±0.15	47.8±0.3	40.2±1.6	23.4±0.2	16.2±0.8
UTA	70.2±1.7	70.7±0.5	68.4±0.4	54.3±0.4	40.2±1.6	31.46±0.14	16.7±0.8
LEE	59.8±0.2	53.2±0.4	44±2	32±2	26.39±0.08	16.8±0.9	9.90±0.10
BOW	70.2±1.7	62.1±0.3	59.5±1.0	45.4±0.2	38±2	27.23±0.12	17.95±0
YOU	----	----	----	----	----	----	----
BA 0.5	70.2±1.7	70.8±0.5	60.0±0.6	51.02±0.14	42.9±0.5	30.45±0.05	17.64±0.07
BA 0.6	70.2±1.7	70.7±0.5	60.0±0.6	54.3±0.4	42.9±0.5	30.45±0.05	17.64±0.07
BA 0.7	70.2±1.7	70.7±0.5	60.0±0.6	54.3±0.4	42.9±0.5	30.45±0.05	17.64±0.07
BA 0.8	70.2±1.7	70.7±0.5	60.0±0.6	54.3±0.4	42.9±0.5	30.45±0.05	17.64±0.07
BA 0.9	70.2±1.7	70.7±0.5	60.0±0.6	54.3±0.4	42.9±0.5	30.45±0.05	17.64±0.07
BA 1	70.2±1.7	70.7±0.5	60.0±0.6	54.3±0.4	42.9±0.5	30.45±0.05	17.64±0.07
CHI	70.2±1.7	70.7±0.5	60.0±0.6	51.02±0.14	42.9±0.5	32.85±0	17.64±0.07
SET	70.2±1.7	70.7±0.5	60.0±0.6	47.9±1.3	40.2±1.6	30.45±0.05	16.3±0
GD_DS	69.9±1.4	70.7±0.5	60.0±0.6	51.02±0.14	37.90±0.02	25±0	17.64±0.07
RLF	----	----	----	----	----	----	----
SCH	65.4±0.2	53.1±0.2	43.6±1.6	39.21±0.03	25.73±0.05	16.8±0.9	17.64±0.07
FUZ	59±2	50.0±1.9	44.16±0.11	32.39±0.14	26.4±0.08	18.8±0.09	19±0

**Tabla 40. Eficiencias para los métodos ARNMF y ACE.  
Problema GL.**

**BASE DE DATOS GL (parte 1)**

Método	1	2	3	4	5	6
BL2	49±3	49±3	48±2	49±3	47±3	52±3
TEKA	44±3	49±3	48±3	49±3	47±3	52±3
CIB	44±3	49±3	36.1±1.4	55±3	47±3	55±3
TEKE	49±3	50±3	47±3	43±3	44±3	52±3
TEKC	36±3	52±3	36.1±1.4	46±2	42±3	48±2
DEV	49±3	49±3	51±3	51±3	47±3	52±3
DER3	40±2	37±3	36±2	39±2	38±2	41.9±0.7
TEKB	44±3	49±3	36.1±1.4	46±2	42±3	52±3
TEK	44±3	49±3	36.1±1.4	46±2	42±3	52±3
BL1	49±3	40±3	48±3	49±3	47±3	52±3
CLO	49±3	49±3	48±3	51±3	47±3	52±3
PRI	44±3	49±3	48±3	51±3	47±3	52±3
SAN	49±3	49±3	48±3	51±3	47±3	52±3
DER2	44±3	49±3	48±3	51±3	47±3	48±3
MAO	44±3	49±3	48±3	55±3	46±3	48±3
UTA	44±3	49±3	48±3	53±2	47±3	44±3
LEE	40±2	43±2	36±2	40.9±0.7	41.9±1.5	37.3±1.8
BOW	55±3	55±3	47±2	51±3	50±3	42±3
YOU	----	----	----	----	----	----
BA 0.5	49±3	47±2	46±3	56±3	53±3	43±3
BA 0.6	49±3	47±2	46±3	56±3	53±3	53±3
BA 0.7	49±3	47±2	46±3	56±3	53±3	44±3
BA 0.8	49±3	47±2	46±3	56±3	52±3	44±3
BA 0.9	59±3	47±2	46±3	56±3	52±3	44±3
BA 1	50±3	47±2	46±3	56±3	52±3	44±3
CHI	44±3	49±3	36.1±1.4	55±3	46±3	52±3
SET	49±3	50±3	47±3	50±3	47±3	52±3
GD_DS	49±3	33±2	46±2	34±3	40±3	38±2
RLF	----	----	----	----	----	----
SCH	40±2	43±2	36±2	40.9±0.7	41.9±1.5	42.5±1.3
FUZ	----	----	----	----	----	----



**Tabla 40. Eficiencias para los métodos ARNMF y ACE.  
Problema GL. (Continuación).**

**BASE DE DATOS GL (parte 2)**

Método	7	8	9	10
BL2	44±3	49±3	56±3	60±3
TEKA	44±3	49±3	56±3	60±3
CIB	42±3	49±3	54±3	49±3
TEKE	54±3	47±3	50±3	37.5±0
TEKC	45±3	37.0±1.9	37.7±1.1	37.2±1.5
DEV	42±3	49±3	49±3	60±3
DER3	41.9±0.9	43.6±1.3	31.6±1.0	32.5±1.5
TEKB	45±3	46±3	56±3	37.2±1.5
TEK	45±3	46±3	56±3	37.2±1.5
BL1	44±3	46±3	56±3	60±3
CLO	44±3	49±3	49±3	60±3
PRI	44±3	46±3	56±3	60±3
SAN	44±3	49±3	56±3	60±3
DER2	47±3	52±3	56±3	37.2±1.5
MAO	47±3	49±3	56±3	60±3
UTA	44±3	49±3	56±3	60±3
LEE	39.5±0.8	40.6±0.8	34.1±0.8	35.9±0.7
BOW	54±3	52±3	45±3	60±3
YOU	----	----	----	----
BA 0.5	47±3	43±2	45±3	60±3
BA 0.6	47±3	43±2	45±3	60±3
BA 0.7	47±3	43±2	45±3	60±3
BA 0.8	47±3	43±2	45±3	60±3
BA 0.9	47±3	43±2	45±3	60±3
BA 1	47±3	43±2	45±3	60±3
CHI	42±3	49±3	56±3	60±3
SET	44±3	46±3	45±3	35.6±1.6
GD_DS	41±2	39±2	38±2	37.2±1.5
RLF	----	----	----	----
SCH	37.2±1.8	36.1±1.2	35.9±1.0	35.9±0.7
FUZ	----	----	----	----

**Tabla 41. Eficiencias para los métodos ARNMF y ACE.  
Problema HE.**

**BASE DE DATOS HE (parte 1)**

Método	1	2	3	4	5	6	7
BL2	76±2	74±2	74±2	81.3±0.8	75±2	79.1±1.0	74±2
TEKA	76±2	74±2	75±2	81.3±0.8	75±2	79.1±1.0	74±2
CIB	76±2	74±3	79.7±0.6	76±3	76±2	79.3±1.1	72±3
TEKE	74±2	76±3	76.5±1.8	75±2	76±2	74±2	76.2±0.5
TEKC	77±3	75±2	75±2	76±2	76±2	78.5±0.6	72±2
DEV	76±2	74±2	74±2	81.3±0.8	75±2	79.1±1.0	74±2
DER3	74±2	74±2	72±2	74±2	73±2	70±3	72±2
TEKB	76±2	74±2	74±2	76±2	75±2	79.1±1.0	74±2
TEK	76±2	78±3	76±2	72±3	75±2	79.9±0.8	76±2
BL1	75±2	74±2	75±2	74.5±1.6	74.6±1.7	79.1±1.0	78±3
CLO	76±2	74±2	75±2	75.9±1.8	74.6±1.7	79.1±1.0	78±3
PRI	75±2	74±2	75±2	74.5±1.6	74.6±1.7	79.1±1.0	78±3
SAN	76±2	74±2	75±2	75.9±1.8	74.6±1.7	79.1±1.0	78±3
DER2	76±2	74±3	77.4±0.9	78±3	73±2	79.1±1.0	74±2
MAO	76±2	74±3	71±3	76±3	76±2	79.3±1.1	73±2
UTA	76±2	74±2	74±2	75±2	75±2	74.4±1.7	77±2
LEE	74±2	75±2	76.9±1.8	70.0±1.9	62±2	65.7±1.4	61.0±1.3
BOW	76±2	78±3	78±3	76±2	75±2	80±2	72±3
YOU	73±2	78±3	77±3	72±3	74±2	79.3±1.1	76±2
BA 0.5	75±2	70±3	75.6±0.7	72±2	69±3	77±3	74±2
BA 0.6	75±2	70±3	75.6±0.7	72±2	69±3	77±3	74±2
BA 0.7	75±2	70±3	75.6±0.7	72±2	69±3	77±3	74±2
BA 0.8	75±2	70±3	75.6±0.7	72±2	69±3	77±3	74±2
BA 0.9	75±2	70±3	75.6±0.7	72±2	69±3	77±3	74±2
BA 1	75±2	70±3	75.6±0.7	72±2	69±3	77±3	74±2
CHI	76±2	76±2	75±3	77±3	73±3	80±3	78±3
SET	76±2	78±3	77±3	72±2	69±3	77±3	75±2
GD_DS	76±2	76±2	80±3	77±3	73.8±1.9	76.4±0.6	76±2
RLF	72±2	74±3	77.4±0.9	75±2	69±3	71±3	74±2
SCH	75±2	73±2	74±2	69.8±1.8	62±2	65.2±1.5	65.2±1.0
FUZ	77±2	74±3	81±3	76±3	70±3	68±3	65±2

**Tabla 41. Eficiencias para los métodos ARNMF y ACE.  
Problema HE. (Continuación).**

**BASE DE DATOS HE (parte 2)**

Método	8	9	10	11	12	13
BL2	72±3	71±3	76±2	89.7±0.8	67±2	71±3
TEKA	72±3	71±3	76±2	79.7±0.8	67±2	68.5±1.6
CIB	69±3	74.1±0.9	71±3	71±3	65±2	73±2
TEKE	74±2	75.5±0.3	73.2±0.5	71±3	71±3	49.9±0.5
TEKC	76.8±1.8	74±2	70±3	66±3	67±2	58.7±0.4
DEV	71±2	71±3	76±2	79.7±0.8	67±2	71±3
DER3	72±2	74.8±0.5	69±3	66.1±1.9	70.7±0.3	66.4±1.4
TEKB	72±3	79.4±1.8	76±2	66±3	67.2±1.6	68.5±1.6
TEK	77±3	79.9±0.8	74.0±1.5	71±3	65±2	73±2
BL1	77±3	76±2	73±2	67±2	67.2±1.6	68.5±1.6
CLO	80±2	75±2	76±2	71±3	67.2±1.6	68.5±1.6
PRI	77±3	76±2	73±2	67±2	67.2±1.6	68.5±1.6
SAN	76±3	75±2	76±2	71±3	67.2±1.6	68.5±1.6
DER2	72±3	82.9±0.5	72±2	69±2	70.1±0.5	66.4±1.4
MAO	75±2	74.1±0.9	71±3	71±3	65±2	66±2
UTA	72±3	76±3	70±3	74±2	66±3	68.5±1.6
LEE	58.9±1.7	62.4±1.5	60.9±1.9	56.7±1.4	58.5±0.7	50.7±0.4
BOW	81.3±1.0	76.6±1.8	73.2±1.9	74±2	66±3	68.5±1.6
YOU	72±3	79.9±0.8	74.4±0.5	72±2	67.2±1.6	66±2
BA 0.5	71.3±1.9	76±2	74.8±1.6	72±3	70.3±0.2	61.14±0.15
BA 0.6	71.3±1.9	73.7±0.5	69±3	72±2	68.6±0.5	61.14±0.15
BA 0.7	71±2	73.7±0.5	69±3	72±2	68.6±0.5	61.14±0.15
BA 0.8	71±2	73.7±0.5	69.8±1.8	68.4±1.6	68.6±0.5	61.14±0.15
BA 0.9	71±2	73.7±0.5	69.8±1.8	68.4±1.6	68.6±0.5	61.14±0.15
BA 1	71±2	73.7±0.5	69.8±1.8	66.9±1.5	68.6±0.5	61.14±0.15
CHI	74±3	76±2	73±2	67±2	71.61±0.17	61.14±0.15
SET	74±2	77.8±1.9	79±2	79.6±0.8	67±2	66.4±1.4
GD_DS	74±2	78.3±0.5	78±3	76±2	67±2	71±3
RLF	72±2	76.6±1.8	73.2±1.9	68±2	66±3	57.7±0.6
SCH	57.9±0.9	61.0±1.0	60.6±1.2	53.9±0.6	55.1±0.4	51.7±0.7
FUZ	64.9±1.5	67.1±1.6	67.4±0.3	64.0±1.7	55.1±0.4	56.9±0.5

**Tabla 42. Eficiencias para los métodos ARNMF y ACE.  
Problema LE.**

**BASE DE DATOS LE (parte 1)**

Método	1	2	3	4	5	6	7	8
BL2	96.8±0.3	96.7±0.4	94.2±1.9	96.2±0.9	94.4±1.8	96.4±0.8	96.6±0.7	93±2
TEKA	96.8±0.3	96.7±0.4	94.2±1.9	96.2±0.9	94.4±1.8	96.4±0.8	94±2	93±2
CIB	96.8±0.3	96.7±0.4	94.2±1.9	96.2±0.9	97.8±0.3	96.4±0.8	94±2	97.3±0.3
TEKE	----	----	----	----	----	----	----	----
TEKC	96.8±0.3	96.6±0.5	91.8±1.8	93±2	94.7±0.4	91±2	94.72±0.15	93.8±0.5
DEV	96.8±0.3	96.7±0.4	94.2±1.9	96.2±0.9	94.4±1.8	96.4±0.8	94±2	93±2
DER3	97.7±0.3	95.8±0.8	96.2±0.5	96.0±0.5	93±3	96.4±0.3	93.7±0.3	93.81±0.19
TEKB	96.8±0.3	96.7±0.4	94.2±1.9	96.2±0.9	94.4±1.8	96.4±0.8	94±2	93±2
TEK	96.8±0.3	96.7±0.4	94.2±1.9	96.2±0.9	94.4±1.8	96.4±0.8	93.8±1.8	95.4±1.0
BL1	96.8±0.3	96.7±0.4	94.2±1.9	96.2±0.9	97.8±0.3	96.5±1.2	94±2	97.3±0.3
CLO	96.8±0.3	96.7±0.4	94.2±1.9	96.2±0.9	97.8±0.3	96.4±0.8	94±2	93±2
PRI	96.8±0.3	96.7±0.4	94.2±1.9	96.2±0.9	97.8±1.2	96.5±1.2	94±2	97.3±0.3
SAN	96.8±0.3	96.7±0.4	94.2±1.9	96.2±0.9	94.4±1.8	96.4±0.8	94±2	93±2
DER2	96.8±0.3	93±2	94.2±1.9	96.2±0.9	97.8±0.3	96.4±0.8	94±2	93±2
MAO	96.8±0.3	96.7±0.4	94.2±1.9	96.2±0.9	94.4±1.8	96.4±0.8	94±2	97.3±0.3
UTA	97.7±0.3	96.0±1.0	94.1±1.8	92±2	97.6±0.2	96.6±0.4	96±2	93±2
LEE	96.3±0.4	94.2±0.3	91±2	93.8±0.4	89.9±1.1	91.42±0.12	89.4±0.6	84.3±0.8
BOW	96.8±0.3	96.7±0.4	96.2±0.9	97.1±0.5	91±2	97.8±0.3	96.8±0.3	96.0±1.1
YOU	96.8±0.3	93±2	96.5±0.8	97.2±0.4	94.1±1.8	96.4±0.8	94±2	93±2
BA 0.5	97.8±0.2	95.8±1.8	95.0±1.6	95.0±1.8	96.8±0.4	96.3±0.5	95.7±0.6	96.3±0.6
BA 0.6	97.8±0.2	95.8±1.8	95.0±1.6	92.8±1.9	96.8±0.4	96.3±0.5	95.7±0.6	96.3±0.6
BA 0.7	97.8±0.2	95.8±1.8	95.0±1.6	92.8±1.9	96.8±0.4	96.3±0.5	95.7±0.6	96.3±0.6
BA 0.8	97.8±0.2	95.8±1.8	95.0±1.6	92.8±1.9	96.8±0.4	96.3±0.5	95.7±0.6	96.3±0.6
BA 0.9	97.8±0.2	95.8±1.8	95.0±1.6	92.8±1.9	96.8±0.4	96.3±0.5	95.7±0.6	96.3±0.6
BA 1	97.8±0.2	95.8±1.8	95.0±1.6	92.8±1.9	96.8±0.4	96.3±0.5	95.7±0.6	96.3±0.6
CHI	93.8±1.2	93±2	94.1±1.8	96.1±0.5	94.6±1.7	94.8±1.0	93±2	93.8±1.8
SET	93.8±1.2	94.8±1.8	96.3±0.3	96.5±0.6	96.3±1.1	96.4±0.7	93.0±1.8	95.7±0.9
GD_DS	----	----	----	----	----	----	----	----
RLF	93.8±1.2	93±2	94.1±1.8	93±2	97.5±0.3	96±2	97.1±0.5	96.1±1.0
SCH	96.3±0.4	97.5±0.3	92.2±1.8	90±2	89.1±1.7	89.3±0.6	85.7±1.9	89.6±0.7
FUZ	----	----	----	----	----	----	----	----

**Tabla 42. Eficiencias para los métodos ARNMF y ACE.  
Problema LE. (Continuación).**

**BASE DE DATOS LE (parte 2)**

Método	9	10	11	12	13	14	15
BL2	96.6±0.9	97.1±0.6	97.0±1.2	96.5±0.6	95.0±0.6	92.3±1.7	91±2
TEKA	96.6±0.9	96.5±1.2	97.0±1.2	96.5±0.6	93.5±1.6	95.5±1.9	94.7±0.8
CIB	94±3	94.8±1.8	96.8±0.6	96.1±0.5	95.5±1.8	94.9±1.8	96.3±0.8
TEKE	----	----	----	----	----	----	----
TEKC	90±2	90±2	93.9±0.2	92.7±1.2	90.4±0.9	87.3±1.9	91.7±0.5
DEV	92±2	97.4±0.3	97.9±0.4	97.0±0.8	96.4±0.9	96.8±0.6	92.5±1.9
DER3	90.9±1.5	88±2	89.7±0.5	86.9±1.7	87.8±0.5	87.2±0.4	89.1±0.6
TEKB	94±3	96.5±1.2	97.0±0.6	95.8±1.6	93.5±1.6	97.2±0.6	95.9±0.7
TEK	96.5±0.4	93.8±1.4	97.1±0.3	93.2±0.9	93.7±1.1	94.2±1.1	94.7±0.7
BL1	94±3	96.5±1.2	94.8±1.8	95.8±1.6	93.5±1.6	95.5±1.9	97.1±0.5
CLO	96.6±0.9	96.5±1.2	94.8±1.8	95.8±1.6	93.5±1.6	95.1±1.8	95.9±0.7
PRI	94±3	96.5±1.2	94.8±1.8	95.8±1.6	93.5±1.6	95.5±1.9	97.1±0.5
SAN	96.6±0.9	96.5±1.2	94.8±1.8	95.8±1.6	93.5±1.6	95.1±1.8	95.9±0.7
DER2	94±3	96.5±1.2	96.3±0.7	94±2	96.7±0.6	95.5±1.9	96.6±0.3
MAO	94±3	94.4±1.8	96.8±0.5	96.1±0.5	95.5±1.8	94.9±1.8	96.3±0.8
UTA	98.32±0.16	95±2	96.1±1.0	95.8±1.6	93.5±1.6	97.2±0.6	92.5±1.9
LEE	82.8±0.4	80±2	79.4±1.9	80.8±0.2	74.6±1.6	76.1±1.3	77.94±0.05
BOW	96.8±0.4	97.2±0.4	95.9±1.8	94.4±1.9	94±2	91±2	97.0±0.4
YOU	96.8±0.6	95±2	97.2±0.6	96.1±0.6	94.3±1.2	95.2±0.9	96.3±0.4
BA 0.5	92.8±1.9	94.7±1.8	94.3±1.6	92±3	93.4±1.8	94.34±0.12	94.3±0.6
BA 0.6	92.8±1.9	94.7±1.8	94.3±1.6	92±3	89±2	92.5±0.6	91.9±0.2
BA 0.7	92.8±1.9	95.8±0.9	93±3	92.4±1.8	95.2±0.7	90.9±1.7	86±2
BA 0.8	92.8±1.9	95.8±0.9	93±3	92.4±1.8	95.2±0.7	90.9±1.7	84±2
BA 0.9	96.1±0.7	95.8±0.9	93±3	92±2	79±3	81±2	84±2
BA 1	96.1±0.7	95.8±0.9	93±3	92±2	79±3	81±2	84±2
CHI	96.3±0.9	91.6±1.8	91.2±1.8	93.1±0.8	93.3±0.7	92.8±0.5	92.2±0.4
SET	90±3	95.7±0.5	95.9±0.4	93.1±0.8	94.0±1.0	94.5±0.4	92.2±0.4
GD_DS	----	----	----	----	----	----	----
RLF	96.3±1.1	95.8±1.8	93.9±1.7	96.5±1.1	95.3±1.8	96.0±0.7	94.6±1.8
SCH	88±3	91.7±1.6	86±3	83.0±1.5	87±2	85±2	87±2
FUZ	----	----	----	----	----	----	----

**Tabla 42. Eficiencias para los métodos ARNMF y ACE.  
Problema LE. (Continuación).**

**BASE DE DATOS LE (parte 3)**

Método	16	17	18	19	20	21	22
BL2	94.1±0.5	91.7±0.7	89±2	93.4±0.2	91.45±0.17	86.67±0	78.07±0
TEKA	95.2±0.6	92.6±1.6	91.4±0.8	93.4±0.2	91.45±0.17	86.67±0	78.07±0
CIB	94.8±0.9	97.2±0.5	90.97±0.14	88±3	89.9±1.3	78.07±0	78.07±0
TEKE	----	----	----	----	----	----	----
TEKC	89.0±0.4	87±0	86.94±0.06	86.5±0.4	86.8±0.2	87±0	73.2±0
DEV	96.2±0.6	93±2	91.4±0.8	93.4±0.2	88.5±1.4	82.7±1.6	78.07±0
DER3	87.8±0.6	84±2	87.11±0.11	87.66±0.18	79.1±1.3	78.4±0.7	75±2
TEKB	94±2	93±2	91.4±0.8	91.9±1.5	87±2	82.7±1.6	62.27±0
TEK	94.8±0.9	94.6±0.7	91.9±1.6	91.293±0.011	89.9±1.3	78.07±0	66.93±0
BL1	96.0±1.6	94±2	97.0±0.3	91.9±1.5	87±2	87±3	62.27±0
CLO	96.0±1.6	94±2	97.0±0.3	91.9±1.5	87±2	87±3	62.27±0
PRI	96.0±1.6	94±2	97.0±0.3	91.9±1.5	87±2	87±3	62.27±0
SAN	94±2	93±2	91.4±0.8	91.9±1.5	87±2	82.7±1.6	62.27±0
DER2	94.9±1.7	95.9±0.5	88±3	89.2±1.7	91.73±0	78.07±0	78.07±0
MAO	97.3±0.2	92±2	90.97±0.14	88±3	89.9±1.3	78.07±0	78.07±0
UTA	94±2	94.5±1.8	93.3±1.8	93.6±0.2	91.73±0	78.07±0	78.07±0
LEE	63.1±0.3	63.1±0.3	62.87±0.16	63.0±0.3	63.6±0.3	62.31±0.04	62.27±0
BOW	95.7±1.6	94.5±1.8	97.0±0.3	91.9±1.5	87±2	87±3	62.27±0
YOU	94.6±0.6	91±2	94.6±0.4	91.293±0.011	89.9±1.3	82.3±1.9	66.93±0
BA 0.5	92.5±1.4	89.8±0.3	87±2	87±3	90.3±0.4	91.3±0.6	77±3
BA 0.6	89±3	89.8±0.3	87±2	87±3	90.3±0.4	91.3±0.6	77±3
BA 0.7	86±2	88±2	72±3	73±3	84±3	80±3	77±3
BA 0.8	86±2	88±2	72±3	73±3	84±3	80±3	77±3
BA 0.9	86±2	88±2	72±3	73±3	84±3	80±3	77±3
BA 1	86±2	88±2	72±3	73±3	83±3	80±3	77±3
CHI	91.4±0.3	89.53±0	87±3	90.1±0.4	89.99±0.19	91.3±0.6	77±3
SET	91.4±0.3	94.2±0.7	87±3	90.1±0.4	86.1±1.0	78±2	77±3
GD_DS	----	----	----	----	----	----	----
RLF	95.7±1.1	95.8±0.4	94.8±1.0	93.9±0.8	90.6±1.2	87±2	62.27±0
SCH	87±3	72.4±1.6	71±2	71±2	72.4±1.6	62.27±0	62.27±0
FUZ	----	----	----	----	----	----	----

**Tabla 43. Eficiencias para los métodos ARNMF y ACE.  
Problema M1.**

**BASE DE DATOS M1**

Método	1	2	3	4	5	6
BL2	68±2	69±2	68.1±1.9	70±2	74±3	71±2
TEKA	68±2	69±2	71±2	70±3	69±3	71±2
CIB	68±2	69±2	71±2	76±3	69±3	71±2
TEKE	69±2	68±3	77.1±10	69±2	69±3	71±2
TEKC	69±2	75.0±1.0	70±2	39.7±0.5	40.4±0.7	40.4±1.6
DEV	68±2	72±3	71±2	76±3	69±3	71±2
DER3	42±2	62±3	44.9±0.9	45.6±1.5	45.0±1.0	45.6±1.5
TEKB	68±2	69±2	72±3	76±3	69±3	71±2
TEK	68±2	69±2	72±3	76±3	69±3	71±2
BL1	68±2	69±2	68.1±1.9	70±2	74±3	71±2
CLO	68±2	69±2	72±3	76±3	69±3	71±2
PRI	68±2	69±2	68.1±1.9	70±2	74±3	71±2
SAN	68±2	69±2	71±2	70±3	69±3	71±2
DER2	68±2	69±2	72±3	67±3	74±3	71±2
MAO	71±2	69±2	71±2	76±3	70±3	71±2
UTA	71±2	69±2	72±3	76±3	69±3	71±2
LEE	43±2	43±2	40.9±1.0	45±2	45.0±1.0	45.6±1.5
BOW	68±2	69±2	72±3	76±3	63±3	48.0±0.3
YOU	68±2	69±2	72±3	69±2	69±3	71±2
BA 0.5	83±2	73.9±0.9	69±2	71±2	69±3	71±2
BA 0.6	83±2	73.9±0.9	69±2	71±2	69±3	71±2
BA 0.7	83±2	73.9±0.9	69±2	71±2	69±3	71±2
BA 0.8	83±2	73.9±0.9	69±2	71±2	69±3	71±2
BA 0.9	83±2	73.9±0.9	69±2	71±2	69±3	71±2
BA 1	83±2	73.9±0.9	69±2	71±2	69±3	71±2
CHI	68±2	70±2	70±2	70±2	74±3	71±2
SET	68±2	70±2	70±2	70±2	74±3	71±2
GD_DS	83±2	67±3	69±2	73±3	76±2	71±2
RLF	70±3	70±2	72±3	61±3	63±3	46±0
SCH	43±2	42.2±1.2	44.5±0.8	44.7±0.9	43.4±0.6	45.6±1.5
FUZ	69±3	75.0±1.0	70±2	72±3	74±3	71±2

**Tabla 44. Eficiencias para los métodos ARNMF y ACE.  
Problema M2.**

**BASE DE DATOS M2**

<b>Método</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>BL2</b>	69.2±1.2	67.4±0.4	66.3±0.6	66.9±0.4	68±0	68±0
<b>TEKA</b>	69.2±1.2	67.4±0.4	66.3±0.6	68±0	68±0	68±0
<b>CIB</b>	69.2±1.2	67.4±0.4	66.3±1.0	68±0	68±0	68±0
<b>TEKE</b>	69.2±1.2	70.6±0.8	67.9±0.7	67.90±0.10	68±0	68±0
<b>TEKC</b>	69.1±0.7	67.6±0.4	68±0	68±0	68±0	68±0
<b>DEV</b>	69.2±1.2	67.4±0.4	66.3±1.0	67.7±0.3	68±0	68±0
<b>DER3</b>	68.0±0.3	68.10±0.18	68±0	68±0	68±0	68±0
<b>TEKB</b>	69.2±1.2	67.4±0.4	66.3±0.6	66.9±0.4	68±0	68±0
<b>TEK</b>	69.2±1.2	67.4±0.4	65.9±0.7	68±0	68±0	68±0
<b>BL1</b>	69.2±1.2	67.4±0.4	66.3±0.6	66.9±0.4	68±0	68±0
<b>CLO</b>	69.2±1.2	67.4±0.4	66.3±1.0	66.9±0.4	68±0	68±0
<b>PRI</b>	69.2±1.2	67.4±0.4	66.3±0.6	66.9±0.4	68±0	68±0
<b>SAN</b>	69.2±1.2	67.4±0.4	66.3±0.6	66.9±0.4	68±0	68±0
<b>DER2</b>	69.1±0.7	67.6±0.4	66.0±0.8	68±0	68±0	68±0
<b>MAO</b>	69.2±1.2	67.4±0.4	66.3±1.0	68±0	68±0	68±0
<b>UTA</b>	69.2±1.2	67.4±0.4	66.3±1.0	68±0	68±0	68±0
<b>LEE</b>	68.1±0.2	68.10±0.18	68±0	68±0	68±0	68±0
<b>BOW</b>	69.2±1.2	68.5±0.4	65.9±0.7	68±0	68±0	68±0
<b>YOU</b>	69.1±0.7	68.6±0.4	68.10±0.18	67.7±0.3	68±0	68±0
<b>BA 0.5</b>	69.1±0.7	68.6±0.4	67.4±0.4	68±0	68±0	68±0
<b>BA 0.6</b>	69.1±0.7	68.6±0.4	67.4±0.4	68±0	68±0	68±0
<b>BA 0.7</b>	69.1±0.7	68.6±0.4	67.4±0.4	68±0	68±0	68±0
<b>BA 0.8</b>	69.1±0.7	68.6±0.4	67.4±0.4	68±0	68±0	68±0
<b>BA 0.9</b>	69.1±0.7	68.6±0.4	67.4±0.4	68±0	68±0	68±0
<b>BA 1</b>	69.1±0.7	68.6±0.4	67.4±0.4	68±0	68±0	68±0
<b>CHI</b>	68.0±0.3	68.10±0.18	68±0	68±0	68±0	68±0
<b>SET</b>	68.0±0.3	68.10±0.18	68±0	68±0	68±0	68±0
<b>GD_DS</b>	69.2±1.2	67.4±0.4	65.9±0.7	67.6±0.4	68±0	68±0
<b>RLF</b>	69.2±1.2	67.0±0.5	66.3±0.6	68±0	68±0	68±0
<b>SCH</b>	68.1±0.2	68.10±0.18	68±0	68±0	68±0	68±0
<b>FUZ</b>	69.7±0.4	66.1±0.7	68±0	68±0	68±0	68±0



**Tabla 45. Eficiencias para los métodos ARNMF y ACE.  
Problema M3.**

**BASE DE DATOS M3**

<b>Método</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>BL2</b>	79±2	75±3	80±2	84±2	89±2	79±2
<b>TEKA</b>	79±2	75±3	80±2	84±2	89±2	79±2
<b>CIB</b>	79±2	75±3	80±2	90.2±1.2	89±2	79±2
<b>TEKE</b>	84±3	83±2	84±3	90.2±1.2	89±2	79±2
<b>TEKC</b>	84±3	79±2	74±3	75±2	84±2	79±2
<b>DEV</b>	81±3	75±3	80±2	84±2	89±2	79±2
<b>DER3</b>	81±3	78±3	48.1±1.0	48.7±1.5	43.6±1.2	49.5±1.6
<b>TEKB</b>	78±3	77±3	80±2	80±3	89±2	79±2
<b>TEK</b>	78±3	77±3	80±2	80±3	89±2	64±3
<b>BL1</b>	79±2	75±3	85±3	90.2±1.2	89±2	64±3
<b>CLO</b>	79±2	75±3	80±2	90.2±1.2	89±2	64±3
<b>PRI</b>	79±2	75±3	85±3	90.2±1.2	89±2	64±3
<b>SAN</b>	79±2	75±3	80±2	90.2±1.2	89.3±2	64±3
<b>DER2</b>	81±3	78±3	80±2	84±2	89±2	64±3
<b>MAO</b>	79±2	75±3	80±2	90.2±1.2	89±2	79±2
<b>UTA</b>	79±2	89.0±1.9	80±3	81±2	89±2	79±2
<b>LEE</b>	77±3	48.6±0.9	47.9±1.6	52.3±0.7	51.7±0.5	51.6±0.4
<b>BOW</b>	79±2	75±3	80±2	84±2	89±2	79±2
<b>YOU</b>	----	----	----	----	----	----
<b>BA 0.5</b>	78±3	77±3	81±3	84±2	89±2	64±3
<b>BA 0.6</b>	78±3	77±3	81±3	84±2	89±2	64±3
<b>BA 0.7</b>	78±3	77±3	81±3	84±2	89±2	64±3
<b>BA 0.8</b>	78±3	77±3	81±3	84±2	89±2	64±3
<b>BA 0.9</b>	78±3	77±3	81±3	84±2	89±2	64±3
<b>BA 1</b>	78±3	77±3	81±3	84±2	89±2	64±3
<b>CHI</b>	91.90±0.10	75±3	80±2	90.2±1.2	89±2	64±3
<b>SET</b>	91.90±0.10	75±3	80±2	90.2±1.2	89±2	64±3
<b>GD_DS</b>	78±3	81±2	79±3	81±2	89±2	64±3
<b>RLF</b>	84±3	80±3	85±3	90.2±1.2	89±2	79±2
<b>SCH</b>	68±2	48.6±0.9	47.9±1.6	54±0	54±0	54±0
<b>FUZ</b>	81±3	68.4±1.5	65.3±1.9	68.7±1.1	64±2	64±2

**Tabla 46. Eficiencias para los métodos ARNMF y ACE.  
Problema PI.**

**BASE DE DATOS PI (parte 1)**

Método	1	2	3	4
BL2	74.8±1.1	76.1±0.2	76.76±0.15	76.16±0.17
TEKA	74.8±1.1	76.1±0.2	76.76±0.15	76.16±0.17
CIB	75.24±0.17	76.1±0.2	76.16±0.18	76.16±0.17
TEKE	66.3±0.6	65.3±0.7	70.8±0.4	70.8±0.9
TEKC	75.24±0.17	75.1±0.3	74.8±0.4	75.5±0.3
DEV	74.8±1.1	76.1±0.2	76.16±0.18	76.16±0.17
DER3	76.3±0.3	75.44±0.14	75.8±0.2	64.8±0.5
TEKB	75.24±0.17	76.1±0.2	76.76±0.15	76.16±0.17
TEK	75.24±0.17	76.1±0.2	76.16±0.18	73.6±1.4
BL1	75.5±0.3	75.72±0.18	76.76±0.15	77.0±0.2
CLO	75.5±0.3	75.72±0.18	76.76±0.15	77.0±0.2
PRI	75.5±0.3	75.72±0.18	76.76±0.15	77.0±0.2
SAN	75.5±0.3	75.72±0.18	76.76±0.15	76.16±0.17
DER2	75.5±0.3	75.72±0.17	74.4±1.1	77.0±0.2
MAO	75.24±0.17	76.1±0.2	76.16±0.18	73.6±1.4
UTA	75.24±0.17	75.72±0.18	76.76±0.15	76.16±0.17
LEE	66.3±0.6	69.7±0.7	66.9±0.5	67.9±0.7
BOW	75.24±0.17	76.04±0.13	73.7±1.4	76.52±0.12
YOU	75.24±0.17	76.1±0.2	76.16±0.18	76.52±0.12
BA 0.5	75.5±0.3	75.60±0.08	75.5±1.2	77.56±0.15
BA 0.6	75.5±0.3	75.60±0.08	75.5±1.2	77.56±0.15
BA 0.7	75.5±0.3	75.60±0.08	75.5±1.2	77.56±0.15
BA 0.8	75.5±0.3	75.60±0.08	75.5±1.2	77.56±0.15
BA 0.9	75.5±0.3	75.60±0.08	75.5±1.2	77.56±0.15
BA 1	75.5±0.3	75.60±0.08	75.5±1.2	77.56±0.15
CHI	75.6±0.2	74.4±1.0	75.20±0.13	76.40±0.17
SET	75.5±0.3	74.4±1.0	75.20±0.13	74.6±1.1
GD_DS	75.5±0.3	74.4±1.0	75.20±0.13	78.8±0.3
RLF	75.24±0.17	75.0±0.4	76.16±0.18	73.6±1.4
SCH	66.3±0.6	65.7±0.6	65.2±0.4	64.7±0.5
FUZ	75.4±0.3	75.44±0.14	74.92±0.13	74.5±0.2

**Tabla 46. Eficiencias para los métodos ARNMF y ACE.  
Problema PI. (Continuación).**

**BASE DE DATOS PI (parte 2)**

<b>Método</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>BL2</b>	77.20±0.18	77.0±0.3	75.76±0.14	74.4±0
<b>TEKA</b>	77.20±0.18	77.0±0.3	75.76±0.14	74.4±0
<b>CIB</b>	75.7±0.4	75.52±0.10	75.76±0.14	74.4±0
<b>TEKE</b>	69.1±0.8	65.9±0.2	66.08±0.13	65.2±0
<b>TEKC</b>	74.6±0.3	73.7±1.0	65.00±0.09	65.2±0
<b>DEV</b>	77.20±0.18	77.0±0.3	75.76±0.14	63.3±0.3
<b>DER3</b>	60.40±0.17	65.9±0.2	65.00±0.09	65.60±0.13
<b>TEKB</b>	77.20±0.18	77.0±0.3	75.76±0.14	63.3±0.3
<b>TEK</b>	75.1±0.3	75.52±0.10	75.76±0.14	74.4±0
<b>BL1</b>	76.56±0.09	75.52±0.10	75.76±0.14	63.3±0.3
<b>CLO</b>	76.56±0.09	75.52±0.10	75.76±0.14	63.3±0.3
<b>PRI</b>	76.56±0.09	75.52±0.10	75.76±0.14	63.3±0.3
<b>SAN</b>	76.56±0.09	75.52±0.10	75.76±0.14	63.3±0.3
<b>DER2</b>	76.56±0.09	75.52±0.10	75.76±0.14	63.3±0.3
<b>MAO</b>	75.7±0.4	75.52±0.10	75.76±0.14	74.4±0
<b>UTA</b>	76.56±0.09	77.32±0.18	75.76±0.14	74.4±0
<b>LEE</b>	67.8±0.5	66.9±0.3	64.44±0.13	65.2±0
<b>BOW</b>	75.1±0.3	73.4±1.3	75.76±0.14	74.4±0
<b>YOU</b>	75.1±0.3	75.52±0.10	75.76±0.14	74.4±0
<b>BA 0.5</b>	76.2±1.2	77.32±0.18	75.76±0.14	74.4±0
<b>BA 0.6</b>	76.2±1.2	77.32±0.18	75.76±0.14	74.4±0
<b>BA 0.7</b>	76.2±1.2	77.32±0.18	75.76±0.14	74.4±0
<b>BA 0.8</b>	76.2±1.2	77.32±0.18	75.76±0.14	74.4±0
<b>BA 0.9</b>	76.2±1.2	77.32±0.18	75.76±0.14	74.4±0
<b>BA 1</b>	76.2±1.2	77.32±0.18	75.76±0.14	74.4±0
<b>CHI</b>	76.2±0.3	77.0±0.3	75.76±0.14	74.4±0
<b>SET</b>	76.2±0.3	77.0±0.3	75.76±0.14	74.4±0
<b>GD_DS</b>	68.3±0.6	69.3±0.6	66.7±0.2	65.56±0.11
<b>RLF</b>	75.1±0.3	73.4±1.3	75.76±0.13	74.4±0
<b>SCH</b>	63.1±0.5	67.56±0.17	67.6±0.4	65.2±0
<b>FUZ</b>	75.7±0.4	74.4±1.0	67.6±0.4	65.40±0.17

**Tabla 47. Eficiencias para los métodos ARNMF y ACE.  
Problema VO.**

**BASE DE DATOS VO (parte 1)**

Método	1	2	3	4	5	6	7	8
BL2	94.4±0.3	91±2	94.8±0.2	92±2	95.9±0.2	89±3	94.20±0.13	95.2±0.2
TEKA	94.4±0.3	94.2±0.3	92±2	92±2	95.9±0.2	89±3	94.20±0.13	94.3±0.2
CIB	94.4±0.3	91±2	94.8±0.2	92±2	95.0±0.3	91±2	94.7±0.3	95.3±0.2
TEKE	93.7±0.2	88±3	92.7±0.4	90±3	92.6±0.3	87.8±1.8	91.1±0.3	88.1±1.6
TEKC	93±2	94.5±0.3	94.0±0.2	91±2	88.3±1.9	85±2	89.0±0.5	87.9±0.5
DEV	94.2±0.4	90±3	94.8±0.2	92±2	95.9±0.2	89±3	93.2±0.4	93.9±0.4
DER3	93.7±0.2	91±3	90±2	92.3±1.5	87±3	88.7±0.3	87.4±0.3	86±3
TEKB	94.9±0.3	92±2	92±2	92±2	93.8±1.6	89±3	94.90±0.18	95.0±0.3
TEK	94.9±0.3	91±2	92±2	91±2	92.9±1.6	92.8±0.3	90±2	94.4±0.2
BL1	94.4±0.3	93.2±1.6	93.3±1.6	91.6±1.8	93.8±1.6	95.0±0.3	95.0±0.4	93.1±1.6
CLO	94.4±0.3	91±2	93.3±1.6	93.8±0.3	92.8±1.6	90±2	94.7±0.4	93.8±0.2
PRI	94.4±0.3	93.2±1.6	93.3±1.6	91.6±1.8	93.8±1.6	95.0±0.3	95.0±0.4	93.1±1.6
SAN	94.4±0.3	91±2	92±2	92±2	92±2	95.0±0.3	95.0±0.4	93.1±1.6
DER2	94.4±0.3	91±2	93.3±1.6	95.1±0.2	92±2	95.0±0.3	93.0±1.6	92.5±1.6
MAO	94.9±0.3	91±2	94.8±0.2	94.9±0.4	94.8±0.3	94.9±0.2	94.7±0.3	94.6±0.2
UTA	94.4±0.3	91±2	94.8±0.2	95.1±0.2	94.6±0.4	94.0±1.5	94.6±0.3	93.1±1.6
LEE	91.3±0.6	90±3	92.4±0.5	91.6±0.3	89±3	92.1±0.4	90.4±0.7	88.7±0.4
BOW	91±2	94.6±0.3	91±3	91±2	94.60±0.16	94.1±0.3	94.60±0.16	91±2
YOU	94.2±0.4	88.9±1.9	94.5±0.5	95.3±0.2	92±2	92.6±1.7	95.2±0.2	90±3
BA 0.5	93.0±1.6	94.5±0.3	91±2	92±2	94.5±0.4	92±2	91.2±1.8	92.4±1.6
BA 0.6	92.8±1.6	94.5±0.3	91±2	92±2	90±2	93.5±0.3	90±3	92.0±1.5
BA 0.7	92.8±1.6	94.5±0.3	91±2	92±2	90±2	93.2±0.2	90±3	92.0±1.5
BA 0.8	92.8±1.6	94.5±0.3	91±2	92±2	90±2	93.2±0.2	90±3	92.8±0.3
BA 0.9	92.8±1.6	94.5±0.3	94.4±0.3	92±2	90±2	93.2±0.2	90±3	92.8±0.3
BA 1	92.8±1.6	94.5±0.3	94.4±0.3	92±2	90±2	89±3	93.3±0.3	93.4±0.3
CHI	91±2	91±2	91±2	90±2	93.4±0.5	90±2	90±3	94.2±0.3
SET	91±2	91±2	91±2	90±2	93.4±0.5	90±2	90±3	94.2±0.3
GD_DS	94.4±0.3	95.4±0.4	91±2	90±2	93.4±0.5	90±2	94.7±0.3	94.2±0.3
RLF	94.4±0.3	94.1±0.3	94.2±0.2	92.8±1.5	92±2	93.0±1.6	92±2	93.8±1.6
SCH	91.3±0.6	90±3	91.0±0.8	89.4±0.4	84±2	80±2	84.5±0.5	87.8±0.6
FUZ	----	----	----	----	----	----	----	----

**Tabla 47. Eficiencias para los métodos ARNMF y ACE.  
Problema VO. (Continuación).**

**BASE DE DATOS VO (parte 2)**

Método	9	10	11	12	13	14	15	16
BL2	89.4±1.3	91.8±0.3	88±3	88±2	92.80±0.13	93.00±0.15	87±2	88.8±1.7
TEKA	91±2	89.0±1.9	93.1±1.6	91±2	92.80±0.13	92.90±0.18	91.5±1.5	88.8±1.7
CIB	95.2±0.2	95.50±0.17	94.3±0.7	92.0±1.4	92.5±1.6	90.1±1.7	92.20±0.13	88.8±1.7
TEKE	86±3	91.8±0.8	89±3	85.5±0.3	72.0±1.0	72.0±1.0	72.60±0.16	74±0
TEKC	83±2	81±2	81.1±0.3	79±3	83.30±0.15	72.2±0.2	68±0	68±0
DEV	92.1±1.5	93.7±0.3	87±2	89±3	88±3	90.9±0.4	87±2	88.8±1.7
DER3	88.1±0.2	87.8±0.4	86.8±0.3	87±0	87±0	82.1±0.4	71±0	71±0
TEKB	89±2	88±2	91.8±0.5	94.10±0.10	92.80±0.13	92.90±0.18	87±2	88.8±1.7
TEK	94.6±0.2	94.1±0.4	91±2	88±3	88.1±1.9	90.1±1.7	87±2	88.8±1.7
BL1	91±3	91.4±1.8	91.6±1.6	91.4±1.8	92±2	94±0	93±0	88.8±1.7
CLO	91±3	91.4±1.8	91.6±1.6	91.4±1.8	92±2	94±0	93±0	88.8±1.7
PRI	91±3	91.4±1.8	91.6±1.6	91.4±1.8	92±2	94±0	93±0	88.8±1.7
SAN	94.5±0.5	91.4±1.8	91.6±1.6	91±2	92±2	94±0	87±2	88.8±1.7
DER2	94.5±0.5	91.4±1.8	95.40±0.16	91.4±1.8	92±2	92.90±0.18	87±2	88.8±1.7
MAO	92±2	95.50±0.17	95.50±0.17	92.0±1.4	87±3	90.1±1.7	92.20±0.13	88.8±1.7
UTA	94.5±0.5	91.4±1.8	95.6±0.5	96±0	92±2	92.90±0.18	87±2	88.8±1.7
LEE	86.5±0.5	83±2	83.0±1.6	81±2	80±2	64.0±0.5	65.8±0.2	66±0
BOW	94.90±0.18	94.8±0.2	94.0±1.6	94.10±0.10	92±2	94±0	93±0	88.8±1.7
YOU	91±3	95.50±0.17	92±3	93.5±1.6	92±2	92.90±0.18	92.3±0.5	88.8±1.7
BA 0.5	94.40±0.16	93.3±0.9	92.6±1.6	92.5±1.6	92.5±1.6	93.00±0.15	93±0	88.8±1.7
BA 0.6	89±2	88±2	90±3	90±2	92.3±1.5	93.10±0.17	93±0	88.8±1.7
BA 0.7	89±2	88±2	90±3	88±2	89±2	93.10±0.18	93±0	88.8±1.7
BA 0.8	89±2	88±2	90±3	88±2	89±2	93.10±0.18	93±0	88.8±1.7
BA 0.9	89±2	88±2	90±3	88±2	89±2	93.10±0.18	93±0	88.8±1.7
BA 1	89±2	93.4±0.2	93.3±0.2	94.1±0.4	92.3±1.5	94.10±0.10	93±0	88.8±1.7
CHI	92.6±0.2	90.5±0.8	89±3	87±3	89±2	87±2	87±2	88.8±1.7
SET	90±3	90.5±0.8	91.5±0.4	87±3	89±2	87±2	87±2	88.8±1.7
GD_DS	90±3	90.5±0.8	89±3	89±3	92.70±0.15	91.5±0.3	90±2	88.8±1.7
RLF	94.4±0.5	93.90±0.10	93.80±0.13	94±0	93.7±0.3	94±0	93±0	88.8±1.7
SCH	83.4±0.7	80.7±0.5	80.6±0.3	72.7±0.6	70.1±0.5	62.90±0.10	63±0	63±0
FUZ	----	----	----	----	----	----	----	----

**Tabla 48. Eficiencias para los métodos ARNMF y ACE.  
Problema WD.**

**BASE DE DATOS WD (parte 1)**

Método	1	2	3	4	5	6
BL2	95±2	94±2	98.5±0.4	98.1±0.7	97.3±1.7	98.93±0.16
TEKA	95±2	98.69±0.14	94±2	98.1±0.7	98.6±0.4	99.06±0.14
CIB	95±2	93±3	98.50±0.19	95.4±1.9	95.6±1.6	96.8±1.7
TEKE	97.5±1.7	99.50±0.08	95±2	94.3±1.8	97.3±1.7	94±2
TEKC	99.00±0.14	95.0±1.9	98.69±0.15	96.0±1.6	94±3	99.00±0.19
DEV	98.69±0.19	98.4±0.5	96±2	98.0±0.6	94±3	93±2
DER3	95±2	98.50±0.14	94.6±1.9	98.4±0.5	97.1±1.7	94±3
TEKB	95±2	93±3	98.4±0.3	98.75±0.13	97.2±1.7	96±2
TEK	95±2	94±2	98.50±0.19	95.4±1.9	95.6±1.6	96.8±1.7
BL1	95±2	98.50±0.10	98.1±0.5	93±3	98.13±0.16	96.7±1.7
CLO	95±2	98.50±0.10	98.1±0.5	96.1±1.8	95±2	98.69±0.17
PRI	95±2	98.50±0.10	98.1±0.5	93±3	98.13±0.16	96.7±1.7
SAN	95±2	98.50±0.10	98.1±0.5	96.1±1.8	98.13±0.16	98.69±0.17
DER2	95±2	93±3	98.4±0.3	94±3	95.6±1.6	98.63±0.13
MAO	97.2±1.7	96.69±0.15	98.5±0.4	99.06±0.10	98.6±0.2	99.06±0.14
UTA	96.4±1.6	93±3	92±2	95±2	98.6±0.2	95±2
LEE	98.7±0.6	96.0±1.4	93±2	98.8±0.3	98.8±0.4	98.4±0.3
BOW	95±2	95±2	94.9±1.9	97.2±1.7	99.13±0.17	92±2
YOU	96.69±0.11	94±2	95±2	98.9±0.2	98.1±1.0	94.8±1.9
BA 0.5	98.75±0.13	94±2	98.75±0	98.81±0.06	97.8±0.9	93±3
BA 0.6	98.75±0.13	96.9±1.7	98.63±0.13	98.81±0.06	97.8±0.9	93±3
BA 0.7	98.75±0.13	96.9±1.7	98.63±0.13	98.81±0.06	97.8±0.9	93±3
BA 0.8	98.75±0.13	96.9±1.7	98.63±0.13	98.81±0.06	97.8±0.9	93±3
BA 0.9	98.75±0.13	96.9±1.7	98.63±0.13	98.81±0.06	97.8±0.9	93±3
BA 1	98.75±0.13	96.9±1.7	98.63±0.13	98.81±0.06	97.8±0.9	93±3
CHI	99.06±0.14	94±3	98.1±0.6	98.50±0.10	98.63±0.18	98.4±0.6
SET	96.8±1.7	98.4±0.3	98.1±0.6	98.4±0.4	98.3±0.5	98.4±0.6
GD_DS	96.8±1.7	98.3±0.5	98.1±0.6	92±2	93±2	98.06±0.17
RLF	98.7±0.2	94±2	98.88±0.08	96.5±1.8	98.7±0.5	99.25±0.18
SCH	97.5±1.7	98.1±1.2	99.25±0.08	97.1±0.9	92±2	97.1±0.9
FUZ	95±2	98.50±0.10	98.19±0.17	96.6±1.7	98.63±0.13	98.50±0.14

**Tabla 48. Eficiencias para los métodos ARNMF y ACE.  
Problema WD. (Continuación).**

**BASE DE DATOS WD (parte 2)**

Método	7	8	9	10	11	12
BL2	97.8±1.0	97.2±1.4	99.00±0.10	95±2	98.1±0.9	98.94±0.19
TEKA	97.1±1.7	96±2	96±2	96.9±1.7	96.8±1.8	99.0±0.4
CIB	98.69±0.15	98.4±0.4	98.9±0.2	96.0±1.6	98.9±0.2	96.5±1.8
TEKE	91±2	94±2	89±3	93±2	87±3	90±3
TEKC	98.4±0.5	99.19±0.13	96.1±1.9	96.3±1.7	98.6±0.8	97.9±1.1
DEV	98.3±0.3	97.2±1.4	94±2	98.2±0.3	98.88±0.16	96.2±1.6
DER3	97.1±1.7	97.0±1.7	93±3	96±2	96.4±1.8	96.3±1.7
TEKB	95±2	97.9±0.8	98.50±0.17	96.2±1.8	98.75±0.16	97.8±0.9
TEK	98.7±0.4	98.4±0.4	98.75±0.16	94±2	96.5±1.5	99.50±0.18
BL1	98.63±0.18	97.9±0.8	95.8±1.6	96.4±1.1	96.8±1.7	97.8±0.9
CLO	95±2	98.81±0.17	95.8±1.6	96.4±1.1	98.81±0.15	98.75±0.16
PRI	98.63±0.18	97.9±0.8	95.8±1.6	96.4±1.1	96.8±1.7	97.8±0.9
SAN	95±2	97.9±0.8	95.8±1.6	96.4±1.1	98.81±0.14	97.8±0.9
DER2	98.88±0.16	98.8±0.2	94±2	98.94±0.13	97.0±1.7	98.6±0.2
MAO	98.94±0.13	98.6±0.2	97.0±1.7	96.0±1.6	95±2	96.5±1.8
UTA	93±3	95±2	98.81±0.17	97.7±0.9	98.81±0.15	98.75±0.16
LEE	96±2	97.5±0.8	92±2	98.4±0.3	96.0±1.8	90±2
BOW	96.8±1.6	97.0±1.7	99.19±0.13	99.1±0.3	98.5±0.3	99.0±0.4
YOU	99.44±0.11	99.4±0.2	97.5±1.8	97.1±1.6	99.50±0.15	98.6±1.2
BA 0.5	90±2	94±2	92±2	88±2	89±3	90±3
BA 0.6	90±2	94±2	92±2	88±2	89±3	90±3
BA 0.7	90±2	94±2	92±2	88±2	89±3	90±3
BA 0.8	92±3	92±2	92±2	89±3	95.6±0.4	95.3±0.2
BA 0.9	92±3	92±2	92±2	89±3	95.6±0.4	95.3±0.2
BA 1	92±3	92±2	92±2	89±3	95.6±0.4	95.3±0.2
CHI	97.56±0.17	94±2	94.0±1.8	97.38±0.18	96.88±0.09	97.69±0.09
SET	97.06±0.09	97.3±0.2	95.5±1.6	97.7±0.2	96.6±1.0	94±2
GD_DS	97.06±0.09	95.5±1.6	96.1±0.2	93±2	88±3	89±2
RLF	98.56±0.16	98.2±0.2	98.44±0.19	98.81±0.15	96.6±1.7	91±3
SCH	88.1±0.8	83.3±1.9	80.6±1.9	82.8±0.4	82.19±0.14	77±2
FUZ	93±3	95.6±1.9	91±2	90±2	94±2	95.6±1.7

**Tabla 48. Eficiencias para los métodos ARNMF y ACE.  
Problema WD. (Continuación).**

**BASE DE DATOS WD (parte 3)**

Método	13	14	15	16	17	18
BL2	95±2	98.2±0.4	93±3	96.4±1.7	96±2	96.6±1.4
TEKA	96±2	98.0±0.8	97.9±1.0	98.2±0.7	97.0±1.7	98.81±0.14
CIB	99.25±0.13	95±2	98.6±0.2	98.2±0.2	95±2	98.56±0.10
TEKE	91±2	91±2	90±2	93±2	91±3	91±2
TEKC	95.7±1.9	98.8±0.2	94±2	96.6±1.9	98.4±0.6	95±2
DEV	98.6±0.2	99.25±0.13	98.4±0.7	94.8±1.8	96.6±0.8	95±2
DER3	90±3	94±2	92±2	94±2	94±2	89±2
TEKB	97.4±1.7	96.9±1.7	98.81±0.15	98.2±0.7	97.0±1.7	98.82±0.15
TEK	98.88±0.18	95±2	96.9±1.8	96.5±1.8	98.94±0.19	99.3±0.5
BL1	97.4±1.7	98.1±0.9	98.81±0.15	98.2±0.7	98.69±0.15	98.63±0.16
CLO	97.4±1.7	98.1±0.9	98.81±0.15	98.2±0.7	98.69±0.15	93±2
PRI	97.4±1.7	98.1±0.9	98.81±0.15	98.2±0.7	98.69±0.15	98.63±0.16
SAN	97.4±1.7	96.9±1.7	98.81±0.15	98.2±0.7	98.69±0.15	93±2
DER2	98.81±0.17	98.9±0.2	99.13±0.10	95±2	98.56±0.09	98.94±0.13
MAO	93±2	95±2	98.5±0.4	97.4±1.7	96.3±1.8	98.56±0.10
UTA	98.7±0.2	98.1±0.9	97.2±1.7	98.81±0.17	92±3	95.0±1.9
LEE	90±2	85±3	90±3	93.8±0.4	87±3	91±2
BOW	96±2	99.50±0.13	97.2±1.8	99.3±0.2	97.8±1.4	98.94±0.16
YOU	97.7±1.8	96.1±1.7	96.69±0.17	98.88±0.08	98.56±0.13	95±2
BA 0.5	90±3	97.4±0.4	93±2	92±2	98.4±0.7	92±3
BA 0.6	90±3	97.4±0.4	93±2	92±2	98.4±0.7	92±3
BA 0.7	90±3	97.4±0.4	93±2	92±2	98.4±0.7	92±3
BA 0.8	90±3	97.4±0.4	93±2	92±2	98.4±0.7	92±3
BA 0.9	90±2	97.4±0.4	93±2	92±2	98.4±0.7	92±3
BA 1	90±2	97.4±0.4	93±2	92±2	98.4±0.7	92±3
CHI	97.81±0.10	98.63±0.13	98.1±0.4	96.9±0.3	97.1±0.5	97.56±0.17
SET	98.19±0.17	96.94±0.06	97.19±0.14	96.9±0.3	97.4±0.3	97.06±0.10
GD_DS	96.75±0.08	95.3±1.6	97.19±0.14	96.9±0.3	97.4±0.3	97.06±0.10
RLF	96±2	95±2	98.88±0.16	99.06±0.19	95±2	98.56±0.16
SCH	94.8±1.4	93±2	92±2	90±3	88±2	88.1±0.8
FUZ	93±2	95±2	91±3	91±3	97.9±0.2	93±3



**Tabla 48. Eficiencias para los métodos ARNMF y ACE.  
Problema WD. (Continuación).**

**BASE DE DATOS WD (parte 4)**

Método	19	20	21	22	23	24
BL2	98.3±1.0	97.7±1.8	94±2	94±2	93±2	93±2
TEKA	97.3±1.7	98.6±0.4	97.1±1.7	99.13±0.17	98.3±0.3	95±2
CIB	98.94±0.10	94.6±1.8	98.50±0.19	99.00±0.10	91±2	91±2
TEKE	95±3	87.9±0.3	81±2	84±2	84.3±0.6	84±3
TEKC	93±2	94.5±1.8	92±2	89±3	90±3	94±2
DEV	93±2	93±2	91±3	95.8±0.7	95.3±1.0	90±3
DER3	92±2	93.0±1.8	91±2	88±2	87±3	90±3
TEKB	99.63±0.17	98.6±0.4	97.1±1.7	99.13±0.17	98.3±0.3	94.7±1.9
TEK	99.3±0.3	94.6±1.8	98.50±0.19	99.00±0.10	91±2	98.3±0.2
BL1	99.63±0.17	98.6±0.4	97.1±1.7	99.13±0.17	98.3±0.3	94.7±1.9
CLO	99.63±0.17	98.6±0.4	97.1±1.7	96.7±1.7	98.3±0.3	94.7±1.9
PRI	99.63±0.17	98.6±0.4	97.1±1.7	99.13±0.17	98.3±0.3	94.7±1.9
SAN	99.63±0.17	98.6±0.4	97.1±1.7	96.7±1.7	98.3±0.3	94.7±1.9
DER2	98.94±0.13	93±3	98.94±0.10	95±2	96.8±1.7	98.44±0.17
MAO	95±2	94.6±1.8	99.13±0.14	99.00±0.10	91±2	98.3±0.2
UTA	94±2	98.94±0.16	95.8±1.7	96.7±1.7	96.8±1.7	94.2±1.8
LEE	87±2	85±3	91±2	85±2	93.1±0.4	78±2
BOW	99.25±0.16	96±2	98.75±0.19	98.3±0.3	96.7±1.4	95±2
YOU	94±2	97.7±1.1	96.5±1.8	92±2	95±2	95±2
BA 0.5	92±2	92±2	97.1±0.3	95.6±0.4	88±2	95.1±0.4
BA 0.6	92±2	92±2	97.1±0.3	95.6±0.4	88±2	95.1±0.4
BA 0.7	92±2	92±2	97.1±0.3	95.6±0.4	88±2	95.1±0.4
BA 0.8	92±2	92±2	89±3	95.6±0.4	95.3±0.2	95.1±0.4
BA 0.9	92±2	92±2	89±3	95.6±0.4	95.3±0.2	95.1±0.4
BA 1	92±2	92±2	89±3	95.6±0.4	95.3±0.2	95.1±0.4
CHI	94±2	94.0±1.8	97.38±0.18	96.88±0.09	97.69±0.10	97.3±0.2
SET	97.3±0.2	95.5±1.6	97.7±0.2	96.6±1.0	94±2	95.7±1.7
GD_DS	95.5±1.6	96.1±0.2	93±2	88±3	89±2	95.9±0.5
RLF	98.2±0.2	98.44±0.19	98.81±0.15	96.6±1.7	91±3	98.6±0.2
SCH	83.3±1.9	80.6±1.9	82.8±0.4	82.19±0.14	77±2	69.1±1.1
FUZ	95.6±1.9	91±2	90±2	94±2	95.6±1.7	93±2

**Tabla 48. Eficiencias para los métodos ARNMF y ACE.  
Problema WD. (Continuación).**

**BASE DE DATOS WD (parte 5)**

Método	25	26	27	28	29	30
BL2	92±2	92±2	90±3	92.00±0.13	84.1±1.7	64.375±0
TEKA	98.44±0.14	95.6±1.6	98.00±0.16	94.50±0.13	92.3±0.3	88±3
CIB	96.6±1.7	94±2	98.50±0.14	89±2	89±3	71.3±0.3
TEKE	80±3	81.4±0.7	79.1±0.2	71.1±0.4	67.6±0.4	67.8±0.4
TEKC	93±3	93±2	94±2	96.81±0.15	88±3	85±2
DEV	90±2	88±2	92±2	87±3	63.6±0.3	64.375±0
DER3	90±3	90±2	89±3	89.9±1.0	89.94±0.06	84±2
TEKB	98.44±0.14	98.31±0.13	98.00±0.16	94.50±0.13	92.3±0.2	88±3
TEK	96.6±1.7	94±2	98.50±0.13	89±2	89±3	91.00±0.10
BL1	98.44±0.14	98.31±0.13	98.00±0.16	94.50±0.13	92.3±0.3	88±3
CLO	91±3	98.31±0.13	98.00±0.16	90±2	92.3±0.3	71.3±0.3
PRI	98.44±0.14	98.31±0.13	98.00±0.16	94.50±0.13	92.3±0.3	88±3
SAN	98.44±0.14	98.31±0.13	98.00±0.16	90±2	92.3±0.3	88±3
DER2	90±3	90±3	92.44±0.06	89±2	75.6±1.9	71.3±0.3
MAO	96.6±1.7	94±2	98.50±0.14	89±2	89±3	71.3±0.3
UTA	91±2	91±2	79±2	76.9±1.8	75.7±1.9	71.3±0.3
LEE	81.3±0.5	81.1±1.9	79±2	76.4±1.4	78.9±0.6	78.0±1.3
BOW	93±2	94.8±1.7	93±2	92.9±0.3	92.3±0.3	71.3±0.3
YOU	97.7±0.7	94±2	98.00±0.16	94.50±0.13	89±3	91.00±0.10
BA 0.5	91.5±1.7	89±2	94.1±0.5	90±3	90±3	91.00±0.10
BA 0.6	91.5±1.7	89±3	94.1±0.5	90±3	90±3	91.00±0.10
BA 0.7	91.5±1.7	89±3	94.1±0.5	90±3	90±3	91.00±0.10
BA 0.8	91.5±1.7	89±3	94.1±0.5	90±3	90±3	91.00±0.10
BA 0.9	91.5±1.7	89±3	94.1±0.5	90±3	90±3	91.00±0.10
BA 1	91.5±1.7	89±3	94.1±0.5	90±3	90±3	91.00±0.10
CHI	97.3±0.3	94.3±1.6	96.69±0.16	94±2	94.2±1.6	91.00±0.10
SET	90±2	90±2	94.6±0.7	95.50±0.08	87±3	76±3
GD_DS	93.3±1.6	95.6±0.2	93.3±1.6	92±2	91±2	92.5±1.3
RLF	98.19±0.15	94±2	98.00±0.16	94.50±0.13	89±3	91.00±0.10
SCH	68.8±0.9	67.3±1.4	62.4±0.5	64.375±0	64.375±0	64.375±0
FUZ	93±2	92±3	92±2	96.6±0.2	88±2	89±2

## II.4. Eficiencias para los métodos ARNMF &WD

*Tabla 49. Eficiencias para los métodos ARNMF&WD.  
Problema BN.*

### *BASE DE DATOS BN (parte 1)*

Método	1	2	3	4	5	6	7	8
BL2	65.1±1.0	65.4±1.2	65.6±0.8	65.9±0.7	66.9±0.8	66.4±1.1	66.8±0.8	66.8±0.8
TEKA	65.1±1.0	65.4±1.2	65.6±0.8	65.9±0.7	66.9±0.8	66.4±0.6	67.1±1.0	66.3±1.0
CIB	65.1±1.0	66.4±0.9	65.1±0.6	66.0±0.8	66.2±0.7	66.3±0.8	66.0±0.7	65.6±0.5
TEKE	----	----	----	----	----	----	----	----
DEV	65.1±1.0	65.6±0.4	66.7±0.9	67.4±0.6	66.9±0.5	67.6±0.7	66.7±0.8	66.7±0.8
DER3	65.1±1.0	72.0±0.9	72.3±0.8	72.4±0.8	71.8±1.1	70.5±0.8	72.7±1.4	70.1±1.1
TEKB	65.1±1.0	65.4±1.2	65.6±0.8	65.9±0.7	65.7±0.9	67.8±1.0	67.1±0.9	66.8±0.8
TEK	65.1±1.0	64.4±0.9	65.1±0.6	66.0±0.8	64.7±0.6	66.3±0.8	66.0±0.7	65.6±0.5
BL1	66.0±1.3	66.8±1.0	67.7±0.8	68.0±0.8	66.9±0.5	67.6±0.7	66.7±0.8	66.8±0.8
CLO	66.0±1.3	67.1±0.7	65.7±0.5	68.0±0.8	66.9±0.5	67.6±0.7	66.7±0.8	66.8±0.8
PRI	66.0±1.3	66.8±1.0	67.7±0.8	68.0±0.8	66.9±0.5	67.6±0.7	66.7±0.8	66.8±0.8
SAN	66.0±1.3	66.8±1.0	67.7±0.8	68.0±0.8	66.9±0.5	67.6±0.7	66.7±0.8	66.9±1.2
DER2	65.7±0.4	67.1±0.7	66.6±0.9	67.4±0.6	66.9±0.5	66.4±0.7	66.0±0.7	66.8±0.8
MAO	63.9±1.0	63.2±0.6	65.6±0.6	66.6±0.9	65.4±1.0	65.4±1.0	66.6±0.7	66.1±1.0
UTA	65.5±0.7	66.2±0.9	64.3±0.8	64.0±1.0	67.0±1.0	66.6±0.8	66.2±1.1	66.3±1.0
LEE	71.5±0.8	71.7±0.9	72.2±0.4	71.3±1.0	69.9±0.6	69.0±0.8	69.2±0.7	69.0±1.2
YOU	63.2±0.6	71.7±0.9	70.6±0.9	70.9±0.3	70.2±0.4	70.9±0.8	70.8±1.0	72.6±1.5

### *BASE DE DATOS BN (parte 2)*

Método	9	10	11	12	13	14	15	16
BL2	67.5±0.8	67.3±0.8	68.5±1.0	65.3±0.4	65.4±1.2	66.2±0.5	65.2±0.8	64.9±0.7
TEKA	66.2±0.6	67.3±0.8	68.5±1.0	65.3±0.4	65.3±1.6	66.2±0.5	65.2±0.8	67.1±0.9
CIB	66.2±0.8	65.2±0.5	67.2±0.6	66.0±0.7	65.3±1.0	64.5±0.9	64.7±0.7	64.8±0.7
TEKE	----	----	----	----	----	----	----	----
DEV	67.5±0.8	67.1±1.2	68.5±1.0	65.3±0.4	64.4±0.3	66.6±0.8	67.4±0.9	68.7±0.7
DER3	74.4±1.0	71.3±0.9	74.0±1.4	70.7±1.2	70.0±1.0	71.4±0.9	66.3±0.9	68.4±0.9
TEKB	67.5±0.8	65.3±0.7	65.5±0.8	65.3±0.4	65.3±1.6	66.2±0.5	65.2±0.8	67.1±0.9
TEK	67.6±0.6	67.4±0.9	64.7±0.6	64.4±0.8	64.4±0.8	65.1±0.6	65.5±1.0	64.8±0.7
BL1	65.2±0.6	67.3±0.8	68.5±1.0	65.3±0.4	65.4±1.2	66.2±0.5	65.2±0.8	67.1±0.9
CLO	65.2±0.6	67.3±0.8	68.5±1.0	65.3±0.4	65.4±1.2	66.2±0.5	65.2±0.8	67.1±0.9
PRI	65.2±0.6	67.3±0.8	68.5±1.0	65.3±0.4	65.4±1.2	66.2±0.5	65.2±0.8	67.1±0.9
SAN	65.2±0.6	67.3±0.8	68.5±1.0	65.3±0.4	65.4±1.2	66.2±0.5	65.2±0.8	67.1±0.9
DER2	66.3±1.1	64.4±0.7	65.4±0.7	65.3±0.4	65.4±1.2	65.6±0.8	66.4±0.6	68.3±0.8
MAO	68.9±1.2	67.1±0.9	66.9±0.8	68.3±0.9	68.4±0.5	67.7±0.9	66.6±0.7	66.4±0.7
UTA	65.1±0.9	67.3±0.8	65.7±0.6	66.1±0.6	67.0±0.7	66.6±0.8	66.8±0.6	67.4±0.8
LEE	69.1±1.2	70.1±0.6	73.2±1.8	69.8±1.0	71.0±1.2	72.3±1.1	72.0±0.9	71.6±0.7
YOU	69.4±1.1	72.9±1.2	72.5±0.7	70.6±0.7	72.9±0.8	72.3±0.8	72.0±0.8	72.8±0.8

**Tabla 49. Eficiencias para los métodos ARNMF&WD.  
Problema BN. (Continuación).**

**BASE DE DATOS BN (parte 3)**

Método	17	18	19	20	21	22	23	24
BL2	65.3±1.0	67.4±0.5	67.4±0.6	64.3±0.8	68.0±0.5	66.7±0.7	67.5±0.6	68.4±0.5
TEKA	65.3±1.0	67.4±0.5	67.4±0.6	64.3±0.8	68.0±0.5	66.7±0.7	65.5±0.9	68.4±0.5
CIB	66.7±0.9	65.7±0.6	66.3±0.9	66.7±0.4	66.8±1.0	66.7±0.7	65.5±0.9	67.5±1.0
TEKE	----	----	----	----	----	----	----	----
DEV	66.1±0.4	68.2±0.9	65.9±0.6	66.6±1.0	68.2±0.6	66.1±1.1	64.6±0.7	67.5±1.0
DER3	69.2±1.1	66.4±1.3	67.8±0.9	67.0±1.1	68.5±1.1	67.0±0.9	66.9±1.0	67.7±0.7
TEKB	65.3±1.0	67.4±0.5	67.4±0.6	64.3±0.8	68.0±0.5	66.7±0.7	64.6±0.7	67.5±1.0
TEK	66.7±0.9	65.7±0.6	66.3±0.9	65.5±0.5	67.2±1.2	66.7±0.8	64.6±0.7	63.6±1.1
BL1	65.3±1.0	68.2±0.9	66.7±0.5	66.6±1.0	68.0±0.5	66.7±0.7	64.6±0.7	67.2±0.7
CLO	65.3±1.0	68.2±0.9	66.7±0.5	66.6±0.5	68.0±0.5	66.7±0.7	64.6±0.7	67.2±0.7
PRI	65.3±1.0	68.2±0.9	66.7±0.5	66.6±1.0	68.0±0.5	66.7±0.7	64.6±0.7	67.2±0.7
SAN	65.3±1.0	67.4±0.5	66.7±0.5	66.6±1.0	68.0±0.5	66.7±0.7	64.6±0.7	67.2±0.7
DER2	67.5±0.7	66.7±0.8	66.8±0.9	67.5±0.8	66.2±0.5	67.0±1.2	64.6±0.7	67.5±1.0
MAO	66.3±0.8	66.2±0.4	67.8±0.7	66.4±0.7	66.0±0.6	66.1±0.6	64.1±0.6	66.9±1.3
UTA	66.8±0.6	66.3±0.7	66.8±0.9	65.1±0.4	66.4±0.6	67.2±0.5	66.1±0.4	67.5±1.0
LEE	69.9±0.6	69.5±0.7	67.8±0.6	67.1±0.7	65.4±0.6	65.6±0.7	67.0±0.7	66.2±0.9
YOU	71.6±0.9	71.3±0.7	70.2±0.4	68.4±0.7	69.7±0.7	70.1±0.6	68.6±0.9	70.7±0.6

**BASE DE DATOS BN (parte 4)**

Método	25	26	27	28	29	30	31	32
BL2	66.9±0.8	65.1±0.6	65.2±1.0	68.6±1.0	65.7±0.5	67.8±0.6	67.7±0.5	66.3±0.8
TEKA	66.6±0.7	65.1±0.6	65.2±1.0	61.3±0.9	65.7±0.5	66.4±0.7	67.0±1.0	67.1±0.5
CIB	63.1±0.9	62.2±0.7	62.3±1.2	64.8±1.8	66.0±1.3	62.2±1.7	65.1±1.2	69.2±0.7
TEKE	----	----	----	----	----	----	----	----
DEV	63.1±0.9	65.1±0.6	65.2±1.0	68.6±1.0	66.1±0.5	66.3±0.5	64.9±0.5	67.0±0.2
DER3	67.9±0.7	68.2±0.6	66.4±0.6	66.4±0.7	65.5±0.6	65.2±0.6	65.4±0.6	65.6±0.5
TEKB	66.9±0.8	65.1±0.6	65.2±1.0	68.6±1.0	65.7±0.5	67.8±0.6	67.7±0.5	66.3±0.8
TEK	68.2±0.9	62.2±0.7	64.6±0.7	66.2±0.6	66.1±1.1	64.4±1.1	65.1±1.2	61.3±1.1
BL1	66.9±0.8	65.1±0.6	65.2±1.0	68.6±1.0	65.7±0.5	67.8±0.6	67.7±0.5	63.9±0.4
CLO	66.9±0.8	65.1±0.6	65.2±1.0	68.6±1.0	65.7±0.5	67.8±0.6	67.7±0.5	63.9±0.4
PRI	66.9±0.8	65.1±0.6	65.2±1.0	68.6±1.0	65.7±0.5	67.8±0.6	67.7±0.5	63.9±0.4
SAN	66.9±0.8	65.1±0.6	65.2±1.0	68.6±1.0	65.7±0.5	67.8±0.6	67.7±0.5	66.3±0.8
DER2	69.9±0.5	69.4±0.4	69.1±0.5	69.3±0.4	68.2±0.5	66.3±0.5	66.4±0.4	68.0±0.7
MAO	63.6±1.2	63.0±1.0	65.2±0.5	63.9±0.5	68.6±0.5	67.4±0.8	66.7±0.4	66.7±0.7
UTA	66.9±0.8	69.1±0.7	68.7±0.5	70.3±1.0	71.5±0.4	68.6±0.5	67.7±0.4	66.3±0.8
LEE	66.8±0.7	67.2±0.5	65.9±0.8	66.2±0.9	62±3	69.1±0.6	69.9±0.6	70.3±0.2
YOU	69.2±0.5	70.1±0.9	68.4±0.7	66.6±0.7	66.8±0.7	68.2±1.0	68.6±0.9	63.0±0.7

**Tabla 49. Eficiencias para los métodos ARNMF&WD.  
Problema BN. (Continuación).**

**BASE DE DATOS BN (parte 5)**

Método	33	34	35	36	37	38	39
BL2	63.7±0.9	61.4±0.6	67.2±0.6	62.3±0.9	62.5±0.3	65.5±1.0	64.4±0.6
TEKA	63.7±0.9	72.0±0.9	67.2±0.6	62.3±0.9	62.5±0.3	65.5±1.0	64.4±0.6
CIB	72.2±0.9	74.6±0.5	65.1±0.3	65.4±0.2	65.2±0.6	66.7±0	66.7±0
TEKE	----	----	----	----	----	----	----
DEV	64.3±0.6	61.4±0.6	60.6±0.5	60.1±1.2	66.0±0.5	65.5±1.0	64.4±0.6
DER3	69.2±0.7	66.8±0.2	61.8±0.5	61.7±0.9	62.1±0.7	60.5±0.4	66.7±0
TEKB	63.7±0.9	61.4±0.6	67.2±0.6	66.0±0.6	62.5±0.3	65.5±1.0	64.4±0.6
TEK	64.3±1.1	67.4±0.7	67.0±0.5	65.5±0.2	65.2±0.6	66.7±0	66.7±0
BL1	64.0±0.9	61.4±0.6	67.2±0.6	66.0±0.6	62.5±0.3	65.5±1.0	64.4±0.6
CLO	64.0±0.9	61.4±0.6	67.2±0.6	66.0±0.6	62.5±0.3	65.5±1.0	64.4±0.6
PRI	64.0±0.9	61.4±0.6	67.2±0.6	66.0±0.6	62.5±0.3	65.5±1.0	64.4±0.6
SAN	64.0±0.9	61.4±0.6	67.2±0.6	66.0±0.6	62.5±0.3	65.5±1.0	64.4±0.6
DER2	65.1±0.7	64.8±0.8	62.8±0.8	59.9±0.6	64.6±0.5	69.8±1.1	64.6±0.5
MAO	64.8±0.8	62.5±1.0	64.4±0.5	70.7±1.5	60.2±0.8	65.5±1.0	66.7±0
UTA	70.5±0.6	62.5±0.5	64.6±0.9	69.2±0.2	69.0±0	72.53±0.12	66.7±0
LEE	67.6±1.1	67.4±0.7	67.0±0.5	65.5±0.2	65.2±0.6	66.7±0	66.7±0
YOU	64.3±0.8	63.6±0.7	66.1±0.4	66.2±0.5	68.7±0.7	69.8±1.1	66.7±0

**Tabla 50. Eficiencias para los métodos ARNMF&WD.  
Problema BU.**

**BASE DE DATOS BU**

Método	1	2	3	4	5	6
BL2	67±2	57.7±1.0	55.7±0.6	57.4±1.1	50.40±0.16	50.5±0.6
TEKA	67±2	57.7±1.0	55.7±0.6	57.4±1.1	50.40±0.16	50.5±0.6
CIB	63.5±1.7	54.9±0.5	55±0.5	51.3±0.4	50.40±0.16	54±0
TEKE	68.3±0.4	62.0±1.1	52.6±0.4	51.4±0.8	51.3±0.5	54.3±0.4
DEV	67±2	65±2	67.4±1.6	54.9±0.5	58.5±0.8	55.5±0.5
DER3	69±2	63.4±0.9	60.6±0.7	57.9±1.4	58.5±0.8	55.5±0.5
TEKB	67±2	57.7±1.0	55.7±0.6	57.4±1.1	50.40±0.16	50.5±0.6
TEK	60.1±1.3	56.3±0.8	54.5±0.5	53.9±0.5	50.40±0.16	54±0
BL1	67±2	65±2	55.7±0.6	57.4±1.1	50.40±0.16	50.5±0.6
CLO	67±2	65±2	55.7±0.6	57.4±1.1	50.40±0.16	50.5±0.6
PRI	67±2	65±2	55.7±0.6	57.4±1.1	50.40±0.16	50.5±0.6
SAN	67±2	65±2	55.7±0.6	57.4±1.1	50.40±0.16	50.5±0.6
DER2	67±2	57.7±1.0	55.7±0.6	54.9±0.5	53.5±0.6	57.6±0.8
MAO	56.9±1.4	60.4±1.3	60.6±0.8	57.4±1.1	50.40±0.16	50.5±0.6
UTA	67±2	65±2	55.7±0.6	54.9±0.5	53.5±0.6	50.5±0.6
LEE	67±2	69.5±0.2	66±2	57.5±1.1	57±0.8	54.3±0.4
YOU	----	----	----	----	----	----

**Tabla 51. Eficiencias para los métodos ARNMF&WD.  
Problema CR.**

**BASE DE DATOS CR (parte 1)**

Método	1	2	3	4	5	6	7	8
BL2	85±3	81±3	85±3	88.2±0.3	83±2	86.5±1.7	85±3	85±3
TEKA	85±3	81±3	85±3	88.2±0.3	85±2	86.5±1.7	85±3	85±3
CIB	83±3	84±3	85±3	87.9±0.4	80±5	84±3	86.8±0.2	80±4
TEKE	----	----	----	----	----	----	----	----
DEV	85±3	81±3	85±3	88.2±0.3	83±3	86.5±1.7	85±3	85±3
DER3	83±2	82±4	78±5	84±4	81±5	85±3	85±3	85±3
TEKB	85±3	81±3	85±3	88.2±0.3	83±3	86.5±1.7	85±3	85±3
TEK	83±3	86.4±1.7	82±3	85±3	85±3	81±4	74±5	84±3
BL1	85±3	81±3	85±3	88.2±0.3	85±2	86.5±1.7	85±3	85±3
CLO	85±3	81±3	85±3	88.2±0.3	85±2	86.5±1.7	85±3	85±3
PRI	85±3	81±3	85±3	88.2±0.3	84±2	86.5±1.7	85±3	85±3
SAN	85±3	81±3	85±3	88.2±0.3	85±2	86.5±1.7	85±3	85±3
DER2	85±3	81±3	85±3	77±5	87.0±0.4	86.8±0.3	87.5±0.3	84±3
MAO	85±3	81±4	83±3	86.1±0.3	79±4	82±3	82±3	82±3
UTA	86.2±1.7	85±3	87.5±0.3	88.2±0.3	85±2	88.2±0.3	84±3	80±4
LEE	77±2	73.5±1.8	74±2	75±2	70±3	74±2	76.0±1.1	76.8±1.2
YOU	84±3	82±4	82±4	84±3	83±3	71±5	80±4	80±4

**BASE DE DATOS CR (parte 2)**

Método	9	10	11	12	13	14	15
BL2	74±6	82±4	81±4	88.10±0.07	88±0	85.8±1.5	88±0
TEKA	85±3	82±4	85±3	88.10±0.07	88±0	88±0	88±0
CIB	81±4	83±4	84±3	83±2	81±3	85.8±1.5	72±3
TEKE	----	----	----	----	----	----	----
DEV	74±6	82±5	81±4	88.10±0.07	88±0	85.8±1.5	88±0
DER3	81±5	85±3	75±5	85±5	88±0	88±0	88±0
TEKB	85±3	82±4	81±4	88.10±0.07	88±0	85.8±1.5	88±0
TEK	81±2	83±3	83±3	83±2	85±3	85.8±1.5	88±0
BL1	74±6	82±4	81±4	88.10±0.07	88±0	85.8±1.5	88±0
CLO	74±6	82±4	81±4	88.10±0.07	88±0	85.8±1.5	88±0
PRI	74±6	82±4	81±4	88.10±0.07	88±0	85.8±1.5	88±0
SAN	74±6	82±4	81±4	88.10±0.07	88±0	85.8±1.5	88±0
DER2	81±4	85±3	81±4	78±5	85±3	85.8±1.5	88±0
MAO	76±7	86.2±0.4	84±3	83±2	81±4	88±0	61.1±0.7
UTA	80±4	87.5±0.2	84±3	85±2	88±0	85.8±1.5	88±0
LEE	71±4	72±3	70±4	74±3	76.7±0.5	65.7±1.3	68.3±0.5
YOU	88.20±0.17	85±3	82±4	85±3	85±3	85.8±1.5	88±0

**Tabla 52. Eficiencias para los métodos ARNMF&WD.  
Problema GL.**

**BASE DE DATOS GL (parte 1)**

Método	1	2	3	4	5	6
BL2	44±3	55±4	45±4	49±3	47±3	51±4
TEKA	44±3	55±4	45±4	49±3	51±5	51±4
CIB	44±3	49±3	47±4	49±3	46±3	48±3
TEKE	44±3	49±3	47±4	43±4	48±4	46±3
DEV	44±3	55±4	45±4	49±3	47±3	51±4
DER3	40±2	43.1±1.4	43.7±1.5	43.9±1.5	49.5±1.9	43.1±1.9
TEKB	44±3	55±4	45±4	49±3	51±5	51±4
TEK	44±3	49±3	36.1±1.4	52±4	46±3	48±4
BL1	44±3	55±4	45±4	49±3	47±3	51±4
CLO	44±3	55±4	45±4	49±3	47±3	51±4
PRI	44±3	55±4	45±4	49±3	47±3	51±4
SAN	44±3	55±4	45±4	49±3	47±3	51±4
DER2	44±3	55±4	47±4	47±4	47±3	51±4
MAO	44±3	55±4	45±4	49±3	47±3	51±4
UTA	49±3	49±3	45±4	49±3	47±3	51±4
LEE	52±4	43±2	36±2	39±2	41.7±0.8	40.2±1.3
YOU	----	----	----	----	----	----

**BASE DE DATOS GL (parte 2)**

Método	7	8	9	10
BL2	44±3	49±3	56±3	60±3
TEKA	44±3	49±3	56±3	60±3
CIB	47±3	49±3	56±3	60±3
TEKE	45.9±1.9	56±4	55±5	60±3
DEV	44±3	49±3	56±3	60±3
DER3	45.9±1.9	49.9±1.0	43.8±1.5	37.2±1.5
TEKB	44±3	49±3	56±3	60±3
TEK	46±4	49±3	56±3	60±3
BL1	44±3	49±3	56±3	60±3
CLO	44±3	49±3	56±3	60±3
PRI	44±3	49±3	56±3	60±3
SAN	44±3	46±3	56±3	60±3
DER2	47±3	52±3	56±3	37.2±1.5
MAO	46±3	49±3	56±3	60±3
UTA	44±3	49±3	56±3	60±3
LEE	37.5±1.3	42.7±0.7	37.3±0.9	37.5±0
YOU	----	----	----	----

**Tabla 53. Eficiencias para los métodos ARNMF&WD.  
Problema HE.**

**BASE DE DATOS HE (parte 1)**

Método	1	2	3	4	5	6	7
BL2	75±2	76±2	74±2	67±4	74±2	79.1±1.0	76±4
TEKA	75±2	76±2	71±4	67±4	74±2	79.1±1.0	76±4
CIB	75±2	76±3	74±2	67±4	73±3	74.4±1.7	74±2
TEKE	74±2	73±3	78.7±0.5	71±4	75±2	75±2	70±4
DEV	75±2	76±3	74±2	67±4	74±2	79.1±1.0	76±4
DER3	74±2	75±2	67±3	72±3	64.0±1.2	66.2±1.4	65.2±1.0
TEKB	75±2	76±2	74±2	67±4	74±2	79.1±1.0	76±4
TEK	75±2	76±3	80.8±0.3	74±3	76±3	79.1±1.0	74±2
BL1	76±3	76±2	71±4	67±4	74±2	79.1±1.0	76±4
CLO	76±3	76±2	71±4	67±4	74±2	79.1±1.0	76±4
PRI	76±3	76±2	71±4	67±4	74±2	79.1±1.0	76±4
SAN	75±2	76±2	74±2	67±4	74±2	79.1±1.0	76±4
DER2	75±2	76±2	74±2	67±4	74±2	74.4±1.7	77±3
MAO	75±2	76.3±0.6	68±2	66±3	62±3	64±3	72±2
UTA	75±2	73±3	71±4	74.5±1.6	74±2	74.4±1.7	77±3
LEE	74±2	72±2	68±2	67±2	62±2	65.7±1.4	61.0±1.3
YOU	73±2	75±3	78±3	70±4	74±4	76±4	73±3

**BASE DE DATOS HE (parte 2)**

Método	8	9	10	11	12	13
BL2	72±3	77±3	73±2	71±3	67.2±1.6	68.5±1.6
TEKA	72±3	77±3	73±2	71±3	67.2±1.6	68.5±1.6
CIB	72±3	77±3	74.0±1.5	71±3	60±3	66±2
TEKE	72±3	67±4	73.7±0.6	71±3	65±2	73±2
DEV	72±3	77±3	73±2	71±3	67.2±1.6	68.5±1.6
DER3	67.1±0.7	66.8±0.7	63.7±1.7	66±3	68±2	66±2
TEKB	72±3	77±3	73±2	71±3	67.2±1.6	68.5±1.6
TEK	72±3	77±3	74.4±0.5	71±3	65±2	66±2
BL1	72±3	77±3	73±2	71±3	67.2±1.6	68.5±1.6
CLO	72±3	77±3	73±2	71±3	67.2±1.6	68.5±1.6
PRI	72±3	77±3	73±2	71±3	67.2±1.6	68.5±1.6
SAN	72±3	77±3	73±2	71±3	67.2±1.6	68.5±1.6
DER2	72±3	77±3	73±2	71±3	67.2±1.6	68.5±1.6
MAO	68±3	68±3	62.4±1.0	66±3	68±2	66±2
UTA	72±3	76±3	72±2	74±2	76.4±0.5	68.5±1.6
LEE	63.0±1.5	60.7±0.9	63.6±0.6	62.8±0.6	62.9±1.7	64.8±0.6
YOU	77.1±0.7	76±2	70.5±0.4	64±2	67.2±1.6	66±2



**Tabla 54. Eficiencias para los métodos ARNMF&WD.  
Problema PI.**

**BASE DE DATOS PI (parte 1)**

Método	1	2	3	4
BL2	75.24±0.17	75.72±0.18	76.76±0.15	77.0±0.2
TEKA	75.24±0.17	76.1±0.2	76.76±0.15	77.0±0.2
CIB	75.24±0.17	75.72±0.18	76.76±0.15	77.0±0.2
TEKE	75.6±0.2	74.7±1.1	75.04±0.19	74.3±0.3
DEV	75.5±0.3	75.72±0.18	74.4±1.1	77.0±0.2
DER3	78.5±0.4	69.7±0.7	67.0±0.5	69.5±0.7
TEKB	75.5±0.3	75.72±0.18	76.76±0.15	77.0±0.2
TEK	75.24±0.17	76.1±0.2	73.7±1.4	77.0±0.2
BL1	75.5±0.3	75.72±0.18	76.76±0.15	77.0±0.2
CLO	75.5±0.3	75.72±0.18	76.76±0.15	77.0±0.2
PRI	75.5±0.3	75.72±0.18	76.76±0.15	77.0±0.2
SAN	75.5±0.3	75.72±0.18	76.76±0.15	77.0±0.2
DER2	75.24±0.17	75.72±0.18	76.76±0.15	77.0±0.2
MAO	75.4±0.3	75.64±0.19	74.7±0.2	74.2±0.8
UTA	75.24±0.17	76.1±0.2	76.76±0.15	77.0±0.2
LEE	66.3±0.6	69.7±0.7	69.0±1.1	70.8±0.9
YOU	76.3±0.3	76.7±0.2	75.8±1.2	76.9±0.2

**BASE DE DATOS PI (parte 2)**

Método	5	6	7	8
BL2	76.56±0.09	75.52±0.10	75.76±0.14	74.4±0
TEKA	76.56±0.09	77.32±0.18	75.76±0.14	74.4±0
CIB	76.56±0.09	75.52±0.10	75.76±0.14	74.4±0
TEKE	75.1±0.3	75.52±0.10	75.76±0.14	74.4±0
DEV	76.56±0.09	75.52±0.10	75.76±0.14	74.4±0
DER3	67.3±0.8	65.3±0.4	65.6±0.3	63.1±0.5
TEKB	76.56±0.09	75.52±0.10	75.76±0.14	74.4±0
TEK	76.56±0.09	75.52±0.10	75.76±0.14	74.4±0
BL1	76.56±0.09	75.52±0.10	75.76±0.14	74.4±0
CLO	76.56±0.09	75.52±0.10	75.76±0.14	74.4±0
PRI	76.56±0.09	75.53±0.10	75.76±0.14	74.4±0
SAN	76.56±0.09	75.52±0.10	75.76±0.14	74.4±0
DER2	76.56±0.09	75.52±0.10	75.76±0.14	74.4±0
MAO	75.2±0.2	75.52±0.10	74.0±0.3	63.1±0.5
UTA	76.56±0.09	77.32±0.18	75.76±0.14	74.4±0
LEE	70.8±0.9	68.0±0.9	65.6±0.8	65.2±0
YOU	76.2±1.2	77.32±0.18	75.76±0.14	74.4±0

II. APÉNDICES DEL CAPÍTULO 3

**Tabla 55. Eficiencias para los métodos ARNMF&WD.  
Problema VO.**

**BASE DE DATOS VO (parte 1)**

Método	1	2	3	4	5	6	7	8
BL2	94.4±0.3	87±6	92±3	94.6±0.2	90±3	93.5±0.3	93.8±0.6	91±3
TEKA	94.4±0.3	87±6	92±3	94.6±0.2	91±3	91±3	93.8±0.6	91±3
CIB	90±4	87±6	92±3	91±3	93.6±0.5	93.5±0.3	93.6±0.4	91±3
TEKE	93.7±0.2	93.4±0.2	87±4	90±3	91.7±0.3	90±3	90±3	93.4±0.3
DEV	94.4±0.3	87±6	92±3	94.6±0.2	90±3	93.5±0.3	93.8±0.6	91±3
DER3	93.0±1.6	91.6±0.3	86±4	84±4	89.7±0.3	87±3	84±3	93.4±0.5
TEKB	94.4±0.3	87±6	92±3	94.6±0.2	90±3	93.5±0.3	93.8±0.6	91±3
TEK	94.4±0.3	87±6	92±3	91±3	93.6±0.5	91±3	93.8±0.6	91±3
BL1	94.4±0.3	87±6	92±3	94.6±0.2	90±3	93.5±0.3	93.8±0.6	91±3
CLO	94.4±0.3	87±6	92±3	94.6±0.2	90±3	93.5±0.3	93.8±0.6	91±3
PRI	94.4±0.3	87±6	92±3	94.6±0.2	90±3	93.5±0.3	93.8±0.6	91±3
SAN	94.4±0.3	87±6	92±3	94.6±0.2	90±3	93.5±0.3	93.8±0.6	91±3
DER2	90±4	87±6	92±3	94.6±0.2	90±3	91±3	93.8±0.6	89±4
MAO	94.2±0.4	95.2±0.2	94.4±0.4	91±3	91±3	92±3	92±3	95.2±0.3
UTA	92.8±1.6	91±3	92±3	91±3	88±4	93.8±0.6	92±3	91±3
LEE	91.3±0.6	91.0±0.2	91.7±0.6	91.6±0.3	92.2±0.6	90.9±0.4	91.0±0.5	86±3
YOU	91±2	91±3	94.6±0.4	94.7±0.9	95.3±0.4	92±3	88±4	94.5±0.2

**BASE DE DATOS VO (parte 2)**

Método	9	10	11	12	13	14	15	16
BL2	94.5±0.5	90±4	95.6±0.5	91±2	92±2	94±0	93±0	88.8±1.7
TEKA	92±3	94.5±0.4	95.6±0.5	91±2	92±2	94±0	93±0	88.8±1.7
CIB	92±3	88±4	91±3	95.7±0.3	92±2	94±0	87±2	88.8±1.7
TEKE	93.2±0.3	88±4	93.5±0.2	91±3	91±3	94±0	93±0	88.8±1.7
DEV	94.5±0.5	90±4	95.6±0.5	91±2	92±2	94±0	93±0	88.8±1.7
DER3	91±3	92.7±0.7	90±3	88±3	91.3±0.4	90.1±0.6	90.2±0.2	69±0
TEKB	94.5±0.5	90±4	95.6±0.5	91±2	92±2	94±0	93±0	88.8±1.7
TEK	92±3	88±4	91±3	91±3	92±2	94±0	87±2	88.8±1.7
BL1	94.5±0.5	90±4	95.6±0.5	91±2	92±2	94±0	93±0	88.8±1.7
CLO	94.5±0.5	90±4	95.6±0.5	91±2	92±2	94±0	93±0	88.8±1.7
PRI	94.5±0.5	90±4	95.6±0.5	91±2	92±2	94±0	93±0	88.8±1.7
SAN	94.5±0.5	90±4	95.6±0.5	91±2	92±2	94±0	93±0	88.8±1.7
DER2	94.3±0.2	91.4±1.8	95.6±0.5	93±3	92±2	94±0	93±0	88.8±1.7
MAO	94.7±0.2	95.2±0.3	90±3	88±3	91.3±0.4	91±0	90.2±0.2	69±0
UTA	94.5±0.5	91.4±1.8	95.6±0.5	93±3	92±2	94±0	93±0	88.8±1.7
LEE	87.8±0.4	88.9±0.4	89.0±0.5	86±3	87±3	89.10±0.10	89±0	84±0
YOU	94.7±0.2	95.6±0.3	94.9±0.2	93.5±1.6	92±2	92.90±0.17	87±2	90±0

**Tabla 56. Eficiencias para los métodos ARNMF&WD.  
Problema WD.**

**BASE DE DATOS WD (parte 1)**

Método	1	2	3	4	5	6
BL2	95±2	98.75±0.19	98.88±0.16	99.25±0.08	95±3	98.6±0.3
TEKA	95±2	98.75±0.19	98.88±0.16	99.25±0.08	95±3	98.6±0.3
CIB	95±3	97.7±1.0	97.8±0.8	99.06±0.10	98.6±0.2	98.88±0.13
TEKE	95±3	95±3	95±3	95±3	96±3	98.0±0.4
DEV	95±2	98.88±0.13	92±5	98.88±0.08	98.5±0.2	98.4±0.5
DER3	98.7±0.6	96.0±1.5	96.4±1.9	98.25±0.16	99.31±0.06	92±4
TEKB	95±2	98.75±0.19	98.88±0.16	99.25±0.08	95±3	98.6±0.3
TEK	95±2	98.1±0.4	97.8±0.8	98.63±0.13	95±3	98.88±0.13
BL1	95±2	98.75±0.19	98.88±0.16	99.25±0.08	95±3	98.6±0.3
CLO	95±2	98.75±0.19	98.88±0.16	99.25±0.08	95±3	98.6±0.3
PRI	95±2	98.75±0.19	98.88±0.16	99.3±0.08	95±3	98.6±0.3
SAN	95±2	98.75±0.19	98.88±0.16	99.25±0.08	95±3	98.6±0.3
DER2	95±2	98.1±0.4	98.1±0.4	98.6±0.2	98.6±0.2	98.6±0.3
MAO	95±2	98.1±0.4	97.8±0.8	98.3±0.4	98.63±0.18	99.06±0.14
UTA	96.4±1.6	98.69±0.17	92±5	98.4±0.4	92±5	92±5
LEE	95±3	92±5	95±3	98.3±0.5	95±3	98.2±0.4
YOU	98.7±0.6	96.0±1.5	99.1±0.3	96±3	95±3	98.63±0.13

**BASE DE DATOS WD (parte 2)**

Método	7	8	9	10	11	12
BL2	98.63±0.18	97.9±0.8	97.3±1.2	96±3	99.00±0.17	98.6±0.3
TEKA	98.63±0.18	97.9±0.8	97.3±1.2	96±3	99.00±0.17	98.6±0.3
CIB	98.94±0.13	95±3	99.06±0.14	98.5±0.6	95±3	95±3
TEKE	88±5	95±3	98.38±0.19	98.50±0.19	97.9±0.4	95±3
DEV	92±5	98.1±0.9	96±3	99.06±0.14	97.9±1.0	97.6±1.0
DER3	95±3	95±3	94±4	94±3	93±3	91±4
TEKB	98.62±0.18	97.9±0.8	97.3±1.2	96±3	99.00±0.17	98.6±0.3
TEK	98.94±0.13	98.3±0.6	99.25±0.13	98.7±0.3	95±3	95±3
BL1	98.62±0.18	97.9±0.8	97.3±1.2	96±3	99.00±0.17	98.6±0.3
CLO	98.62±0.18	97.9±0.8	97.3±1.2	96±3	99.00±0.17	98.6±0.3
PRI	98.62±0.18	97.9±0.8	97.3±1.2	96±3	99.00±0.17	98.6±0.3
SAN	98.62±0.18	97.9±0.8	97.3±1.2	96±3	99.00±0.17	98.6±0.3
DER2	95±3	98.3±0.3	98.75±0.16	98.50±0.19	95±3	98.50±0.19
MAO	96±3	99.13±0.10	93±4	98.6±0.2	98.69±0.17	98.81±0.17
UTA	96±3	99.38±0.16	92±4	99.1±0.3	95±3	99.13±0.17
LEE	88±5	91±4	91±5	91±4	92±3	94±3
YOU	96.2±0.8	87±5	94±3	93±3	95.2±0.3	89±4

**Tabla 56. Eficiencias para los métodos ARNMF&WD.  
Problema WD. (Continuación).**

**BASE DE DATOS WD (parte 3)**

Método	13	14	15	16	17	18
BL2	97.4±1.7	98.1±0.9	97.2±1.7	92±5	95±3	90±6
TEKA	97.4±1.7	98.1±0.9	97.2±1.7	92±5	95±3	90±6
CIB	99.25±0.13	95±2	98.6±0.2	98.2±0.2	95±2	98.56±0.10
TEKE	97.8±0.5	97.4±0.6	87±5	96.7±0.3	93±3	96.2±0.3
DEV	98.94±0.16	95±3	97.9±0.7	98.94±0.13	98.88±0.13	98.63±0.16
DER3	84±5	96.9±0.3	89±4	91±3	86±5	93.4±1.1
TEKB	97.4±1.7	98.1±0.9	97.2±1.7	92±5	95±3	90±6
TEK	99.38±0.16	95±3	98.6±0.2	98.2±0.2	95±2	98.56±0.10
BL1	97.4±1.7	98.1±0.9	97.2±1.7	92±5	95±3	90±6
CLO	97.4±1.7	98.1±0.9	97.2±1.7	92±5	95±3	90±6
PRI	97.4±1.7	98.1±0.9	97.2±1.7	92±5	95±3	90±6
SAN	97.4±1.7	98.1±0.9	97.2±1.7	92±5	95±3	90±6
DER2	98.31±0.19	98.69±0.17	98.63±0.16	98.3±0.3	93±4	95±3
MAO	98.88±0.08	95±3	88±5	95±3	92±3	98.56±0.10
UTA	97.4±1.7	98.94±0.10	97.2±1.7	98.81±0.17	92±3	88±5
LEE	85±5	91±3	88±4	88±4	91±3	90±3
YOU	92±3	90±3	92.1±0.8	87±3	93.38±0.10	87±4

**BASE DE DATOS WD (parte 4)**

Método	19	20	21	22	23	24
BL2	99.63±0.17	98.6±0.4	97.1±1.7	99.13±0.17	98.3±0.3	95±2
TEKA	99.63±0.17	98.6±0.4	97.1±1.7	99.13±0.17	95±3	95±2
CIB	98.94±0.10	94.6±1.8	98.50±0.19	95±3	91±2	91±2
TEKE	95.6±0.3	89±3	92±3	90±4	92±4	87±5
DEV	99.63±0.17	93±4	97.4±0.5	98.38±0.17	98.6±0.2	98.44±0.17
DER3	93±3	92.31±0.16	85.3±1.8	86±2	82±3	85±2
TEKB	99.63±0.17	98.6±0.4	97.1±1.7	99.13±0.17	98.3±0.3	95±2
TEK	98.94±0.10	94.6±1.8	98.50±0.19	95±3	91±2	98.3±0.2
BL1	99.63±0.17	98.6±0.4	97.1±1.7	99.13±0.17	98.3±0.3	95±2
CLO	99.63±0.17	98.6±0.4	97.1±1.7	99.13±0.17	98.3±0.3	95±2
PRI	99.63±0.17	98.6±0.4	97.1±1.7	99.13±0.17	98.3±0.3	95±2
SAN	99.63±0.17	98.6±0.4	97.1±1.7	99.13±0.17	98.3±0.3	95±2
DER2	95±3	98.2±0.2	95±3	97.9±0.3	91±5	90±4
MAO	95±2	94.6±1.8	98.50±0.19	99.00±0.10	91±2	98.3±0.2
UTA	96±2	98.6±0.4	95.8±1.7	96±2	96.8±1.7	94.2±1.8
LEE	91.2±1.5	80.7±0.5	78.8±1.6	78.1±1.9	77±2	79.3±0.4
YOU	92.5±1.4	86±3	85.3±1.8	82±2	82.8±0.7	77±3

**Tabla 56. Eficiencias para los métodos ARNMF&WD.  
Problema WD. (Continuación).**

**BASE DE DATOS WD (parte 5)**

<b>Método</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
<b>BL2</b>	98.44±0.14	98.31±0.13	98.00±0.16	94.50±0.13	92.3±0.3	71.3±0.3
<b>TEKA</b>	98.44±0.14	98.31±0.13	98.00±0.16	94.50±0.13	92.3±0.3	71.3±0.3
<b>CIB</b>	96.6±1.7	94±2	87±4	89±2	89±3	71.3±0.3
<b>TEKE</b>	92±3	92±3	94.88±0.13	89±2	75.7±1.9	71.3±0.3
<b>DEV</b>	91±3	90±4	92±3	86±4	92.3±0.3	71.3±0.3
<b>DER3</b>	85±3	86±3	87±2	84±3	84±3	78.06±0.06
<b>TEKB</b>	98.44±0.14	98.31±0.13	98.00±0.16	94.50±0.13	92.3±0.3	71.3±0.3
<b>TEK</b>	96.6±1.7	94±2	87±4	89±2	89±3	71.3±0.3
<b>BL1</b>	98.44±0.14	98.31±0.13	98.00±0.16	94.50±0.13	92.3±0.3	71.3±0.3
<b>CLO</b>	98.44±0.14	98.31±0.13	98.00±0.16	94.50±0.13	92.3±0.3	71.3±0.3
<b>PRI</b>	98.44±0.14	98.31±0.13	98.00±0.16	94.50±0.13	92.3±0.3	71.3±0.3
<b>SAN</b>	98.44±0.14	98.31±0.13	98.00±0.16	94.50±0.13	92.3±0.3	71.3±0.3
<b>DER2</b>	91±3	91±2	94.88±0.13	92±3	89±3	71.3±0.3
<b>MAO</b>	95±3	94±2	98.50±0.14	89±2	89±3	71.3±0.3
<b>UTA</b>	93.9±0.2	90±3	87±4	86±4	75.7±1.9	71.3±0.3
<b>LEE</b>	78.1±1.6	76±2	78.56±0.13	76.8±1.4	76.9±1.4	62.8±0.5
<b>YOU</b>	81.7±0.7	80.9±1.8	74.2±1.2	75.5±1.2	68.1±0.9	64.1±0.7

## II.5. Comparación entre métodos ARNMF y ACE

Tabla 57. Comparación entre métodos ARNMF y ACE.

		Método BL2
TEKA	Mejor BL2: Iguales: Peor BL2:	LE AB, BL, BN, BU, D1, GL, HE, M1, M2, M3, PI, VO, CR, WD
CIB	Mejor BL2: Iguales: Peor BL2:	AB, BU, CR, GL, HE, M1, PI D1, M2, M3 BL, BN, VO, WD
TEKE	Mejor BL2: Iguales: Peor BL2:	BU, D1, GL, HE, PI, VO, WD M2, M3, AB, BL, LE, M1, M2
TEKC	Mejor BL2: Iguales: Peor BL2:	AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2 BL, BN
DEV	Mejor BL2: Iguales: Peor BL2:	BN, CR, WD AB, BU, D1, GL, HE, M1, M2, M3, PI, VO BL, LE
DER3	Mejor BL2: Iguales: Peor BL2:	BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2 AB, BL, BN
TEKB	Mejor BL2: Iguales: Peor BL2:	BN, GL, HE, VO, WD AB, BU, D1, M1, M2, M3, PI BL, CR, LE
TEK	Mejor BL2: Iguales: Peor BL2:	AB, BN, BU, CR, D1, GL, HE, PI, VO, WD M1, M2, M3 BL, LE
BL1	Mejor BL2: Iguales: Peor BL2:	BN, CR, D1, HE, LE, PI, VO, WD AB, BU, GL, M1, M2, M3 BL
CLO	Mejor BL2: Iguales: Peor BL2:	BN, CR, D1, HE, LE, PI, VO, WD BU, GL, M1, M2, M3 AB, BL
PRI	Mejor BL2: Iguales: Peor BL2:	BN, CR, D1, HE, LE, PI, VO, WD AB, BU, GL, M1, M2, M3 BL
SAN	Mejor BL2: Iguales: Peor BL2:	BN, CR, HE, PI, VO, WD AB, BU, D1, GL, LE, M1, M2, M3 BL
DER2	Mejor BL2: Iguales: Peor BL2:	BN, CR, GL, LE, PI, VO BU, D1, M1, M2, M3 AB, BL, HE, WD
MAO	Mejor BL2: Iguales: Peor BL2:	CR, HE, PI BL, BU, D1, GL, M1, M2, M3 AB, BN, LE, VO, WD
UTA	Mejor BL2: Iguales: Peor BL2:	CR, HE AB, BU, D1, GL, M1, M2, M3, PI BL, BN, LE, VO, WD
LEE	Mejor BL2: Iguales: Peor BL2:	AB, BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2 BL
BOW	Mejor BL2: Iguales: Peor BL2:	AB, BN, BU, CR, D1, HE, LE, M1, PI, VO, WD GL, M2, M3 BL
YOU	Mejor BL2: Iguales: Peor BL2:	CR, HE, PI, WD M1, M2 BN, LE, VO

Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).

		Método BL2
BA 0.5	Mejor BL2: Iguales: Peor BL2:	BU, CR, HE, LE, VO, WD AB, D1, GL, M2, M3, PI BL, BN, M1
BA 0.6	Mejor BL2: Iguales: Peor BL2:	BU, CR, HE, LE, VO, WD AB, D1, GL, M2, M3, PI BL, BN, M1
BA 0.7	Mejor BL2: Iguales: Peor BL2:	BU, CR, HE, LE, VO, WD AB, D1, GL, M2, M3, PI BL, BN, M1
BA 0.8	Mejor BL2: Iguales: Peor BL2:	BU, CR, HE, LE, VO, WD AB, D1, GL, M2, M3, PI BL, BN, M1
BA 0.9	Mejor BL2: Iguales: Peor BL2:	AB, BU, CR, HE, LE, VO, WD D1, GL, M2, M3, PI BL, BN, M1
BA 1	Mejor BL2: Iguales: Peor BL2:	AB, BU, CR, HE, LE, VO, WD D1, GL, M2, M3, PI BL, BN, M1
CHI	Mejor BL2: Iguales: Peor BL2:	BN, BU, CR, HE, LE, VO, WD D1, GL, M1, M2, M3, PI AB, BL
SET	Mejor BL2: Iguales: Peor BL2:	AB, BN, CR, GL, LE, VO, WD D1, HE, M1, M2, M3, PI BL, BU
GD_DS	Mejor BL2: Iguales: Peor BL2:	GL, HE, VO, WD D1, M2, M3 AB, BL, BU, M1, PI
RLF	Mejor BL2: Iguales: Peor BL2:	CR, HE, LE, M1, PI, VO M2, M3 BN, BU, WD
SCH	Mejor BL2: Iguales: Peor BL2:	BN, CR, D1, GL, HE, LE, M1, M3, PI, VO BU, M2 AB, BL, WD
FUZ	Mejor BL2: Iguales: Peor BL2:	BU, CR, D1, HE, M3, PI, WD AB, BL, M1, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método TEKA</b>		
<b>CIB</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	AB, BU, CR, GL, HE, PI BL, D1, M1, M2, M3, WD BN, LE, VO
<b>TEKE</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BU, D1, GL, HE, PI, VO BL, M3 AB, M1, M2, WD
<b>TEKC</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	AB, BU, CR, D1, GL, HE, M1, M3, PI, VO, WD BL, M2 BN, LE
<b>DEV</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	CR, WD AB, BL, BN, BU, D1, GL, HE, M1, M2, M3, PI, VO LE
<b>DER3</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BU, CR, D1, GL, HE, M1, M3, PI, VO, WD BL, M2 AB, BN, LE
<b>TEKB</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BN, CR, GL, HE, LE, VO AB, BL, BU, D1, M1, M2, M3, PI WD
<b>TEK</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	AB, BN, BU, CR, D1, GL, HE, PI, VO, WD BL, M1, M2, M3 LE
<b>BL1</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BN, CR, D1, HE, PI, VO AB, BL, BU, M1, M2, M3, WD LE, WD
<b>CLO</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BN, CR, D1, HE, PI, VO BL, BU, GL, M1, M2, M3 AB, LE, WD
<b>PRI</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BN, CR, D1, HE, PI, VO AB, BL, BU, GL, M1, M2, M3 LE, WD
<b>SAN</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BL, BN, CR, HE, LE, PI, VO AB, BU, D1, GL, M1, M2, M3 WD
<b>DER2</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BN, CR, GL, HE, PI, VO, WD BL, BU, D1, M1, M2, M3 AB, HE, LE
<b>MAO</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	CR, HE, PI BL, BU, D1, GL, M1, M2, M3, WD AB, BN, LE, VO
<b>UTA</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	CR, HE AB, BL, BU, D1, GL, M1, M2, M3, PI BN, LE, VO, WD
<b>LEE</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	AB, BL, BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2
<b>BOW</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	AB, BN, BU, CR, D1, HE, M1, PI, VO, WD BL, GL, M2, M3 LE
<b>YOU</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	CR, HE, LE, PI, WD M1, M2 BN, VO



**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

		<b>Método TEKA</b>
<b>BA 0.5</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BU, CR, HE, VO, WD AB, BL, D1, GL, M2, M3, PI BN, LE, M1
<b>BA 0.6</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BU, CR, HE, VO, WD AB, BL, D1, GL, M2, M3, PI BN, LE, M1
<b>BA 0.7</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BU, CR, HE, VO, WD AB, BL, D1, GL, M2, M3, PI BN, LE, M1
<b>BA 0.8</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BU, CR, HE, VO, WD AB, BL, D1, GL, M2, M3, PI BN, LE, M1
<b>BA 0.9</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	AB, BU, CR, HE, VO, WD BL, D1, GL, M2, M3, PI BN, LE, M1
<b>BA 1</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	AB, BU, CR, HE, VO, WD BL, D1, GL, M2, M3, PI BN, LE, M1
<b>CHI</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BN, BU, CR, HE, LE, VO, WD BL, D1, GL, M1, M2, M3, PI AB
<b>SET</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	AB, BN, CR, GL, LE, VO, WD BL, D1, HE, M1, M2, M3, PI BU
<b>GD_DS</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	GL, HE, VO, WD BL, D1, M2, M3 AB, BU, M1, PI
<b>RLF</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	CR, HE, M1, PI, VO M2, M3 BN, BU, LE
<b>SCH</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BL, BN, CR, D1, GL, HE, M1, M3, PI, VO, WD BU, M2 AB, LE
<b>FUZ</b>	<b>Mejor TEKA:</b> <b>Iguales:</b> <b>Peor TEKA:</b>	BU, CR, D1, HE, M3, PI, WD BL AB, M1, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método CIB</b>		
<b>TEKE</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BU, D1, HE, PI, VO BL, M3 AB, GL, M1, M2, WD
<b>TEKC</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	AB, BU, CR, D1, GL, LE, M1, M3, PI, VO, WD BL, HE, M2 BN
<b>DEV</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BN, LE, VO, WD BL, D1, M2, M3 AB, BU, CR, GL, HE, M1, PI
<b>DER3</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD BL, M2 AB
<b>TEKB</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BN, LE, VO BL, D1, M1, M2, M3 AB, BU, CR, GL, HE, PI, WD
<b>TEK</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BN, D1, LE, PI, VO AB, BL, BU, M2, M3 CR, GL, HE, M1, WD
<b>BL1</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BN, D1, VO BL, LE, M2, M3 AB, BU, CR, GL, HE, M1, PI, WD
<b>CLO</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BN, D1, VO BL, LE, M2, M3 AB, BU, CR, GL, HE, M1, PI, WD
<b>PRI</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BN, D1, VO BL, LE, M2, M3 AB, BU, CR, GL, HE, M1, PI, WD
<b>SAN</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BL, BN, BU, GL, LE, VO D1, M2, M3 AB, CR, HE, M1, PI, WD
<b>DER2</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BN, LE, WD BL, D1, M1, M2, M3 AB, BU, CR, GL, HE, PI, VO
<b>MAO</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	LE, PI BL, BN, D1, HE, M1, M2, M3, WD AB, BU, CR, GL, VO
<b>UTA</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	HE BL, D1, M1, M2, M3 AB, BN, BU, CR, GL, LE, PI, VO, WD
<b>LEE</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	AB, BL, BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2
<b>BOW</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	AB, GL, M2 BL, M3 BU, D1, HE, M1, PI, VO, WD
<b>YOU</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	LE M1, M2, VO BN, CR, HE, PI, WD

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método CIB</b>		
<b>BA 0.5</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BU, HE, LE, VO, WD BL, M2, M3 AB, BN, CR, D1, GL, M1, PI
<b>BA 0.6</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BU, HE, LE, VO, WD BL, M2, M3 AB, BN, CR, D1, GL, M1, PI
<b>BA 0.7</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BU, HE, LE, VO, WD BL, M2, M3 AB, BN, CR, D1, GL, M1, PI
<b>BA 0.8</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BU, HE, LE, VO, WD BL, M2, M3 AB, BN, CR, D1, GL, M1, PI
<b>BA 0.9</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BU, HE, LE, VO, WD BL, M2, M3 AB, BN, CR, D1, GL, M1, PI
<b>BA 1</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BU, HE, LE, VO, WD BL, M2, M3 AB, BN, CR, D1, GL, M1, PI
<b>CHI</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BN, BU, HE, LE, VO, WD BL, M1, M2, M3 AB, CR, D1, GL, PI
<b>SET</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	AB, BN, GL, LE, VO, WD BL, M1, M2, M3 BU, CR, D1, HE, PI
<b>GD_DS</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	GL, VO, WD BL, M2, M3 AB, BU, D1, HE, M1, PI
<b>RLF</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BN, HE, LE, M1, PI, VO M2, M3 BU, CR, WD
<b>SCH</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	BL, BN, CR, D1, GL, HE, LE, M1, M3, PI, VO M2 AB, BU, WD
<b>FUZ</b>	<b>Mejor CIB:</b> <b>Iguales:</b> <b>Peor CIB:</b>	CR, D1, HE, M3, PI, WD BL AB, BU, M1, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

	<b>Método TEKE</b>	
<b>TEKC</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, BU, D1, GL, M1, M2, M3, WD BL, M2 HE, PI, VO
<b>DEV</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, M1, M2 BL, M3 BU, D1, GL, HE, PI, VO, WD
<b>DER3</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, BU, GL, HE, M1, M2, M3, WD BL D1, PI, VO
<b>TEKB</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, HE, M1, M2 BL, M3 BU, D1, GL, PI, VO, WD
<b>TEK</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, M1, M2 BL, M3 BU, D1, GL, HE, PI, VO, WD
<b>BL1</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	HE, M1, M2 BL, M3 AB, BU, D1, GL, PI, VO, WD
<b>CLO</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	HE, M1, M2 BL, M3 AB, BU, D1, GL, PI, VO, WD
<b>PRI</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	M1, M2 BL, M3 AB, BU, D1, GL, HE, PI, VO, WD
<b>SAN</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, BL, BU, GL, HE, M1, M2 M3 D1, PI, VO, WD
<b>DER2</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	M1, M2, WD BL, M3 AB, BU, D1, GL, HE, PI, VO
<b>MAO</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, M1, M2, WD BL, M3 BU, D1, GL, HE, PI, VO
<b>UTA</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	HE, M1, M2 BL, M3 AB, BU, D1, GL, PI, VO, WD
<b>LEE</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, BL, BU, D1, GL, HE, M1, M2, M3, PI, VO, WD
<b>BOW</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, GL, M2 BL, M3 BU, D1, HE, M1, PI, VO, WD
<b>YOU</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	M1, M2  HE, PI, VO, WD

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método TEKE</b>		
<b>BA 0.5</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, HE, M2, WD BL, M3 BU, D1, GL, M1, PI, VO
<b>BA 0.6</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, HE, M2, WD BL, M3 BU, D1, GL, M1, PI, VO
<b>BA 0.7</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, HE, M2, WD BL, M3 BU, D1, GL, M1, PI, VO
<b>BA 0.8</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, HE, M2, WD BL, M3 BU, D1, GL, M1, PI, VO
<b>BA 0.9</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, HE, M2, WD BL, M3 BU, D1, GL, M1, PI, VO
<b>BA 1</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, HE, M2, WD BL, M3 BU, D1, GL, M1, PI, VO
<b>CHI</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, HE, M1, M2, WD BL, M3 BU, D1, GL, PI, VO
<b>SET</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, M1, M2, WD BL, M3 BU, D1, GL, HE, PI, VO
<b>GD_DS</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	GL, M2, WD BL, M3 AB, BU, D1, HE, M1, PI, VO
<b>RLF</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	M1, M2 M3 BU, HE, PI, VO, WD
<b>SCH</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, BL, GL, HE, M1, M2, M3, PI, VO, WD BU, D1
<b>FUZ</b>	<b>Mejor TEKE:</b> <b>Iguales:</b> <b>Peor TEKE:</b>	AB, M1, M2, M3, WD BL, D1 BU, HE, PI

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método TEKC</b>		
<b>DEV</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN, WD BL, M2 AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO
<b>DER3</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	AB, BN, CR, GL, HE, M1, M3, PI, VO BL, BU, M2 D1, LE, VO
<b>TEKB</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN BL, M2 AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>TEK</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN BL, M2 AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>BL1</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN BL, M2 AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>CLO</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN BL, M2 AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>PRI</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN BL, M2 AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>SAN</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BL, BN M2 AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>DER2</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN BL, M2 AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>MAO</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN AB, BL, HE, M2 BU, CR, D1, GL, LE, M1, M3, PI, VO, WD
<b>UTA</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	HE BL, BN, M2 AB, BU, CR, D1, GL, LE, M1, M3, PI, VO, WD
<b>LEE</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	AB, BL, BN, CR, GL, HE, LE, M1, M3, PI, VO, WD BU, D1, M2
<b>BOW</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN BL, M2 AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>YOU</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN M2 CR, HE, LE, M1, PI, VO, WD

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método TEKC</b>		
<b>BA 0.5</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN, HE, WD BL, BU, M2 AB, CR, D1, GL, LE, M1, M3, PI, VO
<b>BA 0.6</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN, HE, VO, WD BL, BU, M2 AB, CR, D1, GL, LE, M1, M3, PI
<b>BA 0.7</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN, HE, VO, WD BL, BU, M2 AB, CR, D1, GL, LE, M1, M3, PI
<b>BA 0.8</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BU, HE, VO, WD BL, BU, M2 AB, CR, D1, GL, LE, M1, M3, PI
<b>BA 0.9</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN, HE, WD BL, BU, M2, VO AB, CR, D1, GL, LE, M1, M3, PI
<b>BA 1</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN, HE, WD BL, BU, M2 AB, CR, D1, GL, LE, M1, M3, PI, VO
<b>CHI</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN, HE, M1, WD AB, BL, M2 BU, CR, D1, GL, LE, M3, PI, VO
<b>SET</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN, M1, WD BL, GL, M2 AB, BU, CR, D1, HE, LE, M3, PI, VO
<b>GD_DS</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	GL, WD BL, M2 AB, BU, D1, M1, M3, PI, VO
<b>RLF</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BN, HE M2 BU, CR, LE, M1, M3, PI, VO, WD
<b>SCH</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	BL, BN, CR, GL, HE, M1, M3, PI, VO M2 AB, BU, D1, LE, WD
<b>FUZ</b>	<b>Mejor TEKC:</b> <b>Iguales:</b> <b>Peor TEKC:</b>	AB, CR, HE, M3, WD BL, D1, M1 BU, M2, PI

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método DEV</b>		
<b>DER3</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD BL, M2 AB, BN
<b>TEKB</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	BN, GL, HE, LE, VO AB, BL, BU, D1, M1, M2, M3, PI CR, WD
<b>TEK</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	AB, BN, BU, D1, GL, HE, PI, VO, WD BL, CR, LE, M1, M2, M3
<b>BL1</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	BN, D1, HE, LE, PI, VO AB, BL, BU, CR, GL, M1, M2, M3 WD
<b>CLO</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	BN, D1, HE, LE, PI, VO BL, BU, CR, GL, M1, M2, M3 AB, WD
<b>PRI</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	BN, D1, HE, LE, PI, VO AB, BL, BU, CR, GL, M1, M2, M3 WD
<b>SAN</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	BN, HE, LE, PI, VO AB, BU, CR, D1, GL, M1, M2, M3 BL, WD
<b>DER2</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	BN, GL, LE, PI, VO BL, BU, D1, M1, M2, M3 AB, CR, HE, WD
<b>MAO</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	CR, D1, HE, PI BL, BU, GL, M1, M2, M3 AB, BN, LE, VO, WD
<b>UTA</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	HE, LE AB, BL, BU, D1, GL, M1, M2, M3, PI BN, CR, VO, WD
<b>LEE</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	AB, BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2 BL
<b>BOW</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	AB, BN, BU, D1, HE, LE, M1, PI, VO BL, GL, M2, M3 CR, WD
<b>YOU</b>	<b>Mejor DEV:</b> <b>Iguales:</b> <b>Peor DEV:</b>	HE, LE, PI, WD CR, M1, M2 BN, VO



**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

		Método DEV
<b>BA 0.5</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	BU, HE, LE, VO, WD AB, BL, D1, GL, M2, M3, PI BN, CR, M1
<b>BA 0.6</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	BU, HE, LE, VO, WD AB, BL, D1, GL, M2, M3, PI BN, CR, M1
<b>BA 0.7</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	BU, HE, LE, VO, WD AB, BL, D1, GL, M2, M3, PI BN, CR, M1
<b>BA 0.8</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	BU, HE, LE, VO, WD AB, BL, D1, GL, M2, M3, PI BN, CR, M1
<b>BA 0.9</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	AB, BU, HE, LE, VO, WD BL, D1, GL, M2, M3, PI BN, CR, M1
<b>BA 1</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	AB, BU, HE, LE, VO, WD BL, D1, GL, M2, M3, PI BN, CR, M1
<b>CHI</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	BN, BU, HE, LE, VO, WD BL, CR, D1, GL, M1, M2, M3, PI AB
<b>SET</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	AB, GL, LE, VO, WD BL, D1, HE, M1, M2, M3, PI BN, BU, CR
<b>GD_DS</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	GL, HE, VO, WD BL, D1, M2, M3 AB, BU, M1, PI
<b>RLF</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	HE, LE, M1, PI, VO M2, M3 BN, BU, CR, WD
<b>SCH</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	BL, BN, CR, D1, GL, HE, LE, M1, M3, PI, VO BU, M2 AB, WD
<b>FUZ</b>	<b>Mejor DEV:</b> <b>Iguals:</b> <b>Peor DEV:</b>	BU, CR, D1, HE, M3, PI, WD BL AB, M1, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

		MétodoDER3
<b>TEKB</b>	Mejor DER3: Iguales: Peor DER3:	AB, BN, LE BL, M2 BU, CR, D1, GL, HE, M1, M3, PI, VO, WD
<b>TEK</b>	Mejor DER3: Iguales: Peor DER3:	AB, BN BL, M2 BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>BL1</b>	Mejor DER3: Iguales: Peor DER3:	AB, BN BL, HE, M2 BU, CR, D1, GL, LE, M1, M3, PI, VO, WD
<b>CLO</b>	Mejor DER3: Iguales: Peor DER3:	BN BL, M2 AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>PRI</b>	Mejor DER3: Iguales: Peor DER3:	AB, BN BL, HE, M2 BU, CR, D1, GL, LE, M1, M3, PI, VO, WD
<b>SAN</b>	Mejor DER3: Iguales: Peor DER3:	AB, BL, BN, LE M2 BU, CR, D1, GL, HE, M1, M3, PI, VO, WD
<b>DER2</b>	Mejor DER3: Iguales: Peor DER3:	BN BL, M2 AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>MAO</b>	Mejor DER3: Iguales: Peor DER3:	BL, M2 AB, BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>UTA</b>	Mejor DER3: Iguales: Peor DER3:	AB BL, M2 BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>LEE</b>	Mejor DER3: Iguales: Peor DER3:	BL, CR, D1, GL, HE, LE, M1, M3, PI BU, M2 AB, BN, VO, WD
<b>BOW</b>	Mejor DER3: Iguales: Peor DER3:	AB, BN BL, M2 BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>YOU</b>	Mejor DER3: Iguales: Peor DER3:	LE M2 BN, CR, HE, M1, PI, VO, WD

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método DER3</b>		
<b>BA 0.5</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	AB BL, BU, LE, M2, WD BN, CR, D1, GL, HE, M1, M3, PI, VO
<b>BA 0.6</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	AB, HE BL, BU, LE, M2, WD BN, CR, D1, GL, M1, M3, PI, VO
<b>BA 0.7</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	AB, HE BL, BU, LE, M2, WD BN, CR, D1, GL, M1, M3, PI, VO
<b>BA 0.8</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	AB, HE BL, BU, LE, M2, WD BN, CR, D1, GL, M1, M3, PI, VO
<b>BA 0.9</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	AB, HE BL, BU, LE, M2, WD BN, CR, D1, GL, M1, M3, PI, VO
<b>BA 1</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	AB, HE BL, BU, LE, M2, WD BN, CR, D1, GL, M1, 3, PI, VO
<b>CHI</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	LE, WD BL, HE, M2 AB, BN, BU, CR, D1, GL, M1, M3, PI, VO
<b>SET</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	AB, BN, LE BL, M2 BU, CR, D1, GL, HE, M1, M3, PI, VO, WD
<b>GD_DS</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	GL BL, M2 AB, BU, D1, HE, M1, M3, PI, VO, WD
<b>RLF</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	M2 BN, BU, CR, HE, LE, M1, M3, PI, VO, WD
<b>SCH</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	BL, CR, D1, GL, HE, M1, M3, PI, VO M2 AB, BN, BU, LE, WD
<b>FUZ</b>	<b>Mejor DER3:</b> <b>Iguales:</b> <b>Peor DER3:</b>	CR, D1, HE, PI BL AB, BU, M1, M2, M3, WD

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método TEKB</b>		
<b>TEK</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	AB, BN, BU, CR, D1, HE, LE, PI, VO, WD BL, GL, M1, M2, M3
<b>BL1</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BN, CR, D1, HE, PI, VO AB, BL, BU, M1, M2, M3, WD GL, LE
<b>CLO</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BN, CR, D1, PI, VO BL, BU, HE, M1, M2, M3, WD AB, GL, LE
<b>PRI</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BN, CR, D1, PI, VO AB, BL, BU, HE, M1, M2, M3, WD GL, LE
<b>SAN</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BL, BN, CR, PI, VO AB, BU, D1, HE, LE, M1, M2, M3, WD GL
<b>DER2</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BN, CR, PI, WD BL, BU, D1, GL, M1, M2, M3 AB, HE, LE, VO
<b>MAO</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	CR, HE, PI, WD BL, BU, D1, M1, M2, M3 AB, BN, GL, LE, VO
<b>UTA</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	AB, BL, BU, D1, M1, M2, M3, PI BN, CR, GL, HE, LE, VO, WD
<b>LEE</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	AB, BL, BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2
<b>BOW</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	AB, BN, BU, D1, M1, PI BL, M2, M3, WD CR, GL, HE, LE, VO
<b>YOU</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	CR, HE, PI, WD LE, M1, M2 BN, VO

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método TEKB</b>		
<b>BA 0.5</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BU, CR, LE, VO, WD AB, BL, D1, HE, M2, M3, PI BN, GL, M1
<b>BA 0.6</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BU, CR, HE, LE, VO, WD AB, BL, D1, M2, M3, PI BN, GL, M1
<b>BA 0.7</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BU, CR, HE, LE, VO, WD AB, BL, D1, M2, M3, PI BN, GL, M1
<b>BA 0.8</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BU, CR, HE, LE, VO, WD AB, BL, D1, M2, M3, PI BN, GL, M1
<b>BA 0.9</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	AB, BU, CR, HE, LE, VO, WD BL, D1, M2, M3, PI BN, GL, M1
<b>BA 1</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	AB, BU, CR, HE, LE, VO, WD BL, D1, M2, M3, PI BN, GL, M1
<b>CHI</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BU, CR, HE, LE, VO, WD BL, D1, M1, M2, M3, PI AB, BN, GL
<b>SET</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	AB, CR, GL, LE, VO, WD BL, D1, M1, M2, M3, PI BN, BU, HE
<b>GD_DS</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	GL, HE, VO, WD BL, D1, M2, M3 AB, BU, M1, PI
<b>RLF</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	CR, HE, LE, M1, PI, WD M2, M3 BN, BU, VO
<b>SCH</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BL, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD BU, M2 AB, BN
<b>FUZ</b>	<b>Mejor TEKB:</b> <b>Iguales:</b> <b>Peor TEKB:</b>	BU, CR, D1, HE, M3, PI, WD BL AB, M1, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

	<b>MétodoTEK</b>	
<b>BL1</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BN, HE BL, CR, D1, M1, M2, M3 AB, BU, GL, LE, PI, VO, WD
<b>CLO</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BN, VO, BL, CR, D1, M1, M2, M3 AB, BU, GL, HE, LE, PI, WD
<b>PRI</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BN, HE BL, CR, D1, M1, M2, M3 AB, BU, GL, LE, PI, VO, WD
<b>SAN</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BN CR, M1, M2, M3 AB, BL, BU, D1, GL, HE, LE, PI, VO, WD
<b>DER2</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	WD BL, GL, M1, M2, M3 AB, BN, BU, CR, D1, HE, LE, PI, VO
<b>MAO</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	CR, HE, WD BL, M1, M2, M3, PI AB, BN, BU, D1, GL, LE, VO
<b>UTA</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	HE BL, M1, M2, M3 AB, BN, BU, CR, D1, GL, LE, PI, VO, WD
<b>LEE</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	AB, BL, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2 BN
<b>BOW</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BN, HE, M1 AB, BL, BU, D1, M2, M3, VO CR, GL, LE, PI, WD
<b>YOU</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	WD CR, HE, M1, M2 BN, LE, PI, VO

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

		Método TEK
<b>BA 0.5</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BU, HE, WD BL, M2, M3 AB, BN, CR, D1, GL, LE, M1, PI, VO
<b>BA 0.6</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BU, HE, VO, WD BL, M2, M3 AB, BN, CR, D1, GL, LE, M1, PI
<b>BA 0.7</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BU, HE, VO, WD BL, M2, M3 AB, BN, D1, GL, LE, M1, PI
<b>BA 0.8</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BU, HE, VO, WD BL, M2, M3 AB, BN, CR, D1, GL, LE, M1, PI
<b>BA 0.9</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BU, HE, VO, WD BL, M2, M3 AB, BN, CR, D1, GL, LE, M1, PI
<b>BA 1</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BU, HE, WD BL, M2, M3 AB, BN, CR, D1, GL, LE, M1, PI, VO
<b>CHI</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BU, HE, LE, VO, WD BL, CR, M1, M2, M3 AB, BN, D1, GL, PI
<b>SET</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	AB, GL, LE, VO, WD BL, M1, M2, M3 BN, BU, CR, D1, HE, PI
<b>GD_DS</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	GL, HE, WD BL, M2, M3 AB, BU, D1, M1, PI, VO
<b>RLF</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	HE, LE, M1, VO, WD M2, M3, PI BN, BU, CR
<b>SCH</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	BL, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2 AB, BN, BU
<b>FUZ</b>	<b>Mejor TEK:</b> <b>Iguales:</b> <b>Peor TEK:</b>	CR, D1, HE, M3, PI, WD BL AB, BU, M1, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

	<b>MétodoBL1</b>	
<b>CLO</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	BL, BN, BU, CR, D1, GL, LE, M1, M2, M3, PI, VO, WD AB, HE
<b>PRI</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	BN AB, BL, BU, CR, D1, GL, HE, LE, M1, M2, M3, PI, VO, WD
<b>SAN</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	BL, HE, LE, PI AB, BN, BU, CR, GL, M1, M2, M3, VO, WD D1
<b>DER2</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	GL, WD BL, BU, LE, M1, M2, M3, PI BN, CR, D1, HE, VO
<b>MAO</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	AB, CR, HE, PI, WD BL, BU, D1, GL, M1, M2, M3 BN, LE, VO
<b>UTA</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	AB, BL, BU, GL, LE, M1, M2, M3 BN, CR, D1, HE, PI, VO, WD
<b>LEE</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	AB, BL, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2 BN
<b>BOW</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	AB, BU, M1, PI BL, BN, D1, GL, M2, M3, WD CR, HE, LE, VO
<b>YOU</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	PI, WD CR, M1, M2 BN, HE, LE, VO



Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).

Método BL1		
<b>BA 0.5</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	BU, LE, VO, WD AB, BL, GL, M2, M3 BN, CR, D1, HE, M1, PI
<b>BA 0.6</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	BU, HE, LE, VO, WD AB, BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 0.7</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	BU, HE, LE, VO, WD AB, BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 0.8</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	BU, HE, LE, VO, WD AB, BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 0.9</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 1</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>CHI</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	AB, BU, LE, VO, WD BL, CR, GL, HE, M1, M2, M3 BN, D1, PI
<b>SET</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	AB, GL, LE, VO, WD BL, M1, M2, M3 BN, BU, CR, D1, HE, PI
<b>GD_DS</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	GL, VO, WD BL, M2, M3 AB, BU, D1, HE, M1, PI
<b>RLF</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	HE, M1, PI, WD M2, M3 BN, BU, CR, LE, VO
<b>SCH</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	BL, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD BU, M2 AB, BN
<b>FUZ</b>	<b>Mejor BL1:</b> <b>Iguales:</b> <b>Peor BL1:</b>	AB, BU, CR, D1, HE, M3, PI, WD BL M1, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

	<b>MétodoCLO</b>	
<b>PRI</b>	<b>Mejor CLO:</b> <b>Iguals:</b> <b>Peor CLO:</b>	AB, BN, HE BL, BU, CR, D1, GL, LE, M1, M2, M3, PI, VO, WD
<b>SAN</b>	<b>Mejor CLO:</b> <b>Iguals:</b> <b>Peor CLO:</b>	AB, LE, PI BN, BU, CR, GL, HE, M1, M2, M3, VO, WD BL, D1
<b>DER2</b>	<b>Mejor CLO:</b> <b>Iguals:</b> <b>Peor CLO:</b>	AB, GL, WD BL, BU, LE, M1, M2, M3, PI BN, CR, D1, HE, VO
<b>MAO</b>	<b>Mejor CLO:</b> <b>Iguals:</b> <b>Peor CLO:</b>	AB, CR, HE, PI, WD BL, BU, D1, GL, M1, M2, M3 BN, LE, VO
<b>UTA</b>	<b>Mejor CLO:</b> <b>Iguals:</b> <b>Peor CLO:</b>	AB BL, BU, GL, LE, M1, M2, M3 BN, CR, D1, HE, PI, VO, WD
<b>LEE</b>	<b>Mejor CLO:</b> <b>Iguals:</b> <b>Peor CLO:</b>	AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2 BL, BN
<b>BOW</b>	<b>Mejor CLO:</b> <b>Iguals:</b> <b>Peor CLO:</b>	AB, BU, M1, PI BL, BN, D1, GL, M2, M3, WD CR, HE, LE, VO
<b>YOU</b>	<b>Mejor CLO:</b> <b>Iguals:</b> <b>Peor CLO:</b>	HE, PI, WD CR, M1, M2 BN, LE, VO

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

		Método CLO
<b>BA 0.5</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	AB, BU, LE, WD BL, GL, HE, M2, M3 BN, CR, D1, M1, PI, VO
<b>BA 0.6</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 0.7</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 0.8</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 0.9</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 1</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	AB, BU, HE, LE, WD BL, GL, M2, M3 BN, CR, D1, M1, PI, VO
<b>CHI</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	AB, BU, HE, LE, WD BL, CR, GL, M1, M2, M3 BN, D1, PI, VO
<b>SET</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	AB, GL, LE, WD BL, M1, M2, M3 BN, BU, CR, D1, HE, PI, VO
<b>GD_DS</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	AB, GL, HE, VO, WD BL, M2, M3 BU, D1, M1, PI
<b>RLF</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	HE, M1, PI, WD M2, M3 BN, BU, CR, LE, VO
<b>SCH</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	AB, BL, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD BU, M2 BN
<b>FUZ</b>	<b>Mejor CLO:</b> <b>Iguales:</b> <b>Peor CLO:</b>	AB, BU, CR, D1, HE, M3, PI, WD BL M1, M2

II. APÉNDICES DEL CAPÍTULO 3

Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).

	<b>Método PRI</b>	
<b>SAN</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	BN, LE, PI AB, BU, CR, GL, M1, M2, M3, VO, WD BL, D1, HE
<b>DER2</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	AB, GL, WD BL, BU, LE, M1, M2, M3, PI BN, CR, D1, HE, VO
<b>MAO</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	AB, CR, HE, PI, WD BL, BU, D1, GL, M1, M2, M3 BN, LE, VO
<b>UTA</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	AB, BL, BU, GL, LE, M1, M2, M3 BN, CR, D1, HE, PI, VO, WD
<b>LEE</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2 BL, BN
<b>BOW</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	AB, BU, M1, PI BL, D1, GL, M2, M3, WD BN, CR, HE, LE, VO
<b>YOU</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	PI, WD CR, M1, M2 BN, HE, LE, VO
<b>BA 0.5</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	BU, LE, VO, WD AB, BL, GL, M2, M3 BN, CR, D1, HE, M1, PI
<b>BA 0.6</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	BU, HE, LE, VO, WD AB, BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 0.7</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	BU, HE, LE, VO, WD AB, BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 0.8</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	BU, HE, LE, VO, WD AB, BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 0.9</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>BA 1</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
<b>CHI</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	AB, BU, LE, VO, WD BL, CR, GL, HE, M1, M2, M3 BN, D1, PI
<b>SET</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	AB, GL, LE, VO, WD BL, M1, M2, M3 BN, BU, CR, D1, HE, PI
<b>GD_DS</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	GL, VO, WD BL, M2, M3 AB, BU, D1, HE, M1, PI
<b>RLF</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	HE, M1, PI, WD M2, M3 BN, BU, CR, LE, VO
<b>SCH</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	BL, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD BU, M2 AB, BN
<b>FUZ</b>	<b>Mejor PRI:</b> <b>Iguales:</b> <b>Peor PRI:</b>	AB, BU, CR, D1, HE, M3, PI, WD BL M1, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

	<b>MétodoSAN</b>	
<b>DER2</b>	<b>Mejor SAN:</b>	GL, WD
	<b>Iguales:</b>	BU, D1, M1, M2, M3
	<b>Peor SAN:</b>	AB, BL, BN, CR, HE, LE, PI, VO
<b>MAO</b>	<b>Mejor SAN:</b>	CR, HE, PI, WD
	<b>Iguales:</b>	BU, D1, GL, M1, M2, M3
	<b>Peor SAN:</b>	AB, BL, BN, LE, VO
<b>UTA</b>	<b>Mejor SAN:</b>	
	<b>Iguales:</b>	AB, BU, D1, GL, M1, M2, M3
	<b>Peor SAN:</b>	BL, BN, CR, HE, LE, PI, VO, WD
<b>LEE</b>	<b>Mejor SAN:</b>	AB, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
	<b>Iguales:</b>	M2
	<b>Peor SAN:</b>	BL, BN
<b>BOW</b>	<b>Mejor SAN:</b>	AB, BU, D1, M1, PI
	<b>Iguales:</b>	BN, GL, M2, M3, WD
	<b>Peor SAN:</b>	BL, CR, HE, LE, VO
<b>YOU</b>	<b>Mejor SAN:</b>	HE, PI, WD
	<b>Iguales:</b>	CR, LE, M1, M2
	<b>Peor SAN:</b>	BN, VO
<b>BA 0.5</b>	<b>Mejor SAN:</b>	BU, VO, WD
	<b>Iguales:</b>	AB, D1, GL, HE, M2, M3
	<b>Peor SAN:</b>	BL, BN, CR, LE, M1, PI
<b>BA 0.6</b>	<b>Mejor SAN:</b>	BU, HE, VO, WD
	<b>Iguales:</b>	AB, D1, GL, M2, M3
	<b>Peor SAN:</b>	BL, BN, CR, LE, M1, PI
<b>BA 0.7</b>	<b>Mejor SAN:</b>	BU, HE, VO, WD
	<b>Iguales:</b>	AB, D1, GL, M2, M3
	<b>Peor SAN:</b>	BL, BN, CR, LE, M1, PI
<b>BA 0.8</b>	<b>Mejor SAN:</b>	BU, HE, VO, WD
	<b>Iguales:</b>	AB, D1, GL, M2, M3
	<b>Peor SAN:</b>	BL, BN, CR, LE, M1, PI
<b>BA 0.9</b>	<b>Mejor SAN:</b>	AB, BU, HE, VO, WD
	<b>Iguales:</b>	D1, GL, M2, M3
	<b>Peor SAN:</b>	BL, BN, CR, LE, M1, PI
<b>BA 1</b>	<b>Mejor SAN:</b>	AB, BU, HE, VO, WD
	<b>Iguales:</b>	D1, GL, M2, M3
	<b>Peor SAN:</b>	BL, BN, CR, LE, M1, PI
<b>CHI</b>	<b>Mejor SAN:</b>	BU, HE, LE, VO, WD
	<b>Iguales:</b>	CR, D1, GL, M1, M2, M3
	<b>Peor SAN:</b>	AB, BL, BN, PI
<b>SET</b>	<b>Mejor SAN:</b>	AB, GL, LE, VO, WD
	<b>Iguales:</b>	D1, M1, M2, M3
	<b>Peor SAN:</b>	BL, BN, BU, CR, HE, PI
<b>GD_DS</b>	<b>Mejor SAN:</b>	GL, HE, VO, WD
	<b>Iguales:</b>	D1, M2, M3
	<b>Peor SAN:</b>	AB, BL, BU, M1, PI
<b>RLF</b>	<b>Mejor SAN:</b>	HE, M1, PI, WD
	<b>Iguales:</b>	M2, M3
	<b>Peor SAN:</b>	BN, BU, CR, LE, VO
<b>SCH</b>	<b>Mejor SAN:</b>	CR, D1, GL, HE, M1, M3, PI, VO, WD
	<b>Iguales:</b>	BU, M2
	<b>Peor SAN:</b>	AB, BL, BN, LE
<b>FUZ</b>	<b>Mejor SAN:</b>	BU, CR, D1, HE, M3, PI, WD
	<b>Iguales:</b>	
	<b>Peor SAN:</b>	AB, BL, M1, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

		<b>MétodoDER2</b>
<b>MAO</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, CR, HE, PI BL, BU, D1, M1, M2, M3, VO BN, GL, LE, WD
<b>UTA</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	CR, HE BL, BU, D1, M1, M2, M3 AB, BN, GL, LE, PI, VO, WD
<b>LEE</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, BL, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2 BN
<b>BOW</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, BU, CR, D1, HE, M1, PI, VO, WD BL, M2, M3 BN, GL, LE
<b>YOU</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	CR, HE, LE, PI, VO M1, M2 BN, WD
<b>BA 0.5</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, BU, CR, HE, LE, VO, WD BL, D1, M2, M3 BN, GL, M1, PI
<b>BA 0.6</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, BU, CR, HE, LE, VO, WD BL, D1, M2, M3 BN, GL, M1, PI
<b>BA 0.7</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, BU, CR, HE, LE, VO, WD BL, D1, M2, M3 BN, GL, M1, PI
<b>BA 0.8</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, BU, CR, HE, LE, VO, WD BL, D1, M2, M3 BN, GL, M1, PI
<b>BA 0.9</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, BU, CR, HE, LE, VO, WD BL, D1, M2, M3 BN, GL, M1, PI
<b>BA 1</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, BU, CR, HE, LE, VO, WD BL, D1, M2, M3 BN, GL, M1, PI
<b>CHI</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, BU, CR, HE, LE, VO, WD BL, D1, M1, M2, M3 BN, GL, PI
<b>SET</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, GL, HE, LE, VO, WD BL, D1, M1, M2, M3 BN, BU, CR, PI
<b>GD_DS</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	GL, HE, VO, WD BL, D1, M2, M3 AB, BU, M1, PI
<b>RLF</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	CR, HE, M1, PI, VO M2, M3 BN, BU, LE, WD
<b>SCH</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, BL, CR, D1, GL, HE, LE, M1, M3, PI, VO BU, M2 BN, WD
<b>FUZ</b>	<b>Mejor DER2:</b> <b>Iguales:</b> <b>Peor DER2:</b>	AB, BU, CR, D1, HE, M3, PI, WD BL M1, M2

Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).

	MétodoMAO	
UTA	Mejor MAO: Iguales: Peor MAO:	HE BL, BU, D1, GL, M1, M2, M3 AB, BN, CR, LE, PI, VO, WD
LEE	Mejor MAO: Iguales: Peor MAO:	AB, BL, BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2
BOW	Mejor MAO: Iguales: Peor MAO:	AB, BN, BU, D1, LE, M1, VO, WD BL, GL, M2, M3 CR, HE, PI
YOU	Mejor MAO: Iguales: Peor MAO:	LE, VO M1, M2 BN, CR, HE, PI, WD
BA 0.5	Mejor MAO: Iguales: Peor MAO:	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
BA 0.6	Mejor MAO: Iguales: Peor MAO:	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
BA 0.7	Mejor MAO: Iguales: Peor MAO:	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
BA 0.8	Mejor MAO: Iguales: Peor MAO:	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
BA 0.9	Mejor MAO: Iguales: Peor MAO:	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
BA 1	Mejor MAO: Iguales: Peor MAO:	AB, BU, HE, LE, VO, WD BL, GL, M2, M3 BN, CR, D1, M1, PI
CHI	Mejor MAO: Iguales: Peor MAO:	BN, BU, HE, LE, VO, WD AB, BL, GL, M1, M2, M3 CR, D1, PI
SET	Mejor MAO: Iguales: Peor MAO:	AB, BN, GL, LE, VO, WD BL, M1, M2, M3 BU, CR, D1, HE, PI
GD_DS	Mejor MAO: Iguales: Peor MAO:	GL, VO, WD BL, M2, M3 AB, BU, D1, HE, M1, PI
RLF	Mejor MAO: Iguales: Peor MAO:	BN, HE, LE, M1, VO M2, M3, PI BU, CR, WD
SCH	Mejor MAO: Iguales: Peor MAO:	AB, BL, BN, CR, D1, GL, HE, LE, M1, M3, PI, VO BU, M2 WD
FUZ	Mejor MAO: Iguales: Peor MAO:	AB, BU, CR, D1, HE, M3, PI, WD BL M1, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

	<b>MétodoUTA</b>	
<b>LEE</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	AB, BL, BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2
<b>BOW</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	AB, BN, BU, CR, D1, LE, M1, PI, VO, WD BL, GL, M2, M3 HE
<b>YOU</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	BN, CR, LE, PI, VO, WD M1, M2 HE
<b>BA 0.5</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	BU, CR, HE, LE, VO AB, BL, D1, GL, M2, M3, PI BN, M1, WD
<b>BA 0.6</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	BU, CR, HE, LE, VO AB, BL, D1, GL, M2, M3, PI BN, M1, WD
<b>BA 0.7</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	BU, CR, HE, LE, VO AB, BL, D1, GL, M2, M3, PI BN, M1, WD
<b>BA 0.8</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	BU, CR, HE, LE, VO AB, BL, D1, GL, M2, M3, PI BN, M1, WD
<b>BA 0.9</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	AB, BU, CR, HE, LE, VO BL, D1, GL, M2, M3, PI BN, M1, WD
<b>BA 1</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	AB, BU, CR, HE, LE, VO BL, D1, GL, M2, M3, PI BN, M1, WD
<b>CHI</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	AB, BN, BU, CR, HE, VO BL, D1, GL, M1, M2, M3, PI LE, WD
<b>SET</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	AB, BN, CR, GL, LE, VO, WD BL, D1, M1, M2, M3, PI BU, HE
<b>GD_DS</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	GL, VO, WD BL, D1, M2, M3 AB, BU, HE, M1, PI
<b>RLF</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	BN, CR, LE, M1, PI, VO M2, M3 BU, HE, WD
<b>SCH</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	BL, BN, CR, D1, GL, HE, LE, M1, M3, PI, VO BU, M2 AB, WD
<b>FUZ</b>	<b>Mejor UTA:</b> <b>Iguales:</b> <b>Peor UTA:</b>	AB, BU, CR, D1, HE, M3, PI, WD BL M1, M2



**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

MétodoLEE		
<b>BOW</b>	Mejor LEE: Iguales: Peor LEE:	BN M2 AB, BL, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>YOU</b>	Mejor LEE: Iguales: Peor LEE:	M2 BN, CR, HE, LE, M1, PI, VO, WD
<b>BA 0.5</b>	Mejor LEE: Iguales: Peor LEE:	M2 AB, BL, BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>BA 0.6</b>	Mejor LEE: Iguales: Peor LEE:	HE, M2 AB, BL, BN, BU, CR, D1, GL, LE, M1, M3, PI, VO, WD
<b>BA 0.7</b>	Mejor LEE: Iguales: Peor LEE:	HE, M2 AB, BL, BN, BU, CR, D1, GL, LE, M1, M3, PI, VO, WD
<b>BA 0.8</b>	Mejor LEE: Iguales: Peor LEE:	HE, M2 AB, BL, BN, BU, CR, D1, GL, LE, M1, M3, PI, VO, WD
<b>BA 0.9</b>	Mejor LEE: Iguales: Peor LEE:	HE, M2 AB, BL, BN, BU, CR, D1, GL, LE, M1, M3, PI, VO, WD
<b>BA 1</b>	Mejor LEE: Iguales: Peor LEE:	HE, M2 AB, BL, BN, BU, CR, D1, GL, LE, M1, M3, PI, VO, WD
<b>CHI</b>	Mejor LEE: Iguales: Peor LEE:	M2 AB, BL, BN, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD
<b>SET</b>	Mejor LEE: Iguales: Peor LEE:	BN, WD M2 AB, BL, BU, CR, D1, GL, HE, LE, M1, M3, PI, VO
<b>GD_DS</b>	Mejor LEE: Iguales: Peor LEE:	WD M2 AB, BL, BU, D1, GL, HE, M1, M3, PI, VO
<b>RLF</b>	Mejor LEE: Iguales: Peor LEE:	HE, M2 BN, BU, CR, LE, M1, M3, PI, VO, WD
<b>SCH</b>	Mejor LEE: Iguales: Peor LEE:	CR, GL, HE, M3, PI BL, M1, M2, VO AB, BN, BU, D1, LE, WD
<b>FUZ</b>	Mejor LEE: Iguales: Peor LEE:	AB, WD D1 BL, BU, CR, HE, M1, M2, M3, PI

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

	<b>Método BOW</b>	
<b>YOU</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	CR, LE, WD M2, PI BN, HE, M1, VO
<b>BA 0.5</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	BU, CR, HE, LE, VO, WD BL, GL, M2, M3 AB, BN, D1, M1, PI
<b>BA 0.6</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	BU, CR, HE, LE, VO, WD BL, GL, M2, M3 AB, BN, D1, M1, PI
<b>BA 0.7</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	BU, CR, HE, LE, VO, WD BL, GL, M2, M3 AB, BN, D1, M1, PI
<b>BA 0.8</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	BU, CR, HE, LE, VO, WD BL, GL, M2, M3 AB, BN, D1, M1, PI
<b>BA 0.9</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	BU, CR, HE, LE, VO, WD BL, GL, M2, M3 AB, BN, D1, M1, PI
<b>BA 1</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	BU, CR, HE, LE, VO, WD BL, GL, M2, M3 AB, BN, D1, M1, PI
<b>CHI</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	BU, CR, HE, LE, VO, WD BL, GL, M2, M3 AB, BN, D1, M1, PI
<b>SET</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	AB, CR, GL, LE, VO, WD BL, M2, M3 BN, BU, D1, HE, M1, PI
<b>GD_DS</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	GL, HE, VO, WD BL, M2, M3 AB, BU, D1, M1, PI
<b>RLF</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	HE, M1, PI, VO CR, M2, M3 BN, BU, LE, WD
<b>SCH</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	BL, CR, D1, GL, HE, LE, M1, M3, PI, VO, WD M2 AB, BN, BU
<b>FUZ</b>	<b>Mejor BOW:</b> <b>Iguales:</b> <b>Peor BOW:</b>	CR, D1, HE, M1, M3, PI, WD BL AB, BU, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

		Método YOU
<b>BA 0.5</b>	Mejor YOU: Iguales: Peor YOU:	HE, LE, VO, WD M2 BN, CR, M1, PI
<b>BA 0.6</b>	Mejor YOU: Iguales: Peor YOU:	HE, LE, VO, WD M2 BN, CR, M1, PI
<b>BA 0.7</b>	Mejor YOU: Iguales: Peor YOU:	HE, LE, VO, WD M2 BN, CR, M1, PI
<b>BA 0.8</b>	Mejor YOU: Iguales: Peor YOU:	CR, HE, LE, VO, WD M2 BN, M1, PI
<b>BA 0.9</b>	Mejor YOU: Iguales: Peor YOU:	CR, HE, LE, VO, WD M2 BN, M1, PI
<b>BA 1</b>	Mejor YOU: Iguales: Peor YOU:	CR, HE, LE, VO, WD M2 BN, M1, PI
<b>CHI</b>	Mejor YOU: Iguales: Peor YOU:	BN, HE, LE, VO, WD M1, M2 CR, PI
<b>SET</b>	Mejor YOU: Iguales: Peor YOU:	BN, LE, VO, WD M1, M2 CR, HE, PI
<b>GD_DS</b>	Mejor YOU: Iguales: Peor YOU:	HE, VO, WD M2 M1, PI
<b>RLF</b>	Mejor YOU: Iguales: Peor YOU:	BN, HE, LE, M1, PI, VO M2, WD CR
<b>SCH</b>	Mejor YOU: Iguales: Peor YOU:	BN, CR, HE, M1, PI, VO, WD M2 LE
<b>FUZ</b>	Mejor YOU: Iguales: Peor YOU:	CR, HE, PI, WD M1, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

		Método BA 0.5
<b>BA 0.6</b>	<b>Mejor BA 0.5:</b> <b>Iguales:</b> <b>Peor BA 0.5:</b>	HE, VO AB, BL, BN, BU, CR, D1, GL, LE, M1, M2, M3, PI, WD
<b>BA 0.7</b>	<b>Mejor BA 0.5:</b> <b>Iguales:</b> <b>Peor BA 0.5:</b>	HE, VO AB, BL, BN, BU, CR, D1, GL, LE, M1, M2, M3, PI, WD
<b>BA 0.8</b>	<b>Mejor BA 0.5:</b> <b>Iguales:</b> <b>Peor BA 0.5:</b>	CR, HE, VO AB, BL, BN, BU, D1, GL, LE, M1, M2, M3, PI, WD
<b>BA 0.9</b>	<b>Mejor BA 0.5:</b> <b>Iguales:</b> <b>Peor BA 0.5:</b>	AB, CR, HE, VO BL, BU, D1, GL, LE, M1, M2, M3, PI, WD BN
<b>BA 1</b>	<b>Mejor BA 0.5:</b> <b>Iguales:</b> <b>Peor BA 0.5:</b>	AB, CR, HE BL, BU, D1, GL, LE, M1, M2, M3, PI, WD BN, VO
<b>CHI</b>	<b>Mejor BA 0.5:</b> <b>Iguales:</b> <b>Peor BA 0.5:</b>	BN, BU, CR, HE, LE, M1, VO, WD BL, D1, GL, M2, M3, PI AB
<b>SET</b>	<b>Mejor BA 0.5:</b> <b>Iguales:</b> <b>Peor BA 0.5:</b>	AB, BN, GL, LE, M1, VO BL, D1, M2, M3, PI BU, CR, HE, VO
<b>GD_DS</b>	<b>Mejor BA 0.5:</b> <b>Iguales:</b> <b>Peor BA 0.5:</b>	GL, WD BL, D1, M1, M2, M3 AB, BU, HE, PI, VO
<b>RLF</b>	<b>Mejor BA 0.5:</b> <b>Iguales:</b> <b>Peor BA 0.5:</b>	BN, HE, M1, PI, VO M2, M3 BU, CR, LE, WD
<b>SCH</b>	<b>Mejor BA 0.5:</b> <b>Iguales:</b> <b>Peor BA 0.5:</b>	BL, BN, CR, D1, GL, HE, M1, M3, PI, VO M2 AB, BU, LE, WD
<b>FUZ</b>	<b>Mejor BA 0.5:</b> <b>Iguales:</b> <b>Peor BA 0.5:</b>	CR, D1, HE, M1, M3, PI, WD BL AB, BU, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

		Método BA 0.6
<b>BA 0.7</b>	<b>Mejor BA 0.6:</b> <b>Iguales:</b> <b>Peor BA 0.6:</b>	AB, BL, BN, BU, CR, D1, GL, LE, M1, M2, M3, PI, VO, WD HE
<b>BA 0.8</b>	<b>Mejor BA 0.6:</b> <b>Iguales:</b> <b>Peor BA 0.6:</b>	CR AB, BL, BN, BU, D1, GL, LE, M1, M2, M3, PI, VO, WD HE
<b>BA 0.9</b>	<b>Mejor BA 0.6:</b> <b>Iguales:</b> <b>Peor BA 0.6:</b>	AB, CR BL, BU, D1, GL, LE, M1, M2, M3, PI, WD BN, HE, VO
<b>BA 1</b>	<b>Mejor BA 0.6:</b> <b>Iguales:</b> <b>Peor BA 0.6:</b>	AB, CR BL, BU, D1, GL, LE, M1, M2, M3, PI, WD BN, HE, VO
<b>CHI</b>	<b>Mejor BA 0.6:</b> <b>Iguales:</b> <b>Peor BA 0.6:</b>	BN, BU, CR, LE, M1, WD BL, D1, GL, HE, M2, M3, PI AB, VO
<b>SET</b>	<b>Mejor BA 0.6:</b> <b>Iguales:</b> <b>Peor BA 0.6:</b>	AB, BN, GL, LE, M1 BL, D1, M2, M3, PI BU, CR, HE, VO, WD
<b>GD_DS</b>	<b>Mejor BA 0.6:</b> <b>Iguales:</b> <b>Peor BA 0.6:</b>	GL, WD BL, D1, M1, M2, M3 AB, BU, HE, PI, VO
<b>RLF</b>	<b>Mejor BA 0.6:</b> <b>Iguales:</b> <b>Peor BA 0.6:</b>	BN, M1, PI, VO M2, M3 BU, CR, HE, LE, WD
<b>SCH</b>	<b>Mejor BA 0.6:</b> <b>Iguales:</b> <b>Peor BA 0.6:</b>	BL, BN, CR, D1, GL, HE, M1, M3, PI, VO M2 AB, BU, LE, WD
<b>FUZ</b>	<b>Mejor BA 0.6:</b> <b>Iguales:</b> <b>Peor BA 0.6:</b>	CR, D1, HE, M1, M3, PI, WD BL AB, BU, M2

II. APÉNDICES DEL CAPÍTULO 3

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método BA 0.7</b>		
<b>BA 0.8</b>	<b>Mejor BA 0.7:</b> <b>Iguales:</b> <b>Peor BA 0.7:</b>	CR AB, BL, BN, BU, D1, GL, HE, LE, M1, M2, M3, PI, VO, WD
<b>BA 0.9</b>	<b>Mejor BA 0.7:</b> <b>Iguales:</b> <b>Peor BA 0.7:</b>	AB, CR BL, BU, D1, GL, HE, LE, M1, M2, M3, PI, WD BN, VO
<b>BA 1</b>	<b>Mejor BA 0.7:</b> <b>Iguales:</b> <b>Peor BA 0.7:</b>	AB, CR BL, BU, D1, GL, HE, LE, M1, M2, M3, PI, WD BN, VO
<b>CHI</b>	<b>Mejor BA 0.7:</b> <b>Iguales:</b> <b>Peor BA 0.7:</b>	BN, BU, CR, LE, M1, WD BL, D1, GL, M2, M3, PI AB, HE, VO
<b>SET</b>	<b>Mejor BA 0.7:</b> <b>Iguales:</b> <b>Peor BA 0.7:</b>	AB, BN, GL, LE, M1 BL, D1, M2, M3, PI BU, CR, HE, VO, WD
<b>GD_DS</b>	<b>Mejor BA 0.7:</b> <b>Iguales:</b> <b>Peor BA 0.7:</b>	GL, WD BL, D1, M1, M2, M3 AB, BU, HE, PI, VO
<b>RLF</b>	<b>Mejor BA 0.7:</b> <b>Iguales:</b> <b>Peor BA 0.7:</b>	BN, M1, PI, VO HE, M2, M3 BU, CR, LE, VO
<b>SCH</b>	<b>Mejor BA 0.7:</b> <b>Iguales:</b> <b>Peor BA 0.7:</b>	BL, BN, CR, D1, GL, HE, M1, M3, PI, VO M2 AB, BU, LE, WD
<b>FUZ</b>	<b>Mejor BA 0.7:</b> <b>Iguales:</b> <b>Peor BA 0.7:</b>	CR, D1, M1, M3, PI, WD BL AB, BU, HE, M2

<b>Método BA 0.8</b>		
<b>BA 0.9</b>	<b>Mejor BA 0.8:</b> <b>Iguales:</b> <b>Peor BA 0.8:</b>	AB BL, BU, CR, D1, GL, HE, LE, M1, M2, M3, PI, WD BN, VO
<b>BA 1</b>	<b>Mejor BA 0.8:</b> <b>Iguales:</b> <b>Peor BA 0.8:</b>	AB BL, BU, CR, D1, GL, HE, LE, M1, M2, M3, PI, WD BN, VO
<b>CHI</b>	<b>Mejor BA 0.8:</b> <b>Iguales:</b> <b>Peor BA 0.8:</b>	BN, BU, CR, LE, M1, WD BL, D1, GL, M2, M3, PI AB, HE, VO
<b>SET</b>	<b>Mejor BA 0.8:</b> <b>Iguales:</b> <b>Peor BA 0.8:</b>	AB, BN, GL, LE, M1 BL, D1, M2, M3, PI BU, CR, HE, VO, WD
<b>GD_DS</b>	<b>Mejor BA 0.8:</b> <b>Iguales:</b> <b>Peor BA 0.8:</b>	GL, WD BL, D1, M1, M2, M3 AB, BU, HE, PI, VO
<b>RLF</b>	<b>Mejor BA 0.8:</b> <b>Iguales:</b> <b>Peor BA 0.8:</b>	BN, M1, PI, VO HE, M2, M3 BU, CR, LE, WD
<b>SCH</b>	<b>Mejor BA 0.8:</b> <b>Iguales:</b> <b>Peor BA 0.8:</b>	BL, BN, CR, D1, GL, HE, M1, M3, PI, VO M2 AB, BU, LE, WD
<b>FUZ</b>	<b>Mejor BA 0.8:</b> <b>Iguales:</b> <b>Peor BA 0.8:</b>	CR, D1, M1, M3, PI, VO BL AB, BU, HE, M2

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método BA 0.9</b>		
<b>BA 1</b>	<b>Mejor BA 0.9:</b> <b>Iguales:</b> <b>Peor BA 0.9:</b>	AB, BL, BN, BU, CR, D1, GL, HE, LE, M1, M2, M3, PI, VO, WD
<b>CHI</b>	<b>Mejor BA 0.9:</b> <b>Iguales:</b> <b>Peor BA 0.9:</b>	BN, BU, CR, LE, M1, WD BL, D1, GL, M2, M3, PI AB, HE, VO
<b>SET</b>	<b>Mejor BA 0.9:</b> <b>Iguales:</b> <b>Peor BA 0.9:</b>	AB, BN, GL, LE, M1 BL, D1, M2, M3, PI BU, CR, HE, VO, WD
<b>GD_DS</b>	<b>Mejor BA 0.9:</b> <b>Iguales:</b> <b>Peor BA 0.9:</b>	GL, WD BL, D1, M1, M2, M3 AB, BU, HE, PI, VO
<b>RLF</b>	<b>Mejor BA 0.9:</b> <b>Iguales:</b> <b>Peor BA 0.9:</b>	BN, M1, PI HE, M2, M3 BU, CR, LE, VO, WD
<b>SCH</b>	<b>Mejor BA 0.9:</b> <b>Iguales:</b> <b>Peor BA 0.9:</b>	BL, BN, CR, D1, GL, HE, M1, M3, PI, VO M2 AB, BU, LE, WD
<b>FUZ</b>	<b>Mejor BA 0.9:</b> <b>Iguales:</b> <b>Peor BA 0.9:</b>	CR, D1, M1, M3, PI, WD BL AB, BU, HE, M2

<b>Método BA 1</b>		
<b>CHI</b>	<b>Mejor BA 1:</b> <b>Iguales:</b> <b>Peor BA 1:</b>	BN, BU, CR, LE, M1, WD D1, GL, M2, M3, PI AB, HE, VO
<b>SET</b>	<b>Mejor BA 1:</b> <b>Iguales:</b> <b>Peor BA 1:</b>	AB, BN, GL, LE, M1 BL, D1, M2, M3, PI BU, CR, HE, VO, WD
<b>GD_DS</b>	<b>Mejor BA 1:</b> <b>Iguales:</b> <b>Peor BA 1:</b>	GL, WD BL, D1, M1, M2, M3 AB, BU, HE, PI, VO
<b>RLF</b>	<b>Mejor BA 1:</b> <b>Iguales:</b> <b>Peor BA 1:</b>	BN, M1, PI HE, M2, M3 BU, CR, LE, VO, WD
<b>SCH</b>	<b>Mejor BA 1:</b> <b>Iguales:</b> <b>Peor BA 1:</b>	BL, BN, CR, D1, GL, HE, M1, M3, PI, VO M2 AB, BU, LE, WD
<b>FUZ</b>	<b>Mejor BA 1:</b> <b>Iguales:</b> <b>Peor BA 1:</b>	CR, D1, M1, M3, PI, WD BL AB, BU, HE, M2

<b>Método CHI</b>		
<b>SET</b>	<b>Mejor CHI:</b> <b>Iguales:</b> <b>Peor CHI:</b>	AB, BN, GL BL, D1, M1, M2, M3, PI, VO BU, CR, HE, LE, WD
<b>GD_DS</b>	<b>Mejor CHI:</b> <b>Iguales:</b> <b>Peor CHI:</b>	GL, WD BL, D1, HE, M2, M3 AB, BU, M1, PI, VO
<b>RLF</b>	<b>Mejor CHI:</b> <b>Iguales:</b> <b>Peor CHI:</b>	HE, LE, M1, PI M2, M3 BN, BU, CR, VO, WD
<b>SCH</b>	<b>Mejor CHI:</b> <b>Iguales:</b> <b>Peor CHI:</b>	AB, BL, BN, CR, D1, GL, HE, LE, M1, M3, PI, VO M2 BU, WD
<b>FUZ</b>	<b>Mejor CHI:</b> <b>Iguales:</b> <b>Peor CHI:</b>	AB, CR, D1, HE, M3, PI, WD BL, M1 BU, M2

II. APÉNDICES DEL CAPÍTULO 3

**Tabla 57. Comparación entre métodos ARNMF y ACE. (Continuación).**

<b>Método SET</b>		
<b>GD_DS</b>	<b>Mejor SET:</b> <b>Iguales:</b> <b>Peor SET:</b>	GL, HE, WD BL, BU, D1, M2, M3 AB, M1, PI, VO
<b>RLF</b>	<b>Mejor SET:</b> <b>Iguales:</b> <b>Peor SET:</b>	HE, M1, PI BU, M2, M3 BN, CR, LE, VO, WD
<b>SCH</b>	<b>Mejor SET:</b> <b>Iguales:</b> <b>Peor SET:</b>	BL, BU, CR, D1, GL, HE, M1, M3, PI, VO M2 AB, BN, LE, WD
<b>FUZ</b>	<b>Mejor SET:</b> <b>Iguales:</b> <b>Peor SET:</b>	CR, D1, HE, M3, PI, WD BL, BU, M1 AB, M2

<b>Método GD_DS</b>		
<b>RLF</b>	<b>Mejor GD_DS:</b> <b>Iguales:</b> <b>Peor GD_DS:</b>	HE, M1, PI, VO BU, M2, M3 WD
<b>SCH</b>	<b>Mejor GD_DS:</b> <b>Iguales:</b> <b>Peor GD_DS:</b>	AB, D1, HE, M1, M3, PI, WD BL, BU M2
<b>FUZ</b>	<b>Mejor GD_DS:</b> <b>Iguales:</b> <b>Peor GD_DS:</b>	AB, BL, D1, HE, M1, M3, PI, VO M2 BU, GL, WD

<b>Método RLF</b>		
<b>SCH</b>	<b>Mejor RLF:</b> <b>Iguales:</b> <b>Peor RLF:</b>	CR, M3, PI, WD BU HE, M1, M2
<b>FUZ</b>	<b>Mejor RLF:</b> <b>Iguales:</b> <b>Peor RLF:</b>	BN, BU, CR, HE, LE, M1, M3, PI, VO, WD M2

<b>Método SCH</b>		
<b>FUZ</b>	<b>Mejor SCH:</b> <b>Iguales:</b> <b>Peor SCH:</b>	BL, CR, HE, M1, M2, M3, PI AB, BU, D1, WD



## II.6. Comparación entre métodos ARNMF con y sin WD

Tabla 58. Comparación de métodos ARNMF y ARNMF&amp;WD.

Método BL1 con Weight-Decay		
BL1	Mejor BL1_WD: Iguales: Peor BL1_WD:	BN, VO CR, GL, HE, PI, WD BU
BL2	Mejor BL1_WD: Iguales: Peor BL1_WD:	VO, WD GL BN, BU, CR, HE, PI
CLO	Mejor BL1_WD: Iguales: Peor BL1_WD:	BN, VO CR, GL, PI, WD BU, HE
PRI	Mejor BL1_WD: Iguales: Peor BL1_WD:	BN, VO CR, GL, HE, PI, WD BU
DEV	Mejor BL1_WD: Iguales: Peor BL1_WD:	VO, WD CR, GL BN, BU, HE, PI
LEE	Mejor BL1_WD: Iguales: Peor BL1_WD:	BU, CR, GL, HE, PI, VO, WD BN
TEK	Mejor BL1_WD: Iguales: Peor BL1_WD:	GL, PI, VO, WD CR BN, BU, HE
SAN	Mejor BL1_WD: Iguales: Peor BL1_WD:	BN, PI, VO CR, GL, WD BU, HE
TEKA	Mejor BL1_WD: Iguales: Peor BL1_WD:	VO, WD GL BN, BU, CR, HE, PI
TEKB	Mejor BL1_WD: Iguales: Peor BL1_WD:	GL, VO WD BN, BU, CR, HE, PI
DER2	Mejor BL1_WD: Iguales: Peor BL1_WD:	GL, WD PI, VO BN, BU, CR, HE
DER3	Mejor BL1_WD: Iguales: Peor BL1_WD:	BU, CR, GL, HE, PI, VO, WD BN
MAO	Mejor BL1_WD: Iguales: Peor BL1_WD:	CR, PI, WD GL, HE, VO BN, BU
UTA	Mejor BL1_WD: Iguales: Peor BL1_WD:	HE, WD GL BN, BU, CR, PI, VO
CIB	Mejor BL1_WD: Iguales: Peor BL1_WD:	CR, GL, PI, VO, WD HE BN, BU
TEKE	Mejor BL1_WD: Iguales: Peor BL1_WD:	GL, HE, PI, VO, WD BU
YOU	Mejor BL1_WD: Iguales: Peor BL1_WD:	PI, VO, WD CR BN, HE

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

Método BL2 con Weight-Decay		
BL1	Mejor BL2_WD: Iguales: Peor BL2_WD:	BN, VO CR, GL, HE, PI, WD BU
BL2	Mejor BL2_WD: Iguales: Peor BL2_WD:	VO, WD GL BN, BU, CR, HE, PI
CLO	Mejor BL2_WD: Iguales: Peor BL2_WD:	BN, VO CR, GL, PI, WD BU, HE
PRI	Mejor BL2_WD: Iguales: Peor BL2_WD:	BN, VO CR, GL, HE, PI, WD BU
DEV	Mejor BL2_WD: Iguales: Peor BL2_WD:	VO, WD CR, GL BN, BU, HE, PI
LEE	Mejor BL2_WD: Iguales: Peor BL2_WD:	BU, CR, GL, HE, PI, VO, WD BN
TEK	Mejor BL2_WD: Iguales: Peor BL2_WD:	GL, PI, VO, WD CR BN, BU, HE
SAN	Mejor BL2_WD: Iguales: Peor BL2_WD:	BN, PI, VO CR, GL, WD BU, HE
TEKA	Mejor BL2_WD: Iguales: Peor BL2_WD:	VO, WD GL BN, BU, CR, HE, PI
TEKB	Mejor BL2_WD: Iguales: Peor BL2_WD:	GL, VO WD BN, BU, CR, HE, PI
DER2	Mejor BL2_WD: Iguales: Peor BL2_WD:	GL, WD PI, VO BN, BU, CR, HE
DER3	Mejor BL2_WD: Iguales: Peor BL2_WD:	CR, GL, HE, PI, VO, WD BU BN
MAO	Mejor BL2_WD: Iguales: Peor BL2_WD:	CR, PI, WD GL, HE, VO BN, BU
UTA	Mejor BL2_WD: Iguales: Peor BL2_WD:	HE, WD GL BN, BU, CR, PI, VO
CIB	Mejor BL2_WD: Iguales: Peor BL2_WD:	CR, GL, PI, VO, WD HE BN, BU
TEKE	Mejor BL2_WD: Iguales: Peor BL2_WD:	BU, GL, HE, PI, VO, WD
YOU	Mejor BL2_WD: Iguales: Peor BL2_WD:	PI, VO, WD CR BN, HE

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

Método CLO con Weight-Decay	
BL1	Mejor CLO_WD: BN, VO Iguales: CR, GL, HE, PI, WD Peor CLO_WD: BU
BL2	Mejor CLO_WD: VO, WD Iguales: GL Peor CLO_WD: BN, BU, CR, HE, PI
CLO	Mejor CLO_WD: BN, VO Iguales: CR, GL, PI, WD Peor CLO_WD: BU, HE
PRI	Mejor CLO_WD: BN, VO Iguales: CR, GL, HE, PI, WD Peor CLO_WD: BU
DEV	Mejor CLO_WD: VO, WD Iguales: CR, GL Peor CLO_WD: BN, BU, HE, PI
LEE	Mejor CLO_WD: BU, CR, GL, HE, PI, VO, WD Iguales: Peor CLO_WD: BN
TEK	Mejor CLO_WD: GL, PI, VO, WD Iguales: CR Peor CLO_WD: BN, BU, HE
SAN	Mejor CLO_WD: BN, PI, VO Iguales: CR, GL, WD Peor CLO_WD: BU, HE
TEKA	Mejor CLO_WD: VO, WD Iguales: GL Peor CLO_WD: BN, BU, CR, HE, PI
TEKB	Mejor CLO_WD: GL, VO Iguales: WD Peor CLO_WD: BN, BU, CR, HE, PI
DER2	Mejor CLO_WD: GL, WD Iguales: PI, VO Peor CLO_WD: BN, BU, CR, HE
DER3	Mejor CLO_WD: BU, CR, GL, HE, PI, VO, WD Iguales: Peor CLO_WD: BN
MAO	Mejor CLO_WD: CR, PI, WD Iguales: GL, HE, VO Peor CLO_WD: BN, BU
UTA	Mejor CLO_WD: HE, WD Iguales: GL Peor CLO_WD: BN, BU, CR, PI, VO
CIB	Mejor CLO_WD: CR, GL, PI, VO, WD Iguales: HE Peor CLO_WD: BN, BU
TEKE	Mejor CLO_WD: GL, HE, PI, VO, WD Iguales: BU Peor CLO_WD:
YOU	Mejor CLO_WD: PI, VO, WD Iguales: CR Peor CLO_WD: BN, HE

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

Método PRI con Weight-Decay		
BL1	Mejor PRI_WD: Iguales: Peor PRI_WD:	BN, VO CR, GL, HE, PI, WD BU
BL2	Mejor PRI_WD: Iguales: Peor PRI_WD:	VO, WD GL BN, BU, CR, HE, PI
CLO	Mejor PRI_WD: Iguales: Peor PRI_WD:	BN, VO CR, GL, PI, WD BU, HE
PRI	Mejor PRI_WD: Iguales: Peor PRI_WD:	BN, VO CR, GL, HE, PI, WD BU
DEV	Mejor PRI_WD: Iguales: Peor PRI_WD:	VO, WD CR, GL BN, BU, HE, PI
LEE	Mejor PRI_WD: Iguales: Peor PRI_WD:	BU, CR, GL, HE, PI, VO, WD BN
TEK	Mejor PRI_WD: Iguales: Peor PRI_WD:	GL, PI, VO, WD CR BN, BU, HE
SAN	Mejor PRI_WD: Iguales: Peor PRI_WD:	BN, PI, VO CR, GL, WD BU, HE
TEKA	Mejor PRI_WD: Iguales: Peor PRI_WD:	VO, WD GL BN, BU, CR, HE, PI
TEKB	Mejor PRI_WD: Iguales: Peor PRI_WD:	GL, VO WD BN, BU, CR, HE, PI
DER2	Mejor PRI_WD: Iguales: Peor PRI_WD:	GL, WD PI, VO BN, BU, CR, HE
DER3	Mejor PRI_WD: Iguales: Peor PRI_WD:	BU, CR, GL, HE, PI, VO, WD BN
MAO	Mejor PRI_WD: Iguales: Peor PRI_WD:	CR, PI, WD GL, HE, VO BN, BU
UTA	Mejor PRI_WD: Iguales: Peor PRI_WD:	HE, WD GL BN, BU, CR, PI, VO
CIB	Mejor PRI_WD: Iguales: Peor PRI_WD:	CR, GL, PI, VO, WD HE BN, BU
TEKE	Mejor PRI_WD: Iguales: Peor PRI_WD:	GL, HE, PI, VO, WD BU
YOU	Mejor PRI_WD: Iguales: Peor PRI_WD:	PI, VO, WD CR BN, HE

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

Método DEV con Weight-Decay	
BL1	Mejor DEV_WD: BN, VO Iguales: BU, CR, GL, HE, PI, WD Peor DEV_WD:
BL2	Mejor DEV_WD: VO, WD Iguales: BU, GL Peor DEV_WD: BN, CR, HE, PI
CLO	Mejor DEV_WD: BN, VO Iguales: BU, CR, GL, PI, WD Peor DEV_WD: HE
PRI	Mejor DEV_WD: BN, VO Iguales: BU, CR, GL, HE, PI, WD Peor DEV_WD:
DEV	Mejor DEV_WD: VO, WD Iguales: BU, CR, GL Peor DEV_WD: BN, HE, PI
LEE	Mejor DEV_WD: BU, CR, GL, HE, PI, VO, WD Iguales: Peor DEV_WD: BN
TEK	Mejor DEV_WD: BU, GL, PI, VO, WD Iguales: CR Peor DEV_WD: BN, HE
SAN	Mejor DEV_WD: BN, PI, VO Iguales: BU, CR, GL, WD Peor DEV_WD: HE
TEKA	Mejor DEV_WD: VO, WD Iguales: BU, GL Peor DEV_WD: BN, CR, HE, PI
TEKB	Mejor DEV_WD: GL, VO Iguales: BU, WD Peor DEV_WD: BN, CR, HE, PI
DER2	Mejor DEV_WD: GL, WD Iguales: BU, PI, VO Peor DEV_WD: BN, CR, HE
DER3	Mejor DEV_WD: BU, CR, GL, HE, PI, VO, WD Iguales: Peor DEV_WD: BN
MAO	Mejor DEV_WD: CR, PI, WD Iguales: BU, GL, HE, VO Peor DEV_WD: BN
UTA	Mejor DEV_WD: HE, WD Iguales: BU, GL Peor DEV_WD: BN, CR, PI, VO
CIB	Mejor DEV_WD: CR, GL, PI, VO, WD Iguales: HE Peor DEV_WD: BN, BU
TEKE	Mejor DEV_WD: BU, GL, HE, PI, VO, WD Iguales: Peor DEV_WD:
YOU	Mejor DEV_WD: PI, VO, WD Iguales: CR Peor DEV_WD: BN, HE

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

Método LEE con Weight-Decay	
<b>BL1</b>	Mejor LEE_WD: BN, BU Iguales: Peor LEE_WD: CR, GL, HE, PI, VO, WD
<b>BL2</b>	Mejor LEE_WD: BU Iguales: Peor LEE_WD: CR, GL, HE, PI, VO, WD
<b>CLO</b>	Mejor LEE_WD: BN, BU Iguales: Peor LEE_WD: CR, GL, HE, PI, VO, WD
<b>PRI</b>	Mejor LEE_WD: BN, BU Iguales: Peor LEE_WD: CR, GL, HE, PI, VO, WD
<b>DEV</b>	Mejor LEE_WD: BU Iguales: Peor LEE_WD: BN, CR, GL, HE, PI, VO, WD
<b>LEE</b>	Mejor LEE_WD: BN, BU, CR, GL, PI Iguales: Peor LEE_WD: HE, VO, WD
<b>TEK</b>	Mejor LEE_WD: BN, BU Iguales: Peor LEE_WD: CR, GL, HE, PI, VO, WD
<b>SAN</b>	Mejor LEE_WD: BN, BU Iguales: Peor LEE_WD: CR, GL, HE, PI, VO, WD
<b>TEKA</b>	Mejor LEE_WD: BU Iguales: Peor LEE_WD: BN, CR, GL, HE, PI, VO, WD
<b>TEKB</b>	Mejor LEE_WD: BN, BU Iguales: Peor LEE_WD: CR, GL, HE, PI, VO, WD
<b>DER2</b>	Mejor LEE_WD: BN, BU Iguales: Peor LEE_WD: CR, GL, HE, PI, VO, WD
<b>DER3</b>	Mejor LEE_WD: BN, BU, GL, VO, WD Iguales: Peor LEE_WD: CR, HE, PI
<b>MAO</b>	Mejor LEE_WD: BU Iguales: Peor LEE_WD: BN, CR, GL, HE, PI, VO, WD
<b>UTA</b>	Mejor LEE_WD: BU Iguales: Peor LEE_WD: BN, CR, GL, HE, PI, VO, WD
<b>CIB</b>	Mejor LEE_WD: BU Iguales: Peor LEE_WD: BN, CR, GL, HE, PI, VO, WD
<b>TEKE</b>	Mejor LEE_WD: BU, GL, PI Iguales: Peor LEE_WD: GL, HE, VO, WD
<b>YOU</b>	Mejor LEE_WD: Iguales: Peor LEE_WD: BN, CR, HE, PI, VO, WD

Tabla 58. Comparación de métodos ARNMF y ARNMF&amp;WD. (Continuación).

Método TEK con Weight-Decay	
BL1	Mejor TEK_WD: BN, HE Iguales: CR, GL, PI Peor TEK_WD: BU, VO, WD
BL2	Mejor TEK_WD: Iguales: GL Peor TEK_WD: BN, BU, CR, HE, PI, VO, WD
CLO	Mejor TEK_WD: BN, HE Iguales: CR, GL, PI Peor TEK_WD: BU, VO, WD
PRI	Mejor TEK_WD: BN, HE Iguales: CR, GL, PI Peor TEK_WD: BU, VO, WD
DEV	Mejor TEK_WD: WD Iguales: CR, GL Peor TEK_WD: BN, BU, HE, PI, VO
LEE	Mejor TEK_WD: CR, GL, HE, PI, VO, WD Iguales: BU Peor TEK_WD: BN
TEK	Mejor TEK_WD: GL, PI, WD Iguales: CR Peor TEK_WD: BN, BU, HE, VO
SAN	Mejor TEK_WD: BN, HE, PI Iguales: CR, GL Peor TEK_WD: BU, VO, WD
TEKA	Mejor TEK_WD: Iguales: GL Peor TEK_WD: BN, BU, CR, HE, PI, VO, WD
TEKB	Mejor TEK_WD: GL, HE Iguales: Peor TEK_WD: BN, BU, CR, PI, VO, WD
DER2	Mejor TEK_WD: GL, WD Iguales: PI Peor TEK_WD: BN, BU, CR, HE, VO
DER3	Mejor TEK_WD: CR, GL, HE, PI, VO, WD Iguales: Peor TEK_WD: BN, BU
MAO	Mejor TEK_WD: CR, PI, WD Iguales: GL Peor TEK_WD: BN, BU, HE, VO
UTA	Mejor TEK_WD: HE, WD Iguales: GL Peor TEK_WD: BN, BU, CR, PI, VO
CIB	Mejor TEK_WD: CR, GL, PI, WD Iguales: Peor TEK_WD: BN, BU, HE, VO
TEKE	Mejor TEK_WD: GL, HE, PI, VO, WD Iguales: Peor TEK_WD: BU
YOU	Mejor TEK_WD: PI, WD Iguales: CR Peor TEK_WD: BN, HE, VO

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

		Método SAN con Weight-Decay
BL1	Mejor SAN_WD: Iguales: Peor SAN_WD:	BN, VO CR, GL, HE, PI, WD BU
BL2	Mejor SAN_WD: Iguales: Peor SAN_WD:	VO, WD GL BN, BU, CR, HE, PI
CLO	Mejor SAN_WD: Iguales: Peor SAN_WD:	BN, VO CR, GL, PI, WD BN, HE
PRI	Mejor SAN_WD: Iguales: Peor SAN_WD:	BN, VO CR, GL, HE, PI, WD BU
DEV	Mejor SAN_WD: Iguales: Peor SAN_WD:	VO, WD CR, GL BN, BU, HE, PI
LEE	Mejor SAN_WD: Iguales: Peor SAN_WD:	BU, CR, GL, HE, PI, VO, WD BN
TEK	Mejor SAN_WD: Iguales: Peor SAN_WD:	GL, PI, VO, WD CR BN, BU, HE
SAN	Mejor SAN_WD: Iguales: Peor SAN_WD:	BN, PI, VO CR, GL, WD BU, HE
TEKA	Mejor SAN_WD: Iguales: Peor SAN_WD:	VO, WD GL BN, BU, CR, HE, PI
TEKB	Mejor SAN_WD: Iguales: Peor SAN_WD:	GL, VO WD BN, BU, CR, HE, PI
DER2	Mejor SAN_WD: Iguales: Peor SAN_WD:	GL, WD PI, VO BN, BU, CR, HE
DER3	Mejor SAN_WD: Iguales: Peor SAN_WD:	BU, CR, GL, HE, PI, VO, WD BN
MAO	Mejor SAN_WD: Iguales: Peor SAN_WD:	CR, PI, WD GL, HE, VO BN, BU
UTA	Mejor SAN_WD: Iguales: Peor SAN_WD:	HE, WD GL BN, BU, CR, PI, VO
CIB	Mejor SAN_WD: Iguales: Peor SAN_WD:	CR, GL, PI, VO, WD HE BN, BU
TEKE	Mejor SAN_WD: Iguales: Peor SAN_WD:	GL, PI, VO, WD BU
YOU	Mejor SAN_WD: Iguales: Peor SAN_WD:	PI, VO, WD CR BN, HE



Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

		Método TEKA con Weight-Decay
BL1	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	BN, PI, VO CR, GL, HE, WD BU
BL2	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	VO, WD GL, PI BN, BU, CR, HE
CLO	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	BN, PI, VO CR, GL, WD BU, HE
PRI	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	BN, PI, VO CR, GL, HE, WD BU
DEV	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	BN, VO, WD CR, GL, PI BU, HE
LEE	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	BN, BU, CR, GL, HE, PI, VO, WD
TEK	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	BN, GL, PI, VO, WD CR BU, HE
SAN	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	BN, PI, VO CR, GL, WD BU, HE
TEKA	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	VO, WD GL, PI BN, BU, CR, HE
TEKB	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	BN, GL, VO PI, WD BU, CR, HE
DER2	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	BN, GL, PI, WD VO BU, CR, HE
DER3	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	BN, CR, GL, HE, PI, VO, WD BU
MAO	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	CR, PI, WD GL, HE, VO BN, BU
UTA	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	HE, WD GL, PI BN, BU, CR, VO
CIB	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	CR, GL, PI, VO, WD HE BN, BU
TEKE	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	BU, GL, HE, PI, VO, WD
YOU	Mejor TEKA_WD: Iguales: Peor TEKA_WD:	PI, VO, WD CR BN, HE

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

		Método TEKB con Weight-Decay
BL1	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	BN, VO CR, GL, HE, PI, WD BU
BL2	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	VO, WD GL BN, BU, CR, HE, PI
CLO	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	BN, VO CR, GL, PI, WD BU, HE
PRI	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	BN, VO CR, GL, HE, PI, WD BU
DEV	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	VO, WD CR, GL BN, BU, HE, PI
LEE	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	CR, GL, HE, PI, VO, WD BU BN
TEK	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	GL, PI, VO, WD CR BN, BU, HE
SAN	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	BN, PI, VO CR, GL, WD BU, HE
TEKA	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	VO, WD GL BN, BU, CR, HE, PI
TEKB	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	GL, VO WD BN, BU, CR, HE, PI
DER2	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	GL, WD PI, VO BN, BU, CR, HE
DER3	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	CR, GL, HE, PI, VO, WD BU BN
MAO	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	CR, PI, WD GL, HE, VO BN, BU
UTA	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	HE, WD GL BN, BU, CR, PI, VO
CIB	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	CR, GL, PI, VO, WD HE BN, BU
TEKE	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	GL, HE, PI, VO, WD BU
YOU	Mejor TEKB_WD: Iguales: Peor TEKB_WD:	PI, VO, WD CR BN, HE

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

		Método DER2 con Weight-Decay
BL1	Mejor DER2_WD: Iguales: Peor DER2_WD:	BN, HE, VO CR, PI BU, GL, WD
BL2	Mejor DER2_WD: Iguales: Peor DER2_WD:	BN, VO  BU, CR, GL, HE, PI, WD
CLO	Mejor DER2_WD: Iguales: Peor DER2_WD:	BN, VO CR, PI BU, GL, HE, WD
PRI	Mejor DER2_WD: Iguales: Peor DER2_WD:	BN, VO CR, PI BU, GL, HE, WD
DEV	Mejor DER2_WD: Iguales: Peor DER2_WD:	BN, VO CR, WD BU, GL, HE, PI
LEE	Mejor DER2_WD: Iguales: Peor DER2_WD:	BN, CR, GL, HE, PI, VO, WD  BU
TEK	Mejor DER2_WD: Iguales: Peor DER2_WD:	BN, PI, VO CR, GL, HE BU, WD
SAN	Mejor DER2_WD: Iguales: Peor DER2_WD:	BN, PI, VO CR BU, GL, HE, WD
TEKA	Mejor DER2_WD: Iguales: Peor DER2_WD:	BN, VO  BU, CR, GL, HE, PI, WD
TEKB	Mejor DER2_WD: Iguales: Peor DER2_WD:	BN, VO GL BU, CR, HE, PI, WD
DER2	Mejor DER2_WD: Iguales: Peor DER2_WD:	BN GL, PI BU, CR, HE, VO, WD
DER3	Mejor DER2_WD: Iguales: Peor DER2_WD:	CR, GL, PI, VO, WD BN, BU, HE
MAO	Mejor DER2_WD: Iguales: Peor DER2_WD:	CR, HE, PI VO BN, BU, GL, WD
UTA	Mejor DER2_WD: Iguales: Peor DER2_WD:	  BN, BU, CR, GL, HE, PI, VO, WD
CIB	Mejor DER2_WD: Iguales: Peor DER2_WD:	CR, HE, PI, VO GL BN, BU, WD
TEKE	Mejor DER2_WD: Iguales: Peor DER2_WD:	GL, HE, PI, VO  BU, WD
YOU	Mejor DER2_WD: Iguales: Peor DER2_WD:	PI, VO CR, HE BN, WD

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

Método DER3 con Weight-Decay	
BL1	Mejor DER3_WD: BN, PI Iguales: CR Peor DER3_WD: BU, GL, HE, VO, WD
BL2	Mejor DER3_WD: BN, PI Iguales: Peor DER3_WD: BU, CR, GL, HE, VO, WD
CLO	Mejor DER3_WD: BN, PI Iguales: CR Peor DER3_WD: BU, GL, HE, VO, WD
PRI	Mejor DER3_WD: BN, PI Iguales: CR Peor DER3_WD: BU, GL, HE, VO, WD
DEV	Mejor DER3_WD: BN, PI, WD Iguales: CR Peor DER3_WD: BU, GL, HE, VO
LEE	Mejor DER3_WD: BN, BU, CR, GL, HE, PI, VO, WD Iguales: Peor DER3_WD:
TEK	Mejor DER3_WD: BN, PI Iguales: BU, CR Peor DER3_WD: GL, HE, VO, WD
SAN	Mejor DER3_WD: BN, PI Iguales: CR Peor DER3_WD: BU, GL, HE, VO, WD
TEKA	Mejor DER3_WD: BN, PI Iguales: Peor DER3_WD: BU, CR, GL, HE, VO, WD
TEKB	Mejor DER3_WD: BN, PI Iguales: Peor DER3_WD: BU, CR, GL, HE, VO, WD
DER2	Mejor DER3_WD: BN, PI, WD Iguales: Peor DER3_WD: BU, CR, GL, HE, VO
DER3	Mejor DER3_WD: Iguales: Peor DER3_WD: BN, BU, CR, GL, HE, PI, VO, WD
MAO	Mejor DER3_WD: CR, PI, WD Iguales: HE Peor DER3_WD: BN, BU, GL, VO
UTA	Mejor DER3_WD: PI, WD Iguales: Peor DER3_WD: BN, BU, CR, GL, HE, VO
CIB	Mejor DER3_WD: CR, PI, WD Iguales: BU Peor DER3_WD: BN, GL, HE, VO
TEKE	Mejor DER3_WD: BU, PI, VO, WD Iguales: Peor DER3_WD: GL, HE
YOU	Mejor DER3_WD: PI Iguales: CR Peor DER3_WD: BN, HE, VO, WD

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

		Método MAO con Weight-Decay
BL1	Mejor MAO_WD: Iguales: Peor MAO_WD:	BN, CR, VO GL BU, HE, PI, WD
BL2	Mejor MAO_WD: Iguales: Peor MAO_WD:	WD BN, GL BU, CR, HE, PI, VO
CLO	Mejor MAO_WD: Iguales: Peor MAO_WD:	BN, CR, VO GL BU, HE, PI, WD
PRI	Mejor MAO_WD: Iguales: Peor MAO_WD:	BN, CR, VO GL BU, HE, PI, WD
DEV	Mejor MAO_WD: Iguales: Peor MAO_WD:	BN, CR, WD GL BU, HE, PI, VO
LEE	Mejor MAO_WD: Iguales: Peor MAO_WD:	BN, BU, CR, GL, PI, VO, WD HE
TEK	Mejor MAO_WD: Iguales: Peor MAO_WD:	BN, CR, GL, VO BU, HE, PI, WD
SAN	Mejor MAO_WD: Iguales: Peor MAO_WD:	BN, CR, VO GL BU, HE, PI, WD
TEKA	Mejor MAO_WD: Iguales: Peor MAO_WD:	BN, GL, WD BU, CR, HE, PI, VO
TEKB	Mejor MAO_WD: Iguales: Peor MAO_WD:	BN, GL, VO BU, CR, HE, PI, WD
DER2	Mejor MAO_WD: Iguales: Peor MAO_WD:	BN, GL, WD BU, CR, HE, PI, VO
DER3	Mejor MAO_WD: Iguales: Peor MAO_WD:	BN, CR, GL, HE, PI, VO, WD BU
MAO	Mejor MAO_WD: Iguales: Peor MAO_WD:	CR GL, WD BN, BU, HE, PI, VO
UTA	Mejor MAO_WD: Iguales: Peor MAO_WD:	HE, WD BN, GL BU, CR, PI, VO
CIB	Mejor MAO_WD: Iguales: Peor MAO_WD:	CR, GL VO, WD BN, BU, HE, PI
TEKE	Mejor MAO_WD: Iguales: Peor MAO_WD:	GL, PI, VO BU, HE, WD
YOU	Mejor MAO_WD: Iguales: Peor MAO_WD:	CR VO BN, HE, PI, WD

**Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).**

		Método UTA con Weight-Decay
BL1	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, PI, VO CR, GL BU, HE, WD
BL2	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, VO GL, PI BU, CR, HE, WD
CLO	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, HE, PI, VO CR, GL BU, WD
PRI	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, PI, VO CR, GL BU, HE, WD
DEV	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, VO, WD CR, GL, PI BU, HE
LEE	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, BU, CR, GL, HE, PI, VO, WD
TEK	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, GL, PI, VO CR BU, HE, WD
SAN	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, HE, PI, VO CR, GL BU, WD
TEKA	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, VO GL, PI BU, CR, HE, WD
TEKB	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, GL, HE, VO PI BU, CR, WD
DER2	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, GL, PI, WD VO BU, CR, HE
DER3	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, BU, CR, GL, HE, PI, VO, WD
MAO	Mejor UTA_WD: Iguales: Peor UTA_WD:	CR, PI, WD GL, VO BN, BU, HE
UTA	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, HE, WD GL, PI BU, CR, VO
CIB	Mejor UTA_WD: Iguales: Peor UTA_WD:	CR, GL, PI, VO, WD BN, BU, HE
TEKE	Mejor UTA_WD: Iguales: Peor UTA_WD:	GL, HE, PI, VO, WD BU
YOU	Mejor UTA_WD: Iguales: Peor UTA_WD:	BN, PI, VO CR HE, WD

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

Método CIB con Weight-Decay		
BL1	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, HE, VO GL, PI BU, CR, WD
BL2	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, VO GL BU, CR, HE, PI, WD
CLO	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, VO GL, PI BU, CR, HE, WD
PRI	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, VO GL, PI BU, CR, HE, WD
DEV	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, VO, WD GL BU, CR, HE, PI
LEE	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, BU, CR, GL, HE, PI, VO, WD
TEK	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, GL, PI, VO, WD HE BU, CR
SAN	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, PI, VO GL BU, CR, HE, WD
TEKA	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, VO GL BU, CR, HE, PI, WD
TEKB	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, GL, VO BU, CR, HE, PI, WD
DER2	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, GL, VO, WD PI BU, CR, HE
DER3	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, CR, GL, PI, VO, WD BU, HE
MAO	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, HE, PI, VO GL BU, CR, WD
UTA	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, BU, CR, HE, PI, WD GL, VO
CIB	Mejor CIB_WD: Iguales: Peor CIB_WD:	BN, GL, HE, PI, VO CR BU, WD
TEKE	Mejor CIB_WD: Iguales: Peor CIB_WD:	GL, HE, PI, VO, WD BU
YOU	Mejor CIB_WD: Iguales: Peor CIB_WD:	PI, VO, WD HE BN, CR

Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

Método TEKE con Weight-Decay		
BL1	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	GL BU, HE, PI, VO, WD
BL2	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	GL BU, HE, PI, VO, WD
CLO	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	GL BU, HE, PI, VO, WD
PRI	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	GL BU, HE, PI, VO, WD
DEV	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	GL BU, HE, PI, VO, WD
LEE	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	BU, GL, PI, VO HE, WD
TEK	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	GL BU HE, PI, VO, WD
SAN	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	HE GL BU, PI, VO, WD
TEKA	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	GL BU, HE, PI, VO, WD
TEKB	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	GL, HE BU, PI, VO, WD
DER2	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	GL BU, HE, PI, VO, WD
DER3	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	BU, GL, HE, PI, VO, WD
MAO	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	GL BU, HE, PI, VO, WD
UTA	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	HE GL BU, PI, VO, WD
CIB	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	GL BU HE, PI, VO, WD
TEKE	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	BU, GL, HE, PI, VO WD
YOU	Mejor TEKE_WD: Iguales: Peor TEKE_WD:	HE, PI, VO, WD



Tabla 58. Comparación de métodos ARNMF y ARNMF&WD. (Continuación).

		Método YOU con Weight-Decay
BL1	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, HE, PI, VO CR WD
BL2	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, VO PI CR, HE, WD
CLO	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, PI, VO CR HE, WD
PRI	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, HE, PI, VO CR WD
DEV	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, VO CR, PI HE, WD
LEE	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, CR, HE, PI, VO, WD
TEK	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, PI, VO CR HE, WD
SAN	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, PI, VO CR HE, WD
TEKA	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, VO PI CR, HE, WD
TEKB	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, VO PI CR, HE, WD
DER2	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, PI CR, HE, VO, WD
DER3	Mejor YOU_WD: Iguales: Peor YOU_WD:	BN, CR, PI, VO HE WD
MAO	Mejor YOU_WD: Iguales: Peor YOU_WD:	CR, HE, PI BN, VO, WD
UTA	Mejor YOU_WD: Iguales: Peor YOU_WD:	PI BN, CR, HE, VO, WD
CIB	Mejor YOU_WD: Iguales: Peor YOU_WD:	CR, HE, PI VO BN, WD
TEKE	Mejor YOU_WD: Iguales: Peor YOU_WD:	HE, PI, VO WD
YOU	Mejor YOU_WD: Iguales: Peor YOU_WD:	PI CR, VO BN, HE, WD