UNIVERSITAT POLITÈCNICA DE CATALUNYA

Departament de Llenguatge i Sistemes Informàtics
Ph.D. Programme: Artificial Intelligence

SYMBOLIC AND CONNECTIONIST LEARNING TECHNIQUES FOR GRAMMATICAL INFERENCE

Autor: René Alquézar Mancho
Director: Alberto Sanfeliu Cortés

March 1997

Chapter 10

Conclusions and future research

In this chapter, the main results of Chapters 5 to 9 are summarized, together with the conclusions drawn from them, the related topics that need further research are specified, and the major contributions of the thesis are enumerated. In addition, the applicability of the developed tools to computer vision problems is discussed, and some already reported applications are briefly described.

10.1 Summary of main results and conclusions

The main results and conclusions of the work, which are included in the following subsections, are split into three subjects:

- symbolic methods for regular grammatical inference (Chapter 5);
- connectionist and hybrid methods for regular grammatical inference (Chapters 6 and 7);
- augmented regular expressions and their inductive inference (Chapters 8 and 9).

10.1.1 Symbolic methods for regular grammatical inference

A type of finite-state machines called *unbiased finite-state automata* (UFSAs), which is a simple extension of classical ("positively-biased") FSAs, has been introduced to represent both data and hypotheses in the problem of regular grammatical inference (RGI) from positive and negative examples. The distinguished features of UFSAs are their symmetry with respect to the representation and processing of positive and negative information, and their capability to discriminate between the negatively classified strings (rejected language) and the subset of strings over the given alphabet whose classification is still uncertain (ignored language). These properties make UFSAs well-suited to constitute the hypothesis space for the problem of RGI from positive and negative examples, allowing

- a formal, explicit and symmetrical inconsistency analysis;
- the generalization of common structures in negative examples (in the same way that is done for positive examples);
- a good framework where to define inductive biases and positive or negative constraints based on a-priori knowledge.

The basic theory for the RGI problem has been reformulated in terms of the representation provided by UFSAs, and the search space has been properly identified.

The (non-incremental) RPNI algorithm by Oncina and García [OnGa:92b] has been generalized using UFSAs to enable the introduction of a variety of inductive biases. One such bias is given by maximizing the generalization of the positive and negative examples in a symmetrical way, taking advantage of the UFSA representation. The resulting method [AlSa:95a] keeps the property of *identification in the limit* and a time complexity that is cubic in the size of the prefix tree of the sample. However, the experiments performed using sparse samples of fifteen regular languages have not

shown a learning improvement (measured by the classification rate of the inferred automaton) with respect to the original RPNI method, although quite good results have been obtained anyway. The new method is able to outperform the original one in the cases where the given positive sample is not structurally complete with respect to the canonical DFA of the target language, but the full (positive and negative) sample is structurally complete with respect to the corresponding canonical UFSA.

The problem of incremental learning of regular languages from both positive and negative examples has been studied and new pseudo-incremental polynomial-time methods have been proposed. It has been claimed that a fully incremental approach, such that the whole sample were not stored, would not ensure the consistency of the hypotheses with previous data, unless trivial (canonical automata) or tricky (generalizations with exception lists of inconsistent examples) solutions were returned. Hence, a pseudo-incremental general algorithm for RGI using UFSAs has been described, in which the whole sample is stored in the form of a prefix tree UFSA, and a record of the partition of its states that corresponds to the current hypothesis is maintained [AlSa:95a].

Two pseudo-incremental methods have been designed and tested, that always return a consistent deterministic UFSA (DUFA), but which differ in the split-and-merge procedure that is performed when an example inconsistent with the current hypothesis is supplied. The first one, *SM_1*, which is characterized by a *maximal splitting* of any current inconsistent hypothesis, returns in the very most part of cases the same result than the non-incremental algorithm, while reducing the average-case time complexity for a sequential presentation. The complexity of processing a new string s by this algorithm is $O(l)$, $O(l \cdot |U|^2)$, or $O(|PTU(S)|^3)$, for consistent, ignored, or inconsistent strings, respectively; where l is the length of s, $|U|$ is the number of states of the current UFSA hypothesis, and $|PTU(S)|$ is the number of states of the sample prefix tree. Furthermore, this pseudo-incremental method keeps the identification in the limit property, i.e. it still guarantees the convergence to the target language if some finite set of representative examples is included in the sample. The size of the representative sample is of $O(n^2)$, where n is the number of states of the target UFSA.

The second pseudo-incremental method, *SM_2*, which is characterized by a conservative *minimal splitting* of any current inconsistent hypothesis, significantly reduces the computation time for large samples (with $O(l \cdot |PTU(S)| + |U|^3)$ worst time complexity), but unfortunately it loses the identification in the limit property. Indeed, the experimental results have shown that this conservative method tends to develop excessively complex solutions whenever the correct minimal UFSA is not obtained in the earliest stages. As a consequence, the classification rates of the inferred UFSAs are clearly worse than those of the *SM_1* method.

It can be concluded that the first pseudo-incremental method provides, in some sense, the *best* (simplest) UFSA inferred from a sequence of positive and negative examples among the family of split-and-merge based methods, but its worst-case time complexity, which is $O(|PTU(S)|^3)$, can be a practical impediment for large samples of long strings. In order to overcome this drawback, a two-phase compromise approach could be suggested, in which the *maximal splitting SM_1* method were used until a significant number of examples were entered, and then the more efficient *minimal splitting SM_2* method were used to adapt the reached hypothesis, that is expected to be approximately correct, to the subsequent examples.

Finally, it should be remarked that both the RGI theory and methods using UFSAs presented in Chapter 5 may be extended for the problem of inducing a *C-class UFSA* from string examples belonging to C distinct non-overlapping classes. This would be useful to infer a single classifier for C-class pattern recognition problems.

10.1.2 Connectionist and hybrid methods for regular grammatical inference

The inference of regular language acceptors from examples using *recurrent neural networks* (RNNs) has been studied, and the learning and generalization capabilities of different types of RNNs have been investigated, following the two usual connectionist approaches to RGI, namely,

1) learning the *next-symbol prediction task* for RGI from positive examples, and
2) learning the *string classification task* for RGI from both positive and negative examples.

For each approach, a method to *extract* an UFSA from a trained RNN has been proposed[1]. Both extraction methods are based on a *hierarchical clustering* of the recurrent hidden unit activations that are produced when the training set is presented to the network, but they differ in the criterion used to stop the merging of clusters.

For the first case above, and to avoid the extraction of a universal automaton, the hierarchical clustering is ended when the distance between the two closest clusters exceeds some threshold, which depends on the dimensionality of the state space (the number of recurrent hidden units). For the second case, the hierarchical clustering

[1]For the first approach (only positive examples) it is equivalent in practice to extract UFSAs or common FSAs.

may be ended when merging the states associated with the two closest clusters yields an inconsistency with the data (i.e. acceptance or rejection of both a positive and a negative example) [AlSa:94b], but a less strict stop criterion has also been proposed that allows some number k of tentative merges at each step. In any case, the extraction of a consistent deterministic automaton (a DUFA that belongs to $Lat(PTU(S))$, the lattice of UFSAs that cover the sample prefix tree) is guaranteed after a single clustering process, this being an improvement with respect to previously reported methods [DaDa:91, GiOm:93, MaFa:94]. Moreover, if RNNs are trained to classify the strings of a sample $S = (S^+, S^-)$, then a DUFA in the deterministic border set $DBS_{PTU}(S)$ (i.e. one of the maximal consistent generalizations) can be extracted by adjusting conveniently the parameter k.

A new feature in the proposed extraction methods is the use of symbolic representation (UFSAs) and processing (*merge* operations) along with the clustering process on the activation space, so that the extraction procedure can be regarded as a state merging process from the sample prefix tree, which is guided by the internal state representation of the trained network. Hence, by considering the two stages of neural learning and UFSA extraction as a single inference process, a hybrid connectionist-symbolic RGI method is obtained, which may be included in the class of heuristic RGI techniques based on selecting a partition of the states of a canonical automaton. However, an outstanding difference should be pointed: the inductive bias is not predetermined explicitly as in the symbolic heuristic methods but determined implicitly by several factors, which include the network size and configuration, the parameters of the neural learning and extraction algorithms, and the statistical properties of the given data. On the other hand, the use of UFSAs facilitates to establish a parallelism between the connectionist and symbolic representations, because UFSAs are able to distinguish between positive, negative and uncertain string classification, in the same manner than RNNs acting as acceptors with a tolerance threshold $\epsilon < 0.5$.

Another topic studied in Chapter 6 has been the influence of the activation function on the computational and learning capabilities of some RNN models, namely, the *single-layer RNN* (or SLRNN) and the *augmented SLRNN* (or ASLRNN) architectures. The analysis of the sensitivity of the network current state, represented by the activations of the recurrent hidden units, with respect to the past history of inputs has revealed that the derivative of the activation function plays an important role in the capacity of the network to take into account ("remember") past events. If the derivative function is significantly greater than zero only for a small interval of the activation function domain, as in the case of the *sigmoid*, then whenever the net-input values of the units lay out of this interval (i.e. the units are saturated), the current state of the network will only depend on the very recent inputs.

Consequently, RNNs with a sigmoid activation function in the recurrent hidden units will typically have a very short memory of previous inputs to perform and learn a trained task. This is an important drawback, since for certain tasks (e.g. prediction in sequences with embedded structures), distant inputs may be relevant to determine the right response of the network at present time. Therefore, by using other non-linear activation functions with a less flat derivative, the learning performance of RNNs should improve, specially for complex tasks. The *antisymmetric logarithm* is such an activation function that has been suggested for the recurrent hidden units in RNNs [AlSa:94a, SoAl:94].

For ASLRNNs, the activation function used in the non-recurrent layer(s) must help the network to learn and approximate a proper output function for the trained task. Hence, activation functions that are adequate for *multilayer feedforward* (MLFF) networks to learn and approximate static mappings should also be adequate for the non-recurrent layers in ASLRNNs. Several interrelated arguments have been given to prefer a *sinusoidal* activation function rather than a sigmoid or a Gaussian function in MLFF nets:

- a net with sinusoidal units is more powerful to make discriminations on an input space, due to the non-monotonicity and periodicity of the sine function, thus easing the learning of discrete classification mappings (e.g. XOR);
- MLFF nets with sinusoidal hidden units can be seen as generalized discrete Fourier series with adjustable frequencies, and therefore, they are also good approximators for continuous mappings;
- the non-flat derivative of the sine function helps to backpropagate errors for weight adjustment during gradient-descent learning, and it contributes to the formation of a smooth landscape (without plateau regions, but possibly many minima) in the error-from-weights function.

For these reasons, MLFF nets and ASLRNNs with sinusoidal units should be more powerful and learn much more quickly than their counterparts with sigmoid units and same size, although, on the other hand, they could generalize worse due to a possible overfitting of the training set.

To test the preceding hypotheses, an empirical study has been carried out that has clearly confirmed the beneficial effects of the proposed alternatives on the learning performance of RNNs. Three first-order RNN architectures (SLRNNs, 2L- and 3L-ASLRNNs) were applied to learn the *next-symbol prediction task* from random sequences of positive strings for the two Reber grammars used as benchmark in several works [SeCM:88, SmZi:89, Fahl:91]. For each model, the effect on the learning performance of the network caused by changing the activation function in two types of units (the recurrent hidden units, which represent state information, and the rest of units, which are aimed at the output function) was evaluated. The comparative

results have shown an impressive improvement in the learning performance of the three architectures, in terms of convergence rate and required training time, when the sigmoid function is replaced by antisym-log and sinusoidal activation functions in those two types of units, respectively, thus confirming the conjectures raised previously. In addition, the results obtained by means of these substitutions and the use of a learning algorithm with true gradient computation [AlSa:94a, SoAl:94] have been notably better than those reported in previous studies on the same problem with different RNNs and sigmoid units [ClSM:89, SmZi:89, Fahl:91b]. This has been particularly true in the case of the symmetrical Reber grammar, for which the memory of a distant input is required to predict correctly.

Just substituting the sigmoid by the anti-symmetric logarithm in the recurrent hidden units also improved the learning behavior of the nets, specially for ASLRNNs. The further improvement caused by the replacement of a sigmoid by a sinusoidal activation function in the rest of units was stronger for 3L-ASLRNNs than for 2L-ASLRNNs. This result agrees with our expectations that, the greater the number of feedforward layers in the net, the bigger the difference in error backpropagation ability between sine-based and sigmoid-based networks, due to the already commented issue of the derivative profile.

The results of the experiments on the Reber grammars have also demonstrated the clearly superior performance achieved by using a full-gradient instead of a truncated-gradient learning algorithm (like the one used in [SeCM:88, Elman:90]) for training ASLRNNs. Indeed, the symmetrical Reber grammar could only be learned when using the former. Curiously, for this grammar, $2L_s$-ASLRNNs learned much better than SLRNNs with the same number of hidden and output units, when both were trained by a true gradient descent algorithm and the sigmoid activation function was substituted in the recurrent units. This would give empirical support to the theoretical findings reported about the necessity of augmenting first-order SLRNNs with an output layer to represent any finite-state machine [AlSa:95b].

Another experimental study has been carried out (Chapter 6) to test the use of RNNs for RGI from positive and negative examples [AlSS:97] and to compare their performance with the one displayed by the UFSA-based symbolic methods on the same benchmark data [AlSa:97b]. Both first- and second-order 2L-ASLRNNs (with the antisym-log activation function in the recurrent layer and either a sigmoid or a sinusoidal activation function in the output layer of one unit) were trained to *classify* sparse samples of 15 target regular languages. The 2L-ASLRNNs were trained by a true gradient-descent algorithm, and consistent DUFAs were extracted from them using the UFSA extraction method aforementioned. The generalization performance of both the trained networks and the extracted DUFAs was evaluated on the benchmark test samples.

First- and second-order 2L-ASLRNNs performed similarly, but networks with the sigmoid output unit generalized significantly better than those with the sinusoidal output unit. This is an interesting result, because nets with sinusoidal units were shown in the previous study to learn more easily than nets with sigmoid units, and therefore it appears that there is an inverse relationship between the learning (i.e. training set approximation) and generalization capabilities, as one might expect according to the theory of data approximation and extrapolation. First- and second-order 2L-ASLRNNs with a sigmoid output unit achieved approximately a 78% of total correct classification rate, but they classified correctly the full test sample only in a 5% (respectively 7%) of the runs.

These results were improved by the extracted DUFAs, which reached an 82% of total correct classification rate on the test samples and a 19% of full success rate, that coincides in this case with the percentage of runs in which the target automaton was inferred (identification rate). This validates empirically the UFSA extraction method proposed and confirms the beneficial effects of extracting an automaton from an RNN with continuous activation function: not only a symbolic representation of the inference result is obtained, but also the generalization performance is improved and the problem of bad classification of long strings due to drifting activations is avoided.

However, even after DUFA extraction, the results of the connectionist RGI methods tested have not been so good as those obtained for the same data by the RPNI algorithm and two of the symbolic RGI methods proposed using UFSAs. For comparison, the RPNI algorithm, which yielded the best results, reached a 96% of total correct classification rate on the test samples and a 78% of identification rate.

The small number of examples in the benchmark training sets (37 strings in average) could be the cause of the middling results obtained by the RNN-based approaches, and it is conjectured that a quite higher generalization performance (above 95%) might be achieved by 2L-ASLRNNs by increasing enough the size of the training set. This hypothesis is supported by the results reported in some previous studies on inferring the Tomita's languages using RNNs carried out by other researchers [WaKu:92, MiGi:93]. Hence, it seems that, at least for the simple regular grammars tested, connectionist RGI methods need larger samples than symbolic RGI methods to reach a similar level of generalization performance. This behavior may be explained by the fact that connectionist methods can be regarded indeed as statistical methods.

An algebraic linear framework to represent finite state machines (FSMs) in discrete-time *single-layer recurrent neural networks* (SLRNNs), which has been termed an FS-SLRNN model, has been presented in Chapter 7 [AlSa:95b]. This model is based on the transformation of the nonlinear constraints imposed on the network dynamics by the given FSM and data encoding into a set of static linear equations to be satisfied by

the network weights. This transformation allows an exact emulation of the FSM by a network built with calculated weights, whenever some stability conditions, which have been stated, are met. Otherwise the linear model is just a static approximation of the network dynamics.

It has been proved, using the FS-SLRNN model, that first-order SLRNNs have some limitations in their representation capability, which are caused by the existence of some linear relations that always hold among the equations associated with the state transitions. In order to overcome this problem, a first-order SLRNN may need to represent a larger equivalent machine with split states. Furthermore, first-order SLRNNs need to be augmented (e.g. with an output layer) to be able to represent every output mapping. According to these requirements, Minsky's method for FSM implementation in first-order augmented SLRNNs [Mins:67] has been generalized.

On the other hand, it has been demonstrated that second-order (or higher-order) SLRNNs can easily implement all the FSMs, since their corresponding FS-SLRNN models, given an orthogonal data encoding, are characterized by the full rank of the system matrix. The actual requirements on the network activation function have been determined, and these have been shown to be quite weak, i.e. a large spectrum of activation functions can be used for FSM implementation in SLRNNs.

The FS-SLRNN model can be used to insert symbolic knowledge into discrete-time SLRNNs and ASLRNNs prior to neural learning from examples. This can be done by initializing the network weights to any of the possible solutions of an underdetermined linear system representing the inserted (partial) FSM with an excess of recurrent units. In comparison with other methods that insert FSMs into RNNs [FrGM:91,OmGi:96], the method proposed here is more general, since it can be applied to a wide variety of both activation and aggregation functions. Moreover, a new distinguishing feature of our method is that it allows the inserted rules to be kept during subsequent learning. To this end, a constrained learning procedure has been devised, in which a subset of free weights are trained using a conventional neural learning algorithm and the rest of weights are updated to force the satisfaction of the linear constraints in the system solution.

Finally, a hybrid connectionist-symbolic methodology for RGI, called *active grammatical inference* (AGI) [SaAl:95], has been described in Chapter 7 which combines the FSM insertion, FSA extraction, and neural learning techniques proposed in this work. The AGI methodology encompasses a large class of heuristic connectionist and hybrid RGI approaches, with the novel feature that the learning process performed by the neural network can be guided using symbolic information (a-priori knowledge). In the most general case, the GI process is conceived as a sequence of learning cycles, using or not a-priori knowledge, but many RGI methods that perform a single inference cycle

are covered as well by the AGI general procedure as particular cases. The feasibility of learning a target DFA in two cycles following all the steps involved in the AGI procedure has been shown.

10.1.3 Augmented regular expressions and their inductive inference

The *augmented regular expressions* (AREs) have been defined, which are compact and intelligible descriptions that represent an interesting non-trivial class of context-sensitive languages (CSLs) capable of expressing complex context relationships, e.g. planar shapes with symmetries and relative size constraints [AlSa:97a]. An ARE is based on an underlying regular expression, in which the star symbols are replaced by natural-valued variables that must satisfy a (maybe empty) set of constraints (linear equations and lower bounds). The recognition of a string as belonging to the language described by an unambiguous ARE, which is based on parsing by the underlying RE and testing the ARE constraints, is efficient, to the contrary of parsing by a context-sensitive grammar [AlSa:95c, SaAl:96]. It has been proved that AREs are not able to represent all CSLs nor all the context-free languages (CFLs), but on the other hand, the class of languages described by AREs properly contains the classes of regular languages (in an obvious way) and pattern languages. The ARE representation may be extended in several ways. For example, the definition of non-linear constraints could be allowed (e.g. quadratic equations); this would not pose special difficulties for recognition, but it would highly complicate the learning procedure. To cope with noisy data, a robust recognizer should use an error-correcting regular parser and a tolerant constraint checker (still to be defined).

A general approach to learn AREs from examples has been proposed that is based on splitting the process in two stages: inferring the underlying RE (through some RGI technique) and inducing the maximal number of constraints afterwards (through the use of a method that has been given) [AlSa:96]. This learning strategy is not conceived as an identification method, but a heuristic method in which the inferred ARE strongly depends on the result of the RGI step. However, it has been demonstrated that if the underlying RE were identified, a target ARE would be identified in the limit using this approach, since the constraint induction process finds the ARE describing the smallest language that contains the examples among the AREs based on the same RE.

On the other hand, if negative examples are supplied, then the learning method cannot know which of them should belong to the underlying regular language and which should not, unless an informant classified them in advance. If this information is not available, several learning cycles may be needed in general to obtain a data compatible

ARE. In the first cycle, only the positive examples are taken into account. Hence, a heuristic or characterizable algorithm for RGI from positive examples has to be used in that step, and it should be biased to infer preferrably small-size descriptions (DFAs or REs) representing a high generalization with respect to the sample, both to ease the parsing of examples by the RE and to permit the discovery of the target constraints. The constraint induction could be impeded if the inferred regular language were too restricted to the given examples (sample overfitting).

A specific method for inferring AREs has been described, in which the RGI procedure consists of three steps: training a recurrent neural network from the positive examples to learn the next-symbol prediction task, extracting a DFA from the net, and selecting an RE equivalent to the extracted DFA [AlSC:96]. This method has been implemented and applied to the inference of eight CSLs describing some planar shapes and signal patterns. The inferred AREs classified quite correctly samples of positive and negative strings not used during learning. However, although the test languages were rather simple, the identification of the target ARE was rarely accomplished.

It must be noticed that the cost of the DFA-to-RE transformation in the worst case is exponential in the number of states (e.g. for a fully-connected DFA). Although most of the DFAs that are typically induced from object contours or other physical patterns are sparsely connected and present a quite limited degree of circuit embedment, thus allowing the computation of the equivalent RE, it is clear that the worst-case behavior may defeat the learning method. A possible alternative is to use an RGI method that directly returns a regular expression (e.g. the uv^kw algorithm [Micl:76]). Finally, it must be remarked that the ARE learning strategy proposed is maybe the most promising attempt by now to infer a large class of CSL acceptors from examples, since there has been a lack of methods to learn contex-sensitive grammars and other CSL acceptors in the field of grammatical inference.

10.2 Applications and future research

The application of the learning techniques developed in this work to pattern recognition problems in computer vision is foreseen, and indeed, it has already been started as commented hereinafter. Grammatical inference methods have been applied by other researchers to speech recognition problems (as reviewed in Chapter 1), and therefore, this is another possible field of application, although we still do not have any experience in it. Likewise, some of the techniques proposed can be adapted for natural language processing tasks; for instance, recurrent neural networks can be used for language parsing and translation, and stochastic automata can be inferred and used as probabilistic models of natural languages.

In the case of computer vision applications of GI, the aim is the automatic generation of models from positive and negative samples and maybe some (but not much refined) a-priori knowledge, if available, to be used in pattern recognition tasks. It must be noted that the models returned by the GI techniques studied here are one-dimensional descriptions, while in some problems 2D or 3D models may be needed. Nonetheless, 1D-models can describe contours of planar shapes and classes of such contours, and thus, the inference of these models can be useful for several applications like character recognition. In these cases, the selection of adequate primitives is of outmost importance.

Font and Sanfeliu have been working in the selection of adequate primitives for contours of curved planar shapes, and preliminary results on the problem of digit recognition have been recently reported using these primitives and the AGI methodology [FoSa:97]. On the other hand, Sainz and Sanfeliu have studied how to extend the CSL 1D-models given by AREs to 2D object models, giving rise to a type of representations called pseudo-bidimensional AREs (or PSB-AREs) [SaSa:96a]. In a PSB-ARE, every row of the 2D model is described by an ARE over an alphabet of symbols representing pixel features (e.g. color), and a further ARE over an alphabet of symbols representing different row-modelling AREs is used to describe some relationships among rows. The ARE learning techniques have been used to construct such models from images, and two different applications using PSB-AREs have been studied: recognition of traffic signs in outdoor scenes [SaSa:96a,SaSa:96b] and human face recognition [SaSa:97]. However, further work is needed (see below) both to widen the scope of computer vision applications and to cope with the noisy and incomplete data found in real problems. For example, an error-correcting parsing scheme for AREs would be very helpful to use AREs in recognition tasks.

Further research should be followed to deepen in some of the theoretical and practical aspects raised. The foreseen subjects of research are classified in four main lines as follows.

a) *Symbolic methods for regular grammatical inference*

- To extend the theory and methods of RGI from positive and negative data using UFSAs to the case of learning C-class UFSAs from positive examples of C classes, in order to approach the inference of a single acceptor for C-class recognition problems.
- To design and test new efficient algorithms for the inference of regular expressions from positive and negative examples (and from only positive examples as well), to avoid the exponential DFA-to-RE transformation in the inference of AREs.

b) *Connectionist methods for grammatical inference and recurrent neural networks*

- To design improved learning algorithms for RNNs, which may be based for instance on conjugate gradient and pseudo-Newton optimization techniques.
- To develop and test constructive RNN architectures, to drop the necessity of fixing the number of neurons in advance.
- To study the introduction of inductive biases in the error function to be minimized during neural training.
- To analyse the effect of data encoding on the learning performance of RNNs.
- To develop suitable architectures and learning methods for the connectionist inference of context-free and context-sensitive languages.

c) *Augmented regular expressions and possible extensions*

- To develop error-correcting parsing techniques for AREs.
- To devise "deformable" models based on AREs.
- To study the extension of the ARE representation and learning to the case of *non-linear* AREs, in which some type of non-linear constraints are allowed.
- To investigate the "augmentation" of the context-free expressions [Salo:73] with constraints, in a similar way to that carried out for REs, to obtain an even more powerful representation.
- To improve the bidimensional model based on AREs [SaSa:96a] and extend it to the 3D case, if possible, for computer vision applications.

d) *Applications*

- To improve and extend the application of the developed GI techniques in computer vision, for tasks such as handwritten character recognition, planar shape classification, and 2D object recognition.
- To study the application of the proposed GI methods for speech recognition.
- To study the application of RNNs for natural language translation tasks.

10.3 List of major contributions

The major contributions of the thesis are enumerated here below. The given numbering does not pretend to order the contributions according to their respective importance; the numbering is simply determined by the order in which they have been presented. Symbolic and connectionist learning techniques for grammatical inference have been studied, developed and tested, and the relationship between symbolic and connectionist representations has been investigated. Moreover, it can be seen that the objectives of the work, stated in Section 1.7, have been basically accomplished.

1. *Unbiased finite state automata* (UFSAs) have been proposed as an adequate type of representation for the problem of *regular grammatical inference* (RGI) from positive and negative examples, and the basic theory for the inductive inference of UFSAs has been established [AlSa:95a].
2. Both non-incremental and pseudo-incremental polynomial-time methods for inferring UFSAs from positive and negative examples have been designed and tested [AlSa:95a, AlSa:97b].
3. Two methods have been presented and empirically validated to extract a *deterministic* UFSA from a trained *recurrent neural network* (RNN), that are applicable respectively to the cases of RGI from only positive examples (*next-symbol prediction task*) and RGI from positive and negative examples (*string classification task*) [AlSa:94b].
4. The influence of the *activation function* on the behavior of RNNs has been studied, some alternatives to the commonly-used sigmoid function have been proposed (and justified) to improve the learning capability of RNNs, and this improvement has been confirmed empirically on a benchmark prediction task for RGI from positive examples [AlSa:94a, SoAl:94].
5. An experimental study has been performed using another benchmark to compare different connectionist and symbolic methods for RGI from positive and negative examples [AlSS:97, AlSa:97b].
6. An algebraic linear framework to represent *finite state machines* (FSMs) in discrete-time *single-layer recurrent neural networks* (SLRNNs) has been presented, which has been used to explain the different representational capabilities of first-order and higher-order SLRNNs and to generalize previous methods for implementing FSMs in SLRNNs and *augmented* SLRNNs (ASLRNNs) [AlSa:95b].
7. An FSM insertion method and a constrained neural learning procedure derived from the preceding algebraic model have been proposed: the former to insert a partial FSM into an ASLRNN (or a second-order SLRNN), and the latter to train the corresponding network from examples while keeping the inserted knowledge [AlSa:95b, SaAl:95].

8. A general methodology, called *active grammatical inference* (AGI), has been defined for the inference of UFSAs (or FSAs) from examples and a-priori knowledge (if available), that combines the proposed techniques for automata insertion, neural learning, and automata extraction, and allows the development of a variety of heuristic RGI methods with the common feature that RNNs are used as the basic learning tool [SaAl:95].
9. The *augmented regular expressions* (AREs) have been defined to represent a subclass of CSLs capable of expressing complex context relationships, and two efficient recognition methods for unambiguous AREs have been designed [AlSa:97a, AlSa:95c, SaAl:96].
10. The inductive learning of AREs has been studied, a general approach for their inference from examples has been proposed, and a specific learning method within this approach has been tested successfully [AlSa:96, AlSC:96].