INSS : an hybrid system for constructive machine learning

Fernando S. OSORIO [1,2], Bernard AMY [1]

[1] Laboratoire LEIBNIZ - IMAG -INPG
46, avenue Félix Viallet 38031 Grenoble Cedex 1 - FRANCE
Web : http://www-leibniz.imag.fr/RESEAUX/
E-mail : osorio@imag.fr - amy@imag.fr

[2] UNISINOS - Computer Science Dept.
Av. Unisinos, 950 - CP 275 - CEP 93022-000 RS- BRAZIL
Web : http://www.unisinos.tche.br/

Abstract

In this paper we present the INSS system, a new hybrid approach based upon the principles of KBANN networks. It represents an important improvement in comparison with its predecessor because the learning and the knowledge extraction process are faster and are accomplished in an incremental way . INSS offers a new approach applicable to constructive machine learning with high-performance tools, even in the presence of incomplete or erroneous data.

Keywords:

Constructive machine learning, hybrid neuro-symbolic systems, Cascade-Correlation algorithm, Artificial Neural Networks, ANN rule insertion and extraction.

1. Introduction

The main argument, and the most used one, to justify the study and the application of hybrid symboli-connectionist systems is the complementarity of symbolic AI methods and sub-symbolic connectionist methods (Artificial Neural Networks - ANN).

Such a justification is a very general one. And it remains to be more precise about the real contribution of the hybrid approach. What exactly provides the combination of neural networks and knowledge based systems ? Researchers claim that hybrid systems take advantage of their respective component strengths. Is it a real property of the existing hybrid system ? And what are these advantages ?

To validate an hybrid system, one have to answer these questions, and to describe what really can be done with this system which was hardly done with just one of its components. Particularly the system has to be given proof of the following properties :

- possibility to use and to take into account several kinds of knowledge representation.

- best efficiency of the global system when compared to each of its components.

- strong coupling between the components, leading to an exchange of knowledge between all of these components. This knowledge has to be proved consistent and useful. The best way for implementing such a coupling is to choose the integration mode called co-processing [10,11], in which the different component of the hybrid system work on the same level and exchange information between both themselves and their environment.

- possibility of global learning. The whole system is able to adapt its various sets of knowledge to the variations of the data domain. This tuning can be done following two ways : either learning (or forgetting) new examples, or modifying the architecture of the neural component.

In this paper we describe a system, called INSS (Incremental Neuro-Symbolic System [12,13]), endowed with these properties. In section 2 we explain the origin of our system and the reasons of our choices. Section 3 describes INNS system. Then, in section 4, to validate the system, we present some practical results allowing to show that INSS has the sought properties. Section 5 presents the application of INSS in a medical domain.

2. The co-processing integration mode.

In the classification task domain, the hybrid neuro-symbolic systems, such as SYNHESYS [8] and KBANN [22], exploit their capacity to use at the same time theoretical knowledge (set of symbolic rules) and empirical knowledge (set of observed examples). These two systems are significant examples of the coprocessing integration mode in hybrid systems, allowing a bi-directional knowledge transfer between the symbolic and connectionist modules. Figure 1 shows the general architecture of this kind of systems.

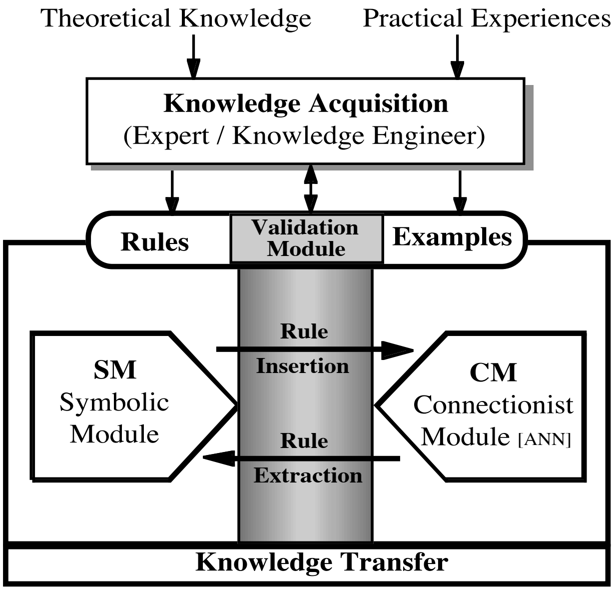

Figure 1 - Hybrid neuro-symbolic systems and knowledge transfer

We chose to base our study on the KBANN model, a well-known hybrid neuro-symbolic system that represents, among others, the state-of-the-art in this domain. This system is able to compile a knowledge base into the form of an ANN. Then, it learns from an example data set, and after that it extracts new rules. This approach allows a refinement of initial knowledge, as we can see in Figure 2. Such a system constructs robust networks: the insertion of a priori theoretical knowledge leads to quicker learning; we can use small data sets during the learning phase; all available knowledge about the problem (whether theoretical or empirical) is used; and thus the system is more adapted to process incomplete and/or erroneous data.

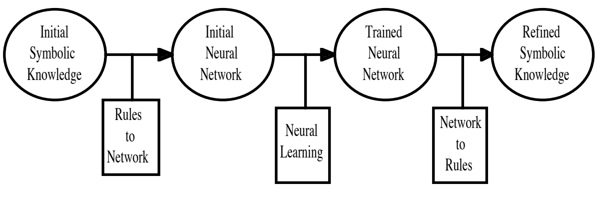

Figure 2 - Knowledge refinement using KBANN

However, the KBANN system has some important drawbacks due to the choice of its ANN model and learning method, the Back-Propagation algorithm [16]. It's the reason why we developed the new system called ***INSS*** to improve KBANN networks and to overcome its main limitations. This new system also authorises insertion, refinement and rule extraction, but, unlike the KBANN system, each process performs incrementally. Moreover, instead of using the Back-Propagation algorithm, based on static networks, INSS uses the Cascade-Correlation learning method [7] which proceeds by adding new units (neurons) during learning. Our approach allows to obtain a constructive network that is able to develop its structure and its knowledge, while keeping unchanged the principal properties of a hybrid neuro-symbolic system. The main feature, that constitutes the originality of our system, is that we are able to perform an incremental rule extraction [5]. We do not know any other neuro-symbolic system able to extract rules in a such incremental way.

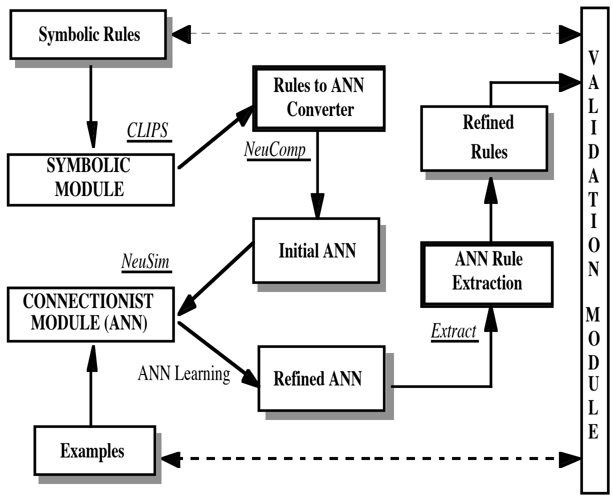

Figure 3 - INSS System: Constructive knowledge refinement

3. The INSS system

The *INSS* system is composed of five modules: *Symbolic-Module* (Symbolic Inference Engine), *NeuComp* (Construction of a network from rules), *NeuSim* (ANN learning and recall), *Extract* (Rule extraction), and *Valid* (Validation of acquired knowledge, by means of study of relations between rules and examples). The INSS system components are represented in Figure 3.

Our system uses the CLIPS language (*C Language Integrated Production System*) [9], developed by the STB-NASA, as its symbolic module. Our system also provides facilities to transfer rules and examples to/from the specific syntax used in this language and the syntax used in our tools (NeuComp/NeuSim/Extract). The

NeuSim module can be also used as a forward-chaining inference engine once the symbolic rules have been transferred to the connectionist module.

The NeuComp module can process elementary production rules of order 0 which are equivalent to IF/THEN forms such as:

IF <Condition>(TRUE/FALSE) *AND*/*OR*
  <Condition> (TRUE/FALSE)...
THEN <Conclusion>

The rule compilation follows the method described by Towell [22,24]. The result of the translation is a network composed of a set of units linked by weighted connections (see Figure 4). The activation of this network, before learning, leads exactly to the same results (outputs) as those obtained with the set of rules.

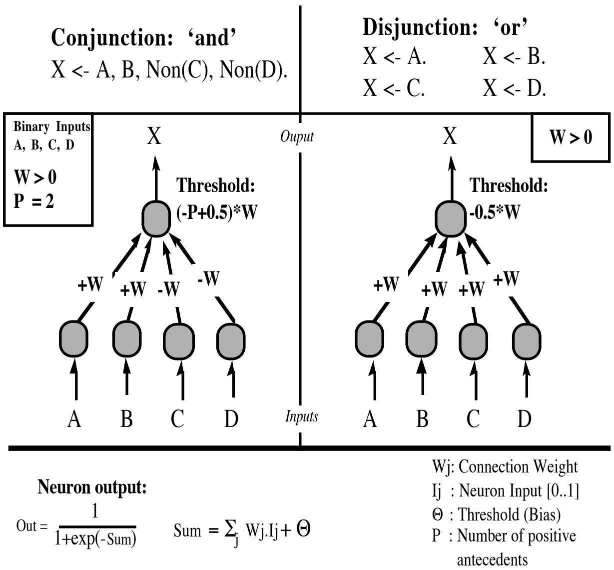

Figure 4 - Rules to network translation ("rule insertion")

We also extended the rules used by KBANN to allow the application of INSS to robotics problems and to study what we called "high level rules" [14,15]. Therefore, NeuComp accepts production rules of order 0^+ (rules including value intervals). We implemented the usage of comparison functions of the following type:

<Feature> <Operator> <Value> or
<Feature> <Operator> <Feature>,
where Operator is *GreaterThan, LessThan or Equal.*

Resulting in rules of this kind:
IF GreaterThan(Sensor_S1, 1.0) *AND*
  LessThan(Sensor_S1, Sensor_S2)
THEN Conclusion_C1

These rules can be compiled into an ANN composed by simple Perceptron like units (we create feed-forward multi-layer networks with sigmoïd based units). A detailed description of all compilation processes, used within INSS, can be found in [13].

As the symbolic rules allow to establish some initial knowledge and then give an initial structure to the network, this approach solves two important problems related to Artificial Neural Networks : on one hand this simplifies the choice of the number and distribution of units, on the other hand we obtain a good assignment of initial values to the connection weights.

The use of the Cascade-Correlation learning algorithm instead of Back-Propagation, in the NeuSim module, allows a quicker learning [7,17], with higher performance results [17,21]. Figure 5 shows an example of the network structure evolution when we apply the Cascade-Correlation learning algorithm. It allows especially constructive learning where the initial knowledge is not mixed with the new acquired knowledge. The importance of such a choice of the learning method is reinforced by studies [18,19] showing that Cascade-Correlation networks can be used to model some aspects of human cognitive development.

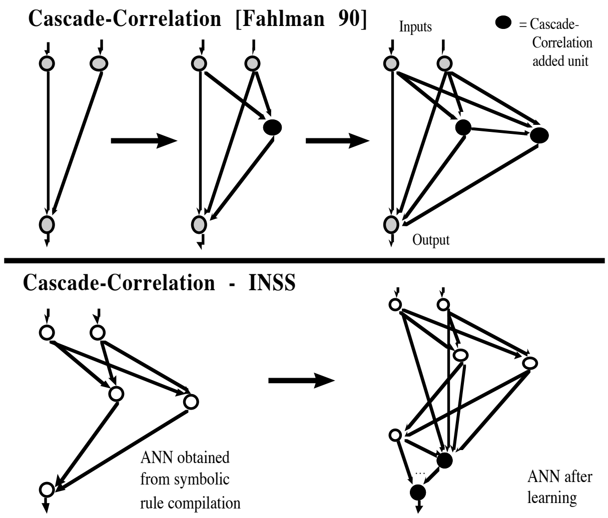

Figure 5 - ANN structure evolution using Cascade-Correlation

The Cascade-Correlation algorithm developed by Fahlman and Lebiere [7], in contrast to static neural learning algorithms such as Back-Propagation [16], is a generative technique to network construction and learning. Instead of merely adjusting weights in a network of fixed topology, Cascade-Correlation starts with a minimal network of input and output units. During learning, it may add hidden units one at a time, installing each on a separate layer. This is done in the following way: if the net is not reducing error fast enough with its current topology, it will select and install a new hidden unit whose output activations correlate best over all training cases with the existing network error. Once one new unit is installed in the network its weights are frozen, and this unit keeps unchanged its learned weights. So, Cascade-Correlation will reduce step-by-step the network output error by a cyclic process of output units learning and hidden unit addition/learning. In essence, Cascade-Correlation searches not only in weight space but also in the space of network topologies.

Learning in a network by adding new units allows to complete, to change, or to refine the initial knowledge. In INSS, using the Extract module, one can be able to analyse only the new added units and the modified output units. The old units always keep their function and their meaning in comparison with the initial rules introduced into the network. As we can preserve unchanged the initial knowledge acquired, this technique makes the main difference of our system in comparison with the KBANN system [5,6,22].

The Extract module [5,6] implements an improved version of the SUBSET algorithm [2,22,23] of rule extraction from neural networks. This algorithm was improved in two ways. First, the extraction process is a lot simpler and quicker since we look only at a small part of the network. We do not need to extract all network knowledge, but just the new acquired knowledge. Second, we developed heuristic methods for network simplification (remove less significant units and links), used before extraction. The use of a simplified network helps us to reduce the complexity of the extraction procedure.

The Valid module [3] finds out the probably incorrect rules and examples. Thus, we will need to submit these inconsistencies to an expert analysis. This module is under development.

In summary, the INSS system presents some important advantages over its predecessor, KBANN. Our system improvements allow us to eliminates some drawbacks of KBANN nets:

- The INSS constructive neural architecture allows to work with incomplete symbolic rule sets and also with incorrect symbolic rule sets. Our system can easily add new rules (neurons) or even make broader changes in the existing ones. The KBANN networks, as they use static networks, restrict learning to less important changes to the rule set. If we need to 'learn' a new rule from examples in KBANN we should add manually specific units for this purpose.
- The KBANN network algorithm tries not to change unit meaning, and tries to keep the symbolic label significance associated to them. We can not be sure that, during the KBANN learning process, its units will not suffer a meaning shift. The Cascade-Correlation, used within INSS, keeps unchanged the initial acquired knowledge (compiled rules) by freezing the network connection weights, and does not have any problem of meaning shift.
- The learning algorithm used in INSS is faster than KBANN's Back-Propagation based algorithm. Besides, this algorithm allows an incremental network construction, by improving the connection weights as well as the network topology.
- Our rule extraction algorithm does not need to analyse all the ANN structure, but instead we just consider the new acquired network knowledge by analysing the new added units. This leads to an important reduction of the rule extraction process complexity.
- We are not restricted to using binary inputs (rules of order 0), nor obligated to pre-process continuous inputs in order to discretizate them. Our system allows symbolic rule compilation of proposition rules of order 0^+.

4. Validation of INSS : practical results

The possibility to use and to take into account several kinds of knowledge representation appears clearly in the description of the functioning of INSS. It’s the same for the possibility of global learning. INSS can not only adapt its various sets of knowledge to the variation of the data domain, but also to learn by modifications of its architecture.

It remains to show that the hybridisation increases the efficiency of the system, and that the knowledge extracted by INSS is a “good” knowledge. With this aim in view we have applied INSS on a relatively simple application, the *Monk's Problem* [21]. This problem is a set of tests developed for performance comparison of different learning algorithms. There are three Monk's problem data sets. Here we will discuss only the results we obtained within the first one, the Monk1 problem, although our tests cover all three problem data sets.

Table 1 - Monk1: Description of the symbolic rule set

<u>Input Features</u>:

HEAD_SHAPE = { ROUND, SQUARE, OCTAGON }
BODY_SHAPE = { ROUND, SQUARE, OCTAGON }
IS_SMILING = { YES, NO }
HOLDING = { SWORD, BALLOON, FLAG }
JACKET_COLOUR = { RED, YELLOW, GREEN, BLUE }
HAS_TIE = { YES, NO }

<u>Symbolic Rules</u>:

Monk1 <- HEAD_SHAPE = ROUND,
BODY_SHAPE = ROUND (I)
Monk1 <- HEAD_SHAPE = SQUARE,
BODY_SHAPE = SQUARE (II)
Monk1 <- HEAD_SHAPE = OCTAGON,
BODY_SHAPE = OCTAGON (III)
Monk1 <- JACKET_COLOUR = RED (IV)

The Monk1 problem data set is composed by one set of four symbolic rules (see table 1 for the complete domain theory), by one generalisation test set of 432 examples (covering all the input space), and by one learning set of 124 examples. The examples are exactly those available in the original data [25]. In our experiments we used portions of the rule set and the examples set in order to study the generalisation capacity of our system. Just the learning set and the rule set were partitioned, for its part, the generalisation test set was preserved unchanged in all experiments.

4.1. First experiment : validity of the extracted knowledge.

This experiment aims at verifying if the system is able to find again the complete rule set from a partial set of knowledge. This is accomplished by means of learning an example base built up with the complete set.

In a first test, we created a network by compiling 75 % of the rules (3 among the 4 available rules). Then we applied the rule extraction method. The extraction process has been applied only on two units, the output and one hidden unit, because one unit only has been added to the network during the learning period. We repeated such a test for all the configurations of the incomplete rule set : one rule eliminated among four available rules. In any case, the extraction method allowed to retrieve the rule removed from the initial set.

In a second test, we used another incomplete rule set constructed by suppressing 50% of the rules contained into the complete set. As in the first test, we refined this initial knowledge by using the original learning data set. The result we obtained is the same one: *we rediscovered all the rules eliminated from the original rule set.*

This set of experiments leads to two remarks :

- In any case, the retrieved rules were found by rule extraction from the ANN added units. That shows the process of modification of the network architecture is consistent.
- The fact we rediscover the eliminated rule means the removed rule was implicitly present in the examples learned by the neural network. The extracted knowledge is sound and not in contradiction to the example set.

4.2. Second experiment : efficiency of the hybridisation.

The results obtained (see table 2) show that INSS is able to treat this problem using all available learning examples, or using a combination of the theoretical knowledge (rules) and empirical knowledge (examples). We showed that we always obtain a superior generalisation rate when we use at the same time rules and examples. Lower generalisation rates are obtained when we used just one information source at the same time.

Table 2 - Monk1 problem:
Using rules and examples to improve generalisation

Portion of Rule Set	Portion of Examples Set	Generalis. using INSS	Generalis. Just rules	Generalis. ANN Just examples
-	100%	**100%**	-	100%
-	75%	**89.21%**	-	89.21%
-	50%	**70.92%**	-	70.92%
100%	-	**100%**	100%	-
75%	-	**83.33%**	83.33%	-
50%	-	**72.22%**	72.22%	-
75%	100%	**100%**	83.33	100%
50%	100%	**100%**	72.22	100%
75%	75%	**100%**	83.33	89.21
75%	50%	**100%**	83.33	70.92
50%	75%	**100%**	72.22	89.21
50%	50%	**89.86**	72.22	70.92

* Generalisation scores represents the average obtained from 5 different runs

+ Our system and the data used in these tests are available (send us e-mail)

5. Medical Diagnosis and other applications.

In order to study the behaviour of INSS on a real application, the system has been also tested on a medical diagnosis application : diagnosis of toxic coma. When a comatose patient is admitted in an emergency care unit, the clinician makes an early tentative diagnosis by collecting clinical and biological parameters. The diagnosis may be later confirmed or rejected by toxicological analysis. So, for the initial therapeutic action to be as adequate as possible, there is a need for an accurate prediction of the toxic cause, without waiting for the toxicological analysis. The use of an intelligent automated system to help in this diagnosis task seems to be very useful. Until now, there is no complete model for describing this knowledge by means of rules.

Our goal was to use INSS to aid to identify the causes of a psychotrope induced coma. We have available a case base of 505 pre-analysed examples of patients. Each example is described with 13 parameters or symptoms obtained directly when the patient is admitted, without waiting for the toxicological analysis. The diagnosis should aid to identify the presence or absence of each one of the 7 individual toxic causes (Alcohol, ADT, Benzodiazepines, Barbiturates, Carbamates, Morphine or Phenothiazines). A more detailed description of this problem can be found in the technical report of the Esprit MIX Project [1].

Table 3 reproduces the results we obtained comparing INSS to other machine learning systems applied to this medical diagnosis problem. All the systems were tested with exactly the same learning and testing data sets, and the results expressed in this table are the average of 10 different runs. The systems we compared with INSS are described in the MIX report□[1].

The scores showed in Table 3, related to the other methods (K-PPV, C4.5 and ProBis), were reproduced from the results obtained by other researchers. As we were constrained to use the same experiment protocol in order to be able to compare these different methods, we show here just a brief performance comparison. Although we published in [1] a more detailed list of the results obtained with INSS related to this problem.

Table 3 - Comparison of the generalisation test rate after learning

Class \ Method	K-PPV	C4.5	ProBIS	INSS
Alcohol -E	66.56%	65.40%	68.94%	**74.50%**
ADT - a	55.39%	55.26%	57.63%	**60.79%**
Barbituriques - B	65.65%	63.32%	64.60%	**82.45%**
Benzodiazepines-b	62.37%	64.34%	63.95%	**83.37%**
Carbamates - c	81.58%	**87.64%**	84.87%	87.28%
Morphine - m	97.23%	97.50%	**97.97%**	97.88%
Phenothiazines - p	66.45%	71.26%	68.95%	**75.36%**

As we can observe from Table 3, the INSS system shows a remarkable performance in this task compared with the other techniques. In some classes the percentage differences between INSS and the other methods are quite small, but always near to the best performance obtained. However we have to remark that in some classes we get relatively poor classification results for all methods. That's due to the intrinsic complexity of this problem and the strong overlapping of the different classes : this kind of complexity is a typical feature of the medical diagnosis.

We also tried to extract rules from the trained ANN. The extracted rules were presented to one expert of this domain, and he immediately recognised them as "valid rules". He also noted that a great part of the rules had captured important relations between certain input features and the presence of one specific toxic substance (e.g., intermediate pupil size, normal eye movement, low core temperature, and prolonged cardiological QT interval, are factors that indicate the possible presence of□ADT).

This research resulted in the development of an experimental user interface to give access to our system through the WWW (World-Wide-Web). This program allows the consultation of the INSS system for toxic coma diagnosis. The user can fill-in a form with the patient's clinical and biological parameters and get back the ANN answer indicating the possible toxic substances absorbed. Presently, the system answer is based upon an ANN trained with the 505 cases database. However, this small number of available cases has proved to be insufficient for a good diagnosis of all toxic substances.

We are currently using the INSS system in two other domains: autonomous robot control and models of human cognitive development (e.g., balance scale problem[18,19]). A description of our preliminary results obtained with these applications can be found in□[13].

6. Conclusion

The INSS system presented here offers many advantages compared to the KBANN system by which it was inspired. This system has a better performance and allows incremental acquisition/extraction of network knowledge. Furthermore, it is based upon an incremental learning method already used to model human cognitive development. This learning method allowed us to develop a system perfectly adapted to the concepts proposed in the framework of constructive machine learning systems. The system was tested on different applications (classification tasks, medical diagnosis, autonomous robot control) obtaining satisfactory results. Actually our main goals are to develop a deeper study of the real-world applications of INSS, as well as to study the aspects related to the constructive acquisition of knowledge.

Acknowledgements

Thanks to Scott Fahlman, Geoffrey Towell, Vincent Danel and Daniel Memmi for their contribution to our research and/or to the production of this article. The research reported here was supported in part by a grant from the CAPES (Brazilian government research agency).

References

[1] AMY, B. et al. (1997). MIX Medical Application (Final Report - Deliverable D16). *Technical Report, Project Esprit-BRA 'MIX' - European Community*. May 1997. Web: http://www-leibniz.imag.fr/RESEAUX/public.html . Ftp: ftp://ftp.imag.fr/pub/LEIBNIZ/RESEAUX-D-AUTOMATES/MIX-FMR.ps.gz

[2] ANDREWS, R.; DIEDERICH, J.; TICKLE, A.B. (1995). A Survey And Critique of Techniques For Extracting Rules From Trained ANN. *Technical Report - Neurocomputing Research Centre, Queensland University of Technology*. Brisbane, Australia. January 1995 (also published in: Knowledge-Based Systems 8[6], pp.378-389, 1995). Web: http://www.fit.qut.edu.au/NRC/ .Ftp: ftp://ftp.fit.qut.edu.au/pub/NRC/tr/ps/QUTNRC-95-01-02.ps.Z

[3] BLOND, P.-Y. (1996). Validation de Connaissances dans un Système Hybride Neuro-Symbolique à Apprentissage Incrémental. *Rapport du D.E.A. en Sciences Cognitives, LEIBNIZ - IMAG, Grenoble - France.*

[4] BOZ, Olcay (1995). Knowledge Integration and Rule Extraction in Neural Networks. *Technical Report, EECS, Lehigh University*. October 1995. Web: http://www.lehigh.edu/~ob00/integrated .html

[5] DECLOEDT, L., OSORIO, F., AMY, B. (1996). RULE_OUT Method: A New Approach for Knowledge Explicitation from Trained Artificial Neural Networks. *Proceedings of the AISB'96 - Workshop on Rule Extraction from Trained Neural Nets*, p.34-42. QUT - Andrews and Diederich Eds. Web: http://www-leibniz.imag.fr/RESEAUX/ public.html . Ftp: ftp://ftp.imag.fr/ pub/LEIBNIZ/ RESEAUX-D-AUTOMATES/osorio.aisb96.ps.gz

[6] DECLOEDT, Loïc (1995). Explicitation de Connaissances dans un Système Hybride d'Intelligence Artificielle. *Rapport du D.E.A. en Informatique, Laboratoire LIFIA - IMAG / INPG*, Grenoble - France, Juin 1995. Web: http://www-leibniz.imag.fr/RESEAUX/public.html . Ftp: ftp://ftp.imag.fr/ pub/LEIBNIZ/RESEAUX-D-AUTOMATES/DEA/decloedt.dea.ps.gz

[7] FAHLMAN, S. E., LEBIERE, C. (1990). The Cascade-Correlation Learning Architecture. *Carnegie Mellon Un., Technical Report - CMU-CS-90-100*. Web: http://www.cs.cmu.edu/Reports/ index.html . Ftp: ftp://archive.cis.ohio-state.edu/ pub/neuroprose/fahlman.cascor-tr.ps.Z

[8] GIACOMETTI, A. (1992). Modèles hybrides de l'expertise. *Thèse de Doctorat, LIFIA - IMAG, Grenoble - France*. Web: http://www-leibniz.imag.fr/ RESEAUX/public.html . Ftp: ftp://ftp.imag.fr/pub/LEIBNIZ/RESEAUX-D-AUTOMATES/giacometti.these.ps.tar.gz

[9] GIARRATANO, Joseph C. (1993). CLIPS User's Guide - Version 6.0. *Lyndon B. Johnson Space Center, Software Technology Branch, NASA* - U.S.A. Web: http://www.jsc.nasa.gov/~clips/ CLIPS.html or http://home.haley.com/ clips.html Ftp: ftp://ftp.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/expert/ systems/clips/0.html

[10] HILARIO, M. (1996). An Overview of Strategies for Neurosymbolic Integration. *In: Connectionist-Symbolic Integration: From Unified to Hybrid Approaches*. Ron Sun (Ed.) - Chapter 2. Kluwer Academic Publishers. Ftp: ftp://cui.unige.ch/AI/

[11] ORSIER, B. (1995). Etude et application de systèmes hybrides neuro-symboliques. *Thèse de Doctorat, UJF - LIFIA, Grenoble - France*. Web: http://www-leibniz.imag.fr/RESEAUX/ public.html . Ftp: ftp://ftp.imag.fr/ pub/LEIBNIZ/ RESEAUX-D-AUTOMATES/orsier.these.ps.gz

[12] OSORIO, F. S. & AMY, B. (1995). INSS: A Hybrid Symboli-Connectionist System that Learns from Rules and Examples (text in portuguese), *Panel'95 - XXI Latin-American Conference on Computer Science,* Canela, Brazil, August 1995. Web: http://www-leibniz.imag.fr/RESEAUX/ osorio/papers/diret.html

[13] OSORIO, F. S. (1998). INSS : Un Système Hybride Neuro-Symbolique pour l'Apprentissage Automatique Constructif. *Thèse de Doctorat, INPG - Laboratoire LEIBNIZ - IMAG, Grenoble - France*. Web: http://www-leibniz.imag.fr/ RESEAUX/public.html . Ftp: ftp://ftp.imag.fr/ pub/LEIBNIZ/RESEAUX-D-AUTOMATES/ osorio.these.ps.gz

[14] REYES, Gerardo (1997). Etude des Connaissances dans les Réseaux de Neurones Artificiels : Représentation et Explicitation de Règles d'Haut Niveau. *Rapport du D.E.A. en Sciences Cognitives, LEIBNIZ - IMAG, Grenoble - France*.

[15] REYES, G.; OSORIO, F.; AMY, B. (1997). Neural Learning of "High Level Rules": The Balance Scale Problem. *HELNET'97 International Workshop on Neural Networks*. Montreux, Swiss, October 1997. Web: http://www-leibniz.imag.fr/RESEAUX/public.html . Ftp: ftp://ftp.imag.fr/pub/LEIBNIZ/ RESEAUX-D-AUTOMATES/osorio.helnet97.ps.gz

[16] RUMELHART, D., HINTON, G., WILLIANS, R. (1986). Learning Internal Representations by Error Propagation. *Parallel Distributed Processing - Explorations in the Microstructure of Cognition, V1. Cambridge: MIT Press.*

[17] SCHIFFMANN, W.; JOOST, M. & WERNER, R. (1993). Comparison of Optimized Backpropagation Algorithms. *Proceedings of the European Symposium on Artificial Neural Networks, ESANN'93*, Brussels, p.97-104, Web: http://www.uni-koblenz.de/~schiff/ publications.html

[18] SHULTZ, T. R., SCHMIDT, W. C. (1991). A Cascade-Correlation Model of Balance Scale Phenomena. *Proceedings of the Thirteenth Annual Conf. of the Cognitive Science Society, Hillsdale, NJ: Erlbaum,* p.635-640. Web: http://www.psych.mcgill.ca/labs/lnsc/html/Lab-Home.html (or Pub-cog-dev.html). Ftp: ftp://ego.psych.mcgill.ca/pub/shultz/balcog.ps.gz

[19] SHULTZ, T. R., MARESCHAL, D., SCHMIDT, W. (1994). Modeling Cognitive Development on Balance Scale Phenomena. *Machine Learning, Kluwer Publishers*, No.16, p.57-86. Web: http://www.psych.mcgill.ca/ labs/lnsc/html/Lab-Home.html (or Pub-cog-dev.html). Ftp: ftp://ego.psych. mcgill.ca/pub/shultz/balml.ps.gz

[20] SUN, Ron & ALEXANDRE, Frederic (1997). Connectionist-Symbolic Integration: From Unified to Hybrid Approaches. *Lawrence Erlbaum Associates.*

[21] THRUN, S. B. et al. (1991). The Monk's Problem - A Performance Comparison of Different Learning Algorithm. *Carnegie Mellon University, Technical Report CMU-CS-91-197*. Web: http://www.cs.cmu.edu/~thrun/ . Ftp: ftp://archive.cis.ohio-state.edu/pub/neuroprose/

[22] TOWELL, G. (1991). Symbolic Knowledge and Neural Networks: Insertion, Refinement and Extraction. *Ph.D. Thesis, University of Wisconsin-Madison - Computer. Science Dept.* Web: http://www.cs.wisc.edu/~shavlik/uwml.html Ftp: ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/ (towell.thesis.*.ps)

[23] TOWELL, G. & SHAVLIK, J. (1993). Extracting Refined Rules From Knowledge-Based Neural Nets. *Machine Learning, Kluwer Academic Publishers*, Boston, pp.71-101, 13. Web: http://www.cs.wisc.edu/~shavlik/ uwml.html . Ftp: ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/ towell.mlj93.ps

[24] TOWELL, G. & SHAVLIK, J. (1994). Knowledge-Based Neural Nets. *Artificial Intelligence*, pp.119-165, 70. Web: http://www.cs.wisc.edu/~shavlik/ uwml.html . Ftp: ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/ towell.aij94.ps

[25] UCI-ML Repository. (1997). *UCI - University of California, Irvine Machine Learning Databases and Domain Theories*. Web: http://www.ics.uci.edu/~mlearn/MLRepository.html . Ftp: ftp://ftp.ics.uci.edu/pub/machine-learning-databases/README