Proceedings of the 6th WSEAS Int. Conf. on NEURAL NETWORKS, Lisbon, Portugal, June 16-18, 2005 (pp39-44)

HLS : HYBRID LEARNING SYSTEM

Jihane BOULAHIA-SMIRANI(1) and Laurent BOUGRAIN(2)
(1) ENSI/LIA. Artificial Intelligence Group Tunis-Tunisie
(2) INRIA/LORIA. Villiers-lès-Nancy, France

ABSTRACT

In this paper, we are interested in a hybrid neuro-symbolic system. We present the HLS (Hybrid Learning System), a new hybrid approach combining a connexionist module, a symbolic module, a rule extraction module and a rule insertion module. It presents an important improvement in comparison with just a connectionist system. HLS provides a new approach applicable to machine learning with high-performance tools, even in presence of incomplete data. The proposed architecture gives a good performance and allows acquisition/extraction of network knowledge.

Keywords:
Hybrid neuro-symbolic system, backpropagation algorithm, Artificial Neural Networks, rule insertion and extraction.

1. INTRODUCTION

The interest in hybrid systems is probably as old as the interest in models of intelligence [1]. Hybrid systems, by definition, is an unusual and imprecise term [2]. It has been used to refer to many areas of computational and engineering systems that mix different techniques, aspects and models [3]. The concept of hybrid systems is very broad. This group of applications includes any method, which merge two different approaches for the solution of a problem. It's often suggested that traditionally serial symbol processing systems of artificial intelligence (AI) and inherently massively parallel artificial neural networks (ANN's) offer two radically, perhaps even irreconcilably different paradigms for modeling minds and brains, both artificial as well as natural. We can note that the two paradigms have forces and weaknesses, which are often complementary. Rather than taking an entirely new-route, it is a question of re-using the assets, from where needed to integrate the two paradigms, thus the birth of the hybrid systems. The basic assumption in a neuro-symbolic hybrid system is that the symbolic system models and connectionist ones are not sufficient by them-selves, but their combination can allow the execution of all types of cognitive operations. It's shown by Church, Kleene, McCulloch, Post, Turning, and others, both AI and ANN represent particular realizations of a universal (Turning- equivalent) model of computation [4].

Most AI systems have been traditionally programmed in languages that were influenced by Von Neumann's design of serial stored program computer. ANN systems on the other hand, have been inspired by models of biological neural networks. The main argument, which is the most used one, to justify the study and the application of hybrid neuro-symbolic systems is the complementarity of symbolic AI (Artificial Intelligence) methods and sub-symbolic connectionist methods .

Such a justification is a very general one. We would like to have a more precise justification about the real contribution of the hybrid approach. What exactly provides the combination of neural networks and Knowledge-based systems? Researches claim that hybrid systems take advantage of their respective component strengths.

In this paper, we are interested in the neuro-symbolic hybrid system [4]. Our contribution is system named HLS : Hybrid Learning System combining a connexionist module with a symbolic module. Artificial neural networks, in particular the multilayer perceptron have proved to be useful for classifying speech isolated words [6,7].

2. HLS : HYBRID LEARNING SYSTEM

The schematic diagram of the overall architecture is proposed for speech recognition [8]. It consists of four major components : a connectionist module, a symbolic module , a rule extraction module, and a rule insertion module (Fig 1). The HLS system provides facilities to transfer rules from a symbolic module to a connectionist one, and examples from the connectionist module to the symbolic one [9,10].

First, we start by initializing the neural network. This initialization is made by the insertion rule module using the databases rules. This approach solves two big problems related to artificial neural networks: on one hand this simplifies the choice of the number of units, on the other hand we obtain a good assignment of initial values to the connection weights. The result is an initialized three layered neural network (architecture, bias and connections weights).

Second the connexionist module is activate to train the neural network. This training is based on a set of examples. After learning the extraction rule module is activate to extract rules from the neural network. The result of this module is a set of rules relating inputs to outputs. Rules extracted are transmitted to the symbolic module for validation.

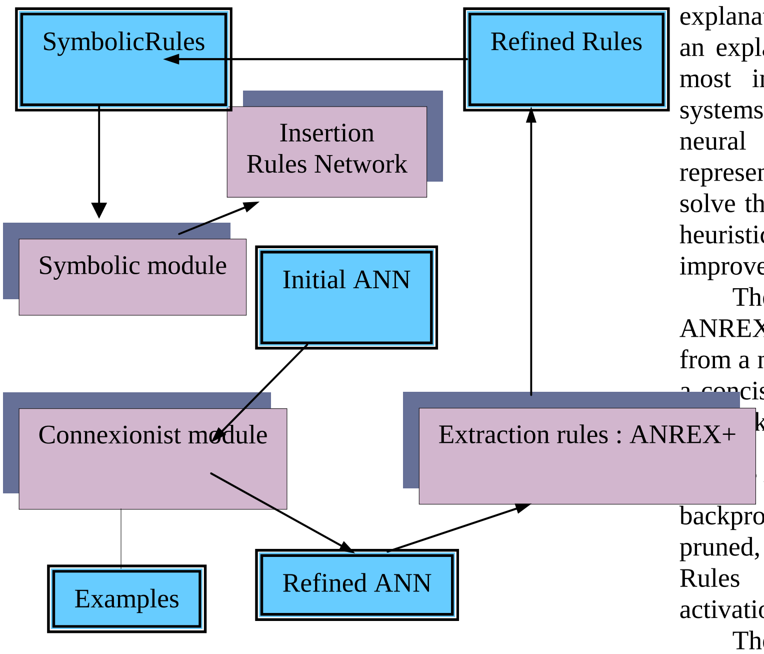


Figure1. HLS system

2.1. Connectionist module

The neural network contained in this module is a multi-layer network. The first question that emerges at the time of the conception of such a network is to know the number of the necessary hidden layers. Theoretically, the perceptron (two-layer network) can classify the space of data only linearly. The exactness of this classification can reach 100% if data is linearly separable. However, a multi-layer network can achieve a non-linear classification. In addition, it was shown [4] that any function, which is approximated by a multi-layer perceptron, can also be approximated by a multi-layer perceptron with only one hidden layer, if its architecture includes sufficient number of hidden units. perceptron. It is know that such a three-layer perceptron can realize arbitrary binary mappings [11]. For the experiments presented in this article, we used a “ feedforward” neural network that is trained using the backpropagation algorithm. Hidden units have a sigmoid activation.

2.2. Rule extraction module

It's becoming increasingly known that, without some forms of explanation capability, the full potential of trained artificial neural networks may not be realized [12]. The merits of including rule extraction techniques, as an adjunct to conventional artificial neural network techniques[13], include the provision of a user explanation capability. In fact, experience has shown that an explanation capability is considered to be one of the most important functions provided by symbolic AI systems. In contrast to symbolic AI systems, artificial neural networks have no explicit knowledge representation [14]. A new module has been defined to solve this problem, this module is named ANREX +: A heuristic Algorithm for Neural Rule extraction+ is an improvement of ANREX.

The algorithm heuristic for neural Rule Extraction ANREX [15] is going to permit a simple rule extraction from a neural network. The result provided by ANREX is a concise and precise description of the internal working [illegible]k. [illegible] standard three layer feedforward network is the [illegible] the algorithm. Weight decay is implemented while backpropagation is carried out. After the network is pruned, its hidden units activation values are discretized. Rules are extracted by examining the descretized activation values of the hidden units.

The basic structure of the neural network in this work is a standard three-layer feedforward network, which consists of an input layer, I, a hidden layer, H, and an output layer, O.

2.3.1. Clustering Algorithm

Rules are not readily extractable because the activation values of the hidden unit are continous. The discretisation of theses values paves the way for rule extraction. Many clustering algorithms can be used for this purpose. The following algorithm discretises the activation values of a hidden unit.

BEGIN

For each hidden unit

1. Let $\varepsilon = 0.2$
2. Start with activation value a_0 generated by the first example in the training set.

3. Cluster the activation values $| a_1 - a_0 | < 0.2$
4. Represent this cluster's activation value by the average of the activation values in the cluster
5. Select next a_0, repeat 3 and 4 for clustering until all activation values are clustered.

END

When the clustering is done, the network's accuracy is checked. A sufficiently small ε guarantees that the network with discretized activation accuracy does not drop and there are still many discrete values. The clustering can be performed again with a larger ε to minimize the number of clusters. Otherwise ε should be reduced to a smaller value.

2.3.1. The ANREX Algorithm

Let f be the sigmoid transfer function applied to the weighted sum of connections, val the activation value function, R1 the first set of rules, R2 the second set of rules and R the final set of rules.

```
BEGIN

/* Rule Extraction between the input and hidden units */

For every hidden unit u  do
   For every input unit  i (where connection u-i exist) do
      If  f(i) =  val(u)   then

        ith input participate in the rule R1
        Add i  with value to R1
      End if

    End For
End For

/* Rule Extraction between the hidden and output units */

For every hidden unit u do
   For  every output j (where connection u-j exist) do
      If  f(u) =  j   then
         jth output participate to the rule R2
         Add  j with value to R2
      End if
   End For
End For

    Merge ( R1,R2) → R

END
```

The algorithm for neural Rule Extraction ANREX [15] is going to permit a simple rule extraction from a neural network. The result provided by ANREX is a concise and precise description of the internal working network. ANREX has been applied to a multi-layer using the back-propagation algorithm. First ANREX has been applied to character recognition, second we applied it to a speech recognition task. In this second application, results show that the performance of ANREX decreases. The number of extracted rules become very large and we have observed the existence of many weights with minimal values so we have decide to study methods of pruning in the neural networks. This extension of ANREX is named ANREX+.

2.3.2. ANREX$^+$ algorithm

ANREX$^+$ is an improvement of ANREX. Before applied ANREX, we pruned the network. Indeed, if only the most significant variables are involved, resulting rules can be simplified, and comprehensibility increased. Methods for measuring unit significance have been developed. The aim is then to remove from the network the least significant units.

The assumption that the influence of a link is proportional to its weight can be wrong. If very little weights usually have little influence, it has been observed, after learning, links with big weights that were not significant for global network response. Som another method, inspired by Optimal Brain Damage Algorithm [?], has been developed to measure influence of each weight on global error; it is based on calculus of global error second derivative. This measure, called saliency, is defined, for a weight w_k as :

$$S(w_k) = (1/2)\ (\sigma^2 ASE\ ) / \sigma w_k \sigma w_k\ (w_k\ )^2$$

where ASE represents the training Average Squared Error. Then weights with low saliency can be pruned, and explanation can be greatly improved by this way.

Symbolic module

The symbolic module finds out the probably incorrect rules and examples.

2.3. Insertion rules module

There are two main benefits from introducing rules in the network : improving generalization to new instances and simplifying learning. Towell and shavlik [16] have proposed methods for mapping a set of prepositional rules into a neural network, refered to as Knowledge Based Artificial Neural Networks. (KBANN).

This module is inspired of this method. The result of the translation is a network composed of a set of units linked by weighted connections.

Conjunctive rules are translated into a neural network by setting weights on all links corresponding to positive (i.e. unnegated) antecedents to the weight value w, by setting weights on all links corresponding to negated antecedents to –w, and the bias on the unit corresponding to the rule's consequent to (-P+0.5)* w; where P is the number of positive antecedents to the rule. Experiments shown that the best value of w is w=4. Figure 2 shows q network which encodes 3B and C and D and not E → A. Intuitively when the rule is activated we have A= f (3w-2.5w) (where f is the sigmoid function). Thus the activation of A is great than 0.5. In all other cases the activation of A is less than 0.5.

To translate a set of rules encoding a disjunction, the insertion module sets the weight of each link corresponding to a disjunctive antecedent to w and the bias on the unit encoding the consequent to –w/2. For example figure 3 shows the network that results from translation of four disjunctive rules.

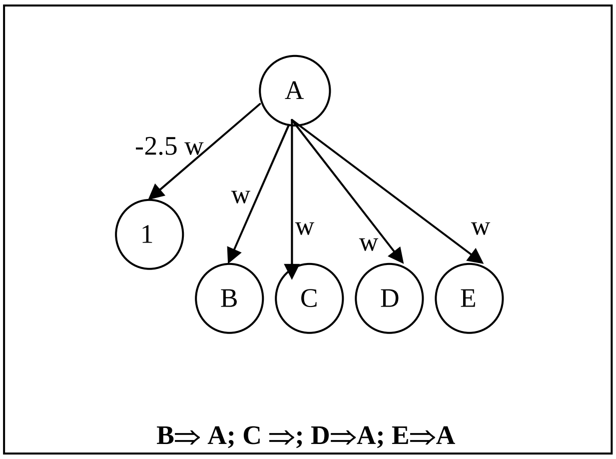


Figure3. Translation of disjunctive rules into neural network

3. EXPERIMENTAL RESULTS

In this paper, we detailed results we obtained in speech recognition and Iris database.

3.1. Speech recognition

The speech signal is digitizes from a head - mounted microphone. A custom interface decimates the 48 kHz digitized speech signal to 16 kHz [15]. The speech is segmented into hamming - window and processed into feature vectors [16]. Each feature vector consists of 12 cepstral coefficients [17].

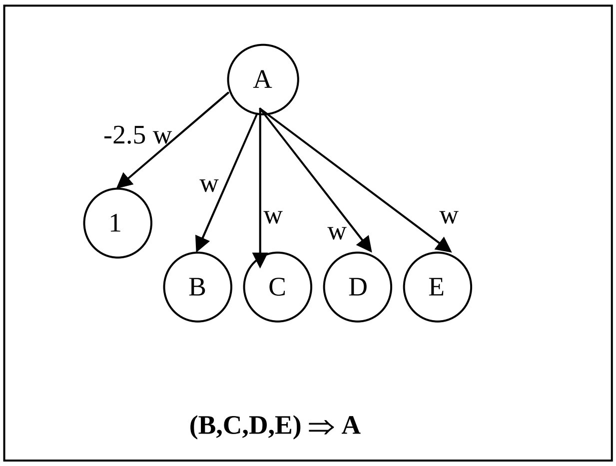


Figure2. Translation of disjunctive rules into neural network

The recognition vocabulary consists of 20 Arabic word. Speech was collected from 50 speakers (28 females and 22 males). The speakers were arbitrary separated into a set of 30 speakers (15 females and 15 males) for training and a set of 20 different speakers (13 females and 7 males) for testing [18].

For speech recognition ,there are two types of experiments : first we begun with testing the time convergence of the system, second the hybrid system recognition with incomplete data. The convergence time of the system is compared with a MLP (Multilayer Perceptron) convergence. The average of the epochs number necessary for the convergence shows clearly that with the HLS system the convergence is more rapid (60 epochs) than the MLP architecture. (980 epochs).

The second experiment has to do with the recognition rate [19] of the system with incomplete date as shown in Table2. This experiment aims at verifying if the system is able to recognize data with incomplete set of examples. In the first test, we created a network with 85% of the examples. Then, we applied the rule extraction module and the insertion module. In a second step, we used another incomplete example set constructed by removing 25% of the examples contained into the complete set. Third a set composed of 65% of the examples, finally only 50% of examples. The results obtained show that HLS is able to deal with this problem using all available learning rules or using a combination of the empirical knowledge (examples) and theoretical knowledge (rules). We showed that we always obtain a best generalization rate when we use at the same time rules and examples. Lower generalization rates are

Proceedings of the 6th WSEAS Int. Conf. on NEURAL NETWORKS, Lisbon, Portugal, June 16-18, 2005 (pp39-44)

obtained just when we use one information source by its own.

Portion of examples	Recognition with a MLP	Recognition with HLS
100%	100%	**100%**
85%	82,1%	**89,4%**
75%	65%	**81%**
65%	61%	**74%**
50%	59%	**62%**

Table 2. Speech recognition rate

3.2. IRIS DATABASE

The iris database is obtainable from the university of California (via anonymous ftp from ics.uci.edu).The summary of these database and results are given below.

The dataset Iris contains 50 examples each of the classes Iris versicolor, Iris setosa and Iris virginica (species of iris). Each example is decibed using four numeric attributes (X_1 , X_2 ,X_3 and X_4 : sepal length, sepal width, petal length and petal width. In the experiment , we start by two type of data, first the database of examples and second an initial set of rules that we have write for the initialization of the neural network as we describe in section 2.3.

Initial rules for inializing neural network :

Rule 1 : If Petal_length < 2.5 then iris-setosa.

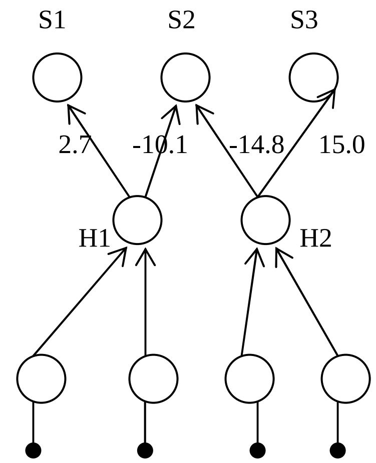
Rule 2 : If 2.5 < Petal_length and Petal_length <5 and 0.75 < Petal_width and Petal_width <1.6 then Iris_versicolor.

Rule3 : If 5 < Petal_length and 1.6 < Petal_width then Iris verginica.

50 fully connected neural networks were used as the starting networks. Each of the trained networks was pruned until its accuracy on the training data dropped below 95%. The weights and topology of networks with the smallest number of connections and an accuracy rate of more than 96% were saved for rule extraction.

One of the pruned networks is describe in Figure 4. It has only 2 hidden units and a total of 8 connections with 98.1% accuracy in the training set and 97.2 % on the testing set. We applied the clustering algorithm described in 2.3.1, we found only 2 discrete values at each of the two hidden units. Qt hidden unit 1, 37 of 50 training examples have activation values equal to 0 and 13 have activation values equal to 1. At hidden2m the activation value of 15 examples is 1 and the activation value of the second hidden unit is equal to –0.5. Since we have two activation values at each of the two hidden units, four different outcomes qt the output unit are possible (table1). From this tqble, it is clear that an example will be classified as Iris setosa as long as its activation value at the second hideden unit is equal to 1. Otherwise, the example is classified as Iris versicolor provided that its first hidden unit activation value H-1=0. The default class will then be Iris virginica.

As seen in figure1 , only two inputs, I1 and I2, determine the activation values of the second hidden unit, H2. Butm since I39 is 1 for all the training data, H2 is effectively determined by I31. Since the weights of the arcs connecting input units 31 and 39 to the second hidden unit are –5.4 and 4.8 respectively, it is easy to conclude thqt if I31=0 then H2 is 1m otherwise , H2 is –0.52. This I;plies that an example will be classified as Iris setosa only if I31 is 0 5hence H2 is 1).



The activation value of the first hidden unitm H1m depends onlyon I26 and I34. The zeignts of the qrcs connecting input units 26 and 34 to the first hidden unit are 5.1 and 8.1 , respectively hence H1 is 0 if and only if I26=I34=0. Other input combinations will yield value 1 for H1. Hence an example with I3=1 I26=I34=0 will be classified as Iris versicolor.

Rules obtained are.

.

Rule 1 : If Petal_length < =1.9 then iris-setosa.

Rule 2 : If Petal_length <=4.9 and Petal_width <=1.6 then Iris_versicolor.

Default Rule : Iris virginica.

1

4. CONCLUSION

We have proposed and implemented a neuro-symbolic model. The proposed architecture named HLS (Hybrid Learning System) provides a good performance and allows acquisition/extraction of network knowledge. The results are very interesting when comparing them to those of a neural network.

REFERENCES

[1] ANDREWS, R.; DIEDERICH, J; TICKLE, A.B. A Survey And Critique of Techniques For Extracting Rules From Trained ANN. *Technical Report – Neurocomputing Reasearch Centre, Queensland University of Technology.* Brisbane, Australia. January 1995.

[2] HILARIO, M. An overview of Strategies for Neurosymbolic Integration.In . Connectionist Symbolic Integration . From Unified to Hybrid Approaches. Ron Sun (ED.)- Chapter 2. Kluwr Academic Publishers.1996.

[3] J.Sima. "Neural expert systems". Neural networks, Vol 8 N:2, pp 261.1194.

[4] ORSIER, B. Etude et application de systèmes hybrides neuro-symboliques. Thèse de doctorat , UJF-LIFIA, Grenoble- France.

[5] SUN, Ron & ALEXANDRE, Frederic . Connectionist-Symbolic Integartion : From Unified to Hybrid Approaches.. *Lawrence Erlbaum Associates.*

[4] Y. Lallement. "Intégration Neuro-Symbolique et Intelligence Artificielle: Applications et Implantation Parallèle". Thèse de Doctorat en Informatique, CRIN-INRIA Lorraine, Université Henri Poincaré – Nancy I, France, Juin 1996.

[8] F. Mouria Beji and J. Boulahia Smirani. "Extraction and Insertion rules during the training process of a neural network". ACIDCA'2000. 22-24 March 2000, Monastir-Tunisia.

[10] C. L. Giles and C. W. Omlin. "Extraction, insertion, and refinement of symbolic rules in dynamically driven recurrent neural networks", Connection Sci., vol. 5, n°3, pp. 307-337,1993.

[11] D.E. Rumelhart, G. E. Hinton and R. J. Williams. "Learning internal representations by error propagation". In parallel distributed Processing : Explorations in the Microstructures of Cognition, volume 1. Cambridge, MA : MIT Press, 1986.

[12] D. Angluin and C. H. Smith. " A survey of inductive inference : Theory and methods". ACM Comput. Survey, vol, 15, pp. 237-269, Sept. 1983.

[15] F. Mouria-Béji, J. Boulahia and M.Marrakchi. *ANREX : an algorithm for neural network rule extraction*. In IEEE International Multiconference on Computational Engineering in Systems Applications IEEE-SMC. April, 1998.

[16] Z. Wu and M. L .*Introduction to Experimental Speech.* The high Education Press, 1987. Beijing.

[17]G. Towell and J. W. Shavlik, *"Extracting refined rules from knowledge-based neural networks"*, Machine learning, vol. 13, pp. 71-101, 1993.

[18] J. W. Shavlik, *"Combining symbolic and neural learning".* Machine Learning, vol. 14, n°3, pp. 321-331, 1994.

[19] Boulahia-Smirani J. *A new universal algorithm for neural rule extraction.* WSEAS Transactions on systems . Issue 2, Volume 3, April 2004.

[20] Boulahia-Smirani J et Mouria-Beji F. *A hybrid learning algorithm.* Brain Inspired Cognitive Systems. University of Stirling, Scotland. United Kingdom. Aout 2004.