Neuro-Fuzzy Systems: A Survey

JOSÉ VIEIRA [*], FERNANDO MORGADO DIAS [**],ALEXANDRE MOTA [***]

* Departamento de Eng.ª Electrotécnica, Escola Superior de Tecnologia de Castelo Branco,
Avenida do Empresario 6000-767 Castelo Branco, PORTUGAL
zevieira@est.ipcb.pt

** Escola Superior de Tecnologia de Setúbal do Instituto Politécnico de Setúbal,
Departamento de Eng.ª Electrotécnica, Campus do IPS, Estefanilha, 2914-508 Setúbal, PORTUGAL
fmdias@est.ips.pt

*** Departmento de Eng.ª Electronica e de Telecomunicações, Universidade de Aveiro,
Campus Universitário de Santiago, 3810 Aveiro, PORTUGAL
alex@det.ua.pt

Abstract: – The techniques of artificial intelligence based in fuzzy logic and neural networks are frequently applied together. The reasons to combine these two paradigms come out of the difficulties and inherent limitations of each isolated paradigm. Generically, when they are used in a combined way, they are called Neuro-Fuzzy Systems. This term, however, is often used to assign a specific type of system that integrates both techniques. This type of system is characterised by a fuzzy system where fuzzy sets and fuzzy rules are adjusted using input output patterns. There are several different implementations of neuro-fuzzy systems, where each author defined its own model. This article summarizes a general vision of the area describing the most known hybrid neuro-fuzzy techniques, its advantages and disadvantages.

Key Words – Hybrid Systems, Cooperative Systems, Concurrent Systems, Neuro-Fuzzy Architectures, Non-Linear Modelling.

1 Introduction

The modern techniques of artificial intelligence have found application in almost all the fields of the human knowledge. However, a great emphasis is given to the accurate sciences areas, perhaps the biggest expression of the success of these techniques is in engineering field. These two techniques neural networks and fuzzy logic are many times applied together for solving engineering problems where the classic techniques do not supply an easy and accurate solution. The neuro-fuzzy term was born by the fusing of these two techniques. As each researcher combines these two tools in different way, then, some confusion was created on the exact meaning of this term. Still there is no absolute consensus but in general, the neuro-fuzzy term means a type of system characterized for a similar structure of a fuzzy controller where the fuzzy sets and rules are adjusted using neural networks tuning techniques in an iterative way with data vectors (input and output system data).

Such systems show two distinct ways of behaviour. In a first phase, called learning phase, it behaves like neural networks that learns its internal parameters off-line. Later, in the execution phase, it behaves like a fuzzy logic system.

Separately, each one of these techniques possess advantages and disadvantages that, when mixed together, theirs cooperage provides better results than the ones achieved with the use of each isolated technique.

1.1 Fuzzy Systems

Fuzzy systems propose a mathematic calculus to translate the subjective human knowledge of the real processes. This is a way to manipulate practical knowledge with some level of uncertainty. The fuzzy sets theory was initiated by Lofti Zadeh [16], in 1965. The behaviour of such systems is described through a set of fuzzy rules, like:

IF <premise> THEN <consequent>

that uses linguistics variables with symbolic terms. Each term represents a fuzzy set. The terms of the input space (typically 5-7 for each linguistic variable) compose the fuzzy partition.

The fuzzy inference mechanism consists of three stages: in the first stage, the values of the numerical inputs are mapped by a function according to a degree of compatibility of the respective fuzzy sets, this operation can be called fuzzyfication. In the second stage, the fuzzy system processes the rules in accordance with the firing strengths of the inputs. In the third stage, the resultant fuzzy values are transformed again into numerical values, this operation can be called defuzzyfication. Essentially, this procedure makes possible the use fuzzy categories in representation of words and abstracts ideas of the human beings in the description of the decision taking procedure.

The advantages of the fuzzy systems are:

- capacity to represent inherent uncertainties of the human knowledge with linguistic variables;
- simple interaction of the expert of the domain with the engineer designer of the system;
- easy interpretation of the results, because of the natural rules representation;
- easy extension of the base of knowledge through the addition of new rules;
- robustness in relation of the possible disturbances in the system.

And its disadvantages are:

- incapable to generalize, or either, it only answers to what is written in its rule base;
- not robust in relation the topological changes of the system, such changes would demand alterations in the rule base;
- depends on the existence of a expert to determine the inference logical rules;

1.2 Neural Networks

The neural networks try to shape the biological functions of the human brain. This leads to the idealisation of the neurons as discrete units of distributed processing. Its local or global connections inside of a net also are idealized, thus leading to the capacity of the nervous system in assimilating, learning or to foresee reactions or decisions to be taken. W. S. McCulloch, W. Pits, described the first Neural Network model and F. Rosenblatt (Perceptron) and B. Widrow (Adaline) develop the first training algorithm. The main characteristic of the neural networks is the fact that these structures can learn with examples (training vectors, input and output samples of the system). The neural networks modifies its internal structure and the weights of the connections between its artificial neurons to make the mapping, with a level of acceptable error for the application, of the relation input/output that represent the behaviour of the modelled system.

The advantages of the neural networks are:

- learning capacity;
- generalization capacity;
- robustness in relation to disturbances.

And its disadvantages are:

- impossible interpretation of the functionality;
- difficulty in determining the number of layers and number of neurons.

2 Neuro Fuzzy Systems

Since the moment that fuzzy systems become popular in industrial application, the community perceived that the development of a fuzzy system with good performance is not an easy task. The problem of finding membership functions and appropriate rules is frequently a tiring process of attempt and error. This lead to the idea of applying learning algorithms to the fuzzy systems. The neural networks, that have efficient learning algorithms, had been presented as an alternative to automate or to support the development of tuning fuzzy systems.
The first studies of the neuro-fuzzy systems date of the beginning of the 90's decade, with Jang, Lin and Lee in 1991, Berenji in 1992 and Nauck from 1993, etc. The majority of the first applications were in process control. Gradually, its application spread for all the areas of the knowledge like, data analysis, data classification, imperfections detection and support to decision-making, etc.
Neural networks and fuzzy systems can be combined to join its advantages and to cure its individual illness. Neural networks introduce its computational characteristics of learning in the fuzzy systems and receive from them the interpretation and clarity of systems representation. Thus, the disadvantages of the fuzzy systems are compensated by the capacities of the neural networks. These techniques are complementary, which justifies its use together.

3 Types of Neuro-Fuzzy Systems

In general, all the combinations of techniques based on neural networks and fuzzy logic can be called neuro-fuzzy systems. The different combinations of these techniques can be divided, in accordance with [10], in the following classes:

Cooperative Neuro-Fuzzy System: In the cooperative systems there is a pre-processing phase where the neural networks mechanisms of learning determine some sub-blocks of the fuzzy system. For instance, the fuzzy sets and/or fuzzy rules (fuzzy associative memories [8] or the use of clustering algorithms to determine the rules and fuzzy sets position [3]). After the fuzzy sub-blocks are calculated the neural network learning methods are taken away, executing only the fuzzy system.

Concurrent Neuro-Fuzzy System: In the concurrent systems the neural network and the fuzzy system work continuously together. In general, the neural networks pre-processes the inputs (or pos-processes the outputs) of the fuzzy system.

Hybrid Neuro-Fuzzy System: In this category, a neural network is used to learn some parameters of the fuzzy system (parameters of the fuzzy sets, fuzzy rules and weights of the rules) of a fuzzy system in an iterative way. The majority of the researchers uses the neuro-fuzzy term to refer only hybrid neuro-fuzzy system.

4 Cooperative Neuro-Fuzzy Systems

In a cooperative system the neural networks are only used in an initial phase. In this case, the neural networks determines sub-blocks of the fuzzy system using training data, after this, the neural networks are removed and only the fuzzy system is executed. In the cooperative neuro-fuzzy systems, the structure is not total interpretable what can be considered a disadvantage.

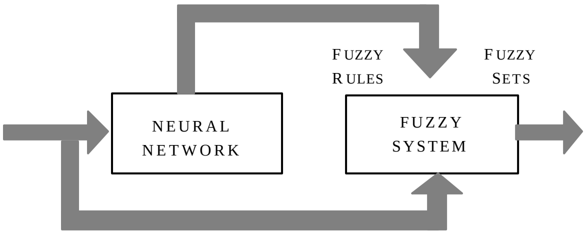

Figure 1. Cooperative Systems

5 Concurrent Neuro-Fuzzy Systems

A concurrent system is not a neuro-fuzzy system in the strict sense, because the neural network works together with the fuzzy system. This means that the inputs enters in the fuzzy system, are pre-processed and then the neural network processes the outputs of the concurrent system or in the reverse way. In the concurrent neuro-fuzzy systems, the results are not completely interpretable, what can be considered a disadvantage.

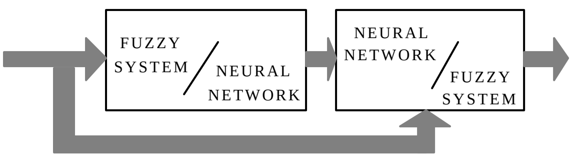

Figure 2. Concurrent Systems

6 Hybrid Neuro-Fuzzy Systems

In Nauck [10] definition: "A hybrid neuro-fuzzy system is a fuzzy system that uses a learning algorithm based on gradients or inspired by the neural networks theory (heuristical learning strategies) to determine its parameters (fuzzy sets and fuzzy rules) through the patterns processing (input and output)".

A neuro-fuzzy system can be interpreted as a set of fuzzy rules. This system can be total created from input output data or initialised with the *à priori* knowledge in the same way of fuzzy rules. The resultant system by fusing fuzzy systems and neural networks has as advantages of learning through patterns and the easy interpretation of its functionality.

There are several different ways to develop hybrid neuro-fuzzy systems, therefore, being a recent research subject, each researcher has defined its own particular models. These models are similar in its essence, but they present basic differences.

Many types of neuro-fuzzy systems are represented by neural networks that implement logical functions. This is not necessary for the application of an learning algorithm in to a fuzzy system, however, the representation trough a neural networks is more convenient because it allows to visualise the flow of data through the system and the error signals that are used to update its parameters. The aditional benefit is to allow the comparison of the different models and visualise its structural differences. There are several neuro-fuzzy architectures like:

Fuzzy Adaptive Learning Control Network (FALCON) C. T. Lin and C. S. Lee [9];
Adaptive Network based Fuzzy Inference System (ANFIS) R. R. Jang [5];
Generalized Approximate Reasoning based Intelligence Control (GARIC) H. Berenji [2];
Neuronal Fuzzy Controller (NEFCON) D. Nauck & Kruse [11];
Fuzzy Inference and Neural Network in Fuzzy Inference Software (FINEST) Tano, Oyama and Arnould [15];
Fuzzy Net (FUN) S. Sulzberger, N. Tschichold and S. Vestli [14];
Self Constructing Neural Fuzzy Inference Network (SONFIN) Juang and Lin [6].
Fuzzy Neural Network (NFN) Figueiredo and Gomide [4];
Dynamic/Evolving Fuzzy Neural Network (EFuNN and dmEFuNN) Kasabov and Song [7];

A summarised description of the five most popular neuro-fuzzy architectures is made in next section.

6.1 FALCON Architecture

The ***Fuzzy Adaptive Learning Control Network*** FALCON [9] is an architecture of five layers as it is shown in figure 3. There are two linguistics nodes for each output. One is for the patterns and the other is for the real output of the FALCON. The first hidden layer is responsible for the mapping of the input variables relatively to each membership functions. The second hidden layer defines the antecedents of the rules followed by the consequents in the third hidden layer. FALCON uses an hybrid learning algorithm composed by a unsupervised learning to define the initial membership functions and initial rule base and it uses a learning algorithm based on the gradient descent to optimise/adjust the final parameters of the membership functions to produce the desired output.

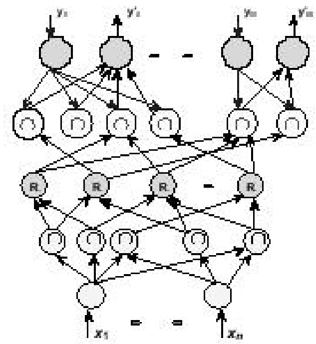

Figure 3. FALCON architecture.

6.2 ANFIS Architecture

The ***Adaptive Network based Fuzzy Inference System*** ANFIS [5] implements a Takagi Sugeno fuzzy inference system and it has five layers as shown in figure 4. The first hidden layer is responsible for the mapping of the input variable relatively to each membership functions. The operator T-norm is applied in the second hidden layer to calculate the antecedents of the rules. The third hidden layer normalizes the rules strengths followed by the fourth hidden layer where the consequents of the rules are determined. The output layer calculates the global output as the summation of all the signals that arrive to this layer.

ANFIS uses backpropagation learning to determine the input membership functions parameters and the least mean square method to determine the consequents parameters. Each step of the iterative learning algorithm has two parts. In the first part, the input patterns are propagated and the parameters of the consequents are calculated using the iterative minimum squared method algorithm, while the parameters of the premises are considered fixed. In the second part, the input patterns are propagated again and in each iteration, the learning algorithm backpropagation is used to modify the parameters of the premises, while the consequents remain fixed.

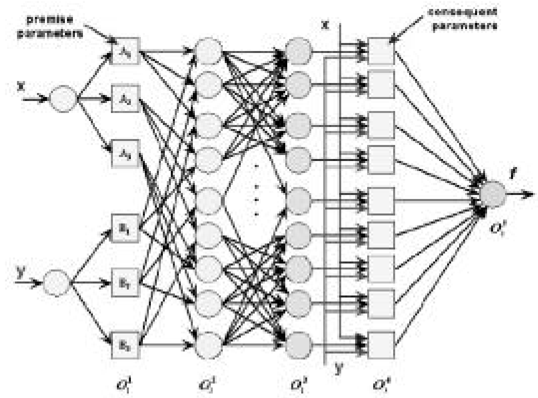

Figure 4. ANFIS architecture.

6.3 GARIC Achitecture

The ***Generalized Approximate Reasoning based Intelligence Control*** GARIC [2] implements a neuro-fuzzy system using two neural netwoks modules, ASN (Action Selection Network) and AEN (Action State Evaluation Network). The AEN is an adaptative evaluator of ASN actions. The ASN of the GARIC is an advanced network of five layers. Figure 5 illustrates GARIC-ASN structure. The connections between the layers are not weighted. The first hidden layer stores the linguistics values of all input variables. Each input can only connect to the first layer, which represent its associated linguistics values. The second hidden layer represents the fuzzy rule nodes that determine the compatibility degree of each rule using a softmin operator. The third hidden layer represents the linguistics values of the output variables. The conclusions of each rule are calculated depending on the strength of the rules antecedents calculated in the rule nodes. GARIC uses the mean of local mean of maximum method to calculate the output of the rules. This method needs for a numerical value in the exit of each rule. Thus, the conclusions should be transformed from fuzzy values for numerical values before being accumulated in the final output value of the system. GARIC uses a mixture of gradient descending and reinforcement learning for a fine adjustment of its internal parameters.

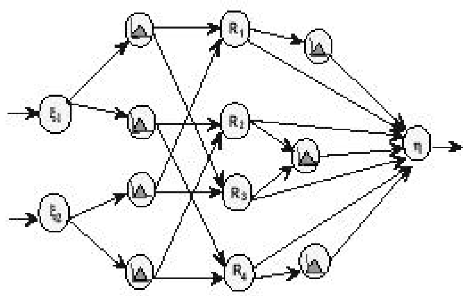

Figura 5. GARIC architecture.

6.4 NEFCON Architecture

The ***Neural Fuzzy Controller*** NEFCON [11] was drawn to implement a Mamdani type inference fuzzy system as illustrated in figure 6. The connections in this architecture are weighted with fuzzy sets and rules using the same antecedents (called shared weights), which are represented by the drawn ellipses. They assure the integrity of the base of rules. The input units assume the function of fuzzyfication interface, the logical interface is represented by the propagation function and the output unit is responsible for the defuzzyfication interface. The process of learning in architecture NEFCON is based in a mixture of reinforcement learning with backpropagation algorithm. This architecture can be used to learn the rule base from the beginning, if there is no *à priori* knowledge of the system, or to optimise an initial manually defined rule base. NEFCON has two variants NEFPROX (for function approximation) and NEFCLASS (for classification tasks) [14].

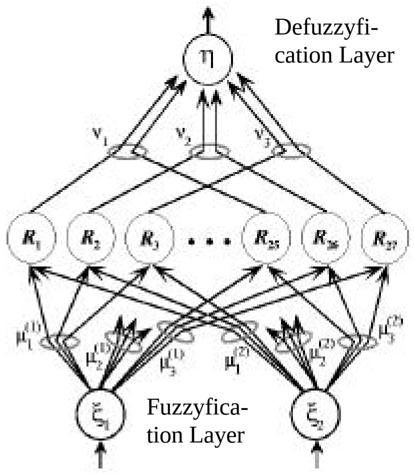

Figure 6. NEFCON architecture.

6.5 EFuNN Architecture

In ***Evolving Neural Fuzzy Network*** EFuNN [10] all nodes are created during the learning phase. The first layer passes data to the second layer that calculates the degrees of compatibility in relation to the predefined membership functions. The third layer contains fuzzy rule nodes representing prototypes of input- output data as an association of hyper-spheres from the fuzzy input and fuzzy output spaces. Each rule node is defined by two vectors of connection weights, which are adjusted through a hybrid learning technique. The fourth layer calculates the degree to which output membership functions are matched the input data and the fifth layer carries out the defuzzyfication and calculates the numerical value for the output variable. ***Dynamic Evolving Neural Fuzzy Network*** (dmEFuNN) [10] is a modified version of the EFuNN with the idea of not only the winning rule node's activation is propagated but a group of rule nodes that is dynamic selected for every new input vector and their activation values are used to calculate the dynamical parameters of the output function. While EFuNN implements Mamdani type fuzzy rules, dmEFuNN implements Takagi Sugeno fuzzy rules.

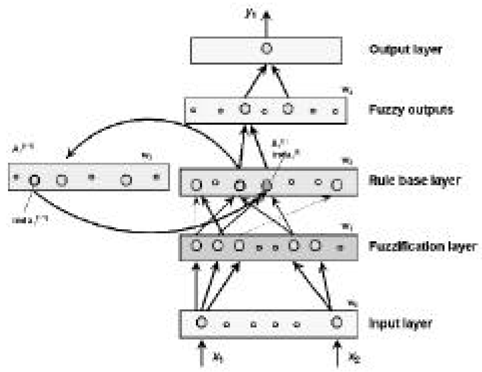

Figure 7. EfuNN architecture.

To get a more detail description of this architectures, beyond the specific pointed references made in this paper, a detailed survey was made by Abraham [1] in 2000 where it can be found a detailed description of several well known neuro-fuzzy architectures theirs respective learning algorithms.

7 Discussion and Application

The hybrid neuro-fuzzy systems present an interpretable model and they have learning capacities in a supervised way. In FALCON, GARIC, ANFIS, NEFCON, SONFIN and FINEST the learning process only concerns the adaptation of internal parameters of a fixed structure of the system. For complex problems, it will be computational demanding to determine all the parameters (of premises parameters, consequents parameters, number of rules, etc) because the parameters will grow exponentially.

An important characteristic of the architecture dmEFuNN and EFuNN is to make the training only in one iteration. This characteristic will allow the implementation of on-line adaptation in a simple way.

Abraham proposed [1] a evolutionary approach based on genetic algorithms for the optimisation of all parameters of the structure of a neuro-fuzzy system (type of fuzzy system, number of rules, parameters, inference operators, rules and membership functions).

In the industrial field, initially these architectures were applied in modelling non-linear systems and control engineering. Actually, however these architectures are used in almost all knowledge areas where a non-linear function should be approximated. The actual neuro fuzzy systems application areas are medicine, economy, control, mechanics, physics, chemistry, etc.

8 Conclusions

This article presents in a summarize way, the last decade of investigation in the area of the modelling non-linear functions through neuro-fuzzy systems. Duo to the vast number of common tools it continues to be difficult to compare conceptually the different architectures and to evaluate comparatively their performances. In generic terms the bibliography points that neuro fuzzy systems that implement Takagi-Sugeno type fuzzy inference systems get more accurate results than the approaches that implement neuro fuzzy inference systems of Mamdani type, although its bigger computational complexity. As a guide line for implementing highly efficient neuro-fuzzy systems they should have the following characteristics; fast learning; on-line adaptability; self-adjusting with the aim of obtaining the small global error possible; small computational complexity.

The data acquisition and the pre-processing of input training data are also very important for the success of the application of the neuro-fuzzy architectures. All the neuro-fuzzy architectures use the gradient descent techniques for the learning its internal parameters. For a faster convergence of the calculation of these parameters it would be interesting to explore other efficient algorithms of neural networks learning as the conjugated gradient or Levenberg-Marquardt search in spite of the backpropagation algorithm.

References

[1] A. Abraham and Baikunth Nath, " Hybrid Intelligent Systems: A Review of a decade of Research", School of Computing and Information Technology, Faculty of Information Technology , Monash University, Autralia, Technical Report Series, 5/2000, 2000, pp. 1-55.

[2] H. R. Berenji and P. Khedkar, "Learning and Tuning Fuzzy Logic Controllers through Reinforcements", IEEE Transactions on Neural Networks, 1992, Vol. 3, pp. 724-740.

[3] E. Czogala and J. Leski, "Neuro-Fuzzy Intelligent Systems, Studies in Fuzziness and Soft Computing", Springer Verlag, Germany, 2000.

[4] M. Figueiredo and F. Gomide; "Design of Fuzzy Systems Using Neuro-Fuzzy Networks", IEEE Transactions on Neural Networks, 1999, Vol. 10, no. 4, pp.815-827.

[5] R. Jang, "Neuro-Fuzzy Modelling: Architectures, Analysis and Applications", PhD Thesis, University of California, Berkley, July 1992.

[6] F. C. Juang, T. Chin Lin, "An On-Line Self Constructing Neural Fuzzy Inference Network and its applications", IEEE Transactions on Fuzzy Systems, 1998, Vol. 6, pp. 12-32.

[7] N. Kasabov e Qun Song, "Dynamic Evolving Fuzzy Neural Networks with 'm-out-of-n' Activation Nodes for On-Line Adaptive Systems", Technical Report TR99/04, Departement of Information Science, University of Otago, 1999.

[8] B. Kosko, "Neural Networks and Fuzzy Systems: A Dynamical System Approach to Machine Intelligence", Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[9] T. C. Lin, C. S. Lee, "Neural Network Based Fuzzy Logic Control and Decision System", IEEE Transactions on Computers, 1991, Vol. 40, no. 12, pp. 1320-1336.

[10] D. Nauck, F. Klawon; R. Kruse, "Foundations of Neuro-Fuzzy Systems", J. Wiley & Sons, 1997.

[11] D. Nauck, R, Kurse, "Neuro-FuzzySystems for Function Approximation", 4th International Workshop Fuzzy-Neuro Systems, 1997.

[12] D. Nauck, "Beyond Neuro-Fuzzy Systems: Perspectives and Directions". Proc. of the Third European Congress on Intelligent Techniques and Soft Computing (EUFIT'95), Aachen, 1995.

[13] D. Nauck; "A Fuzzy Perceptron as a Generic Model for Neuro-Fuzzy Approaches". Proc. Fuzzy-Systems, 2nd GI-Workshop, Munich, 1994.

[14] S. Sulzberger, N. Tschichold e S. Vestli, "FUN: Optimization of Fuzzy Rule Based Systems Using Neural Networks", Proceedings of IEEE Conference on Neural Networks, San Francisco, March 1993, pp. 312-316.

[15] S. Tano, T. Oyama, T. Arnould, " Deep Combination of Fuzzy Inference and Neural Network in Fuzzy Inference", Fuzzy Sets and Systems, 1996, Vol. 82(2), pp. 151-160.

[16] L. A. Zadeh; "Fuzzy Sets", Information and Control, 1965, Vol. 8, pp. 338-353.