FUSION BETWEEN FUZZY SYSTEMS, GENETIC ALGORITHMS AND ARTIFICIAL NEURAL NETWORKS

Angel Garrido

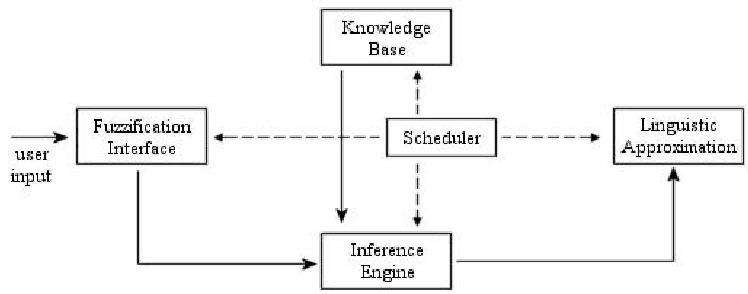
Abstract. Fuzzy System (FS) and Artificial Neural Networks (ANN) are complementary methods. But while ANN can learn from data FS cannot.

Also, Genetic Algorithms (GA) are complementary to FS. While the FS are easy to understand, the GA are not, although they have the ability to learn, and so on. Many researches have been devoted to its fusion. Our purpose is to give a survey of these questions.

2000 *Mathematics Subject Classification*: Fuzzy Systems, Genetic Algorithms, Artificial Neural Networks, Artificial Intelligence.

1.Introduction to Fuzzy Systems

An Expert System (ES) is a program which contains human expert knowledge. It gives answers to the user queries. For this, we need inference methods. A very useful generalization of the mentioned ES is the Fuzzy Expert System (FES). It consists of an ES which can deal with fuzzy information, that is, with some degree of uncertainty. In the "real world" the human expert can express his or her knowledge by means of linguistic terms. So, we represent, in a natural way, such knowledge by fuzzy rules and linguistic modifiers (that reflect terms as "almost", for instance). Therefore, we need to apply fuzzy inference methods. The form in which we usually store the knowledge is the base rule, generally in the logical form: "if-then".



Observe that this structure is similar to that of a Fuzzy Logic Controller (FLC). As in such case, we have:

- Inference Engine

- Fuzzification Interface

- Knowledge Base

Remember that the *Inference Engine* (IE) works as the motive driving the knowledge. For this, it is also called a *Decision-Making Logic.*

But it is very different, because instead of the Defuzzification Module, we have now the *Linguistic Approximation Module (LAM).* Obviously, related to the aforementioned linguistic modifiers, acting on the belonging function.

Such LAM is devoted to attempting to find a linguistic term next to the obtained fuzzy set. The closer to it, the better. How do you work? By a measuring technique, using the distance between fuzzy sets.

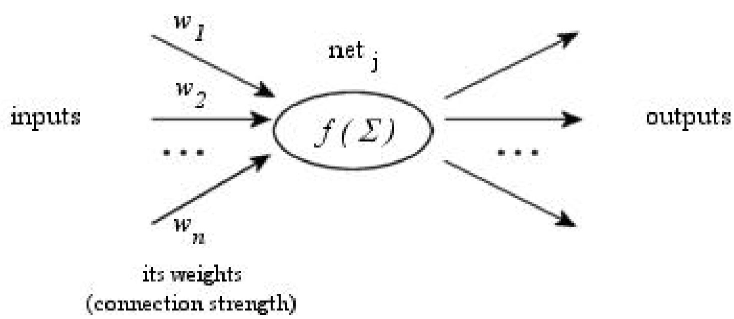
2.Fusion between Fuzzy Systems and ANN

Both methods have advantages and disadvantages, being complementary, because the FS cannot learn from data, whereas NN have this as a basic property. Recently, many studies have been devoted to attempting the fusion of them.

Remember some concepts:

An *Artificial Neural Network* (ANN) is a computational model of the operation of the human brain. It is composed of a set of *nodes,* connected by arcs between nodes, also called *links.*

Their essential characteristic is the assignation of a "weight" (numerical value) to each link. The variation of this associate value is the seminal idea of giving the capability of long-term storage to a NN, and therefore the possibility of learning. Because such learning, usually, takes place by updating the weights (see figure below).



Node of ANN

Each node has a set of input links, coming from other nodes and a set of output links, from our node to the subsequent nodes. And a function, *f*, which is non-linear, and it is called *activation function.*

In this way, each output of a node is generated from three subsets: the inputs of the precedent layer connected with it, the respective weights of the nodes of the layer that are linked to our node, and the action on these links of the function f, our activation function.

If we denote:

o_i, as the output of the node i and therefore, an input of the node *j*, connected with i

w_{ij}, as the weight of the link between both nodes, i and j
f, as the activation function

then we have:

$$net_j = \sum_i w_{ij} o_i$$

and so, the output of the node j (net_j) results:

$$o_j = f\left(net_j\right)$$

As such activation function, *f*, can be used *any nonlinear function*. For instance, the *sigmoidal function*, as you can see later:

$$f\left(x\right) \equiv \left(1 + e^{-x}\right)^{-1}$$

It is the most frequently used, because of this good property:

$$f\,' = f\ \left(1 - f\right)$$

3.Learning Algorithms

The ANN can be classified into:

feedforward ANN

and

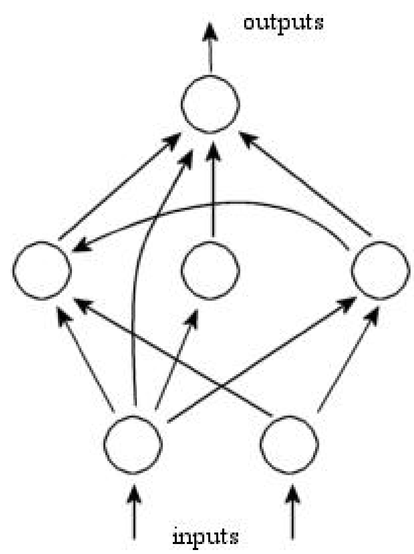
feedback ANN

The difference consists of the presence or absence of feedback cycles.

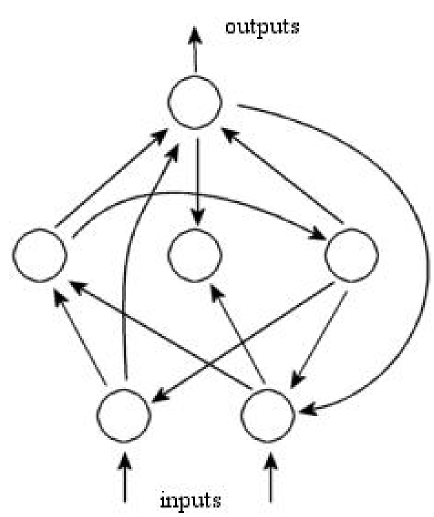
So:

The ANN of the first type has only feedforward links. This means that such ANN does not have feedbak cycles. In this mode, we can not use the output of a node as an input of that node. Nor directly neither indirectly.

The ANN of the second type, admits the possibility of feedback cycles. So, in this case, it is possible to use the output of a node as the input of that node.



Feedforward ANN



Feedback ANN

In such case, there is no guarantee of stability for the ANN. It is so because of the feedback cycle. For this reason, is possible that some converge to a stable point, or to a limit-cycle, or perhaps show a chaotic or divergent behaviour. Circumstances typical of non-linear systems with the presence of feedback.

It is possible to classify the learning algorithms ANN into *two classes:*

- the first, when is *supervised learning*

- the second, the *unsupervised learning*

In the supervised case, we have the training data as a set of pairs, composed by corresponding values of input and output. During the process of learning, the weights are modifying, in such way that the input - output mapping is, in each step, closest to the training data.

The second type differs in that while the supervised *present* target answers for each input, the unsupervised *has* such answers. In this way, the weights are only adjusted when the patterns of inputs have not got target answers. The most usual, in this situation, is the classification of the input patterns in similarity classes, or categories.

4.Multilayer Perceptron

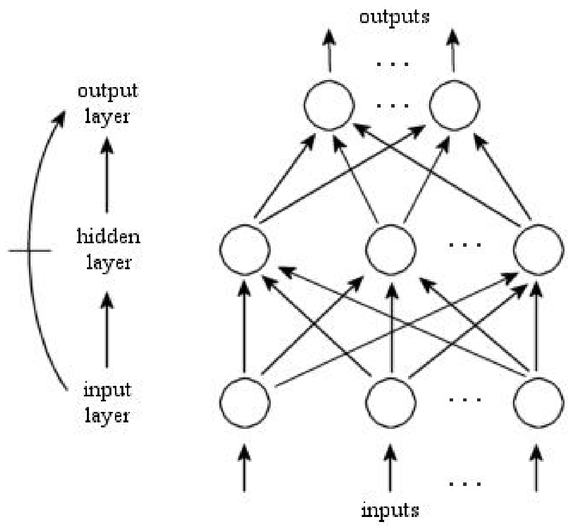
Between the ANN and its learning algorithms, the most usual are the Multilayer Perceptron Networks (MPN). Their learning algorithms are called "error backpropagation method".

Each MPN is composed by several layers, and must be included into the layered feedforward ANN. Remember that a layer is a set of nodes which do not have interconnection, through links. Each node admits only connection with the precedent or subsequent layer, but not more.

About the most usual activation function, as we said, it is the *sigmoidal function,* defined by:

$$f(x) \equiv \left(1 + e^{-x}\right)^{-1}$$

As its derivative is very simple:



Multilayer Perceptron
(three layers, in this case)

$$f'(x) = f(x) - f^{2}(x)$$

The error backpropagation algorithm is more accessible. And so, we can control the change of the weights of the links gradually. Such changes allow the minimization of the error, E.

We have:

$$E \equiv \frac{\sum (t_k - o_k)^2}{2}$$

where t_k and o_k are the expected and the actual outputs, respectively, regarding the node k, in the output layer.

The variation in the weight can be evaluated through the derivative of E with regard to $w_{ji.}$ Then, we change to the opposite direction of the derivation. In this way, if we denote as α the real small number, the learning rate, such variation will be:

$$\Delta w_{ji} = -\alpha \frac{\partial E}{\partial w_{ji}}$$

A related problem is their convergence. If the modification is made gradually, E must decrease, converging finally to a stable point, but perhaps not to the global optimum point. This occurs because the method is only based in derivations. This can be obviated by the introduction of momentum term.

How could be the value of α? The greater is α, the higher is the learning speed, but also the oscillations. And reciprocally, if α is less each time, the learning is slower, but the size of the oscillations decreases. Such election can be made, therefore, in function of our purposes.

5.Fusion of FS with ANN

As we know, both technologies are complementary. Whereas ANN can learn from data, for the FS this is not possible. As counterpart, the knowledge represented by ANN is usually very difficult. Not in the case of FS, very easy because it can be expressed by linguistic terms, and the rules are of the type "if - then".

We can establish four categories into the fusion of both systems:

1) Construct ANN by FS

2) modify FS by supervised ANN learning

3) introducing membership functions through ANN

4) by the concatenation of FS and ANN.

6.Fusion of FS with GA

Also we found here two complementary techniques. Whereas the FS are easy of understand, the GA not, but they have the ability of learning.

GA can be interpreted as an optimization method, a learning mechanism or a general-purpose method.

The set of solutions is called the *population,* and each element is a *chromosome,* also a *individual.* The most usual representation of such individuals are as binary strings of fixed length.

These provides a very effective method in difficult problems of optimization.

The more usual sequence over a population is as this:

1) we depart of a population of size n, initializing such population

2) we evaluate the fitness, into the population of each chromosome

3) if we reach the stop condition, we return the best chromosome

4) we must select $\frac{n}{2}$ pairs from the population

5) we create n new chromosomes, by the crossover operator

6) we apply the mutation operator to such new chromosomes

7) we replace the old with the new chromosomes

8) return to the step number 2.

This produces a cycle which finishes when we reach a predetermined limit, or by a convergence criterion.

The attempts to obtain such fusion between FS and GA can be classified in two groups:

1) the control of parameters in GA by FS

and

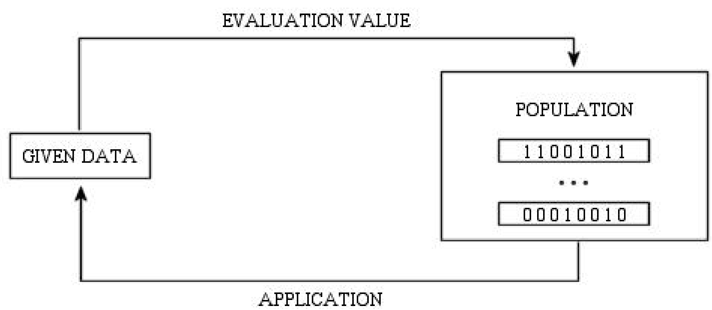
2) the identification of GA with FS

As you know, the FS are not learning algorithms, but the GA can be used as such l.a. for the FS. The presence of parameters in GA makes its modification during the process possible.

7. Identification between GA and FS

A very difficult problem is to take the adequate selection of the fuzzy membership function. Because this election admits many subjectivity aspects and perhaps the need of too much time. Nevertheless, the FS are very useful indeed, in the control of systems.

We can see the scheme of such identification in the last figure.



Taking as representation, into the population, the known codification through binary strings of prefixed length.

The researches can be categorized in two groups:

1) Through the modification of the parameters, in a pre-existing Fuzzy System. Generally, they are the Fuzzy Rules, or perhaps the membership function, f, or both. Because the tuning or modification of such function in GA is analogue to NFS (Neuro Fuzzy Systems). We encode the membership function in chromosomes, and the search of the better f is reached by GA.

2) The second procedure is made by the construction of a FS through GA. Therefore, we do not need a pre-existing FS, as in the precedent case. So, the determination of the parameters, by GA, is fixed directly. The chromosomes which we use include the parameters, as can be the membership functions (through linguistic terms) or their numerical values.

References

[1] Borrajo et al., *Inteligencia Artificial: Métodos y técnicas,* Centro de Estudios Ramón Areces, Madrid, 1993.

[2] Fernández Galán et al., *Problemas resueltos de Inteligencia Artificial Aplicada. Búsqueda y Representación,* Addison-Wesley Iberoamericana, Madrid, 1998.

[3] Garrido, Angel, *Problemas Abiertos y Problemas Resueltos en Inteligencia Artificial,* Madrid, 2005.

[4] Garrido, Angel, *Some connections between Linear Algebra and Fuzzy Logic*, *ILAS 04*, "International Conference on Linear Algebra and Its Applications", held in the University of Coimbra, July 2004. Proceed., special volume of *LAA*.

[5] Garrido, Angel, *Logical Foundations of AI and Matrix Theory.* International Conference on Foundations of the Formal Sciences V: Infinite Games (FotFS V), held in the Rheinisches Friedrich Wilhelm Universität, *Bonn*, Nov. 2004, *Trends in Logic.*

[6] Garrido, Angel, "MT & A I". *Int. Conference on Matrix Theory*, held at Technion, Haifa (Israel), January 2005, *LAA,* special volume of Proceedings.

[7] Garrido, Angel, *Complexity of Algorithms*, EpsMsO ´05, Intern. Conf. held at Athens, Greece, July 2005, full text disposable in CD-Rom edition of the Congress.

[8] Garrido, Angel, *Logics in A. I.*, International Conference *Logic in Hungary,* as celebration of 100th anniversary of Laszlo Kalmar and Rosza Peter, organization: Janos Bolyai Mathematical Institut, held at Uni Corvinus, Budapest, August 2005.

[9] Garrido, Angel, *P vs. NP (Complexity Classes and Graph Theory)*, Intern. Confer. EuroComb05, TU Berlin, September 2005.

[10] Ginsberg, *Essentials of artificial intelligence,* Morgan Kaufmann Publ., San Francisco, Cal., 1993.

[11] Hertz, J., *Introduction to the Theory of Neural Computation,* Santa Fe Institute,1991.

[12] Hajek, P., *Metamathematics of Fuzzy Logic,* Kluwer Academic Publishers, Boston, 1998.

[13] Kandel, A., *Fuzzy Mathematical Techniques with Applications,* Addison-Wesley Publishing Co., Mass., 1986.

[14] Kosko, B., *Neural Networks and Fuzzy Systems*, Prentice Hall, Englewood Cliffs, NJ, 1992.

[15] Michalewicz & Fogel, *How to Solve it: Modern Heuristics,* Springer-Verlag, Corrected Third Printing, Berlín, 2002, Chap. 13: "Fuzzy Systems", pages 363-384.

[16] Mira et al., *Aspectos Básicos de la Inteligencia Artificial,* Ed. Sanz y Torres, Madrid, 1995.

[17] Yager, R. R., and Zadeh, L. A., *Fuzzy Sets, Neural Networks and Soft Computing,* Van Nostrand Reinhold, New York, 1994.

[18] Zadeh, L. A., and Kacprzyk, J., *Fuzzy Logic for the Management of Uncertainty,* John Wiley & Sons, New York, 1992.

[19] Zurada, J. M., *Introduction to Artificial Neural Systems,* West Publ, St. Paul, 1994.

Author:
Ángel Garrido Bullón
Department of Fundamental Mathematics.
Faculty of Sciences UNED Senda del Rey, 9. 28040-Madrid (Spain)
email:algbmv@telefonica.net