

**BELDJEHM Mokhtar**

National Amman University

Faculty of Information Technoloy(FIT)

Departement of Computer Science

Postal Code 19328

Amman-JORDAN

E-mail: mbeldjehm@ammanu.edu.jo  
beldjmok@hotmail.com

## ***Machine Learning based on the Possibilistic-Neuro Hybrid Approach:Design and Implementation***

**Abstract:** This paper deals with the fusion of fuzzy set theory and neural networks. The first fundamental design part consisted in elaborating a new general approach to combine or to fuse fuzzy set theory(or possibility theory) and neural networks approach, which enabled us to develop the *fennec* tool, a possibilistic-neural system. This system uses a feed-forward possibilistic-neuro system, where neuron activation is computed through possibility (and necessity) measure and/or combination with MIN-MAX Compositional rule, and a new fast supervised fuzzy learning mechanism based on the original idea of approximation of fuzzy MIN-MAX relational equations systems so as to minimize some cost function expressed by an error signal. In order, to ensure portability, the *fennec* tool is totally re- implemented in ANSI C++ language . The last practical part consisted in applying *fennec* tool to a real world biomedical diagnosis on proteins/biological inflammatory syndroms.

**Key words:** Machine learning, possibilistic-neuro system, supervised learning, possibility theory, fast MIN-MAX approximation learning algorithm, learnability, validity, biomedical diagnosis.

## 1. Introduction

The adoption of hybrid approach combining several paradigms constitutes an interesting tool in fuzzy modeling, especially useful in extracting or tuning fuzzy *if-then* rules for complex real-world problems. This hybrid approach has recently emerged as very promising area in Artificial Intelligence (AI) fields[1-7, 11-18, 20, 22-29]. It consists to design hybrid learning models combining fuzzy set theory, neural nets and genetic algorithms. Such models rather than conventional ones, are well-suited to resolve complex real-world problems, because of learning, transparency and tolerance. This new field of research is called *Soft-Computing*, according to Zadeh L. A.(Berkeley university): "*It may be argued that is soft-computing rather than hard-computing that should be viewed as the foundation of Artificial Intelligence(AI)*".

The basic idea in the *neuro-fuzzy* or *fuzzy-neuro* approach is to design a feed-forward *fuzzy-neural* network used as a knowledge representation system, it allows incorporating initial fuzzy knowledge in connections, this constitutes a better starting point for learning a special task. Unlike "blank" network approach where learning requires more examples and hence it is extremely time consuming, learning with a fuzzy-neural network does not require more examples because of the incorporated initial knowledge. The key problem in designing such systems is to find efficient learning algorithms. Most of these systems use (fuzzy) backpropagation algorithm, however this algorithm is based on numerical optimisation techniques and, hence unsuited for dealing fuzzy knowledge. We believe that the key idea in a learning process is the nature of computations used. We describe herein through *fennec* : a fuzzy-neuro tool, a new approach to combine fuzzy sets theory (see Zadeh [30-32]) and neural networks. It uses a fast *MIN-MAX supervised fuzzy learning algorithm* to train a feed-forward fuzzy-neural network from I/O examples so as to minimize some cost function. This fuzzy-neural network is designed to extract or to tune *if-then* fuzzy rules. The *fennec* tool is implemented in C with application to *biomedical diagnosis* on proteins/biological inflammatory syndroms.

## 2. Presentation of the *fennec* tool

The hybrid tool that we have developed is designed to extract or to tune fuzzy *if-then* rules by supervised learning from I/O examples. Software practical structure of the tool is illustrated in figure 1. It is composed of five basic parts.

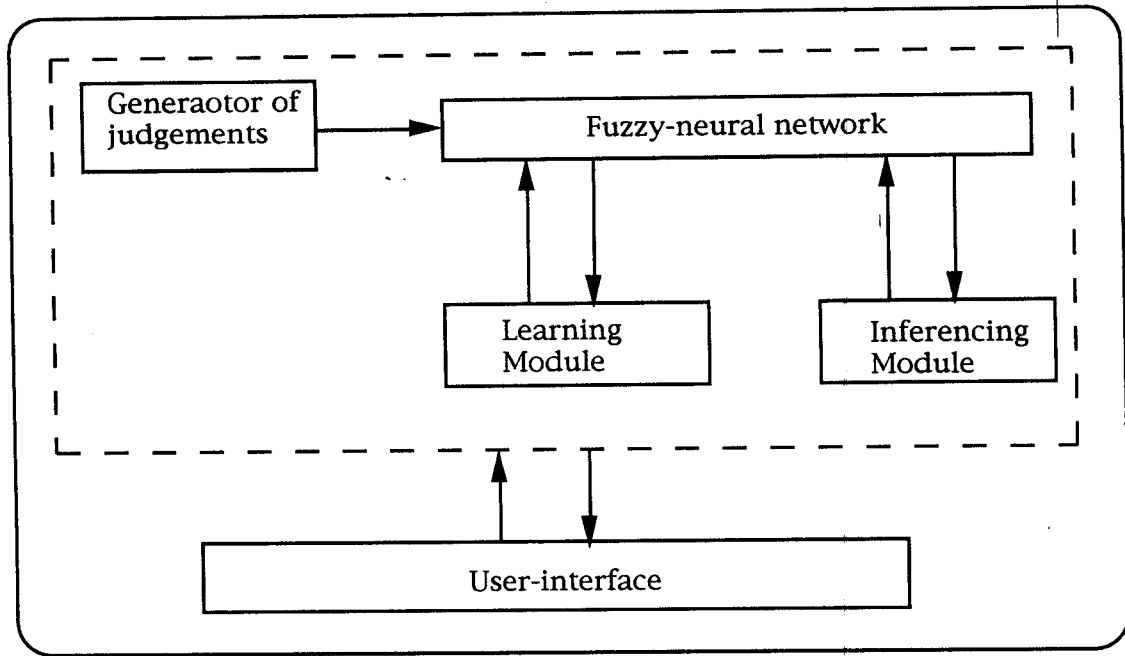


Fig.1. Architecture of the fennec tool

- i) A feed-forward fuzzy-neural network, used as a knowledge representation system. It allows incorporating initial fuzzy knowledge in connections, this constitutes a better starting point for learning a special task. This fuzzy-neuro model in addition to providing a useful structure for representing knowledge allows for a natural framework for learning fuzzy *if-then* rules.
- ii) A generator of judgements which consist, given the initial linguistic knowledge incorporated in the network (represented by membership functions), to generate automatically by performing appropriate adjustment operations, a fuzzy hypothesis from such knowledge during the learning process. Such hypothesis constitutes the new knowledge in learning. The main idea is to start learning with an initial judgement less specific as possible, a such judgement is used to preweight the neural network.
- iii) A learning module which uses a new fast supervised learning mechanism based on the resolution of MIN-MAX fuzzy relational equations. This module allows to construct a learning table and to find the topology of the network used.
- iv) An inferencing module which consist on a fuzzy reasoning mechanism to perform fuzzy inference for an arbitrary pattern presented at the input layer of the network. The fuzzy-neuro network emulates the compositional rule of inference.

v) A user interface which allows the communication between the user and the system.

### 3. Model overview

We consider herein to design a fuzzy-neural network according to the scheme Fuzzy to Neural(or to switch from fuzzy systems to neural networks). Fuzzy *if-then* weighted rules have long been used by Dubois et al.[9], such a rule(diagnosis type) looks like this :

if  $(e_1 \text{ is } w_{k1}, c_{k1})$  And  $(e_2 \text{ is } w_{k2}, c_{k2})$  And  $(e_3 \text{ is } w_{k3}, c_{k3})$  And  $(e_5 \text{ is } w_{k5}, c_{k5})$  then  $s_k$ .

Let  $c_{kj}$  weight represents the grade of importance of "  $e_j$  is  $w_{kj}$  ".

Inversely  $a_{kj} = 1 - c_{kj}$  represents the grade of no-importance of "  $e_j$  is  $w_{kj}$  ".

We propose herein (as shown in Figure 2) a fuzzy-neural feed-forward network. From the semantic point of view, such figure reflects a neural representation of a fuzzy *if-then* weighted rule (diagnosis type). We begin with a brief description of the model : two types of weights are associated with the connections.

Type 1 : Direct connections between input cells ( $e_j$ ) and output cell ( $s_k$ ) with only linguistic weights ( $w_{kj}$ ), interpreted as labels of fuzzy sets, characterizing the variations of the input cells ("  $e_j$  is  $w_{kj}$  ") with the output cell ( $s_k$ ), in this case we have  $a_{kj} = [0, 0] = 0$ .

Type 2 : Connections between input cells ( $e_j$ ) and output cell ( $s_k$ ) via intermediate cells ( $H_{kj}$ ), weights associated to connections between input cells ( $e_j$ ) and intermediate cells ( $H_{kj}$ ) are linguistic ( $w_{kj}$ ), weights associated to connections between intermediate cells ( $H_{ik}$ ) and output cell ( $s_k$ ) are herein numerical intervals ( $[0, 1] \supseteq a_{kj}$ ), instead of scalar value ranging in the interval  $[0, 1]$  ( $a_{kj} \in [0, 1]$ ), this extension provides much flexibility for the network, such interval reflects a domain of possible values of no-importance for the corresponding connection.

Let us consider now cell activation for an arbitrary output cell ( $s_k$ ), as illustrated in figure 2, where only connections used in activation of  $s_k$  appear. From the semantic point of view, such figure reflects a neural representation of an *if-then* fuzzy weighted rule.

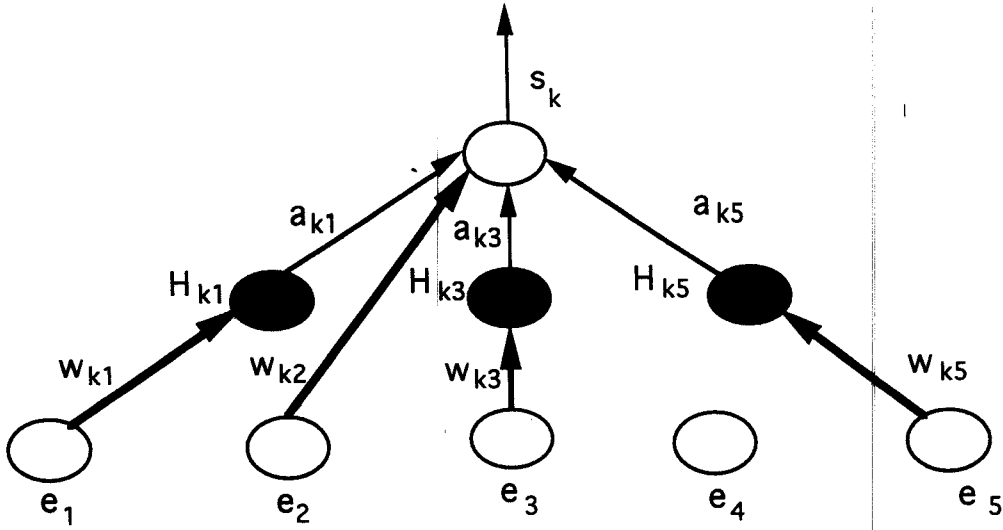


Fig.2. Schematic representation of the model

Let  $\Pi(e_j; w_{kj}) = \text{Sup}[w_{kj} \cup e_j]$  be possibility measure associated to fuzzy sets  $w_{kj}$  and  $e_j$ . Let  $N(e_j; w_{kj}) = \text{Inf}[w_{kj} \cup \text{Not } e_j]$  be necessity measure associated to fuzzy sets  $w_{kj}$  and  $e_j$  (see Zadeh [32] and Dubois & al.[9] for more details). In general our model is regid by three fuzzy equations, for the example of figure 2 :

$$\pi_k = \bigwedge_{j \in \{1, 2, 3, 5\}} (\Pi(e_j; w_{kj}) \vee a_{kj}) \quad (1)$$

$$\eta_k = \bigwedge_{j \in \{1, 2, 3, 5\}} (N(e_j; w_{kj}) \vee a_{kj}) \quad (2)$$

$$s_k = [\eta_k, \pi_k] \quad (3)$$

Observe that MAXIMUM ( $\vee$ ) limits lower amplitudes of inputs, we have

$(\Pi(e_j; w_{kj}) \vee a_{kj}) = a_{kj}$  , if  $\Pi(e_j; w_{kj}) \leq a_{kj}$  , and amplifies higher ones  $(\Pi(e_j; w_{kj}) \vee a_{kj}) = \Pi(e_j; w_{kj})$  , if  $\Pi(e_j; w_{kj}) \geq a_{kj}$  , so the MIN-MAX composition indicates a somewhat excitatory character. It is worthwhile to notice that MIN-MAX composition as containing MIN and MAX operations is strongly nonlinear.

Observe that When  $a_{kj}=1$ , the term  $\Pi(e_j; w_{kj}) \vee a_{kj}$  (respectively  $N(e_j; w_{kj}) \vee a_{kj}$ ) is deleted in the application of MINIMUM( $\wedge$ ). So, it is now clear that  $a_{kj}$  reflects a notion of no-importance, we point out that it is strongly hard if not impossible to make values assignment to grades of no-importance in practical applications, bearing this in mind, we will propose a mechanism to learn such grades of no-importance. In order to reduce learning time, we have used only possibility measure, necessity measure may be used in inferencing. For an arbitrary pattern presented at the input layer of the network, the inferencing module uses the set of extracted *if-then* fuzzy rules to deduce an output conclusion, each output cell ( $S_k$ ) performs the same inferencing algorithm. The feed-forward network used may be seen as a physical device, it emulates the compositional rule of inference, many iterations may be necessary for computing the response (decision) of the network.

## 4. The statement and the resolution of the learning problem

### 4.1 The statement of the learning problem

Learning concern both the associated membership functions of linguistic( $w_{kj}$ )(or the deep structure) and numerical weights( $a_{kj}$ ), from the practical point of view, learning is implemented using *Generate-and-Test* strategy(performed over initial membership functions associated to the initial judgement by the application of appropriate adjustment operations). Generation procedure adopted is automatic, during a learning session there is no need of human operator intervention.

Numerical learning consists for each output cell( $S_k$ ) to update its corresponding numerical weights by the resolution(or the approximation) of a system of MIN-MAX equations so as to minimize some local cost function.

We have used for the network as a cost function the global error over the training set and over the set of output cells. Learning consists to prefer from the set of exact or approximate solutions a class of solutions which minimize the error, in other terms during a session the learning mechanism try to prefer the

corresponding deep structure which minimize the global error, bearing this in mind, we will propose our learning mechanism.

Informally, the learning problem for an arbitrary output cell ( $S_k$ ) can be stated as follows:

Given a) and b)

a) One(or several) initial configuration of membership functions associated to the linguistic weights(or initial judgement)

$$\mu_{w_{ki}} = \mu_{w_{ki}}^{(0)}, i=1 \dots m$$

b) A target set(obtained from the training set) composed of pairs

$$\{ \langle E_j = (e_{1j}, e_{2j}, \dots, e_{mj})^t, b_{kj} \rangle, j=1 \dots n \}, \text{ where } n \text{ is the number of examples}$$

Fuzzy learning consist to find c) and d) so as to minimize some local cost function(defined by a local error)

c) A unique configuration of membership functions associated to the linguistic weights(generated by the generator of hypothesis), let  $l$  such that ( $l \in [0, r]$ )

$$\mu_{w_{ki}} = \mu_{w_{ki}}^{(l)}, i=1 \dots m$$

where  $l \in [0, r]$  and  $r$  denotes the total number of generated hypothesises

d) A unique configuration of numerical weights

$$a_{ki}, i=1 \dots m$$

## 4.2 The resolution of the learning problem

In order to elaborate a learning mechanism exhibiting some "artificial intelligence", we have to deal with three key questions:

- How to generate a fuzzy hypothesis by the generator ?
- How to prefer one hypothesis than another one ?
- How to approximate a MIN-MAX system of equations ?

- How to establish a learning algorithm, which allows to get the "best" generalization with the corresponding linguistic(the deep structure) and numerical weights configurations ?

#### 4.2.1 Functioning of the generator of judgements

The generation process combined with numerical learning may be seen as a fuzzy learning, or from a computing point of view a new kind of fuzzy optimization techniques, the main idea is how to find the "best" deep structure for a special task, and hence how to approximate well the empirical knowledge of the target set.

We have used to operate the generator three appropriate adjustment operations

##### hybrid hedges :

Two hybrid hedges illustrated in figure 3, they combine shift and power operations, they are defined by :

$$\mu_{gc}(A)(u) = (\mu_A(u+y))^{1/2}, \quad \mu_{dr}(A)(u) = (\mu_A(u-x))^2$$

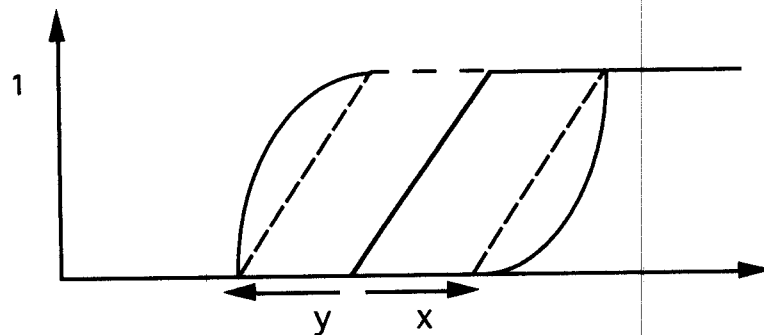


Fig . 3. Hybrid hedges

##### Contrast intensification

The operation of contrast intensification is illustrated in figure 1, it is defined by :

$$\mu_{int}(A)(u) = \begin{cases} 2 \mu_A^2(u) & \text{if } \mu_A(u) < 0.5 \\ 1 - 2(1 - \mu_A(u))^2 & \text{if } \mu_A(u) \geq 0.5 \end{cases}$$



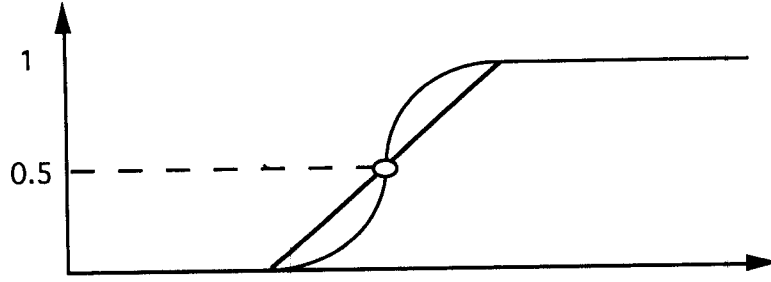


Fig .4. Contrast intensification

As pointed out by Zadeh [31] this operation has the effect of reducing the entropy suggested by De Luca & Termini [8] of  $\text{int}(A)$ .

#### 4.2.2 Resolution of MIN-MAX system

The resolution of *fuzzy relations equations* constitutes a good tool in fuzzy modeling. The theory[2, 10, 19-20] provides us with a set of explicit formulas expressing solutions for various types of equations and their systems. However, the existence of solutions of the system is not known in advance. This makes any preliminary analysis rather tedious if not impossible. In this part of our work we reformulate the problem of solving a system of MIN-MAX from *interpolation-like* format to *approximation-like* one. This means that instead of trying to find exact solution, we try to find the best approximate solution.

Any scalar and any element of vectors or matrices are assumed to have its value in the interval  $[0, 1]$ .

##### i) The statement of interpolation-like format

Our problem can be stated as follows : " Given an  $m \times n$  matrix  $R$  and an  $n$  vector  $b$ , find all  $m$  vectors  $a$  such that  $a \Delta R = b$ ", where  $\Delta$  is the MIN-MAX composition. In the case of the existence of solution, it can be shown that the set of solution has the structure of *inf-semi-lattice*.(see Beldjehem[2] for proof and more details). We propose herein a fast new resolution analytic algorithm which reduces the complexity of resolution in practical cases.

##### ii) Definitions and properties

Operators of resolution :

$$x \varepsilon y = \begin{cases} 0 & \text{si } x \geq y \\ y & \text{si } x < y \end{cases} \quad x \theta y = \begin{cases} 1 & \text{si } x > y \\ y & \text{si } x \leq y \end{cases}$$

\* Derivation:

Let  $a = (a_1, a_2, \dots, a_m)^t$  be a column vector such that  $a_i = \hat{a}$  ou 1,  $i=1, \dots, m$ , where  $\hat{a} = \wedge a$ . We define the set  $GN(a)$  of vectors  $a^*$  derivited from  $a$ , such that  $\exists k$  unique :  $a_k^* = a_k = \hat{a}$  and  $\forall i \neq k \ a_i^* = 1$

For example,

if  $a = (0.5, 1, 0.5, 1, 1)^t$ ,  $GN(a) = \{(0.5, 1, 1, 1, 1)^t, (1, 1, 0.5, 1, 1)^t\}$

>> Relation :

Let  $C$  an  $m \times n$  matrix,  $C_h$  and  $C_k$  two column of  $C$ , we define the >>relation " $C_h$  deletes  $C_k$ " by :

$C_k >> C_h$  i.e  $(C_{ih} \neq 1 \Rightarrow (C_{ik} \neq 1 \text{ and } C_{ih} \leq C_{ik}), C_{ih} = 1 \Rightarrow C_{ik} \text{ is arbitrary})$ .

For example,  $(0.2, 1, 1, 0.2, 1)^t >> (0.1, 1, 1, 1, 1)^t$

iii) Description of the resolution algorithm

Algorithme Resolve (S)

Begin /\* compute the lower bound of the inf-semi-lattice by

$S = \{ \}; \underline{a} = \vee (R \varepsilon b)$  matrix multiplication with the  $\varepsilon$  operator and by selecting the MAX of each row of  $(R \varepsilon b)$  \*/

$G_0 = \underline{a} \theta (R \theta b)$  /\* matrix multiplication with the  $\theta$  operator \*/

Reduce  $(G_0, G_f)$  /\* obtaining  $G_f$  from  $G_0$  by deleting column unities and by performing the >> relation \*/

For each  $G^* \in GN [G_f]$  Do

Begin /\* compute the upper bounds of the inf-semi-lattice

$z = \wedge G^*$  by selecting the MIN of each row of  $G^*$  \*/

$S = S \cup \{a : \underline{a} \leq a \leq z\}$  /\* include  $a$  in the set of solutions  $S$  \*/

End

End

iv) Example

$$R = \begin{bmatrix} 0.6 & 1 & 0.1 & 0.4 & 0.2 \\ 0.3 & 0.2 & 0.7 & 0 & 0.5 \\ 0.4 & 0.6 & 0.7 & 0.6 & 0.1 \\ 0.8 & 0 & 0.5 & 0.2 & 0.6 \end{bmatrix}$$

$$b = [ \quad 0.4 \quad 0.5 \quad 0.1 \quad 0.4 \quad 0.2 \quad ]$$

$$R \varepsilon b = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.5 & 0 & 0.4 & 0 \\ 0 & 0 & 0 & 0 & 0.2 \\ 0 & 0.5 & 0 & 0.4 & 0 \end{bmatrix}$$

$$\underline{a} = [0 \quad 0.5 \quad 0.2 \quad 0.5]$$

$$R \theta b = \begin{bmatrix} 1 & 1 & 0.1 & 0.4 & 0.2 \\ 0.4 & 0.5 & 1 & 0.4 & 1 \\ 0.4 & 1 & 1 & 1 & 0.2 \\ 1 & 0.5 & 1 & 0.4 & 1 \end{bmatrix}$$

$$G_0 = \underline{a} \theta (R \theta b) =$$

$$\begin{array}{ccccc} & C1 & C2 & C3 & C4 & C5 \\ \begin{bmatrix} 1 & 1 & 0.1 & 0.4 & 0.2 \\ 1 & 0.5 & 1 & 1 & 1 \\ 0.4 & 1 & 1 & 1 & 0.2 \\ 1 & 0.5 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

$C4 \gg C3 \quad C5 \gg C3$ , by reduction of  $G_0$  we have

$$G_f =$$

$$\begin{array}{ccc} & C1 & C2 & C3 \\ \begin{bmatrix} 1 & 1 & 0.1 \\ 1 & 0.5 & 1 \\ 0.4 & 1 & 1 \\ 1 & 0.5 & 1 \end{bmatrix} \end{array}$$

$$GN[G_f] = \{$$

$$\left[ \begin{array}{ccc} 1 & 1 & 0.1 \\ 1 & 0.5 & 1 \\ 0.4 & 1 & 1 \\ 1 & 1 & 1 \end{array} \right], \left[ \begin{array}{ccc} 1 & 1 & 0.1 \\ 1 & 1 & 1 \\ 0.4 & 1 & 1 \\ 1 & 0.5 & 1 \end{array} \right] \}$$

$$z_1 = [0.1 \quad 0.5 \quad 0.4 \quad 1]$$

$$z_2 = [0.1 \quad 1 \quad 0.4 \quad 0.5]$$

the set of solutions is then,  $S = \{ a : \underline{a} \leq a \leq z_1, \underline{a} \leq a \leq z_2 \}$

$\underline{a}$  is the lower bound of the inf-semi-lattice  $S$

$z_1, z_2$  are the upper bounds of the inf-semi-lattice  $S$ .

v) The statement of the approximation-like format

We are interested herein by the approximation-like format, our problem can be stated as follows : " Given an  $m \times n$  matrix  $R$  and an  $n$  vector  $b$ , find all  $m$  vectors  $a$  such that  $a \Delta R \supset b$ ", where  $\Delta$  is the MIN-MAX composition and  $\supset$  denotes the *fuzzy inclusion* operation.

Let us consider the case when there is no solution for the system(it does not satisfy the necessary condition).

$$R = \begin{bmatrix} 0.5 & 0.6 & 0.1 & 0.3 & 0.6 \\ 0.7 & 0 & 0.8 & 0.4 & 0.7 \\ 0.8 & 0.3 & 0.5 & 0.7 & 0.6 \\ 0.4 & 0.8 & 0.6 & 0.8 & 0.7 \\ 0.4 & 0.4 & 0.7 & 1 & 0.6 \\ 0.9 & 1 & 1 & 1 & 0.8 \end{bmatrix}$$

$$b = [ 0.3 \quad 0.3 \quad 0.5 \quad 0.4 \quad 0.5 ]$$

*Theorem 1.*

The best approximator(from the fuzzy inclusion point of view)correspond to the lower bound( $\underline{a}$ ) of the inf-semi-lattice. It can be computed easily by the above algorithm( see Beldjehem[2] for proof and more details). It can be shown that our computing algorithm has a polynomial complexity  $O(m \times n)$ .  
by the application of the same resolution algorithm, we compute

$$\underline{a} = [ 0.5 \quad 0.3 \quad 0 \quad 0 \quad 0 \quad 0 ]$$

$$z_1 = [ 0.5 \quad 0.3 \quad 1 \quad 1 \quad 1 \quad 1 ]$$

$$z_2 = [ 1 \quad 0.3 \quad 0.5 \quad 1 \quad 1 \quad 1 ]$$

$$z_3 = [ 1 \quad 0.4 \quad 0.3 \quad 1 \quad 1 \quad 1 ]$$

By performing the MIN-MAX composition, we have

$$b = [ 0.3 \quad 0.3 \quad 0.5 \quad 0.4 \quad 0.5 ] \text{ (the target vector)}$$

$$\underline{a} \Delta R = [ 0.4 \quad 0.3 \quad 0.5 \quad 0.4 \quad 0.6 ]$$

$$z_1 \Delta R = [ 0.5 \quad 0.3 \quad 0.5 \quad 0.4 \quad 0.6 ]$$

$$z_2 \Delta R = [ 0.7 \quad 0.3 \quad 0.5 \quad 0.4 \quad 0.7 ]$$

$$z_3 \Delta R = [ 0.7 \quad 0.3 \quad 0.5 \quad 0.4 \quad 0.6 ]$$

we have  $z_2 \Delta R \supset z_3 \Delta R \supset z_1 \Delta R \supset \underline{a} \Delta R \supset b$  (fuzzy inclusion)

We observe that  $\underline{a} \Delta R$  is the best approximator of  $b$ , concerning the  $z_1$ , the comparison shows that  $z_1 \Delta R$  approximate well  $b$  than  $z_2 \Delta R$  and  $z_3 \Delta R$ , so the algorithm selects  $z_1$  which is better than  $z_2$  and  $z_3$ , the algorithm takes into account a class (W) of approximate solutions defined by

$W = \{ a : \underline{a} \leq a \leq z_1 \}$ , which is a vector with interval components.

In practical applications the set  $W$  may be reduced only to one vector with numerical component ranging in  $[0, 1]$ , formally in this case the algorithm takes the vector  $W = \{ \underline{a} \}$ , which allows to reduce the complexity of computations performed during resolution (polynomial time).

Since our algorithm is valid for both interpolation-like and approximation-like formats, it allows to resolve the more general following problem :

" Given an  $m \times n$  matrix  $R$  and an  $n$  vector  $b$ , find all  $m$  vectors  $a$  such that  $a \Delta R \supseteq b$ ".

#### 4.2.3 Formulation of a learning session

Formally, from the computing point of view, for each output cell ( $s_k$ ), a learning session consist to resolve or to approximate  $(r + 1)$  systems of MIN-MAX equations, as follows:

$$a \Delta R^{(0)} \supseteq b$$

$$a \Delta R^{(1)} \supseteq b$$

.

$$a \Delta R^{(r)} \supseteq b$$

Learning consist to prefer the configuration of the best approximate solution (from the fuzzy inclusion point of view) which minimize the local cost function and hence the corresponding deep structure. In other terms the learning process find the "best" deep structure which correspond to the matrix  $R^{(l)}$  ( $l \in [0, r]$ ) such that :

$$a \Delta R^{(j)} \supseteq a \Delta R^{(l)} \supseteq b, \forall j=0 \dots r$$

Learning try progressively to minimize the local cost function by the generation and the resolution of a new system. It can be shown that the *fennec* system is an

*universal approximator*, furthermore it is now clear that the ultimate aim of learning is to generate a consistant system which correspond to exact solution (or to establish a *universal interpolator*), however it seems that is not alwayse the case in practical applications. In general the value of the local cost function may be seen as a quality index for a learning session. Learning has high speed due to its simplicity and analytic nature.

## 5. Application to biomedical diagnosis

We have been interested with an application to real-world data, in order to study the general behavior of the system and to illustrate its possibilities. The *fennec* tool is implemented in C language with application to biomedical diagnosis on proteins/biological inflammatory syndroms B.I.S [21]. The same approach can be used in general for pattern recognition, fuzzy control, quality control and managerial applications.

### 5.1 Presentation of the problem

We begin by a brief description of the problem. The complexity of proteins/B.I.S problem is due to the complexity of inflammatory process. Linguistic weights are interpreted by labels of fuzzy sets (as shown in figure 5 for the protein Haptoglobin in relation with Vasculitis). The proteins/B.I.S problem can be stated as fellows : Given a protein profile composed of five normalized values (represented by the input pattern to the network). Assign the appropriate diagnosis group (or B.I.S) from eleven groups (represented by the output of the network).

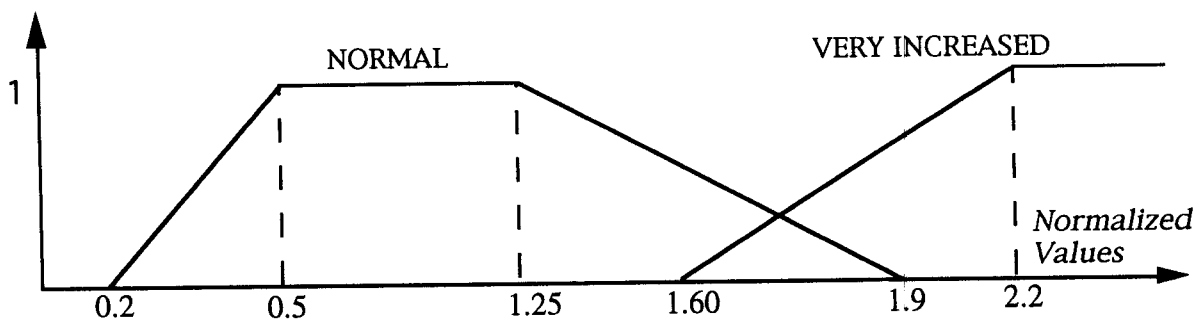


Fig. 5. Haptoglobin is *VERY INCREASED* in relation with Vasculitis

## 5.2 Simulations results

We give herein the learning table obtained (Table 6) at the end of learning session by presenting 163 I/O examples of the problem to the network.

	Hi1	Hi2	Hi3	Hi4	Hi5
S <sub>1</sub>	[0.5, 0.5]	[0.0, 0.0]	[0.0, 0.0]	[0.0, 0.0]	[0.0, 0.0]
S <sub>2</sub>	[0.8, 0.8]	[0.4, 0.4]	[0.0, 0.0]	[0.0, 0.0]	[0.4, 1.0]
S <sub>3</sub>	[0.9, 0.9]	[0.0, 0.0]	[0.2, 0.2]	[0.7, 0.7]	[0.2, 0.2]
S <sub>4</sub>	[0.8, 0.8]	[0.2, 0.2]	[0.0, 0.0]	[0.2, 0.2]	[0.8, 0.8]
S <sub>5</sub>	[0.8, 1.0]	[0.3, 0.3]	[0.3, 0.3]	[0.0, 0.0]	[0.2, 0.2]
S <sub>6</sub>	[0.3, 0.3]	[0.3, 0.3]	[0.2, 0.2]	[0.8, 0.8]	[0.0, 0.0]
S <sub>7</sub>	[0.3, 0.3]	[0.2, 0.2]	[0.2, 0.2]	[0.7, 0.7]	[0.0, 0.0]
S <sub>8</sub>	[0.3, 0.3]	[0.3, 0.3]	[0.3, 0.3]	[0.0, 0.0]	[0.0, 0.0]
S <sub>9</sub>	[0.3, 0.3]	[0.3, 0.5]	[0.1, 0.1]	[0.3, 0.3]	[0.0, 0.0]
S <sub>10</sub>	[0.1, 0.1]	[0.0, 0.0]	[0.0, 0.0]	[0.0, 1.0]	[0.2, 0.5]
S <sub>11</sub>	[0.5, 0.5]	[0.0, 0.0]	[0.0, 1.0]	[0.0, 0.0]	[0.0, 0.0]

Table 6. The learning table

Simulations show that by increasing training set cardinality, extracted fuzzy rules became progressively discriminants (The range of intervals representing numerical weight diminishes progressively) and finally stabilize (to each output cell only one fuzzy rule is associated, and most numerical weights stabilize with numerical scalar values as shown in Table 6). This confirm our intuition that the network try to satisfy constraints imposed by examples. In general the value of entropy decreases progressively when increasing the number of examples. The main idea in learning is to partition the input space into fuzzy regions taking into account both the initial fuzzy judgement(explicit knowledge) and the training set(empirical knowledge), this is the main advantage of hybrid fuzzy-neuro modeling approach.

## 6. Concluding remarks

The *fennec* tool is designed to extract or to tune *if-then* diagnosis fuzzy rules by supervised learning from I/O examples and initial fuzzy knowledge, the advantages of the tool are :

i) dealing with input pattern composed of fuzzy numbers (in addition to numbers).

ii) exhibiting generalizing capabilities and fault tolerance, this is due to the nature of computations used : MIN, MAX,  $\epsilon$ ,  $\theta$  and possibility measures.

iii) The learning algorithm implemented in *fennec* has high speed due to its simplicity and analytic nature.

iv) Transparency : interval numerical weights may be interpreted as grades of relative no-importance for the corresponding connections.

v) Unlike "blank" network approach where learning requires more examples and hence it is extremely time consuming, learning with *fennec* does not require more examples because of the incorporated initial knowledge.

There is a main difference between backpropagation algorithm which is based on gradient descent methods, and our learning mechanism which is based on the resolution algorithm of a system of MIN-MAX equations used as an optimization procedure. The basic idea here is that due to the nature of operators MIN, MAX,  $\epsilon$  and  $\theta$  performed in resolution, the learning process makes a projection of the target vector over connections associated to numerical weights (some components of the target vector are affected to numerical weights). More even, learning in our system runs very fast, even on standard computers, this makes our approach well suited for real time applications such as on-line process control. We conclude that our learning algorithm may be seen as a new kind of fuzzy optimization technique, we propose in future work to investigate and develop a study of *value approximation* and *tolerance analysis* of our system. This study is also related to the *validity and learnability* problems : "Find all fuzzy functions learned by the system by the presentation of a polynomial number of examples to the network ?", This is an open problem in fuzzy learning. Currently we look to use our system to resolve quality control and managerial applications.



## References

- [1] Amano A., Arisuka T., Hataoka N., Ichikawa A. (1989), On the use of neural networks and fuzzy logic in speech recognition, *IJNN'89*, Vol. 1, pp. 1301 - 1306.
- [2] Beldjehem M.(1993), Un apport à la conception des systèmes hybrides neuro-flous par algorithmes de résolution d'équations de relations floues en MIN-MAX : *le système fennec*, Ph.D. Thesis(in french), University of Aix-Marseille 2(UFR of Luminy).
- [3] Beldjehem M. (1993), *Le système fennec*, *BUSEFAL*, 55, pp. 95 - 104.
- [4] Beldjehem M.(1993), *fennec*, un générateur de systèmes neuro-flous, dans les Actes, Applications des ensembles flous, Nimes'93, pp. 209 - 218.
- [5] Beldjehem M.(1994), The *fennec* system, ACM Symposium on Applied Computing (SAC' 94), Track on fuzzy logic in Applications (Marsh 6-8, phoenix, AZ).
- [6] Beldjehem M.(1994), *fennec*, a fuzzy-neuro system, Submitted to Fuzzy Sets and Systems.
- [7] Berenji H. R. (1991), A reinforcement learning-based architecture for fuzzy logic control, *International journal of approximate reasoning*, pp. 267 - 292.
- [8] De Luca A., Termini S. (1972), A definition of a nonprobabilistic entropy in the setting of Fuzzy sets theory, *Information and Control*, Vol. 20, pp. 301 - 312.
- [9] Dubois D., Prade H. (1988), Testemale C., Weighted fuzzy pattern matching, *Fuzzy sets and systems*, Vol. 28, pp. 313 - 331.
- [10] Di Nola, Pedrycz W., Sanchez E. and Sessa S., *Fuzzy relation equations and their Applications to Knowledge engineering* (Kluwer Academic publishers, Dordrecht, 1989).
- [11] Eppler W. (1990), Implementation of fuzzy rules production systems with neural networks, *Parallel processing in neural systems and computers*, Eckmiller R., Hartman G. and Hausk G. (Eds.), Elsevier Science publishers B. V. (North-Holland), pp. 249 - 252.
- [12] Furuya T., Kokubu A., Sakamoto T. (1988), NFS : Neuro fuzzy inference system, *Iizuka'88*, pp. 219 - 230.
- [13] Gupta M., Gurzalczy M. (1991), Fuzzy neuro-computational technique and its application to modeling and control, *IFSA'91-AI*, pp. 46 - 49.
- [14] Keller J. M., Hant D. J. (1985), Incorporating fuzzy membership functions into perceptron algorithm, *IEEE trans. In pattern analysis and machine intelligence*, Vol. Pami-7, N°6, pp. 693 - 699.
- [15] Kosko B. (1988), Bidirectional associative memories, *IEEE Trans. Sys. Man & Cybernetics*, Vol. 18, N° 1.
- [16] Kosko B. (1992), Fuzzy systems are universal approximators, *Proc. of IEEE Int. Conf. on Fuzzy Systems*, San Diego'92, pp. 1153 - 1162.

- [17] Lee C. C. (1990), A self-learning rule-based controller employing approximate reasoning and neural networks concepts, *Int. J. Intell. Syst.* Vol. 5, N° 3.
- [18] Lee S. C., Lee E. T. (1974), Fuzzy sets and neural networks, *J. Cybern.* Vol. 4, N° 2, pp. 83 - 103.
- [19] Pappis C. P., Adamopoulos G. I. (1992), A software routine to solve the generalized inverse problem of fuzzy systems, *Fuzzy Sets and Systems* 47, pp. 319 - 322.
- [20] Pedrycz W. (1993), Fuzzy neural networks and Neurocomputations, *Fuzzy Sets and Systems* 56, pp. 1 - 28.
- [21] Sanchez E., Bartolin R. (1990), Fuzzy inference and medical diagnosis, a Case study, *Biom. Fuzzy Syst. Bull.*, Vol. 1, pp. 4 - 21.
- [22] Saito T., Mukaidono M. (1991), A learning algorithm for max-min network and its application to solve fuzzy relation equations, *proceedings, IFSA'91, Lowen R. Roubens M., Eds.* pp. 46 - 49.
- [23] Takagi H. (1990), Fusion technology of fuzzy theory and neural networks, *Survey and Future Directions, Proc. of the Int. Conf. on Fuzzy Logic & Neural Network (Iizuka, Japon, July 20 - 24)*, pp. 13 - 27.
- [24] Takagi H., Hayashi I. (1988), Artificial-neural-network-driven fuzzy reasoning, *International workshop on fuzzy systems applications (Iizuka'88)*, pp. 183 -184.
- [25] Wang L. X. (1992), Fuzzy systems are universal approximators, *Proc. of IEEE Int. Conf. on Fuzzy Sytems, San Diego'92*, pp. 1163 - 1170.
- [26] Watanabe N., Kawamura A., Masuaka R., Ohwada Y., Asakawa K. (1990), Fuzzy control using neural network, *40th of meeting IPSJ 3C-10*, pp. 148 - 149.
- [27] Yamakawa T., Tomoda S. (1989), A Fuzzy neuron and its application to pattern recognition, *Proc. of the 3rd IFSA'89 Cogress, Seattle, WA*, pp. 30 - 38.
- [28] Yamaguchi T., Imasaki N., Haruki K. (1990), Fuzzy rule realisation on associative system, *IJCNN'90, Vol. 2*, pp. 720 - 723.
- [29] Yager R. R. (1988), On a semantics for neural networks based on linguistic quantifiers *Tech. Report-MII-1103, IONA College, Machine Intelligence Institute, New Rochelle, New York 10801.*
- [30] Zadeh L. A. (1965), Fuzzy sets, *Information and Control*, Vol. 8, pp. 338 -353
- [31] Zadeh L. A. (1973), Outline of e new approach to the analysis of complex systems and decision processes, *IEEE Trans. Sys. Man & Cybernetics* 3, pp. 28 - 44.
- [32] Zadeh L. A. (1978), Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets & Systems*, Vol. 1, pp. 3 - 28.