

# FUZZY -NEURAL PATTERN RECOGNITION

M.Engel, M.Leclercq, C. Pradels

UFR Sciences University of Rouen. 76821 Mont Saint Aignan Cedex, France  
CNRS, G.R. Commande Symbolique et neuromimétique  
phone: 35.14.65.89 fax: 35.14.66.18

**Abstract** - The common techniques of neural networks seem to can be apply perfectly to the classification and to the pattern recognition, but the learning can sometime be long and the complexity of the final network very big. An association between the neural networks and the fuzzy way seems feasible and looks to can make significantly easier the learning period and to allow also to simplify its structure. The fuzzy logic expresses the knowledge in an explicit manner by its rules of inferences, while the neural networks assimilate that knowledge in the weight during the learning, the knowledge is so here implicit. In merging these two approaching, we can introduce the connectionnist techniques in the fuzzy logic operating in the best way for every underlying analogies between these two approaching.

It is that last possibility that is handled in that work. We can, by that association build a neural network of simple structure where we meet the stage of the fuzzy reasoning. The work here described tries in a first time to identify a paraboloid by a simple structure in the aim to well understand the developed association in the aim to apply it subsequently to the recognition of characters.

## INTRODUCTION

Fuzzy Logic is used in a lot of fields (Controllers, pattern recognition, ...) since its apparition in 1974. Some applications have even been marketed in Japan in particular in controlled appliances. But its used in pattern recognition remains modest, results not being very promoted.

At the opposite, neural networks (NNs) make the object of a lot of researches. They bring up a different approach than linguistic or algorithmic systems. Their two main qualities are their parallel structure which offer a good noise resistivity and their ability to learn from examples. NNs have, nowadays, a large field of applications from model and control systems prediction to pattern recognition.

Since little time, researchers work about NNs and Fuzzy Logic association. In fact, both try to reproduce the human reasoning. There are several ways to include their own qualities: We can introduce Fuzzy Logic in the elaboration of the network, or put connections into fuzzy rules, but it could be possible to make a fuzzy-neural network. This is this third solution that we've study.

The target of this article is to suggest a feasible association for pattern recognition and more specifically for characters recognition. Beyond NNs, we'll tackle, in a first time, the Fuzzy Logic linguistic to display in a second time, a fuzzy-neural network. A third part consists of testing this association by applications, in particular by the recognition of a paraboloid and by a type characters recognition approach.

## THE NEURAL NETWORKS

### Introduction

The target of NNs is at once to understand and then to try to reproduce the human brain working which always intrigues a lot of researchers by its incredible quickness and its fabulous ability to work in spite of the lack of explicit instructions. Then, we've tried to

reproduce the mechanics of human brain apprenticeship thanks to an artificial network. There are different productions of NNs distributed in two basic groups: layers networks and loops networks. We'll do not here study these different types. We'll just describe in part layers networks and more especially multi-layers networks which will be used for our association.

### Multi-layers network:

The **multi-layers network** is characterized by a neural distribution in layers. We distinguish three sorts of layers: • **The input layer** which transmits only data from input to neural of the first hidden layer. • **One or more hidden layers** which extract characteristics from input vector which square with the treatment of the inpatternation. • **The output layer** which give the answer of the network.

Each neural is connected either to the nearer layer (then we talk about **simple layers networks**), or to any layer located downstream (these networks are then called **not simple layers networks**). The connections between neural could also be done from a layer to itself but they almost look like loops networks connections.

### Apprenticeship:

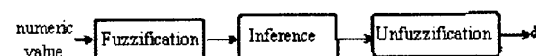
The big problem of NNs is the apprenticeship of weights  $w_i$  because all the knowledge is memorized in the synaptic coefficients which are fixed during the apprenticeship step. We distinguish two sorts of apprenticeship: The supervised apprenticeship where input-output associations are presented to the network by a "teacher", and the not-supervised apprenticeship where we don't know outputs in relation to inputs. There are different learning methods (the minor square method, the retropropagation of gradient, the Windrow-Hoff method, ...) which target is to minimize a criterion (often called "cost function") by weights adjustment. We won't describe here these methods.

## THE FUZZY LOGIC

### Introduction:

The human reasoning bases generally on imperfect knowledge, either because we have a doubt about their validity, then they are uncertain ("*perhaps she'll come tomorrow*"), or because we meet a difficulty to express them clearly, then they are inexplicit ("*she'll come tomorrow but I don't know when*"). Probabilistic methods grapple to uncertain events when the Fuzzy Logic is a concept which allows to treat the human inexplicit reasoning. Fuzzy sets theory permits to generalize events from natural causes appealing to mathematic tools. It presents at least 2 interests: It allows to modelize inexplicit or vague knowledge; It is the only mean to treat at the same time numeric and symbolic knowledge.

Here is a scheme (see figure 1) representing the general methodology for the Fuzzy Logic utilization:



- Figure 1-

## Fuzzification

This is a stage of transition from a numerical to a symbolic field. This stage is indispensable as soon as we want to manipulate precise or not physical data thanks to the fuzzy set theory. It allows for one thing to determinate the belonging function to a fuzzy set.

**Fuzzy set** Here is the definition of a fuzzy set according to ZADEH :

Given a set  $X$  countable or not, and  $x$  an element of  $X$ , then a fuzzy set  $A$  member of  $X$  is a set of couples such as equation (1):

$$\forall x \in X \{ (x, \mu(x)) \} \quad (1)$$

with  $\mu(x)$  the belonging function which takes data in the interval  $[0;1]$ .

If this interval is reduced to the  $\{0;1\}$  value, then the set  $A$  is an ordinary set.

**Belonging Function** It determinates 3 parts :

- One area where referential elements belong completely to a fuzzy set ;  $\Rightarrow \mu(x)=1$
- One area where elements are excluded,  $\Rightarrow \mu(x)=0$
- One fuzzy area (called transition area) where belonging to the set is estimated at different degrees.  $\Rightarrow 0 < \mu(x) < 1$

The belonging function is the belonging degree of  $x$  to the fuzzy set. Its part consists in precisising numerically the meaning of a fuzzy set expressed in principle in linguistic pattern. This is the transition of numerical to symbolic field. There are different belonging functions such as function in trapezium pattern ; triangular function ; Gaussian function (continually derivable). Pattern and parameters of different belonging functions must be adjusted in particular case to treat.

## Inference

It consists in treating the inpatternation by fuzzy rules.

**Fuzzy rules** They reveal a link between elementary fuzzy propositions : "The error is big **and** positive, **then** the control is zero" ... They pattern the inference systems core used in control and they relate the reasoning a human could make about an imperfectly defined knowledge. A fuzzy rule breaks up in 2 parts :

**The premises** : The place where one or several fuzzy propositions are set out.

**The conclusion or the consequence** : The answer.

A rule describes a causal link between the premises of data  $x_i$  and the conclusion thanks to fuzzy implication operators.

**Fuzzy implication operators** There are different operators of fuzzy implication : T\_norm, T\_conorm, Minimum of Mandani and Zadeh, Maximum of Lukasiewicz and Giles, product of Goguen, ...

The 2 operators usually used in the applications of the fuzzy sets theory are the Max. and Minus functions. Nevertheless, T\_norm functions could be used as conjunction operators instead of minus functions, while T\_conorms could take place of the Max. in disjunctions.

## Unfuzzification

Unfuzzification is the operation which allows to get through a linguistic datum pattern perception, a numerical variable physically applicable : this is the opposite step of fuzzification. There are several feasible methods including the Max. and the center of gravity. When the conclusion of the rule is distinct, specific and sure (that is when all of the fuzzy rules facing one to another have a no-fuzzy conclusion), unfuzzification is then implicit.

Example :

**If the outside temperature is high**  $\Rightarrow$  fuzzy proposition.

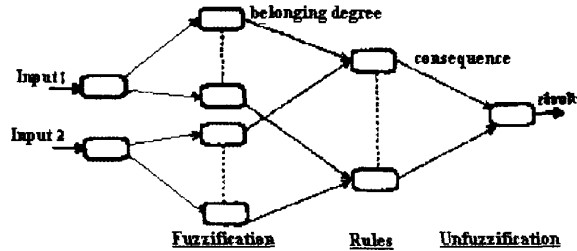
**Then turn off the heating.**  $\Rightarrow$  distinct, specific and sure decision.



The fuzzification will give the belonging degree which allows to obtain the consequence thanks to fuzzy rules. The unfuzzification will give the numerical result.

## Association

We can represent this analyze in the pattern as follow (figure 2)



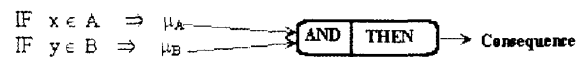
- Figure 2-

The first hidden layer neural represent different areas of the Fuzzy Logic. Then, the different outputs of this layer give us the belonging degree correspondent to the different areas. We introduce in the second hidden layer, rules of the Fuzzy Logic inference.

And then, we obtain the rule:

**IF  $x$  AND IF  $y$  THEN Consequence.**

The output layer will give us the result which could be obtained by unfuzzification. The scheme above (see figure 3) highlight the mass parallelism between the fuzzy reasoning and the connectionist perception. We recognize NNs in the weights incorporation by apprenticeship.



- Figure 3-

We can add more or less predefined weights which give a tipping between **explicit knowledge** ( $\Leftrightarrow$  Fuzzy Logic : The weights are all predefined) and **implicit reasoning** ( $\Leftrightarrow$  Neural Networks : Weights are established by apprenticeship).

## Conclusion:

This association seems to bring about the apprenticeship which is thus less complex (some weights being preestablished) and about the comprehension of the knowledge course in NNs (raising gradually the degree of complexity, we could see the evolution of the reasoning).

## APPLICATIONS

In a first time, we have test this fuzzy-neural network to identify a paraboloid :

RECOGNITION OF A CURVE : represented by the equation (2)

$$y = x_1^2 + x_2^2 \quad (2)$$

This network will be composed of :

- two input  $x_1$  and  $x_2$  such as :  $-1 \leq x_1 \leq +1$  and  $-1 \leq x_2 \leq +1$

- a output  $y : 0 \leq y \leq 2$

This application which gives a simple structure, will allow us to well understand the Fuzzy-neural association for a subsequent implementation for characters recognition.

#### Specification

First, we must built the fuzzy-neural network correspondent at this example . That's why, we 'll follow Fuzzy Logic reasoning courses.

**Fuzzification** It's the matter to break up the sets composed by  $x_1$  and  $x_2$  in fuzzy areas.  $x_1$  and  $x_2$  have the same probability on their set  $[-1;+1]$ ; So, there's no reason to privilege a part rather than another : That's why each area  $[-1;+1]$  is decomposed in equal parts . We have cut up the area in 5 equal parts :

- the "Zero" area : center (0);
- the "Middle Positive " area : center (0.5);
- the "Big Positive "area : center (+1);
- the "Middle Negative " area : center (-0.5);
- the "Big Negative " area : center (-1);

There are several manners to represent areas in Fuzzy Logic. We have choose the Gaussian function which eliminates the problem of derivability (important for weights apprenticeship), of intersection, of disymmetrical area, ... and which presents the advantage to have no distinct cutting up at the area (limits at the ends).

The equation of a Gaussian is like the model as follow (see equation (3)):

$$y = \exp[-(a \cdot x + b)^2] \quad (3)$$

$$\text{with } a = \frac{\sqrt{-\log(\varepsilon)}}{dx} \text{ and } b = -a \cdot m$$

The symmetrical curve of Gauss function is then represented by three parameters ( $m, dx, \varepsilon$ ) which correspond at :

- $m_i$  : center of the fuzzy area  $i$  considered (= localization of the area)

each parameter  $m_i$  of the 5 fuzzy areas have been fixed as :

- $m_0 = -1 \Leftrightarrow$  center of the "Big Negative" area.
- $m_1 = -0.5 \Leftrightarrow$  center of the "Middle Negative" area.
- $m_2 = 0 \Leftrightarrow$  center of the "Zero" area.
- $m_3 = 0.5 \Leftrightarrow$  center of the "Middle Positive" area.
- $m_4 = 1 \Leftrightarrow$  center of the "Big Positive" area.

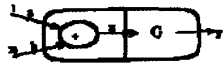
- $\varepsilon$  : residual value allowed at the  $dx$  distance from the Gaussian center. We've chosen  $\varepsilon = 0.1$  for  $dx = 0.5$  , Then  $\varepsilon$  and  $dx$  will be constant because all the areas are identical.

$b_i = -a \cdot m_i$ ,  $b_i$  is then fixed for each area,  $a$  being a constant

The output of a neural in the first hidden layer must answer to equation (4) :

$$\text{output\_neural } j = \exp[-(a \cdot \text{input\_neural } j + b_j)^2] \quad (4)$$

That well corresponds to a belonging degree of an input  $x_i$  to this area. A neural, making in general two tasks well specified (1.A weighted sum; 2.A processing). will be represented by the next pattern:



with equation (5)  $G = \exp[-z^2]$   
- Figure 4-

The weights  $a$  and  $b$  (see figure 4) are here preestablished. So, it won't be weights apprenticeship between the input layer and the first hidden one. We see that each neural of the first hidden layer will have 2 input :  $x_j$  and 1.

We have then created 5 areas by input data that we 'll connected two by two, this which corresponds at the creation of 25 neural (5x5) in the second hidden layer. This one corresponds to the fuzzy inference rule : If...AND...THEN....

**Inference** : we had to choose a rule of inference adaptable to our problem. That's why, we have executed a reasoning which gave us the type of decision to take in contact with the different areas. The 5 Gaussian functions which determine the belonging degree, are very displayed into the space; Then, a point of the space will belong to several Gaussian functions but the more even to insignificant way. We'd try to eliminate these residues. That's why , at once, the belonging degree will be added to each component. Then, the insignificant data will be deleted thanks to the Max.function (see equation (6)).

$$\text{Max}(1, \mu_a + \mu_b) \quad (6)$$

If we standardize this equation, we recognize the **rule of inference of Lukasiewicz** such as equation (7).

$$0 \leq \text{Max}(0, \mu_a + \mu_b - 1) \leq 1 \quad (7)$$

$$\mu = \mu_a + \mu_b - 1 \quad (8)$$

$$f = \text{Max}(0, \mu) \quad (9)$$

In fact, there are several possibilities (We could take a  $T_{\text{conorm}}$  for example) but this case seems to suit to our problem because :

- It's simple and easily computable,
- It fits perfectly itself to neural case : = **weighted input sums** ( $\Leftrightarrow$  equation (8) ) following by the **f function** ( $\Leftrightarrow$  equation (9)).

For the moment, we restrict to this inference rule.

Now, we incorporate this method to the network. The activation of the neural which implements the Max function of Lukasiewicz, is realized by a weighted sum corresponding of the equation (8). The processing is then represented by the function  $f$  (equation (9)). The implementation of this rule could be represented by the next scheme (see figure 5):



- Figure 5-

The synaptic weights are preestablished and equals to -1, 1 and 1 (see figure 4). There won't be weights learning between the first and the second hidden layer. Moreover, we notice each neural of this hidden layer have three inputs.

**Unfuzzification** That's the fuzzification opposite operation, that is a symbolic variable will be transpattern in a numerical language. In general, this function comply at the end of the processing, so, it's situated at the output layer. The two more standard methods use either the Maximum or the Barycenter function. Taking the last one permits to take all the receipted inpatterns into account whereas the Maximum method just take the more important inpatternation into consideration. That's why we have choose to take the center of gravity method. As there is just one output (see equation (2)), the unfuzzification hold only one neural. We reach then the final network (see figure 2).

The difference between Fuzzy Logic and NNs is the add of weights between the different links. The output  $y$  will take different

inpatterns into account (barycenter) but with various degrees thanks to the weights. Which data to give to these synapses? That's the part of the apprenticeship.

**Apprenticeship** : The apprenticeship consists in determining the weights in order that the network respond correctly to our problem (two inputs  $x_1$ ,  $x_2$  and one output  $y$  such as  $y$  resolve the equation (2)). We'll use a supervised method which the target is to minimize a cost function by weights progressive adjustment. The weights raised during the transition between the input layer and the first hidden layer and between the first hidden layer and the second one, are preestablished by the Fuzzy Logic rules. Apprenticeship is situated between the last hidden layer and the output neural.

The first hidden layer represents fuzzy areas set. As there are two inputs, a representation of these areas could be done in a two dimensions space. Each parcel will correspond to an intersection between a Gaussian for the  $x_1$  input and another for the  $x_2$  one. The second hidden layer executes at once the sum of the equation (8) which is equivalent to a bumbs field (3D Gaussian function) filed down by the Max function. We notice here there will have intersections between the different "bumbs". The barycenter method for unfuzzification will allow to take the different areas superposition into account.

Weights will modify the height of the bumbs so as to draw closer in the representation of a paraboloid.

This 3D representation allows us to estimate the value of different weights (in particular at the ends and in the center) so that the "bumbs field" tends towards a paraboloid. The weights of bumbs center in (1,1), (1,-1), (-1,1), (-1,-1) will approximately be equal to 2 (They'll probably be slightly below on account of the barycenter method which will take the neighboring bumbs into consideration). In the same way, the weights of the "central bump" ( $m=0$ ) will be nearer to zero. We compared these weights with different apprenticeship methods.

**Comparison of different apprenticeship method** At once, we try to measure the error  $E$  between the NNs output and the wished one (see equation (10)). We assign it a cost function  $J$  we'll try to minimize by an apprenticeship algorithm.

$$E = \text{wished\_output} - \text{Nns\_output} \quad (10)$$

The target of this part is to determinate the best learning algorithm for our case. In fact, no method seems to prevail. We have then tested the different methods we know.

In a first time, we have been interested by the **lowers square method**. The cost function is equal to the equation (11).

$$J = (d-y)^2 = E^2 \quad (E : \text{square Error; this equation is equivalent to equation (10)}) \quad (11)$$

The minimization of this criterion corresponds to try to find the weights value which cancels out the cost derivative. We then come up against a inversion problem of the matrix contained the different second hidden layer output values.

Another prestige method used for the NNs apprenticeship is the **Gradient retropropagation method**. It minimizes the same function than Lower Square method (at about coefficient). The difference is situated in level of the weights modification : The Lower Square method tries to minimize the function for the weights set so that the Gradient retropropagation method works on the weights taken individually.

The **Windrow Hoff method** is another learning method of the same kind of the Gradient retropropagation one which more to minimize the error, takes the previous weights variation into account. That's why this method is commonly called method of the retropropagation of the gradient of the moment. The advantage using this method instead of the Gradient retropropagation method is because between two weights modifications, the inputs change.

In order to visualize their advantages and their draw backs , we have tested these two methods simultaneously.

**Analysis of different apprenticeship methods results** After the network has been studied, we wonder about the real advantage of the barycenter method during the unfuzzification. In fact, this method don't make a linear interpolation between the different "bumbs" but rather add a non-linearity into our processes. We have think that a simple weighted sum would accurately smoothed our "field to bumbs". We have then testing these two cases. Moreover, the quick convergence of our network induced us to question ourselves about the initialization of weights. In order to check the good working for each learning method, we've voluntarily decided to initialize the weights at a random value contained between 0 and 10. This interval has judiciously been chosen to check the convergence known that the final values set of different weights is contained approximately between 0 and 2. The superior value of the interval has been chosen for not increase too much the duration of the apprenticeship.

An idea as regard size of results is the same for most of the parameters. So, we can't draw any conclusion regarding the number of iterations. We notice that however the method, weights values are practically identical. The error variation is because every the weights haven't yet reached their ended value. The barycenter method using Windrow Hoff always gives a middle error bellow  $10^{-6}$ . It seems the best method of apprenticeship. It would probably increase the base of recognition to rule on the recognition level.

**Possible Improvements** Our fuzzy-neural network gives good results : Level of not recognition reach about to  $10^{-3}$ . Nevertheless, we tried to find feasible mutations to improve our system.

**Trend to the NNs** One advantage of NNs is a good immunity to failures thanks to the repartition of the knowledge. In our network, we lost this benefice. It happens then naturally to add connections between the different layers. But, the only feasible links adjunctions could intelligently be done between two hidden layers. This mutation entails necessary the modifiable weights number increasing. The Retropropagation of Gradient method for the connections adjustment must be extended to the previous layer. Then, the correspondent equation must be generalized. But this correction requires the utilization of the derivative of the function of the second hidden layer. Now, the function Maximum not being continually derivable on its studied set, it needs being smooth by an equivalent but derivable function. As this function is necessarily contained into the  $[0,1]$  set, a good approximation consists in using the sigmoid function (shifted forward right beforehand). The problem as our opinion, is that the add of links inhibits in part the fuzzy reasoning , we tend then towards a nearly neural network. The association won't be there interesting except for the network construction.

Rather than adding connections which weigh down the network, we could modify the weights preestablished by the Fuzzy Logic. We can : • Modify the weights situated between the two hidden layers. That would consist in giving more or less importance to some fuzzy areas. In fact, that entails "filing down" to different "field of bumbs" by the Max function which data won't belonging to  $[0,1]$  set. We can suppose that the second hidden layer will tend towards already to a approximation crude of the paraboloid or rather she'll carry out a final curve adjustment estimated by the output layer weights. A draw back then appears : That's the introduction of not-linearity into the network. In fact, this weights modification will be done individually on each "bumb" and it' will have then discontinuities in level of the bumbs intersection. • Modify the input layer weights. That would consist in adjusting the coefficients  $a$  and  $b$  of the Gaussian function of the fuzzification. This correction allows then possibly another cutting out of the working interval in fuzzy areas, this which will be in part equivalent touse a scale factor. This improvement is the only one solution proposed which don't hit the fuzzy constituting of the network. • Modify either the

input layer weights and the weights situated between the two hidden layers, which would well strengthen either the fuzzy side and the neural side of our network.

We could modify too the number of fuzzy areas of the first hidden layer. It will be interesting analysis obtained results. In fact, the number of "bumps" (25) seems slight to describe all the paraboloid function by a well smooth curve (the field will always be a little granular). More the number of bumps is, more we could act on the curve. In adding under fuzzy sets, we would then reduce the error. A comparative study of these diverse improvements and possibly of their association will be profitable.

**Conclusion** We've obtained very good results (middle Level of no-recognition about  $10^{-3}$ ; middle learning error of  $10^{-5}$  -  $10^{-6}$  for a number of iteration contained between 4 and 50). The quick apprenticeship (15 minutes maximum) allows us to test different methods. The results show the fuzzy-neural association is positive not only by the reached results but also by the 3D-representation of the network.

## RECOGNITION TO CHARACTERS

### Introduction

The recognition of characters is the subject for many researches in particular in level of handwritten characters. The association neural-fuzzy seems to bring about good results for the form recognition and could then be applied in case of characters. Nevertheless this recognition is very powerful by connectionist methods although the results are less interesting with the Fuzzy Logic. In fact, the learning of classical methods of neural networks could sometime be long and the complexity of final network very big. It must be then study the interest of a so network for the recognition of characters. At first, these tests will be applied to a mono-font, mono-size recognition, to simplify the problem.

### Network's structure.

To determinate a network, it is necessary at once to do a study of that we want obtain ( $\Leftrightarrow$  outputs of network), of that we have as inputs then on the structure of network.

### network's outputs

It must be in less as so many outputs as wished classes (the different letters of the alphabet and added eventually the figures). We shall add another class of rejection for the subjects which have not been recognized. A class could be active (output=1) or passive (output=0). But one among the advantages of neural networks is that a class could be more or less active. Each outputs of network will be then contained between 0 and 1. It must be then another processing to decide the valid class when several are actives. In case of equality, we shall choose another class that we shall call confused class (the network does not know to take a decision). We can also complicate the method: IF several classes have nearby outputs ( $\Leftarrow$  a term to make explicit may-be with the Fuzzy Logic ...) to the maximal value, THEN to do another processing to differentiate the different classes. The processing which was differentiating the confused classes could be made by the device of other neuro-fuzzy networks which would function in a parallel way with the principal network. Our system will be then composed of several cells in parallel, each of them would be of simple composition and would make a specific processing. Each of networks could then have in output a boundary number of classes containing several characters. An interconnection between them in a decision's cell would allow then to select the opportune character. That would require less outputs for each networks, the cell of decision treating then the interconnection thanks to the Fuzzy Logic IF... AND... THEN....

### Network's input

In order to build the network, it is necessary to define the inputs exactly we shall enter the characters that we want recognize. That is certainly the heart of problem to the recognition because it is necessary to define: The number of inputs which will constitute our network; The perception of used characters. The difficulty comes from the nearly freedom of choice. We must research the criterion which seems to be the best adapted in problem so that the system obtain a good recognition ratio. The alone obligation comes from subsequent processing of data by the fuzzy methodology: IF ..... AND ..... THEN ..... It must be then defined inputs from which we can calculate the degree of belonging to a set. In a first time, we have decide to not use a preprocessing of data and then to work directly with the pixels of picture. In order to use the previously done work, we have wanted use networks with two inputs. It comes then of course to mind that these two inputs could express the coordinates of a pixel of the letter; Each letter will be then described by a succession of "lighted" pixels. The work is then made "in series" in the aim to use only that two inputs. The rule of inference could be: IF coordX AND coordY THEN select ion the classes XYZ. But a problem comes then: How to memorize the information. It could easily be made by the device of a counter by class which counts the times when number of that the class is active. we shall choose then the one whose the counter will be maximal ( with a eventual processing for two classes having the same probability: Cf. previous paragraph). Two inconveniences appear here: The time of processing: it is necessary to read pixel after pixel all the whole picture image (even if we treat only the "lighted" pixels); The network has only that two inputs; But there are as many "ways" as pixels constituting each letters. That would render the network very complex in spite of its two inputs. We must then: treat the picture in a parallel way; ( $\Rightarrow$  to decrease the number of ways), cut it in areas and not in pixels. ( $\Rightarrow$  to decrease the time of processing) We can then calculate the coordinates of barycenter of pixels of the letter to recognize for each areas. Each area would have two inputs corresponding to the coordinates x1 and x2 of barycenter. (The processing will then be made by pixel in each area but as the sub-networks will be parallel, the time of processing will be so decreased).

The problem is to determinate the number of necessary and sufficient cuts to differentiate the characters one from the others (eventually using fuzzy areas that is with intersections); to cut each areas in fuzzy sub-sets. The pixels of the neighborhood of outline of the picture do not mostly correspond to the character to recognize. The cut-out of the picture must then be thinner around the center of the picture. To determinate the number of divisions, it is firstly necessary to analyze the set of characters (= classes) to recognize. Another method that the we could use for the recognition of characters would consist to use a preprocessing which build an histogram of the picture. Each class will be then the recognition of a curve representing the histogram. We should not use then the Fuzzy Logic for the preprocessing but we recognize very heavily the recognition of curves done in the previous chapter. It must then as many networks as characters to recognize. We could also use these two methods in parallel. We have then in input: The barycenter of each area (= medium position); The number of lighted pixels in each area (= density). The set of these characteristics must allow to distinguish in a fine way all of the classes.

### General structure

The association of inputs and outputs gives us the next schemes when we have:

A preprocessing by histogram:

A fuzzy preprocessing.

This second method seems more complex but it is not obvious that the number of networks could be more considerable than with the first method. It does a processing by fragment of each character then that the first method studies the picture in its set. We could then envisage the processing of characters of different sizes (the position of barycenter could be relative), and of different fonts (if 9 fragments between 10 recognize 1 character, even if the tenth selection the rejection class then the character will be identified. It seems then that the second method would be more attractive. We must however make tests to verify these results and to test the neural-fuzzy association in the recognition of characters.

#### Conclusion

One of advantages of the neural-fuzzy association for the form recognition will be the decomposition in sub-classes. In fact, the fuzzy rule IF ... AND ... THEN... allows to determinate different classes thanks of characteristics: IF characteristic1 AND characteristic2 THEN Class X. We can then envisage to elaborate the neuro-fuzzy network in a rest of classes of more and more specialized. This method have the advantages to obtain a series of simples networks, to know approximately the localization of the knowledge ... Even if we smooth and add connections in each of sub-networks.

#### CONCLUSION

The two advantages of neuro-fuzzy network were: The simplification of the learning which is progressively established t layer after layer; The approximate localization of the knowledge. As it appears that the best method of learning is a method which adapt itself perfectly to the connectionist networks (without Fuzzy Logic). The advantage in this field does not seem evident so many more, the learning in the neural networks is without big problems. In an other part, the knowledge is "deflected" at the time of adding new others connections. We don't know more then where she is situated. Two obvious advantages would be: The decomposition in sub-classes (for the recognition of character); The not-complexity of network (  $\Rightarrow$  more memory, bigger quickness,...) This study could bring about an interest in front of the use of a such network to the recognition of handwritten characters.

#### BIBLIOGRAPHY

Mr GLORENNEC. *Les réseaux neuro-flous évolutifs : un pont entre le flou et le neuronal*, GR CSN,CNRS 92 France

Mr LAMOTTE. *Le flou : quelques définitions et applications*, Logique floue et réseaux de neurones, école d'hiver de St Sorlin d'Arves.1993, *Apprentissage et réseaux de neurones* Pour la Science n°181 (Nov. 92)

Melle LEGENDRE *Reconstitution de mesurandes à l'aide d'un réseau de neurones multicouche, appliquée à un système de mesure de pression à fibres optiques*, Rapport de DESS AII de (Sept. 92)

Melle PAKKER, *Reconnaissance des caractères alphanumériques par une analyse structurale hétérarchique*, Thèse de (Mars 85)

Mme BELLON & Mr BOSCH & Mr PRADE , *Le boum du flou au Japon* , GR CSN,CNRS 92 France

Mr DUBOIS , *Typologie et utilisation de règles floues* , GR CSN,CNRS 92 France

Mr GLORENNEC & Mr BARRET & Mr BRUNET , *Application of neuro-fuzzy networks to identification and control of nonlinear dynamical systems*, La lettre du club logique floue n°2 (Fév. 92)

Mr BERENJI & Mr KHEDKAR *Fuzzy rules for guiding reinforcement learning*, La lettre du club logique floue n°2 (Fév. 92)

N Wakami, S Araki, H Nomura, *Recent Applications of Fuzzy Logic to Home Appliances*, IEEE - 93 pp155-160

D.T. Pham, D. Karaboga, *Design of Neuromorphic Fuzzy Controllers*, IEEE International Conference on System and Cybernetics 1993 pp103-107

P Freidel, *Allier techniques neuronales et conception floue*, Electronique #34, 1-94, pp73-76