Fuzzy Boolean Networks Learning Behaviour

José Alberto Tomé
INESC-id/IST
josé.tome@inesc-id.pt

João Paulo Carvalho
INESC-id.pt/IST
joao.carvalho@inesc-id.pt

Abstract

In this paper one studies the Learning behaviour of an entire Rule Base in Fuzzy Boolean Networks. It is analyzed the influence of a set of factors such as number of inputs per neuron, granularity of antecedent spaces and number of teaching experiments on learning effectiveness without cross influence between rules and on interpolation capabilities of the network. Both one dimensional problems and two dimensional problems are tested and results interpreted using theoretical results also presented.

1. Introduction

Neural network s are known to easily apprehend, through experimental learning, the relations between systemic variables. On the other side Fuzzy Logic has good capabilities to explain system behaviour through a set of qualitative rules. A benefic cooperation may be achieved when fuzzification of the neural net components takes place,[1]-[5]. Another kind of synergetic cooperation may also appear under the form of Neural Fuzzy Systems, when the fuzzy inference system is improved with learning capabilities through the use of neural nets as system components [6]-[10]. In both cases, however, the human intervention on the conceptual models is evident and neither the topology of real world neural systems, which is mainly random at microscopic level [11], nor some of their emergent properties are present on these hybrid systems. However, these type of topology and emergent properties appear on Fuzzy Boolean Networks or FBN's [12]-[14], which architecture and behaviour has been inspired on natural reasoning systems.

FBN's, as animal brains, operate on a great number of neurons and on a greater number of connections between them. Moreover those neurons are organized into so called cards and each card is associated with a specific concept or variable. Neurons are Boolean in the sense that their outputs belong to the set [0, 1] (similarly to the fire do not fire real neurons). Other Boolean neural nets have been studied for some time [15] but with different objectives such as applications in pattern recognition problems and logical synthesis.

On FBN's the "value" of a concept is given by the percentage of activated neurons on the card associated with that concept which is called *activation ratio*. It has been proved [16] that these FBN´s are capable of learning and reasoning qualitatively, as natural systems, on a similar way to the qualitative Fuzzy reasoning using If Then rules. The internal architecture of such nets is based in meshes of connections linking randomly chosen neuron outputs from antecedent cards to neuron inputs of consequent cards. In addition, each neuron has a remembrance capability, given by a set of binary memories (in hardware model, flip-flops). Actually, there is one such binary memory per possible antecedent rule; if there are A antecedent variables with a granularity of k each (defined as the number of linguistic terms of each variable) then each neuron will have a possible maximum of k^A such binary memories. During the learning phase (during which, activation of both antecedents and consequent are externally imposed), for each consequent neuron of such a net, these flip-flops can be set or reset or they can maintain their previous state depending on the activation of the antecedent binary values of the neuron inputs and of the consequent neuron. Each experiment allows each one of the neurons to update just one of the qualitative rules.

2. FBN Architecture

A brief description of the network architecture and the deduction of its reasoning and learning characteristics follows. The model assumes that each consequent neuron is an N.m input neuron, where N is the number of antecedents and m the number of inputs

This work was supported in part by the Portuguese Foundation for Science and Technology under project POSI/SRI/47188/2002 and by European Commission Grant IMPROVE (IST-2003-004785).

0-7695-2976-3/07 $25.00 © 2007 IEEE
DOI 10.1109/ISDA.2007.72

coming from the same antecedent area. Each input I_{kn} (k=1,N;

n=1,m) is connected to a randomly chosen neuron output from antecedent area k. As an interpretation, one may say that each consequent neuron "observes" each of the antecedent areas through a sample of m binary values. Moreover, each of these samples is taken as a simple count on the number of activated inputs.

Consider dij the detection of i_j, that is the Boolean function which takes value "1" if and only if there are i activated inputs coming from antecedent j. Similarly, $d(i_1, ..., i_N)$ is considered the joint detection of i_1 activated inputs from antecedent 1,..., and i_N activated inputs from antecedent N. In a hardware model, each single neuron is designed in order to implement the following Boolean function:

$$\underset{i1=0}{\overset{m}{\mathrm{OR}}} \ldots \underset{iN=0}{\overset{m}{\mathrm{OR}}}\ d(i_1,...,i_N)\ \mathrm{AND}\ \ ff(i_1,...,i_N),$$

with $d(i_1,...,i_N) = \underset{j=1}{\overset{N}{\mathrm{AND}}}\, d_{ij}$

The term ff($i_1,...,i_N$) represents the memory of the neuron regarding that particular input count configuration and it will be established during a training phase. Each one of these terms (ff) is associated with a flip-flop in a possible hardware neuron implementation.

Define *activation ratio* of any area as the ratio of activated neurons and the total number of neurons of that area. This is the same as the probability of a randomly chosen neuron in that area to be activated. Let p_j represent the activation ratio of antecedent area j and pr(i1,..,iN) the probability of ff(i1,..,iN) to be activated. Then, since for a randomly chosen neuron, ν, one and only one of the d(i1,..,iN) is activated, it follows that the activation ratio of the consequent area of ν becomes:

$$\sum_{i1=0}^{m} .. \sum_{iN=0}^{m} \prod_{j=1}^{N} \binom{m}{k_{ij}}\, p_j^{\,k_{ij}} .(1-p_j)^{m-k_{ij}} .\mathrm{pr}(i1,..,iN)$$

or simply:

$$\sum_{i1=0}^{m} .. \sum_{iN=0}^{m} \mathrm{P^a r}(i1,..iN).\mathrm{pr}(i1,..,iN) \qquad (1)$$

Using the algebraic product and the bounded sum for t-norm and t-conorm respectively[17], it follows that microscopic neural operations as defined above emerge, at the macroscopic or network level, as fuzzy qualitative reasoning. To this purpose the equations above may be interpreted as follows:

Input variables, the activation ratios pj, are fuzzified through binomial membership functions of the form $\binom{m}{kij} pi^{kij}.(1-pi)^{m-kij}$. The evaluation of the expression for a given p_j represents the membership degree of p_j in that fuzzy set.

The product of the terms, the $\prod_{j=1}^{N} \binom{m}{k_{ij}}\, p_j^{\,k_{ij}} .(1-p_j)^{m-k_{ij}} .\mathrm{pr}(i1,..,iN)$, represents the fuzzy intersection of the antecedents (i=1,N), by definition of the above t-norm. Considering the consequent fuzzy sets as singletons (amplitude "1") at the consequent UD values pr(i1,..,iN), it follows that the equations represent the defuzzification by the Center of Area method.

One may conclude that the network implements a set of production rules of the type:

IF A is A1 AND B is B1 AND… …. THEN C is C1

where A and B are Antecedent variables , C is the Consequent variable, A1, B1,… are linguistic terms of fuzzy sets defined by the count samples on m inputs and C1 is a fuzzy set (singleton) at the consequent defined by the probability of the flip-flop ff($\mathbf{I_1}$) to be at "1". This probability is set during the learning process, being $\mathbf{I_1}$ the N element vector of the above counts.

3. Learning

The setting of logical values at the neuron flip flops establishes the learning phase of the network. Macroscopically, this turns to be the setting of the pr(k1,...,kN) probabilities of the internal flip-flops in (1). During this learning phase the network is activated (both in antecedent and consequent areas) by a collection of experiments and for each experiment a particular input configuration is presented to each consequent neuron. This configuration addresses one and only one internal flip-flop of each neuron. Updating of each flip-flop value depends on its selection (or not) and on the logic value of the consequent neurone. This may be considered an Hebbian type of learning [16] if pre and post-synaptic activities are, in the present model, given by the activation ratios: p_j for antecedent area j and p_{out} for the consequent area. For each neuron, the m+1 different counts are the meaningful parameters to take into account for pre synaptic activity of one antecedent. Thus, in a given experiment, the correlation between posterior synapse activity (p_{out}) and pre synaptic activity -the probability of a given $d(i_1, ..., i_N)$ to be activated- can be represented by the probability of the different flip-flops to be activated. In practical terms, for each teaching experiment and for each consequent neuron, the state of flip-flop ff($i_1,\ldots,i_N$) is determined by, and only by, the Boolean values of decoder output d(i_1, ..., i_N) and of the output neuron state considered.

Here, one is considering the interesting case when non-selected flip-flops maintain their state and selected flip-flops take the value of consequent neuron, which corresponds to updating equation (2), where P^a is the probability of activating, in the experiment, the decoder output associated with p:

$$p(t+1)-p(t) = P^a.(p_{out} - p(t)) \qquad (2)$$

It is quite easy to see that the network converges to the taught rule, if every experiment teaches the same rule, say Pout as the consequent activation ratio for the given antecedent P^a [13].

To deal with a complete Rule Base consider then a sequence of experiments, each one teaching a different rule. In such a case it is necessary to consider not only P^a, but the probabilities P_k^j , of activating any generic rule k antecedent part (on a neuron, each corresponds to different flip-flops), when teaching rule j with consequent $P_{out}j$ on a given experiment. Learning rule i, on time step t, when rule t is being taught, is given by the activation probability of the corresponding internal flip-flops: $p_i(t+1)= p_i(t) + P_i^t (P_{out}t- p_i(t))$. This is known as the flywheel equation and its solution, [17], is well known when P_i^t is constant with t. The solution for $p_i(t)$ is $P_{out}t$, if $1/ P_i^t$ is large. However, since P_i^t varies with t, this can not be applied directly and a study must be done on the influence of varying P_i^t and $P_{out}t$ with time. Suppose a finite set of R+1 consequent singleton positions on the consequent UD, and assume an equal number of rules.

Consider this set of R+1 rules and assume (for commodity) that any rule k, where $k \in \{0,1,2,..,R\}$, is taught at time steps k, k+R+1, …, k+r.(R+1). Focusing on the learning of rule i one obtains, by the successive application of the equation above, and after a complete cycle of teaching each rule once:

$p_i(i+R+1) = p_i(i).(1-P_i^i) + \alpha . (\mathbf{P_{out}j}^T - \mathbf{p_i(t)}^T) . (1-P_i^i) + P_i^i P_{out}i$

where:

$\mathbf{p_i(t)} = [p_i(i)\ \ p_i(i)\ \ p_i(i)\ \ldots\ p_i(i)]$

$\alpha = [(P_i^{i+1})\ \ (P_i^{i+2})\ldots(P_i^{i-1})\ldots-(P_i^{i+1}.P_i^{i+2})\ldots(-1)^R(P_i^{i+1}.P_i^{i+2}.\ldots.P_i^{i-1})]$

$\mathbf{P_{out}j} = [P_{out}i+1\ \ \ P_{out}i+2\ \ \ldots\ P_{out}i-1]$

If the learning of any generic rule **i** is efficient, this means that $p_i(i+r.(R+1))/P_{out}i$ approximates 1 with r (meaning that p_i equals what is being taught, without interference of other rules).From this one obtains:

$p_i(i+R+1)/P_{out}i = p_i(i) . (1-\Sigma\alpha). (1-P_i^i) /P_{out}i + \alpha . \mathbf{P_{out}j}^T .(1-P_i^i) / P_{out}i + P_i^i$

where $\Sigma\alpha$ means the sum of every element of vector α. Since $(1-\Sigma\alpha). (1-P_i^i) <1$ there is a limit point for $p_i(i+R+1)/P_{out}i = p_i(i)/P_{out}i$, giving the solution, $p_i(t)$, when enough time steps have been passed:

$p_i(t)= \alpha . \mathbf{P_{out}j}^T . (1-P_i^i)/(P_i^i + \Sigma\alpha - \Sigma\alpha.P_i^i) + P_i^i . P_{out}i / (P_i^i + \Sigma\alpha - \Sigma\alpha.P_i^i)$

Considering a worst case where every rule distinct from i is taught the same consequent Pj, expression above becomes:

$p_i(t)= Pj.\Sigma\alpha . (1-P_i^i)/(P_i^i + \Sigma\alpha - \Sigma\alpha.P_i^i) + P_i^i . P_{out}i / (P_i^i + \Sigma\alpha - \Sigma\alpha.P_i^i)$

In order to obtain $p_i(t)= P_{out}i$ condition (3) is obtained.

$$P_i^i \gg \Sigma\alpha . (1-P_i^i) \qquad (3)$$

Consider the case of one antecedent. Taking, generically, P_i^i as the sum of the binomials around i (meaning that antecedent rule i actually comprise counts from i-k/2 to i+k/2, for a given k) one can use the relation between binomials and Gaussian distribution [18]:

$$P(a <= x <= b) = \Phi(\beta) - \Phi(\alpha) = \sum_{x=a}^{b} \binom{m}{x} p^x . q^{(m-x)}$$

where α=(a-m.p-0.5)/Sqrt(m.p.q)

and β=(b-m.p+0.5)/Sqrt(m.p.q)

In our case α=(i-j-0.5)/Sqrt(j.(1-j/m)), since a and b equal i, p=j/m and q=1-p. P_i^i becomes then:

$$-\frac{1}{\sqrt{2\pi}} \left(\sum_{j=0}^{k} \int_{(i-(i-k/2+j)-0.5)/\sqrt{(i-k/2+j).(1-(i-k/2+j)/m)}}^{(i-(i-k/2+j)+0.5)/\sqrt{(i-k/2+j).(1-(i-k/2+j)/m)}} e^{-\left(x^2/2\right)} dx \right)$$

Making the simplifying consideration that the Gaussian variances of the parcels are equal (which is not exactly true but approximately), and noting that the numerator of the lower limit of an integral parcel is exactly the upper limit of the next integral parcel, one may consider, instead of the above sum, a single integral between two limits, as follows:

$$-\left(1/\sqrt{2\pi}\right) \int_{(k/2-0.5)/\sqrt{(i-k/2)(1-(i-k/2)/m)}}^{(-k/2+0.5)/\sqrt{(i+k/2)(1-(i+k/2)/m)}} e^{-\left(x^2/2\right)} dx$$

or:

$$\left(1/2\right) \left| erf\left(x/\sqrt{2}\right) \right|_{(-k/2+0.5)/\sqrt{(i+k/2)(1-(i+k/2)/m)}}^{(k/2-0.5)/\sqrt{(i-k/2)(1-(i-k/2)/m)}}$$

Taking the worst case, when i=m/2, one gets:

$$erf\left|\frac{k-1}{\sqrt{m-\frac{k^2}{m}}}\right| = erf\ |y1|$$

A similar deduction can be made for the $\Sigma\alpha$. In this case, and as i=m/2, one gets two identical terms, one giving the contributions from every count j from 0 to m/2-k/2-1, the other from m/2+k/2+1 to m :

$$\left.\frac{m-k-1}{\sqrt{k.(2-k/m)}}\right| - erf\left|\frac{k+1}{\sqrt{(m-k^2/m)}}\right| = erf\,|y2| - erf\,|y3|$$

Finally, expression (3) implies that:

$$\frac{erf|y1|}{(1-erf|y1|).(erf|y2|-erf|y3|)} -> \infty \quad \text{with m.}$$

From erf properties one obtains from (3) the expression (4) tending to infinity for any k equal to a power of m greater then ½. Table I exhibits the values of erf(y1), erf(y2), erf(y3) and the limit of the expression (4) which should tend be much greater than 1, for an effective rule separation.

$$expr=P_i^i\ /(\Sigma\alpha\ .\ (1\text{-}P_i^i\ )) \qquad (4)$$

One concludes that $k=m^{1/2}$ defines the limit granularity under which there is no effective rule separation and above which learning of any set of rules is effective.

TABLE 1

ERF FUNCTIONS AND LIMIT

	k=j	$k=m^{1/2}$	$k=m^{1/3}$	$k=m^{2/3}$	k=m/j
erf(y1)	0	0.52	0	1	1
erf(y2)	1	1	1	1	1
erf(y3)	0	0.52	0	1	1
expr	0	2.6	0	∞	∞

For an A-Dimensional problem (A antecedents) (4) takes the same form, but P_i^i will be the product of the sums of binomials, each one around a given i (in the limit, a A-dimensional Gaussian, when m->∞). Similarly for $\Sigma\alpha$. Expression (4) will take the same limits as above, since the power of A will not affect the arguments of function f (0, 1 or ∞). An adequate granularity of the input space for learning any A-dimensional (A antecedents) Rule Base without cross influence is an aggregation of k^A different counts for each joint antecedent membership function, with k being a power of m between ½ and 1

4. A Study on FBN Error

When studying the error of FBN's one has to take into account two factors to understand its behaviour. The first is the condition to effectively learn the Rule Base without cross influence, which conducted to the above relation $k=m^x$ with x>1/2. The second is the capability of the Net to interpolate correctly, that is, to give right consequent values for non taught antecedent values. One may interpret the consequent space as being built of known taught samples placed on consequent points driven by the antecedent rules, thus having a number of samples equal to the number of rules. For the one antecedent case and inspired on the sampling theorem, one should expect the number of linguistic terms of the antecedent (also designated by NR and corresponding to the samples that reconstruct the output form) to be at least 2*f (where f is the maximum consequent frequency considering the output domain as the unit time). Actually these samples should be equally spaced, a condition not guaranteed but taken as approximated. These two conditions put different limits for k: an upper limit from sampling theorem, since k=m/NR, and a lower limit from $k=m^x$.

In figure 1 plots of k=m/2, k=m/6, k=m/10 and k=SQRT(m) are shown, being the last one with a dashed style. The first three functions give the sampling upper limit for k according the sampling theorem, respectively for f=1, f=3 and f=5; being the last one the theoretical lower limit of k to prevent cross learning in any circumstance.

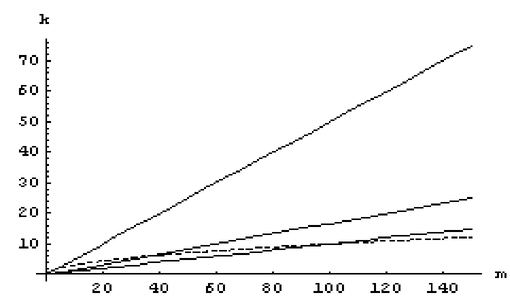

Fig 1. Upper and Lower limits for k, function of m.

It is seen that both conditions are obeyed for f=1 with virtually every m (m>4), but only for m>36 (with f=3) and m>100 (with f=5). On practical terms, the condition resulting from $k=m^x$ is weaker because it has been calculated on the worst possible conditions, which are: every rule different from the one being taught (Rt) has the same consequent value and that value is as far as possible from the one being taught to rule Rt. In order to evaluate the FBN capabilities to effectively learn and thus adequately separate the rules and to interpolate one generated artificial parameterized functions and taught the Net with random points. On a further stage, inputs (parameter values of the functions) are applied to the Net and the results are used to calculate error. To test the 1-antecedent case one used the function f= (cos 2Πx+1.2)/2.4, both for x belonging to the interval (0,1) and (0,3). The interesting parameters, from which the net behaviour depends, are the number of teaching experiments, the number of inputs per consequent neuron and per antecedent area (m) and the number of rules (NR), where NR=m/k being k the number of adjacent counts that are aggregated to form one antecedent linguistic term. This NR, and also k, is a form to refer to the granularity of antecedent space. In figure 2 it is shown the net output and the real values of f for m= 80, Number of Rules = 40 and using 512 different teaching experiments, with x on the range (0,1) .It is also shown on figure 3 the net output and actual values for m=144, NR=48, using 250 teaching experiments and with x on the range (0,3) The mean squared error and absolute error for these parameter values have been: MSE= 0.000597, AE= 0.0315 for the first case and MSE=0,0058 and AE=0,094 for the second case. The definitions of MSE and AE are: MSE=Σ(True Value-FBN Output)^2/(Nº of points*2) and AE = ΣABS(True Value-FBN Output)/ (Nº of points).

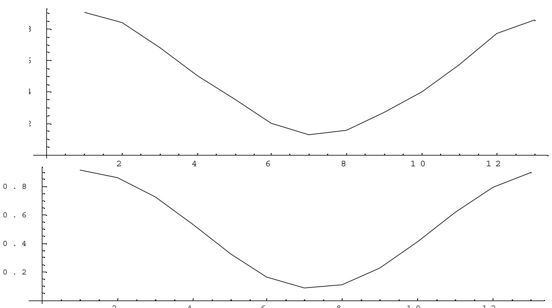

Fig.2 Net output and taught values – m=80, NR=40, 512 experiments and x on range (0,1).

In order to have a better understanding of the overall behaviour of the error with the net parameters one has obtained graphs of the error (MSE) as function of the number of teaching experiments for m and Number of Rules (or Granularity on antecedent space) as parameters, error as function of the granularity with m

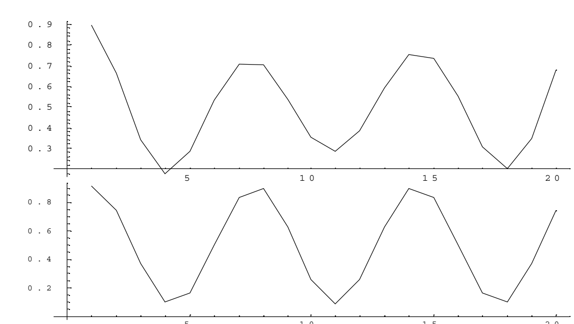

Fig 3. Net output and taught values – m=144, NR=48, 250 experiments and x on range (0,3).

and number of experiments as parameters and error function of m for the best achieved result on the Number of Rules space for different values of number of teaching experiments.

Following, in figure 4 is an example of MSE as function of the number of experiments. One used m=99 and Number of Rules=10 (approximately $m^{1/2}$). The numbers of experiments used for teaching – x axis-were 25, 63,100,250, 512 and 1000.

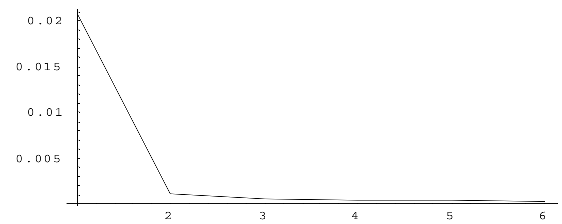

Fig. 4 MSE versus number of experiments – m=99, NR=10

This behaviour is typical: the error greatly diminishes with the number of experiments. The error is substantially higher for small values of NR, indicating that in order to capture the consequent function more rules (samples) are necessary.

The evolution of the MSE with NR may also be seen in figures 5 and 6. The examples with m=144, 250 experiments with NR=2,4,6,8,9,12,16,24,36,48,72,144 are displayed for f=1 and f=3.

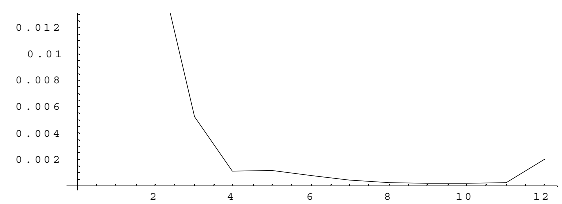

Fig.5. MSE function of NR – m=144 and f=1.

The minimum for the MSE are 0.0001840 and 0.00577 respectively and both obtained for NR=48 (k=3). Again the high values for error when NR is low are due to the inability of the net to capture the output

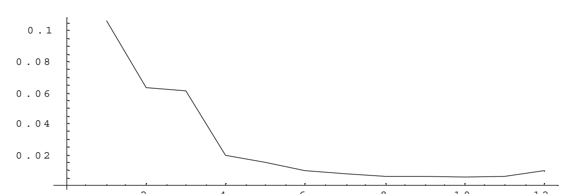

Fig. 6 MSE function of NR – m=144 and f=3.

function dynamics (averaging or low pass effect). On the other extreme NR is too high, that is k is very far from $m^{1/2}$.

In figure 7 it is shown the MSE function of m for the best achieved value among the different NR. The number of experiments on that graph is 512 and the values of m on x axis were 15,24,49,63, 80, 99 and 144. As expected the error diminishes almost systematically with m, and similar figures are obtained with other number of teaching experiments.

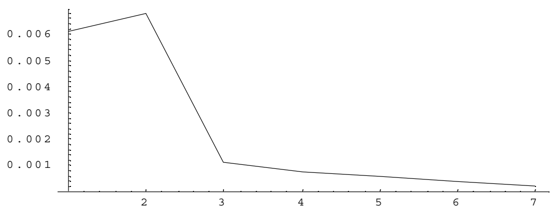

Fig 7. MSE in function of m for best NR.

One has also used the FBN network to learn a two dimension problem, a function of the form f = (Abs[Cos[x] Sin[y]]+1)/2.2. As an example the figure 8 illustrates the true values and those given by the FBN after teaching with NR= 25 and m= 224.

On that figure one has used a single dimension (for each x value the different y values are consecutively placed). One has obtained for this example MSE = 0.000342409 and AE = 0.0213149.

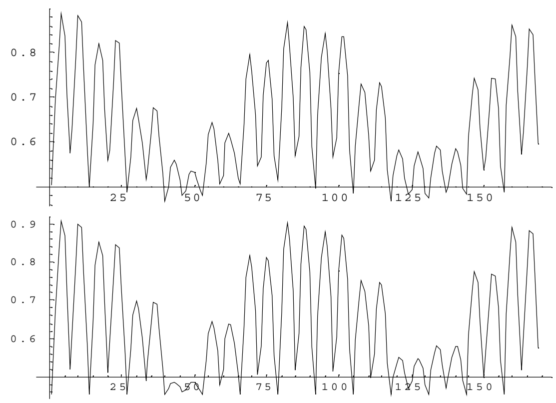

Fig 8. Net output and taught values for 2-dimension problem – NR=25, m=224

5. Conclusions

The learning and interpolating (thus generalization) capabilities of Fuzzy Boolean Networks have been investigated in this paper with the purpose of finding the influence of the net parameters on their effectiveness.

It has been found that the granularity of antecedent spaces is related with the number of inputs per neuron by two different and in some way contradictory relations. Some tests have been carried out and optimum values for those parameters, in terms of minimum error, have been found. The influence of the number of teaching experiments has also been investigated and curves giving the consequent error established. One concludes that this study establishes the basis for parameter selection for the FBN optimization error.

6. References

[1] GUPTA, M. and QI (1991). "On fuzzy neurone models" *Proc. Int. Joint Conf. Neural Networks*, volII, 431-436, Seattle.

[2] HAYASHI, Y., E. CZOGALA and J.BUCKLEY (1992). "Fuzzy neural controller" *Proc. IEEE Int. Conf. Fuzzy Systems*, 197-202,San Diego.

[3] KELLER,J.M. and D.J.HUNT(1985). "Incorporating fuzzy membership functions into the perceptron algorithm." *IEEE Transactions Pattern Anal.Mach.Intell.*. PAMI-7(6):693-699.

[4] LIN, CHIN-TEN and LEE,C.S.(1996). *A Neuro-Fuzzy Synergism to Intell. Systems*.New Jersey : Prentice Hall.

[5] PEDRYCZ,W. (1992). "Fuzzy neural networks with reference neurones as pattern classifiers" *IEEE Trans.Neural Networks* 3(5):770-775.

[6] HORIKAWA, S., T. FURUHASHI and Y. UCHIKAWA (1992). "On fuzzy modelling using fuzzy neural networks with the back propagation algorithm." *IEEE Transactions Neural Networks* 3(5): 801-806.

[7] KELLER,J. , R. KRISHNAPURAM and F. RHEE (1992). "Evidence aggregation networks for fuzzy logic interface." *IEEE Trans. Neural Networks* 3(5): 761-769.

[8] LIN, CHIN-TEN and LEE,C.S.(1996). *A Neuro-Fuzzy Synergism to Intell. Systems*.New Jersey : Prentice Hall.

[9] TAKAGI,H. and HAYASHI, (1991). "NN-driven fuzzy reasoning." I*nt. J. Approx.e Reasoning* 5(3):191-212.

[10] YAGER,R. (1992). "OWA neurones: A new class of fuzzy neurones". *Proc. Int. Joint Conference Neural Networks*, vol I, 226-231, Baltimore.

[11] EDELMAN,G.M. (1992) *Bright Air,Brilliant Fire: on the Matter of the Mind*. Basic Books.*on the Matter of the Mind*. Basic Books.

[12] TOMÉ,J.A. (1993,a)."A simple Neural Model for Fuzzy Reasoning."*Proc. IEEE Int.Conf. Fuzzy Systems*,San Fr.

[13] TOMÉ,J.A. (1993,b)."Learning in Binary Neural Nets which have Fuzzy Reasoning." *Proc. Int. Joint Conf. Neural networks*,Nagoya

[14] TOMÉ,J.A. (1994)."A connectionist model for recurrent fuzzy reasoning."*Proc. Int. Conf. Neural Information Processing*, Seul.

[15] ALEKSANDER, I. (1971). *Microcircuit Learning Computers* Mill & Boon Ltd.

[16] TOMÉ,J.A. (1993,b)."Learning in Binary Neural Nets which have Fuzzy Reasoning." *Proc. Int. Joint Conf. Neural networks*,Nagoya

[17] LIN, CHIN-TEN and LEE,C.S.(1996). *A Neuro-Fuzzy Synergism to Intell. Systems*.New Jersey : Prentice Hall.

[18] KREYSZIG,E.(1979). *Advanced Engineering Mathematics*. John Wiley & Sons.