Learning in Fuzzy Boolean Networks – Rule Distinguishing Power

José A.B. Tomé

INESC,
Rua Alves Redol nº9, 1000 Lisboa Portugal
jose.tome@inesc.pt

Abstract. Fuzzy Boolean Networks are Boolean networks with nature like characteristics, such as organization of neurons on cards or areas, random individual connections, structured meshes of links between cards. They also share with natural systems some interesting properties: relative noise immunity, capability of approximate reasoning and learning from sets of experiments. An overview of the processes involved in reasoning supported on an hardware architecture are presented, as well as how Hebbian-Grossberg learning can be achieved. An interesting problem related with these nets is the number of different rules that they are able to capture from experiments without cross interferences, that is, their rule capacity. This work establishes a lower bound for this number, proving that it depends on the number of inputs per consequent neurons and its relation to consequent granularity. An application for traffic problems is also provided.

1 Introduction

The known capabilities of Fuzzy Systems to explain system behavior through qualitative rules may strongly benefit the global performance of neural networks, adding new capabilities to their classical possibilities of experimental learning variable relations. This can be achieved by fuzzification of the neural net components [2,3,6,7,8]. Another form of cooperation between the two paradigms is to build a Fuzzy System with components that are neural nets. To the usual inference and explanation properties of the Fuzzy System are thus added the learning capabilities of neural networks [5,7,11]. This synergy needs not to be achieved through this kind of "adding components" from the two paradigms. Instead they may be embedded together in a common structure, as is the case of author's work presented elsewhere [9, 10]. In that work it is presented a Boolean neural network where variable (or concept) values are represented by the activation level of neurons on associated neural areas. These networks present other interesting similarities with natural neural systems, including an intrinsic immunity to noise in individual neurons or connections (since the density of neural activation is not disturbed by individual errors), fire/do not fire individual neuron operations, random

G. Bittencourt and G. Ramalho (Eds.): SBIA 2002, LNAI 2507, pp. 270-279, 2002.

connections between neurons and structured macro connections between neural areas or cards. An emergent property of such networks is their fuzzy reasoning capabilities (qualitative rules implementation), despite none fuzzy concept was placed at the micro (neural) level. Moreover, the model is also an Universal Approximator [10]. These structures are also capable to automatically adapt to the granularity of the input variables and to decide about the relevance of input variables to a given problem. Concerning their learning capabilities, the Fuzzy Boolean Nets can learn from sets of experiments in a non-supervised way, using elements of binary memory embedded on the neuron internal structure.

An interesting question related with such networks is their capacity in memorizing different rules (fuzzy rules) , or the achievable granularity of antecedent and consequent variables. In this work it is shown how such questions may be answered and a lower bound for the consequent granularity (and thus for the rule capacity) is established.

2 Architecture

The complete description of the network architecture and the deduction of its reasoning and learning characteristics may be found in [9], but a brief résumé follows.

Neurons are aggregated in areas (one area per variable) and connections are established between the outputs of antecedent neurons (those on antecedent areas) to the inputs of the neurons on the consequent area. The model here considered postulates that each consequent neuron is an N.m input neuron, where N is the number of antecedents and m the number of inputs 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 can 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, since there is no reason to differentiate between two different samples with the same number of activated binary variables (the same number of "ones").

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. Each single neuron is designed in order to implement the following Boolean function:

$$\underset{i1=0}{\overset{m}{OR}} \ldots \underset{iN=0}{\overset{m}{OR}} d(i_1,\ldots,i_N) \;\; \text{AND} \;\; ff(i_1,\ldots,i_N), \quad \text{with} \quad d(i_1,\ldots,i_N) = \underset{j=1}{\overset{N}{AND}} d_{ij}$$

The term $ff(i_1,\ldots,i_N)$ 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. A possible hardware implementation is given in fig. 1.

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:

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

or simply:

$$\sum_{i1=0}^{m}..\sum_{iN=0}^{m} Pa_r(i1,..iN).pr(i1,..,iN) \qquad (1)$$

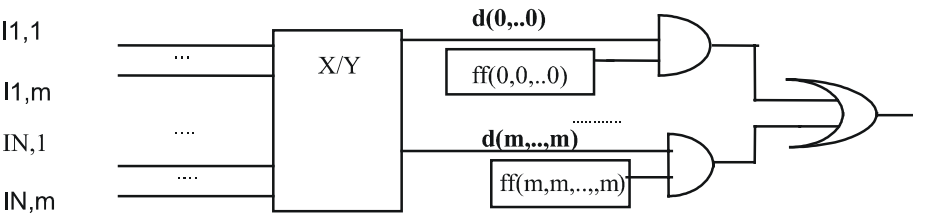

Fig. 1. Neuron architecture

Using the algebraic product and the bounded sum for t-norm and t-conorm respectively, 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 $\prod_{j=1}^{N}\binom{m}{kij}\ p_j^{kij}.(1-p_j)^{m-kij}.pr(i1,..,iN)$, 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

It is the setting of logical values at the neuron flip flops that establishes the learning phase of the network. Macroscopically, this turns to be the setting of the pr(k1,...,kN) probabilities in the above expression, (1), of the internal flip-flops. 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 [4] 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,...,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.

Considering then the pr(k1,...,kN),in expression (1), as the synaptic strengths, one may have different learning types, depending on how they are updated (for simplicity the indexes are omitted in what follows). 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 a kind of Grossberg based learning. It corresponds to the following updating equation (where indexes are not represented and p is used in place of pr, for simplicity,), and where P^a is the probability of activating, in the experiment, the decoder output associated with p:

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

First, 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.

Considering any initial p different from P_{out}, it will converge to P_{out} with experiments teaching the same rule (that is, with P_{out} as the consequent activation ratios and

the same P^a). To prove this, take p(t+2)-p(t+1) and consider a consequent activation ratio of P_{out} for all experiments:

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

It is a simple matter to verify that:

$$|p(t+2)-p(t+1)| < |p(t+1)-p(t)| \text{ for any t}$$

Thus it may be concluded that with a set of coherent experiments -teaching the same rule- the net converges. It will establish the p with the same value as P_{out}, and in each experiment it will approach that value P_{out} proportionally to the distance between the present value of p and P_{out} itself, that is, with approaching zero decreasing steps.

A more difficult problem is the study of convergence in such a network, if a set of different rules is taught, not just one. And what about the number of different rules it can accommodate without interference, that is, cross learning? Obviously such a problem is equivalent to the one of studying the granularity of variables, in particular the consequent, since antecedents are limited by m.

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, [1], 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. This implies that a single rule is considered for each consequent fuzzy set. This is not a restriction, since it will be proved in what follows that, if there is a number of different antecedent rules with the same consequent, the system learns properly, as was the case for a single rule for each consequent singleton.

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:

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

where:

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

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

$\mathbf{P_{out}j}$ =[$P_{out}i+1$ $P_{out}i+2$ … $P_{out}i-1$] and the dimension of these vectors being $R+\binom{R}{2}+\binom{R}{3}+...+\binom{R}{R}$.

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). Dividing both members of the equation by $P_{out}i$ one obtains a linear equation relating the efficiency of the rule i before and after a complete cycle of teaching every rule:

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

and where $\Sigma\alpha$ means the sum of every element of vector $\boldsymbol{\alpha}$. 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:

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

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 the case of every different antecedent rule having the same consequent ($Pj= P_{out}i$) the expression above just reduces to $p_i(t)= P_{out}i$, and the system learns as expected. For the more general case of different rules with different consequents it is easily seen that the condition for $p_i(t)=P_{out}i$ (first parcel of second member tends to zero and second parcel to $P_{out}i$) is:

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

Considering the behavior when m is large, the P_i^i and P_i^j are multidimensional Gaussians, calculated on the point defined by rule i antecedent. If generic rule i has A antecedents, its antecedent part can be characterized by an A-dimensional vector **i**=(i1,i2,…,iA). Then it follows:

$$P_i^i = N(i1, \sqrt{i1.(1-i1/m)}\,) \,.\, N(i1, \sqrt{i2.(1-i2/m)}\,) \ldots N(i1, \sqrt{iA.(1-iA/m)}\,)$$

$$= \prod_{k=i1}^{iA} N(k, \sqrt{k.(1-k/m)}$$

$$\Sigma\alpha = \sum_{p\in R'} (\prod_{q=1}^{A} N(k_p^q, \sqrt{k_p^q.(1-k_p^q/m)}\,)\,) + \text{“higher order terms”} \,,$$

and where R’ is the set of rules with exception of rule i and each of them being defined by vector $\mathbf{k_p} = (k_p^1, k_p^2, \ldots, k_p^A)$.

Since the “higher order terms” give a negative contribution to the sum one may discard them, when condition (3) is investigated. Using a polynomial approximation [12] for the Gaussian distributions one obtains:

$$P_i^i = Z(0)^A \cdot \prod_{i1}^{iA} \left(\frac{i.(m-i)}{m}\right)^{-1/2} = 0.401^A \cdot \prod_{i1}^{iA} \left(\frac{i.(m-i)}{m}\right)^{-1/2}$$

$$\Sigma\alpha = \sum_{p\in R'} \left(\prod_{q=1}^{A} (k_p^q.(m-k_p^q)/m)^{-1/2} \cdot \left\langle POL6\left[k_p^q.(m-k_p^q)/m)^{-1/2}.(i_q - k_p^q)\right]\right\rangle^{-1} \right),$$

with POL6 representing a 6 degree polynomial. If one is interested on the capacity of the system to learn different rules, it is necessary and sufficient to prove that any two consecutive antecedent parts are able to distinguish between any two consequents. This worst case is when $(i_q - k_p^q)$ represents the difference between two consecutive counts (antecedent parts), that is m/R, where R+1 is the total number of rules. In order to satisfy inequality (3) it is sufficient guarantee the relation between the minimum value for the first member and the maximum for the second. As $\left(\frac{i.(m-i)}{m}\right)^{-1/2}$ has a minimum for i=m/2, with value $2/m^{1/2}$, the minimum for the first member is $0.401^A.(2/m^{1/2})^A$. The second member is R. $(\Upsilon.\Upsilon^{-6}.(m/R)^{-6})^A$, with $\Upsilon=(k_p^q.(m-k_p^q)/m)^{-1/2}$ varying between $2/m^{1/2}$ and $R^{1/2}/m^{1/2}$. The maximum value for the second member of (3) is then : $R.\ 2^{-5A}/((m^{1/2})^{-5A}).\ (m/R)^{-6A}$ and inequality (3) becomes $0.401^A.\ (2/m^{1/2})^A >> 2^{-5A}.R^{6A+1}.\ m^{-7/2A}$, that is: $.m^{-1/2A} >> 0.039^A.\ m^{(6A+1).X-7/2A}$. From this, making $R=m^X$ and taking the limit situation when m increases, one extracts: X<1/2 that is $R< m^{1/2}$.

As R represents the consequent granularity, i.e., the number of distinguishable (for effective learning) consequent singletons, if one considers different rules those with different consequent parts (that is, of type: IF (A is A1 AND B is B1 AND ....) OR (A is A2 AND B is B2 AND...) OR...THEN Z is Zi), the following sufficient condition has been obtained :

It is a sufficient condition for Boolean neural nets to learn any set of rules, that the number of rules increases with the square root of m.

4 Traffic Application

There are real applications of this work, despite its main objective is simply to present a possible model for explanation of reasoning and learning in natural neural networks. Thus, one could use it in any context where each neuron can be associated with an "agent" that partially observes local samples of the real world. Different experiments could be simply different observations by the set of "agents" (neurons).

Consider, as a possible very simple example, that it is pretended to establish a qualitative base rule, capable of relating traffic conditions on three one lane main roads entering a town with the traffic on an internal one lane street of the same town. This is, clearly, a three antecedent one consequent problem. The town administration could admit a number of persons -"the agents" and separate them by the three roads and the street to observe the traffic – N persons per street or road. Each one of them positioned on the roads (antecedents) should note down if there is a car passing in

front of him at times t_1, t_2, ..., t_m, count that number and record it, and repeat the experiment on a previously accorded schedule defined by a set of t_1´s and where intervals between t_j and t_{j+1} are previously defined (X experiments which could be distributed along the day or days). As for those positioned on the street (consequent), each one should note down if there is a car passing in front of him or not at times t_1. This should be a completely decentralized set of experiments (one experiment considered to be the set of the N observations both for the consequent and antecedents), very easy to implement and where it is not asked to anybody to make, only by himself and dependent of subjective errors, any global assessment of the traffic conditions. If it is admitted that the time slots (t_m-t_1) are short enough to consider that the traffic conditions do not vary during each one of them and that people are randomly distributed along the roads one gets the conditions to use the net. At the end of the set of experiments, the data of each experiment - N recorded values (C_i, i=1,N) of the street observers together with 3 random chosen person records (^iA1_j, ^iA2_k, ^iA3_l), one from each road – is used to teach the net. The process is repeated for every experiment.

A simulation of this example has been carried out, where 40 neurons per antecedent and consequent have been use and a set of 4 cycles of teaching experiments was used. In each cycle traffic has been generated on the roads and street using three traffic situations both for antecedents and consequent. For the antecedents, and for each experiment, it has been used a binomial distribution to generate the number of active antecedent neurons on each set of 8 neurons (m=8), with the following partition of activated neurons: S0≡{0,1,2}; S1≡{3,4,5} and S3≡{6,7,8}. The traffic conditions were randomly generated using as individual probabilities, for the binomial, p=0.1; p=0.5 and p=0,9, which can be labeled as **L**ow, **M**edium and **H**igh respectively. For the consequent random traffic has been originated with three different probabilities: 0.15, 0.5 and 0.85, which may also be labeled as **L**ow, **M**edium and **H**igh, respectively. Using these settings, each cycle taught the following set of traffic conditions: **LLL_L**; **LLM_L**; **LLH_L**; **LML_L**; **LMM_L**; **LMH_L**; **LHL_L**; **LHM_L**; **LHH_L**; **MLL_M**; **MML_M**; **MLH_L**; **MMH_L**; **MHL_M**; **MHM_M**; **MHH_M**; **HLL_H**; **HML_H**; **HHL_H**; **HLH_M**; **HMH_M** and **HHH_M**, where the first three letters indicate the traffic conditions of the three antecedents and the fourth the consequent traffic condition. As it is seen there is a set of rules not taught (**MMM_?**, for example), in order to evaluate the behavior of the net in those conditions, and for that purpose a third state (not taught) was used on the simulation for each flip-flop. The results, expressing the activation ratio of the internal flip-flops for each antecedent rule, which is the same as the learnt consequent activation ratio for that antecedent rule, can be seen on figures 2(a), 2(b) and 2(c), respectively for traffic conditions on Road1 “Low”, “Medium” and “High”. It may be noticed that the network has apprehended the taught rules and illustrates the fact that some traffic conditions are much better defined (more “crisp”) then others. It is easily noticed that, for traffic conditions not taught, the ratio of the third state (not taught) is much higher and, in such cases, the activation ratios for 1’s and 0’s are influenced by neighbor taught rules. As a conclusion, the network has been able to learn the set of rules presented (not explicitly) during the teaching experiments.

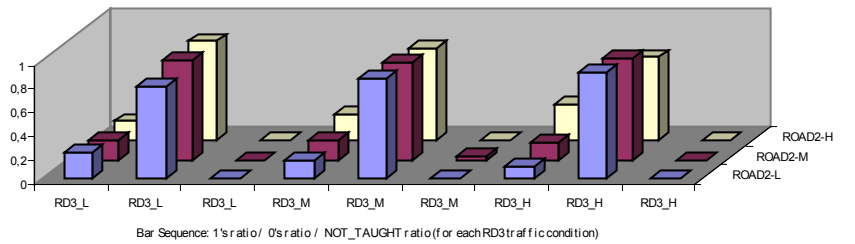

Fig 2(a) - Traffic Low on Road1

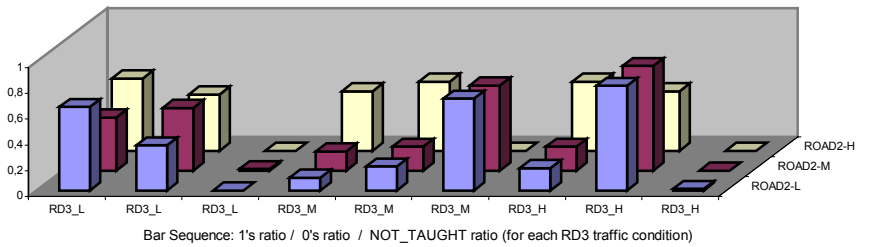

Fig 2(b)-Traffic Medium on Road1

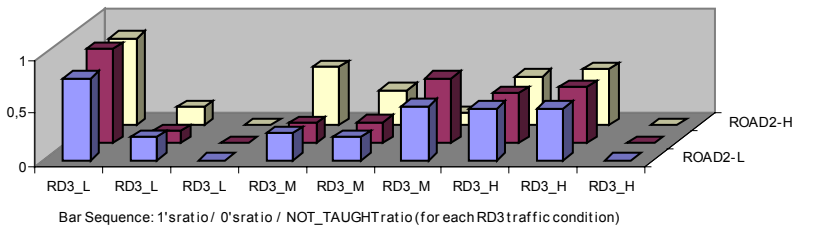

Fig 2(c)-Traffic High on Road1

5 Conclusions

A class of neural nets which functionality seems to be more similar to natural nets than the classic neural nets has been presented elsewhere[9]. In this class of networks variables or concepts are associated with different neural areas, meshes of links are established from areas to other areas depending on antecedent-consequent dependence, individual links are randomly established between neurons of those areas, individual neurons have only a two state space dimension (fire/do not fire) and the notion of "amplitude" of each variable is given by a natural activation ratio (that is the fraction

of activated neurons on a given area). The net is also totally digital (binary); there are no weights. Robustness is an intrinsic property of such nets: any number of neurons or connections may be deleted or corrupted, provided the remaining neurons are enough to define accurately the activation ratios

Moreover these nets are capable of reasoning, implementing qualitative rules. It appears that fuzzy reasoning is a natural emergent property of these networks. Mechanisms for non-supervised learning of fuzzy rules from real experiments use Hebbian like concepts. It has been shown that the learning process converges, not only for one single rule but also for a repetitive sequence of different rules. In such a case it has been concluded that, in the limit, the system is able to learn a number of rules compatible with a consequent granularity increasing with the square root of m, the number of inputs per neuron and per antecedent.

References

[1] Goldberg, S. Introduction to Difference Equations, Dover Publications, NY. (1958).

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

[3] Hayashi, Y., E. Czogala and J.Buckley "Fuzzy neural controller" Proc. IEEE Int. Conf. Fuzzy Systems, 197-202,San Diego. (1992).

[4] Hebb, D. The Organization of Behaviour: A Neuropsychological Theory. John Wiley &Sons. (1949).

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

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

[7] Lin, Chin-Ten and Lee,C.S. A Neuro-Fuzzy Synergism to Intelligent Systems.New Jersey : Prentice Hall. (1996).

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

[9] Tomé,J.A.."Neural Activation ratio based Fuzzy Reasoning."Proc. IEEE World Congress on Computational Inteligence, Anchorage,May 1988,pp 1217-1222. (1998,a).

[10] Tomé,J.A."Counting Boolean Networks are Universal Approximators" Proc. of 1988 Conference of NAFIPS, Florida, August 1988, pp 212-216. (1998,b)

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

[12] Handbook of Mathematical, Scientific and Engineering Formulas, Tables, Functions, Graphs, Transforms. Research and Education Association, New Jersey.