AN APPROACH TO THE CONSTRUCTION OF FUZZY NEURAL NETWORK BASED ON FUZZY RELATION*

Z.M. ZHANG A.DI NOLA S. SESSA

Istituto Di Matematica
Facolta Di Architettura
Universita Di Napoli
Via Monteoliveto 3,80134 Napoli
Italy

ABSTRACT:
Fuzzy Relation is an important concept in Fuzzy Sets Theory. In this paper, based on Fuzzy Relation, we propose a method to build a Fuzzy Neural Network for fuzzy system. The method is different from those presented by other papers, and the structure of the fuzzy neural network seems more reasonable and flexible. A numeric example has proved its effectiveness.

[KEYWORDS] FUZZY MODEL, FUZZY NEURAL NETWORK, FUZZY RELATION

I. INTRODUCTION

Since Zadeh firstly proposed the concept of "Fuzzy Sets" in 1965 , fuzzy sets theory and its applications, have been developed quickly and widely [4, 5, 6]. Especialy in such fields as control and expert system, it has been proved that fuzzy logic is a powerful mathematical tool for dealing with the modelling and control aspects of complex processes, which transparently expresses the conflicting character of the precision of the model and the degree of its generality, i.e., the principle of incompatibility [1].

On other hand, recently, Neural Network has become a hot research topic after having been silent for more than twenty years. Its most important feature is that a system based on neural network is endowed with learning function from environmental facts so that its adaptivity to environment is largely enforced.

Providing a fuzzy logic system with the constructure of neural network makes it possesses both the ability of dealing with uncertainty of knowledge and self-completing function (learning function). Such a system is called Fuzzy Neural Network (FZNN) usually. Up to now, many research works concerned on FZNN and its applications have been done [2]. The common methodology of those works about how to construct a FZNN, is to follow the structure of fuzzy reasoning rules of "if-then", so that the constructure of the system is fixed on how many rules there are and how many linguistic values there are on input & output universes[3].

In this paper, based on the Fuzzy Relation, we put forward an approach of how to construct Fuzzy Neural Network so that it has a flexible constructure regardless of how many rules and how many linguistic values there are on input & output universes. Hence, it becomes easy to build a fuzzy logic system by use of the FZNN proposed by us as its main part.

---

*Supported By C.N.R in Italy

II. ABOUT FUZZY RELATION AND FUZZY CONTROL

Fuzzy relation has become an important concept in fuzzy sets theory, even in the research fields of control, modelling, and knowledge engineering, etc. [6]. In fact, only if we recall how the first fuzzy controller proposed by Mamdani was run successfully, we can easily find that fuzzy relation, really has taken an important part in the fields of control, despite that the reasoning result now is obtained not from the composition operation about fuzzy relation but from so-called "SIMPLIFIED FUZZY REASONING METHOD" or " DIRECT METHOD FOR FUZZY REASONING " for the sake of speed-up and memory-save.

Hence, it is worthwhile to state the mechnism of the first fuzzy controller simply. For simplicity, assume that a fuzzy controller or even a fuzzy system has one input and one output and the fuzzy rules are following:

Rule i : if x is Ai then y is Bi. (1)

where, $Ai \in F(X)$, $Bi \in F(Y)$, $i = 1, 2, \ldots, n$. $\forall x \in X$, $\forall y \in Y$.
X, Y are universes respectively.

The procedure of constructing a fuzzy controller by use of fuzzy relation usually is devided into two steps :

Step 1: Find a fuzzy relation matrix implicated in Eq(1)

$$R = \bigcup_{i=1}^{n} Ri \qquad (2)$$

where, $Ri = Ai \times Bi$, $i = 1, 2, \ldots, n$ (3)

or denotes Eq(1) in the grade degree of membership:

$$R(x, y) = \bigvee_{i=1}^{n} Ri(x, y) \qquad (4)$$

where, $Ri(x, y) = Ai(x, y) \wedge Bi(x, y)$, $i = 1, 2, \ldots, n$
for all $\forall x \in X$, $\forall y \in Y$
"\/", "/\" means max and min operations respectively

Step 2: Give an practical input A, find output of fuzzy system by using the composition of fuzzy relation, i.e., we have output B:

$$B = A \circ R \qquad (5)$$

Where, "o" denotes a composition operator of fuzzy relation. There are numerous composition operators [4,5]. Here, we explain "o" as sum-* operator, i.e.

$$B(y) = \sum_{\forall x \in X} [\ A(x) * R(x,y)\ ] \qquad (6)$$

Eq(5) can be expressed as in Fig.1, imagically. From Fig.1, it tells us that Fuzzy Relation R plays a role in describing dynamic characteristics of a system as the transfer function in control theory does.

III. CONSTRUCTING A FUZZY NEURAL NETWORK BASED ON FUZZY RELATION

3.1 Network Explanation for Eq(6)

Since a fuzzy relation expresses the dynamic features of a system described by fuzzy model, it is a natural idea to build a Fuzzy Neural Network (FZNN) based on the fuzzy relation. Suppose that the universe of discourse X and Y are discreted respeetively, say, $X \supset (x1, x2, \ldots, xn)$, $Y \supset (y1, y2, \ldots, ym)$, so that we can denote Eq(6) in the form of network structure shown in Fig.2.

Obviously, Fig.2 can be also named as a kind of neural network which has only two layers, i.e., input and output layer. But it should be pointed that the connection weights of the network between two layers are all elements of a fuzzy relation which expresses the dynamic features of a fuzzy system. This point is very consistent with the main feature of a neural network that the knowledge about a system is distributed on its all connection weights. That is to say, physical sense expressed by Fig.2 is very clear and easy to understand. Of course, it is also a kind of fuzzy neural network.

3.2 Constructing a Fuzzy Neural Network using Fig.2 as a unit

What Eq(6) or Fig.2 expresses is only a simplest case of a fuzzy system, Here, we begin to discuss how to construct a fuzzy neural network for a more complex fuzzy system. Given a fuzzy system with multi-input and one-output which consists of following fuzzy rules:

If x1 is Ai1 and x2 is Ai2 and .....and xm is Aim then y is Bi (7)

where, $Aij \in F(Xj)$, $Bi \in F(Y)$. $i=1,2,\ldots,n$, $j=1,2,\ldots,m$.
Xj, Y are universes respectively, $\forall x \in Xj$, $\forall y \in Y$

We can decompose a complex fuzzy model like Eq(7) into consisting of several singlest models like Eq(1) based on the equivalence of two models as follows:

$$\left\{\begin{array}{l} \text{If } x1 \text{ is } Ai1 \text{ then } y \text{ is } Bi \\ \text{and} \\ \ldots\ldots \\ \text{and} \\ \text{If } xm \text{ is } Aim \text{ then } y \text{ is } Bi \end{array}\right\} \quad Ri \qquad (8)$$

And we have the fuzzy reasoning results from Eq(8)

$$B^*(x) = \left(\bigcap_{j=1}^{m} A_j^*\right) \circ R \qquad (9)$$

where, $$R = \bigcup_{i=1}^{n} \left(\bigcap_{j=1}^{m} Rij \right) \qquad (10)$$

$$Rij = Aij \times Bi, \quad B^* \in F(Y), \; i=1, 2, \ldots, n, \; j = 1, 2, \ldots, m \qquad (11)$$

For the sake of simplicity, we can approximately obtain following Eq(12) from Eq(9):

$$B^*(x) \approx \bigcap_{j=1}^{m} (A_j^* \text{ o } R) \quad (12)$$

It should be pointed that the approximation between Eq(12) and Eq(9) can be neglected because R in Eq(12) will be completed after learned.

Therefore, by use of Fig 2. as a unit, it is easy for us to build a Fuzzy Neural Network which can describe the dynamic features of fuzzy model like Eq(7) shown in Fig.3, in which TRAN_NN, which is from Fig.2, is a neural network used to finish the transfer from universe of input to universe of output and it is composed of several TRAN-NN units shown in Fig.4 and Fig.5. On the other hand, Min Operation Array is used to realize min operation in Eq(12), its more detail description is given in Fig.6.

Meanwhile, in a practical problem, input and output of a system usually are real numbers. In this case, we can easily change Fig.3 into Fig.7, by adding quantification and defuzzification device to Fig.3. Other parts remain the same.

Up to here, a new approach to the building of a fuzzy neural network based on the fuzzy relation for fuzzy system has been proposed by Fig.3.- Fig.7. And its constructure is simple and flexible. In next section, let us give detail explanation of the learning method of the FZNN.

IV. THE LEARNING METHOD OF THE FZNN

Suppose that a fuzzy system has two input and output, say, x1, x2 and y, respectively, and that the discreted level of their universe are all 4. According to the constructing method of the FZNN shown in Fig.3-Fig.7, we give the constructure of the FZNN with two-input and one-output as given in Fig.8, in which Part I is responding to quantification, Part II is responding to TRAN_NN, Part III is responding to Min Operation Array, Part IV is responding to defuzzification in Fig,7.

Following is to discuss the learning method of the FZNN, since the learning method is an important problem for a Neural Network. As well-known, the learning method based on error-back propagation for a neural network is a best one so far. For the FZNN, such a learning method can be used as well. However, because it has some special types of neuros, their learning method should be discussed in detail by use of the idea of error-back propagation and with considering the convergence of the learning algorithm at the same time.

If the expected output is y, and the practical output is O^6, and the evaluation index of error is:

$$E = (1/2) (y - O^6)^2 \quad (13)$$

Hence, for the output layer the error(i.e., 6th layer) is:

$$d^6 = (O^6 - y) \quad (14)$$

$$O^6 = wy * O^5 \quad (15)$$

The error of 5th layer is:

$$d^5 = wy * d^5 \quad (16)$$

The connection weight between 6th layer and 5th layer is wy, which change the output of the defuzzification into practical universe. Its modifying formula is as follows:

$$wy = - \varepsilon * d^6 * o^5 + \eta * wy \quad (17)$$

$$o^5 = \sum_{i=1}^{4} (y_j * o_j^4) \;/\; \sum_{i=1}^{4} o_j^4 \quad (18)$$

where,

ε : the learning rate of the FZNN.
η : the forgetting rate.
y : discreted value of output universe y.

However, there is a limitation for the learning of wy, i.e., if wy < 0 after modified, then let wy remain unchanged at this learning since it is no meaning for a negtive wy.

The connection weights between 5th layer and 4th layer, i.e, Part IV and Part III, are fixed as y1,...,y4, which are discreted level of universe of discourse of output y. For the 4th layer, we have the error:

$$d_j^4 = d^5 * y_j \;/\; (y4 - y1) \quad (19)$$

Where, j = 1, 2, ...,4, the number of neuros of 4th layer.
y1 : min value of output universe y.
y4 : max value of output universe y.

Because the neuros of 4th layer are a kind of min neuros, we should define the back propagating method of its error.

Since, its input and output function is:

$$o_j^4 = \min(i_{j1} , i_{j2}), \quad j = 1, 2, 3, 4 \quad (20)$$

Where, i_{j1} , i_{j2}, are two input of jth neuros of 4th layer respectively.

That is to say, the error of jth neuro of 4th layer may be caused by i_{j1} or i_{j2}, but for the conservative consideration, it is better to take the errors of both the two input as the same, i.e., d_j^4. Since the connection weights of 3rd layer, i.e., between Part III and Part II in Fig.8, are equal to 1, the errors of neuros of 3rd layer are as follows:

$$d_{kj}^3 = d_j^4 * f'(i_{kj}^3) \quad (21)$$

$$f'(i_{kj}^3) = o_{kj}^3 * (1 - o_{kj}^3) \quad (22)$$

$$o_{kj}^3 = 1 \;/\; [1 + \exp(-i_{kj}^3)], \quad j = 1, 2, 3, 4 \quad (23)$$

$$i_{kj}^3 = \sum_{j=1}^{4} (o_{kj}^2 * W_{kjl}) \quad (24)$$

where, k = 1, 2, the number of input of the FZNN.

The error of 2nd layer can be given as follows:

$$d^2_{kj} = \sum_{l=1}^{4} (d^2_{kj} * W_{kjl}) \qquad (25)$$

Therefore, we can easily obtain the modifying formula of the connection weights between 3rd layer and 2nd layer, which in fact mean all elements of fuzzy relation matrix R.

$$W_{kjl} = - \varepsilon * d^2_{kj} * O^2_{kj} + \eta * W_{kjl} \qquad (26)$$

$$O^2_{kj} = O^1_{kj} \qquad (27)$$

$k=1,2, \quad j=1,2,3,4. \quad l=1,2,3,4$

where, O^2_{kj}: output of jth neuro of kth input of 2nd layer.

O^1_{kj}: output of jth neuro of kth input of 1st layer.

W_{kjl}: means the connection weight between jth neuro of kth input of 2nd layer and lth neuro of 3rd layer.

Since W_{kjl} is in fact a element of relation matrix R, that is, W_{kjl} should be less than or equal to 1. Hence, there is also a limitation for the learning of W_{kjl}, i.e., if $W_{kjl} < 0$ after modified, then let $W_{kjl} = 0$, if $W_{kjl} > 1$ after modified, then let $W_{kjl} = 1$.

The neuros of 1st layer, i.e., are special one whose function is to distribute its input to only one of its next neuros with the value of 1. That is, its function can be described as follow:

$$O^1_{kj} = \begin{cases} 1 & \text{if } x_{k(j-1)} < i'_k < x_{kj} \\ 0 & \text{otherwise} \end{cases} \qquad (28)$$

$$i'_k = sck * i_k \qquad (29)$$

Where, $k = 1, 2, \quad j = 1, 2, 3, 4.$

x_{k0}: the minimum value of universe X_k

x_{kj}: discreted valve of universe X_k

i_k : kth input of the FZNN

sck: scalers of input

Hence, the error of neurals of 1st layer is:

$$d^1_k = \sum_{j=1}^{4} (O^1_{kj} * d^2_{kj} * x_{kj} / (x_{kmax} - x_{kmin}) \qquad (30)$$

Where, $k = 1, 2. \ \forall x_{kj} \in X_k$, discreted value of universe X_k

x_{kj} : discreted value of universe X_k

x_{kmax}: minimum value of universe X_k

x_{kmin}: maximum value of universe X_K

· The modifying formula of input scalers (sc1 & sc2 in Fig.8) is given as follows:

$$sck = -\xi * d'_K * i_K + \eta * sck \qquad k = 1, 2 \tag{31}$$

Like the modification of wy, there is also a limitation for the learning of sck, i.e., if sck < 0 after modified, then let sck remain unchanged at this learning since it is no meaning for a negtive sck.

REMARK: the learning rate and forgetting rate is written the same in Eq(17), Eq(26) and Eq(31). Actually, they can have different values according to the needs of practical applications.

Thus we have the learning method of the FZNN like Eq(14)---Eq(31) by taking two-input & one-output fuzzy system as a simple example. But the learning method can be used into the FZNN with multi-input and multi-output as well.

V. A NUMERIC EXAMPLE

In this section, we shall give a numeric example by use of data given in Table 1 [7] which is often used to evaluate the precision of the model designed or learned and is obtain from following non-linear function:

$$y = (1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5})^2 \tag{32}$$

In Table 1, there is a dummy variable x4, which can be used as a noise to the model. In this example, we do not take it into account as some other papers do.

The evaluation criterion about the model is :

$$E = \frac{1}{N} \sum_{i=1}^{N} \frac{|\, yi - yi' \,|}{yi} * 100\% \tag{33}$$

where, yi is output of the real model (or from function)
yi' is output of the model, i.e., the output of the FZNN in this paper
N is number of data

Like other papers, we use 20 data as learning data(right 20 data in Table 1) and another 20 data as evaluating data (left 20 data in Table 1). The learning method is given in last section, and the difference is only that there are three input in this example. The initial weights(including thresholds) are given randomly between 0 and 0.5. The discreted level of input are all 7 and 17 for output. However, in this example, we let the scalers of inputs and output fixed as 1.

Only after learning for 100 times with the learning data, the FZNN obtains acceptable results given in Table2 in which some results from other papers are also listed in order to give a convenient comparision. That is, the FZNN can be really convergent.

From Table 2, we can say that the FZNN proposed by us has not so good result about learning data, but it has a satisfying one about evaluating data, which means that the FZNN possesses a good robustness for its unknown environment. So, we think that the FZNN has more applicational value since even for unknown inputs, it can almostly remain the same precision from the learning data (its known environment). However, the precision of some models identified from other papers(see Table 1) becomes several times worse about evaluating data than about learning data, which may be dangerous in the practical applications.

On the other hand, generally speaking, for a learning system, the more its initial weights (or parameters) can be given randomly, the easier it can be used. However, some fuzzy neural networks such as proposed by [3] should be given a better initial parameters (e.g., good primary rules). Otherwise they can not be used. Conversely, it is easy to use the FZNN since its random initial weights. Therefore, the effectiveness of the FZNN proposed by us in this paper is proved.

VI. CONCLUSION & FUTURE WORK

In this paper, based on Fuzzy Relation, we propose a method to build a Fuzzy Neural Network for fuzzy system. The method is different from these presented by other papers, and the structure of the fuzzy neural network seems more reasonable and flexible. Compared with other fuzzy neural networks, the effectiveness of the FZNN proposed by us are proved by use of a numeric example.

Our future work about the FZNN is to use it into a control system in order to build a self-learning system.

REFERENCES

[1] W.Pedryce, Selected Issues of a Frame of Knowledge Representation Relized by Means of Linguistic Labels, J. of Intelligent Systems, submitted.

[2] J.Owan, D.H.Sharp, Neural Nets, Quarterly Reviews of Biophysics 21, 3(1988), pp365-427 (Printed in Great Bitain)

[3] S.I.Horikawa, T.Furuhashi, S.Okuma, Y.Uchikawa, A Learning Fuzzy Controller Using a Neural Network, Journal of the Society of Instrument and Control Engineering, No.2, Vol.27, 1991, pp208-215 (in Japanese)

[4] Zhi-ming Zhang, Studies on the Implementation of Fuzzy Inference Machine on Hardware and Its Applications, Ph.D Thesis, Beijing Normal University, 1990.

[5] M.Mizumoto, Fuzzy Sets and Their Operators II, Information and Control, 50(1981), p166-174.

[6] A.Dinola, S.Sessa, W.Pedrycz, E,Sanchez, Fuzzy Relation Equations and Their Applications to Knowledge Engineering, Kluwer Academic Bublishers, 1989.

[7] M.Sugeno, et al, Fuzzy Modelling, Transaction of the Society of Instrument and Control Engineering, (Japan), 23-6, 1987.

[8] M.Yamaoka, M.Mukaidono, A Learning Method of Fuzzy Inference Rules with a Neural Network, IFSA'91, Brussels.

Table 1 Input and Output Data for Evaluation

No.	x1	x2	x3	x4	y	No.	x1	x2	x3	x4	y
1	1	3	1	1	11.11	21	1	1	5	1	9.545
2	1	5	2	1	6.521	22	1	3	4	1	6.043
3	1	1	3	5	10.19	23	1	5	3	5	5.724
4	1	3	4	5	6.043	24	1	1	2	5	11.250
5	1	5	5	1	5.242	25	1	3	1	1	11.110
6	5	1	4	1	19.02	26	5	5	2	1	14.360
7	5	3	3	5	14.15	27	5	1	3	5	19.610
8	5	5	2	5	14.36	28	5	3	4	5	13.650
9	5	1	1	1	27.42	29	5	5	5	1	12.430
10	5	3	2	1	15.39	30	5	1	4	1	19.020
11	1	5	3	5	5.724	31	1	3	3	5	6.380
12	1	1	4	5	9.766	32	1	5	2	5	6.521
13	1	3	5	1	5.87	33	1	1	1	1	16.000
14	1	5	4	1	5.406	34	1	3	2	1	7.219
15	1	1	3	5	10.19	35	1	5	3	5	5.724
16	5	3	2	5	15.39	36	5	1	4	5	19.020
17	5	5	1	1	19.68	37	5	3	5	1	13.390
18	5	1	2	1	21.06	38	5	5	4	1	12.680
19	5	3	3	5	14.15	39	5	1	3	5	19.610
20	5	5	4	5	12.68	40	5	3	2	5	15.390

Table 2 Evaluation value of designed (or learned) model

Model	Input Variables	E1(%)	E2(%)
FZNN	x1, x2, x3	2.02	1.96
Fuzzy Controller using NN [3]	x1, x2, x3	0.6	2.41
Fuzzy Model I [7]	x1, x2, x3	1.5	2.1
Fuzzy Model II [7]	x1, x2, x3	1.1	3.6
Artificial NN-driven fuzzy reasoning [3]	x1, x2, x3	0.47	4.79
Fuzzy Model using NN [8]	x1, x2, x3	2.15	6.99
GMDH	x1, x2, x3	4.7	5.7

where, E1: the evaluation value which is calculated by use of the designed model(or learned) from learning data

E2: the evaluation value which is calculated by use of the designed model(or learned) from evaluating data

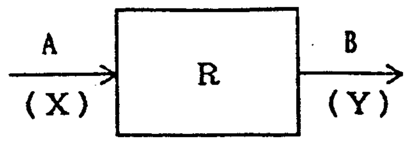

Fig. 1 The transfer process of universe based on Fuzzy Realtion

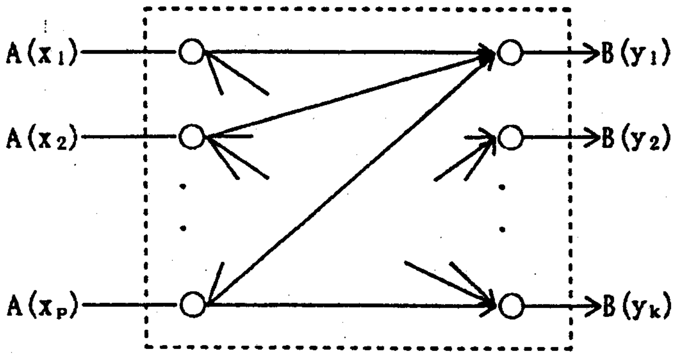

Fig. 2 Network Expression of Egu. (7)

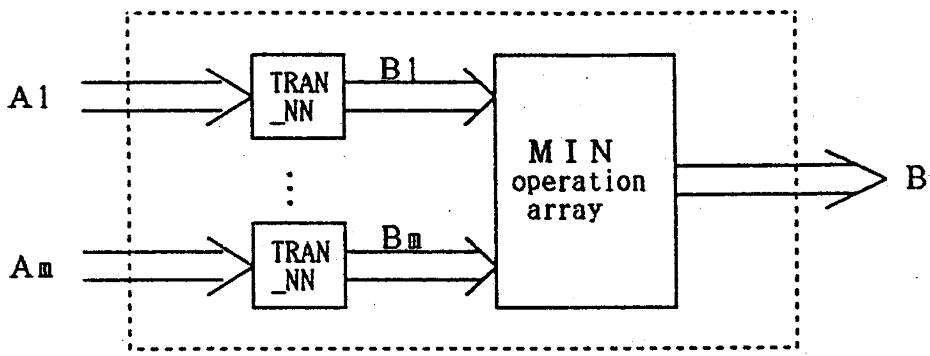

Fig. 3 An block diagram of Fuzzy Neural Network (FZNN)

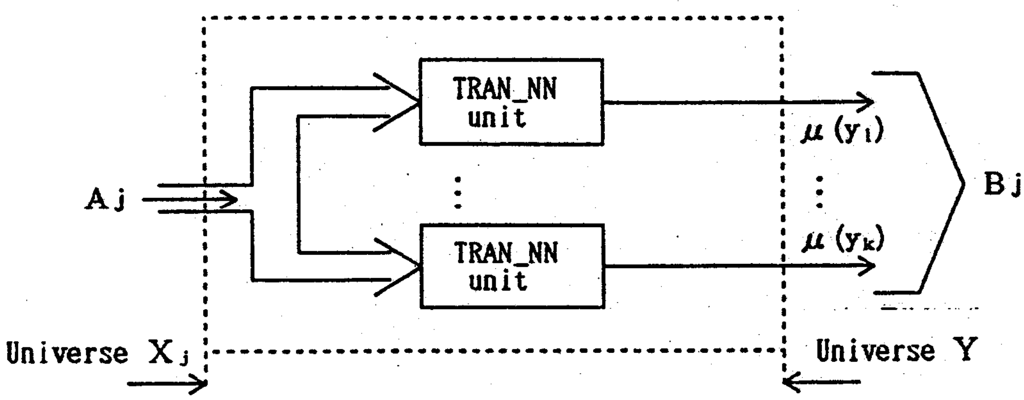

Fig. 4 A Block Diagram of a TRAN_NN in FZNN

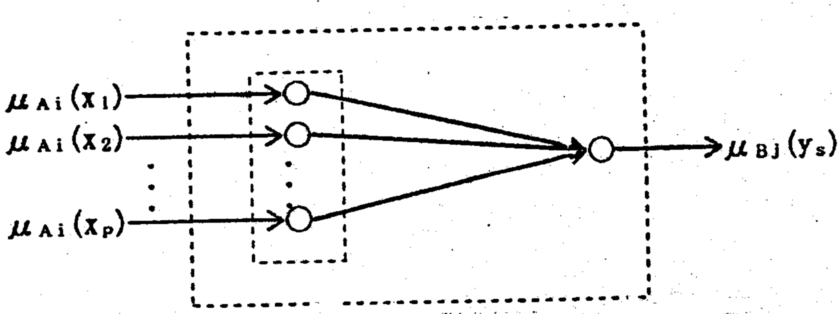

Fig. 5 The constructure of a TRAN_NN unit in TRAN_NN

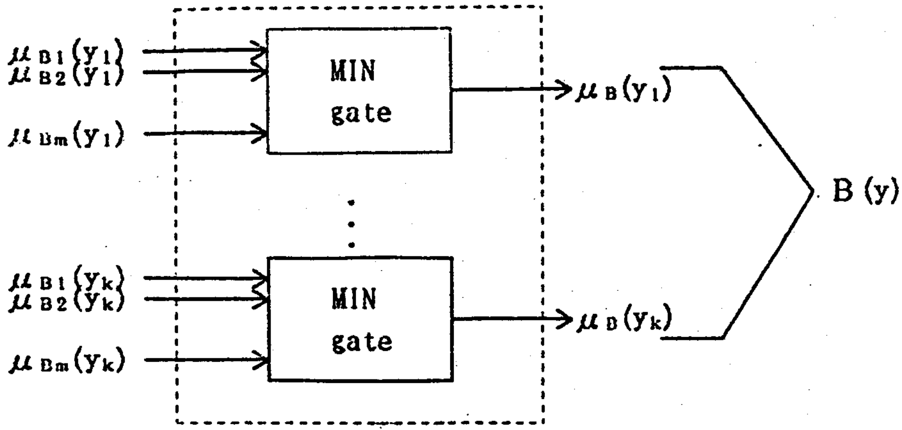

Fig. 6 The internal architecture of MIN operation array in FZNN

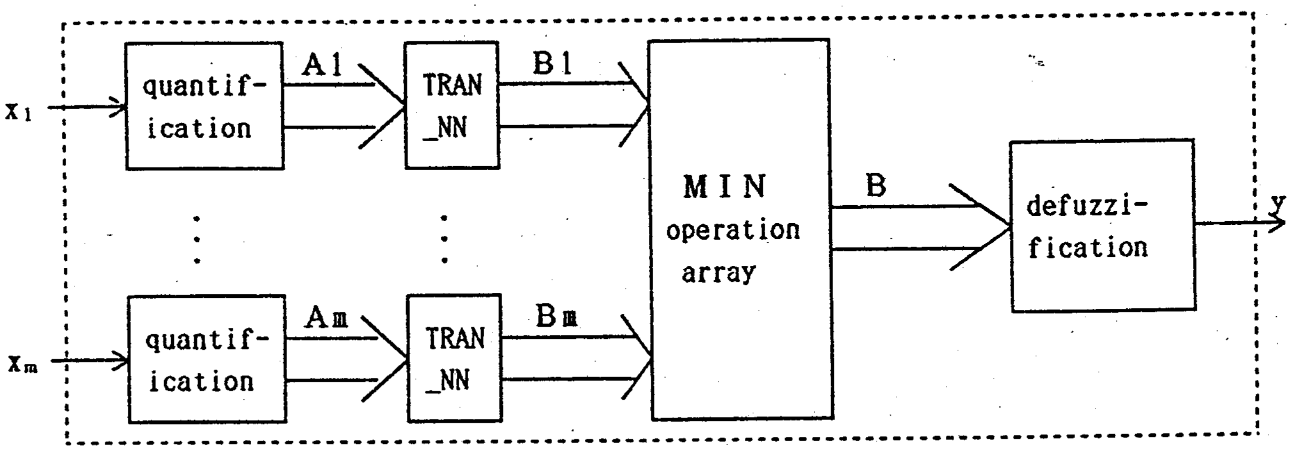

Fig. 7 An block diagram of FZNN with real input & output

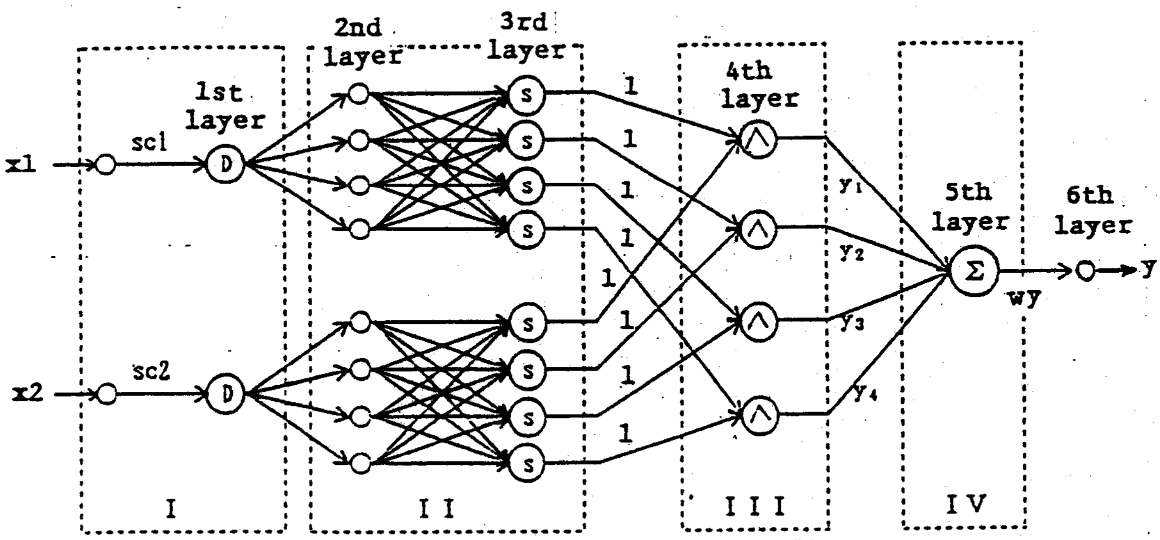

Fig. 8 An Example of FZNN