A Fuzzy Inference Network for Classification

Y. Cai and H. K. Kwan
Department of Electrical Engineering, University of Windsor
401 Sunset Avenue, Windsor, Ontario, Canada N9B 3P4

Abstract — In this paper, a fuzzy inference network (FIN) is proposed. The proposed FIN preserves the advantages of both fuzzy classification algorithm and neural networks. It can learn membership functions directly from training samples and classify patterns according to the membership values. An efficient self-organizing learning algorithm is also presented.

I. INTRODUCTION

Non-fuzzy classification techniques assume that a pattern **X** belong to only one class while fuzzy classification algorithms assign a pattern **X** with a distributed membership value to each class. The fuzzy labels can be defuzzified and then a crisp decision can be reached by using fuzzy idea in the model. We can find a better solution to a crisp problem by looking in a larger space at first, which allow the algorithm more freedom to avoid errors. The key point of fuzzy classification is to find the fuzzy boundaries between all the classes. Most of the existing fuzzy classification algorithms cannot learn membership functions directly from training data with fuzzy labels. A few of them using backpropagation learning algorithm to learn the parameters of the membership functions. The main problem of BP algorithm is that its learning speed is slow. In this paper, a classifier based on fuzzy inference network (FIN) is proposed. It can learn membership functions of all the fuzzy classes directly from fuzzy labelled training data. The proposed FIN utilizes the learning ability of neural networks, and preserves the advantages of fuzzy classification algorithms. An efficient self-organizing learning algorithm is also proposed.

II. FUZZY INFERENCE NETWORK

A neuron is called a fuzzy neuron (FN) if it has N weighted inputs and M outputs. Moreover, we have:

$$z = h(w_1 x_1, w_2 x_2, \cdots, w_N x_N) \tag{1}$$

$$s = f(z, t) \tag{2}$$

$$y_j = g_j(s) \qquad j = 1, 2, \cdots, M \tag{3}$$

where z is the net input; h( ) is the aggregation function; s is the state; f( ) is the activation function; t is the activating threshold; and $\{g_j(\ ) \mid j=1,2,\ldots,M\}$ represent M output functions of the FN.

The proposed FIN is constructed in a three-layer structure and different types of fuzzy neurons are used in different layers (shown in Fig. 1). The first layer accepts input patterns and transfers the features of the patterns into membership values. The second layer represents training patterns. The third layer gives the membership values of all the fuzzy classes. The algorithm of the FIN is:

$$y_{ij}^1 = g_{ij}(x_i) = \begin{cases} 1 + \alpha_{ij}^1 (x_i - \theta_{ij}) & \text{if } 0 \ge \alpha_{ij}^1 (x_i - \theta_{ij}) \ge -1 \\ 1 - \alpha_{ij}^2 (x_i - \theta_{ij}) & \text{if } 1 \ge \alpha_{ij}^2 (x_i - \theta_{ij}) \ge 0 \\ 0 & \text{if otherwise} \end{cases}$$

$$i = 1, 2, \ldots, N; \qquad j = 1, 2, \ldots, M \tag{4}$$

$$y_j^2 = \underset{i=1}{\overset{N}{MIN}} (y_{ij}^1) \qquad j = 1, 2, \ldots, M \tag{5}$$

$$y_p^3 = \begin{cases} \dfrac{\sum_{j=1}^{M} (w_{jp} \cdot y_j^2)}{\sum_{j=1}^{M} w_{jp}} & \text{if} \sum_{j=1}^{M} w_{jp} \neq 0 \\ 0 & \text{if} \sum_{j=1}^{M} w_{jp} = 0 \end{cases} \tag{6}$$

$$p = 1, 2, , \ldots, P$$

0-7803-2431-5/95 $4.00

where x_i is the input of the *ith* FN in the first layer which represents the *ith* feature value of the input pattern. $g_{ij}(x_i)$ is the *jth* output function of the *ith* FN as shown in Fig. 2. Parameters M, Θ_{ij}, α^1_{ij}, α^2_{ij} and w_{jp} (for all the i, j, and p) are to be determined by the learning algorithm. The FNs in the first layer are TRAN-FNs. The FNs in the second layer are called as MIN-FNs and SUM-FNs are used in the third layer.

During the learning procedure, the fuzzy inference network is automatically established to match all the training samples. We define d_{mp} as the *pth* desired output of the *mth* training sample (p=1 to P, m=1 to M) and DX_i (i=1 to N) as the largest distances of the feature values in the *ith* dimension. The learning algorithm is:

Step 1: Create N TRAN-FNs in the first layer, and P SUM-FNs in the third layer. Give the values of DX_i (i=1 to N). Set m=1.

Step 2: Create the *mth* MIN-FN in the second layer. Set $\Theta_{im}=x_{im}$ (i=1 to N), $\alpha^1_{im}=DX_i$, $\alpha^2_{im}=DX_i$. and $w_{mp}=d_{mp}$ (p=1 to P).

Step 3: Adjust α^1_{ir} and α^2_{ir} (r=1 to m) according to the following rule: For i=1 to N and r=1 to m-1, find out those Θ_{ij} that satisfy $\Theta_{ij}<\Theta_{im}$. If Θ_{iJ} belongs to this group and $\Theta_{im}-\Theta_{iJ}$ $=\mathrm{MIN}(\Theta_{im}-\Theta_{ij})$, set $\alpha^1_{im}=1/(\Theta_{im}-\Theta_{iJ})$ and $\alpha^2_{iJ}=1/(\Theta_{im}-\Theta_{iJ})$. For i=1 to N, and r=1 to m-1, find out those Θ_{ik} that satisfy $\Theta_{ik}>\Theta_{im}$. If Θ_{iK} belong to this group and $\Theta_{iK}-\Theta_{im}=\mathrm{MIN}(\Theta_{ik}-\Theta_{im})$, set $\alpha^2_{im}=1/(\Theta_{iK}-\Theta_{im})$ and $\alpha^1_{iK}=1/(\Theta_{iK}-\Theta_{im})$.

Step 4: Set m=m+1, if m≤M, go to **Step 2**. If m>M, the learning procedure is finished.

In the proposed self-organizing learning algorithm, the fuzzy inference network is trained to match all the training samples. The proposed learning algorithms determine the number of MIN-FNs and adjust the parameters of the FNs in each layer to remember the information of the training samples. The training will be finished in one epoch. This is different from those backpropagation type learning algorithms that have to be performed for many epochs in order to achieve the optimal results. Moreover, this learning algorithm can constitute hypersurface membership functions from training data.

III. SIMULATIONS

The proposed FIN was simulated on PC-486 (33MHz) using C language. In the simulations, the proposed FIN was trained by four sets of training data from fuzzy Exclusive OR (quasi-XOR) problem [4] and the circle-in-the-square (CIS) problem [5]. There are 2 fuzzy classes in the quasi-XOR problem, Y^1 and Y^2 . Training Data Set 1 has 36 input-output samples and Training Data Set 2 has 121 input-output samples (see Table I) from quasi-XOR problem. Training Data Sets 3 and 4 are samples from the CIS problem. The CIS problem is a typical classification problem. There are two crisp classes in the square: inside the circle and outside the circle. The area inside the circle equals to the area outside the circle. There are 121 samples in Training Data Set 3 and 1024 samples in Training Data Set 4. These samples are regularly distributed in the square. The classification results of the FIN are compared with those of the fuzzy ARTMAP [5].

Table II gives the learning results of the FIN. In Table II, M is the number of MIN-FNs in the second layer and t is the learning time. Table III gives the classification results of the FIN and fuzzy ARTMAP. In Table III, K is the number of training samples, RTR is the recognition rate on training data and RTE is the recognition rate on testing data. Fig. 3 shows the derived 2-D membership functions of the quasi-XOR problem. Fig. 4 gives the classification results of the FIN for the CIS problem.

From the simulation results, we can see that the number of MIN-FNs in the second layer equals to the number of training samples. The learning of the proposed FIN is very fast. The classification results of the FIN are better than those of fuzzy ARTMAP. The FIN can recall all the training samples while the fuzzy ARTMAP cannot. The classification result of the FIN with 121 training samples is better than that of the fuzzy ARTMAP with 1000 training samples, and the classification result of the FIN with 1024 training samples is better than that of the fuzzy ARTMAP with 10000 training samples. If more training samples are learned, the derived membership functions will be smoother and the classification results will be better.

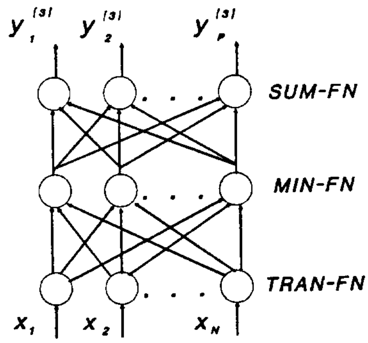


Fig. 1 The proposed FIN

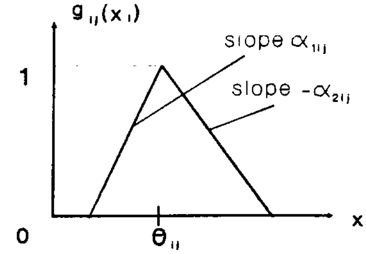
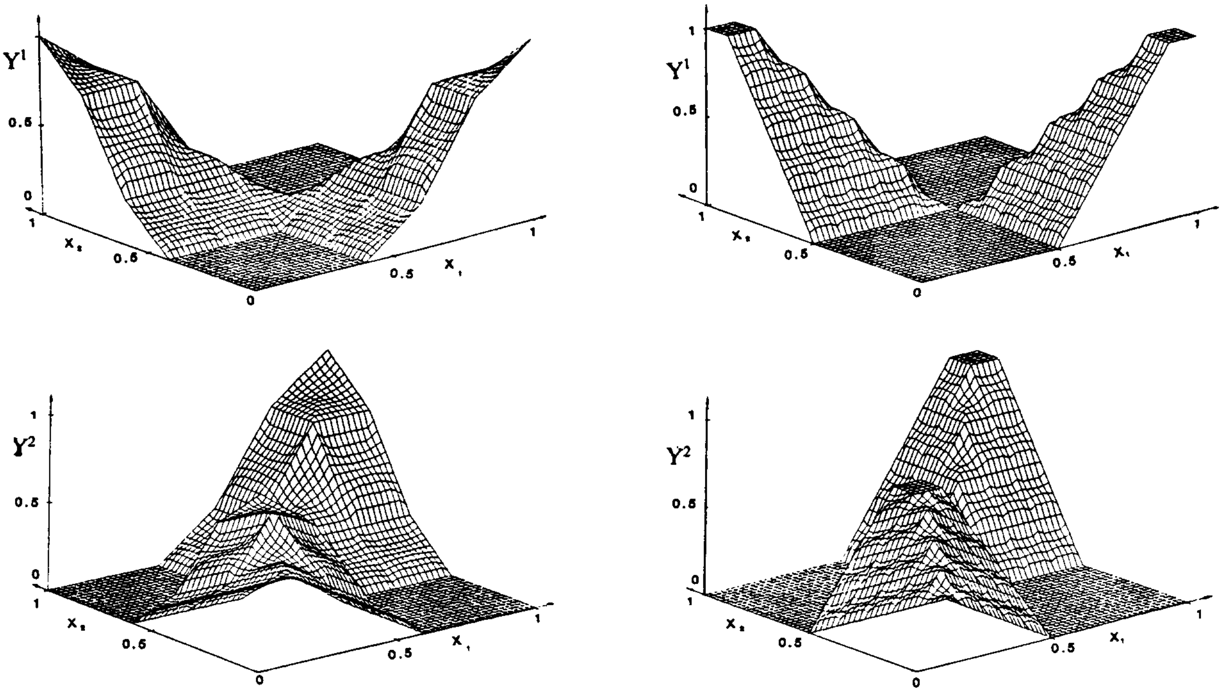


Fig. 2 Output function of TRAN-FNs



(a) with 36 training samples (b) with 121 training samples

Fig.3 The derived membership functions of the FIN for the quasi-XOR problem

(a) with 121 training samples (b) with 1024 training samples

Fig.4 Classification result of FIN for the Circle-in-the-Square problem

VI. CONCLUSIONS

A fuzzy inference network has been proposed in this paper. The proposed FIN can learn membership functions and determine the fuzzy and non-fuzzy partitions according to the membership values. An efficient self-organizing learning algorithm for the FIN has also been presented. The learning is fast. The classification results of the FIN is better than those of the fuzzy ARTMAP. The proposed FINs can be used in pattern recognition systems and fuzzy control systems.

REFERENCES

[1] Y. Cai and H. K. Kwan, "A fuzzy neural classifier for pattern classification", in *Proceedings of International Symposium on Circuits and Systems,* Chicago, May 3-6, 1993, pp.2367-2370.

[2] H. K. Kwan and Y. Cai, "A fuzzy neural network and its application to pattern recognition," *IEEE Transactions on Fuzzy Systems,* No. 3, Vol. 2, 1994, pp. 185-193.

[3] H. K. Kwan, Y. Cai and B. Zhang, "Membership function learning in fuzzy classification," *Int. Journal of Electronics,* Vol.74, No.6, 1993, pp.845-850.

[4] I. Enbustsu, K. Baba, and N. Hara, "Fuzzy rule extraction from a multilayer neural network", in *Proceedings of IJCNN,* July, 1991, Vol.2, pp.461-465.

[5] Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., and Rosen, D. B., "Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, September, 1993, pp. 698-713.

TABLE I Fuzzy labels of quasi-XOR for Y^1 and Y^2

x_2 \ x_1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.0	0, 1.00	0, 1.00	0, 0.75	0, 0.50	0, 0.25	0, 0	0.25, 0	0.50, 0	0.75, 0	1.00, 0	1.00, 0
0.1	0, 1.00	0, 1.00	0, 0.75	0, 0.50	0, 0.25	0, 0	0.25, 0	0.50, 0	0.75, 0	1.00, 0	1.00, 0
0.2	0, 0.75	0, 0.75	0, 0.75	0, 0.50	0, 0.25	0, 0	0.25, 0	0.50, 0	0.75, 0	0.75, 0	0.75, 0
0.3	0, 0.50	0, 0.50	0, 0.50	0, 0.50	0, 0.25	0, 0	0.25, 0	0.50, 0	0.50, 0	0.50, 0	0.50, 0
0.4	0, 0.25	0, 0.25	0, 0.25	0, 0.25	0, 0.25	0, 0	0.25, 0	0.25, 0	0.25, 0	0.25, 0	0.25, 0
0.5	0, 0	0, 0	0, 0	0, 0	0, 0	0, 0	0, 0	0, 0	0, 0	0, 0	0, 0
0.6	0.25, 0	0.25, 0	0.25, 0	0.25, 0	0.25, 0	0, 0	0, 0.25	0, 0.25	0, 0.25	0, 0.25	0, 0.25
0.7	0.50, 0	0.50, 0	0.50, 0	0.50, 0	0.25, 0	0, 0	0, 0.25	0, 0.50	0, 0.50	0, 0.50	0, 0.50
0.8	0.75, 0	0.75, 0	0.75, 0	0.50, 0	0.25, 0	0, 0	0, 0.25	0, 0.50	0, 0.75	0, 0.75	0, 0.75
0.9	1.00, 0	1.00, 0	0.75, 0	0.50, 0	0.25, 0	0, 0	0, 0.25	0, 0.50	0, 0.75	0, 1.00	0, 1.00
1.0	1.00, 0	1.00, 0	0.75, 0	0.50, 0	0.25, 0	0, 0	0, 0.25	0, 0.50	0, 0.75	0, 1.00	0, 1.00

TABLE II Learning results of the FIN

Training Data Set	N	M	P	t(sec)
1	2	36	2	0.05
2	2	121	2	0.06
3	2	121	2	0.33
4	2	1024	2	8.74

TABLE III Classification results of the FIN comparing with Fuzzy ARTMAP for CIS problem

	FIN		fuzzy ARTMAP		
K	121	1024	100	1000	10000
RTR	100%	100%	99.0%	95.5%	97.7%
RTE	93.6%	98.5%	88.6%	92.5%	96.7%