A LEARNING RULE FOR FUZZY ASSOCIATIVE MEMORIES *

Fan Junbo, Jin Fan and Shi Yan
Inst. of Neural Networks & Information Techniques
Southwest Jiaotong University
Chengdu, Sichuan 610031, P. R. China

Abstract

In this paper a learning rule for multiple pattern pairs in fuzzy associative memories (FAMs) with Max-Min composition units is presented. Under a certain condition, the proposed rule can efficiently encode multiple fuzzy pattern pairs in a single FAM and perfect association of these pairs can be achieved. The correctness of the proposed rule is proved and illustrative examples are given.

1 Introduction

It is well known that neural networks can efficiently model subjective phenomena such as perception, image and memory while fuzzy logic is another powerful tool to model phenomena associated with perception and thinking. In recent years, several models that combine neural networks and fuzzy logic, called fuzzy neural networks, have been developed and widely applied to prospective fields such as control, pattern recognition, decision making and expert systems [1 -5]. The FAM introduced by Kosko, which involves Max-Min composition units, is one of the most useful and important paradigms [1, 2]. Such a FAM is characterized by the subset recall when trained using the fuzzy Hebian rule. The inability to reliably store more than one pair is its enssential drawback.

Some endeavors have given brith to overcoming this drawback. Hirota and Pedrycz used a gradient descent algorithm with two-value derivatives under some constraints [6]. Authors of this paper ever suggested an iterative learning algorithm that employed a so-called hill -climbing search strategy to conduct connection weight adjustments [7]. However, both algorithms may easily get struck at local minima of a defined index function although satisfactory performence of the trained FAM can usually be achieved. We propose here another learning rule that employs a suitable fuzzy logical-implication operation different from the fuzzy Hebbian rule used by Kosko. The proposed rule is capable of encoding multiple pattern pairs in the weight matrix of a FAM if there exists some weight matrix that properly forms the fuzzy relation between the input and output of the given pairs.

2 FAMs with Max-Min Composition Units

A FAM is designed to store p fuzzy (unit-interval valued) pattern pairs (A_1, B_1), ..., (A_p, B_p) in the weight matrix W, where the kth pair is represented by the vectors $A_k = (a_1^k, ..., a_n^k)$ and $B_k = (b_1^k, ..., b_m^k)$ respectively. The topology of a FAM is shown in Fig. 1.

In the FAM with Max-Min composition units, perfect association implies that eq. (1) should hold

---

$$A_k \circ W = B_k \qquad (k=1, 2, \ldots, p) \qquad\qquad (1)$$

where the symbol "$\circ$" denotes the Max-Min composition operation. Generally, there are two kinds of methods to find the weight matrix W. One is to define a performence index, for instance, the sum of squared errors

$$E = 1/2 \sum_{k=1}^{p} \sum_{j=1}^{m} (b_j^k - b_j)^2 \qquad\qquad (2)$$

where b_j^k and b_j are the desired output and actual output of the jth output unit. Weight adjustment is aimed at minimizing the index function[6]. The other is to encode the single pair (A_k, B_k) into the matrix W_k and then to superimpose the matrices W_k s in a single matrix W, i.e., W may be found using the following equation

$$W = \bigcap_{k=1}^{p} W_k$$

$$= \bigcap_{k=1}^{p} (A_k \to B_k) \qquad\qquad (3)$$

where the symbol "$\to$" denotes some implication operation, and "$\cap$" denotes a generalized superimposition (e.g., maxima, minima etc). Kosko adopted a fuzzy Hebbian (the minimum implication R_c) rule pointwise to encode the kth pair (A_k, B_k) in the W_k, then superimposed the matrices W_ks in a single matrix W by the maximum operation. That is

$$w_{ij} = \bigvee_{k=1}^{p} w_{ij}^k$$

$$= \bigvee_{k=1}^{p} (a_i^k \xrightarrow{R_c} b_j^k)$$

$$= \bigvee_{k=1}^{p} (a_i^k \wedge b_j^k) \qquad\qquad (4)$$

where the symbols "$\vee$" and "$\wedge$" stand for the maximum and minimum operations respectively, and w$_{ij}$s are components of the W. Obviously, multiple pairs can not be reliably stored in the FAM. An approach to overcoming the problem is that a bank of FAMs is used, each of which seperately store only one pair. But this results in consuming space of storage.

3 The Proposed Learning Rule

As usual, we wish to distributively and reliably store multiple fuzzy pattern pairs (A_k, B_k) in a single FAM such that each B_k can be perfectly recalled when each corresponding A_k is presented. Instead of using the minimum implication R_c, we associate here B_k with A_k usng the implication R_g, and we replace the maximum superimposition by the minimum one as well. In this way, we obtain the following learning rule

$$W = \bigwedge_{k=1}^{p} W_k$$

4274

$$= \bigwedge_{k=1}^{p} (A_k \xrightarrow{Rg} B_k) \tag{5}$$

In pointwise notation, that is

$$w_{ij}^k = a_i^k \xrightarrow{Rg} b_j^k$$

$$= \begin{cases} b_j^k & \text{if } a_i^k > b_j^k \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

$$w_{ij} = \bigwedge_{k=1}^{p} (a_i^k \rightarrow b_j^k) \tag{7}$$

Obviously, the complexity of the proposed learning rule is equivalent to the rule used by Kosko.

As we know, there doesn't always exist the weight matrix of a FAM, which makes eq. (1) hold for arbitrary set of input-output pairs. In other words, these pairs have to satisfy a certain condition for the purpose of perfect association. On the other hand, even if there exists some desired weight matrix, the Kosko's rule is not guaranteed to yield the desired matrix, as shown in [1, 2]. Based on analyzing the condition for existence of desired weight matrix, we show below that the proposed learning rule can yield the desired matrix if it indeed exists.

Since the m units in the output layer of the FAM are independent one another, we can consider only one output unit. For the jth output unit, we have

$$\begin{bmatrix} a_1^1 & a_2^1 & \dots & a_n^1 \\ a_1^2 & a_2^2 & \dots & a_n^2 \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ a_1^p & a_2^p & \dots & a_n^p \end{bmatrix} \circ \begin{bmatrix} w_{1j} \\ w_{2j} \\ . \\ . \\ . \\ w_{nj} \end{bmatrix} = \begin{bmatrix} b_j^1 \\ b_j^2 \\ . \\ . \\ . \\ b_j^p \end{bmatrix} \tag{8}$$

We define the following sets and value

$$K = \{ 1, 2, \dots, p\} \tag{9}$$

$$I = \{ 1, 2, \dots, n\} \tag{10}$$

$$G_{ij} = \{ k \in K \mid a_i^k > b_j^k \} \tag{11}$$

$$E_{ij} = \{ k \in K \mid a_i^k = b_j^k \} \tag{12}$$

$$m_{ij} = \begin{cases} \bigwedge_{k \in G_{ij}} b_j^k & \text{if } G_{ij} \neq \varphi \\ 1 & \text{if } G_{ij} = \varphi \end{cases} \tag{13}$$

$$S_{ij} = \{ k \in G_{ij} \cup E_{ij} \mid b_j^k \leqslant m_{ij} \} \tag{14}$$

Then the necessary and sufficient condition for existence of the desired weight vector that enables eq. (8) to hold can be expressed as follows[8]

$$\bigcup_{i \in I} S_{ij} = K \tag{15}$$

4275

If there exists some weight vector enabling eq. (8) to hold such that some $k' \in K$ but $k' \notin S_{ij}$ for all $i \in I$. By definition, $a_i^{k'} < b_j^{k'}$ or $w_{ij} < m_{ij} < b_j^{k'}$. Hence we have $b_j^{k'} = \bigvee_{i \in I} (a_i^{k'} \wedge w_{ij})$

$= \bigvee_{i|k \notin S_{ij}} (a_i^{k'} \wedge w_{ij}) < b_j^{k'}$. This is obviously a contradiction. Vice versa, assume that

$\bigcup_{i \in I} S_{ij} = K$. For any $k \in K$, $a_i^k < b_j^k$, or $w_{ij} < m_{ij} < b_j^k$ for all i as $k \in S_{ij}$, but $a_i^k = b_j^k$ and

$m_{ij} > b_j^k$ or $a_i^k > b_j^k$ and $m_{ij} = b_j^k$ for all i as $k \in S_{ij}$. Hence, if let $w_{ij} = m_{ij}$, we have

$\bigvee_{i \in I} (a_i^k \wedge w_{ij}) = \{ \bigvee_{i|k \in S_{ij}} (a_i^k \wedge w_{ij}) \} \vee \{ \bigvee_{i|k \notin S_{ij}} (a_i^k \wedge w_{ij}) \} \vee_{i|k \in S_{ij}} (a_i^k \wedge w_{ij})$

$= \bigvee_{i|k \in S_{ij}} (a_i^k \wedge m_{ij}) = b_j^k$. This means that there exists a weight vector enabling eq. (8) to hold.

So, there exists a weight matrix W enabling eq. (1) to ho, if and only if eq. (15) holds for any j. The proposed learning rule expressed by eq. (6) and (7) is in accordance with eq. (13). From above analysis, we can conclude that the proposed rule can yield the desired weight matrix enabling eq. (1) to hold if it exists.

4 Examples

Two sets of fuzzy pattern pairs (Data1 and Data2) to be stored are shown in Tab. 1 and 2.

For Data1, we can obtain the weight matrix of the FAM using the proposed rule

$$W = \begin{bmatrix} 0.6 & 0.5 & 1.0 & 0.9 & 1.0 & 0.5 \\ 0.4 & 0.5 & 0.4 & 0.4 & 0.6 & 0.4 \end{bmatrix}^T \tag{16}$$

The FAM with the above matrix can perfectly each output pattern B_k when presented with each input pattern A_k. But the Kosko's rule produces the following weight matrix

$$W = \begin{bmatrix} 0.9 & 0.7 & 0.6 & 0.9 & 0.7 & 0.7 \\ 0.6 & 0.6 & 0.6 & 0.6 & 0.6 & 0.6 \end{bmatrix}^T \tag{17}$$

With the matrix, the trained FAM can not perfectly all pairs in Data1.
For Data2, two rules yield matrices

$$W = \begin{bmatrix} 0.2 & 0.8 & 1.0 & 1.0 & 1.0 & 0.4 \\ 0.6 & 0.5 & 1.0 & 0.4 & 0.4 & 0.5 \end{bmatrix}^T \tag{18}$$

and

$$W = \begin{bmatrix} 0.4 & 0.8 & 0.5 & 0.7 & 0.6 & 0.6 \\ 0.6 & 0.5 & 0.5 & 0.4 & 0.5 & 0.5 \end{bmatrix}^T \tag{19}$$

It is easy to confirm that eq. (1) holds with eq. (18), but it dosen't with the weight matrix eq. (19).

5 Conclusion

A learning rule for multiple fuzzy pattern pairs in FAMs with Max-Min composition units is presented. Under a certain condition, multiple pairs can be reliably stored in a single FAM and perfect associations can be achieved. The correctness of the proposed rule is demonstrated. Due to the simple complexity , the proposed rule may be much of use in the design of fuzzy systems where multiple inference rules need to be reliably stored.

References

[1] Kosko, B. (1987). Fuzzy Associative Memories, In Kandel, A. (Ed.) , Fuzzy Expert Systems Reading, MA: Addison-Wesley.

[2] Kosko, B. (1991). Neural Networks and Fuzzy Systems: a dynamical systems approach to machine intelligence.

[3] Jin, F., FAN, J. B. and TAN, Y. D. (1991). Neural Networks and Neurocomputers: Principles and Applications, Southwest Jiaotong University Press.

[4] Fan, J. B., Jin, F. and Yuan, X. (1992) . A New Neuro- Fuzzy Network for Pattern Classification, Proc. of the ICIIPS'92 (Beijing).

[5] Fan, J. B., Jin, F. and Shi, Y. (1992). A New Model of Fuzzy Associative Memories, Proc. of the IJCNN'92 (Beijing), Vol. 3.

[6] Hirota, K. and Pedrycz, W. (1991). Fuuzy Logic Neural Networks: Design and Computations, Proc. of the IJCNN'91 (Singapore).

[7] Fan, J. B., Jin, F. and Yuan, X. (1992). A Learning Algorithm for Multiple Fuzzy Pattern pair Associations, Proc. of IJCNN'92 (Beijing), Vol. 3.

[8] Saito, T. and Mukaidono, M. (1991). A Learning Algorithm for Max- Min Network and Its Application to Solve Fuzzy Relation Equation, Proc. of the IFSA'92 (Bussels).
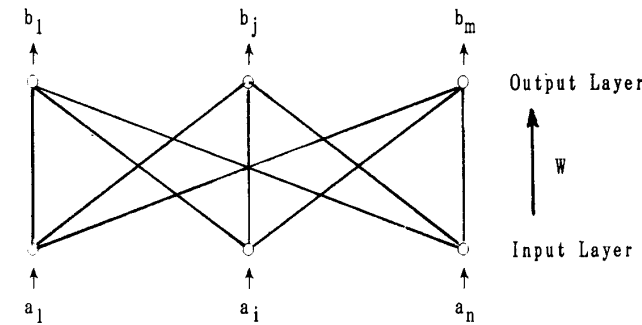
Fig. 1 The topology of a FAM

Tab. 1 Data1

k	A_k	B_k
1	(0. 7 0. 2 0. 4 0. 6 0. 3 0. 8)	(0. 6 0. 4)
2	(0. 9 0. 4 0. 5 1. 0 0. 7 0. 0)	(0. 9 0. 6)
3	(0. 3 0. 6 0. 4 0. 5 0. 2 0. 9)	(0. 5 0. 5)
4	(0. 1 0. 7 0. 4 0. 7 0. 7 0. 7)	(0. 7 0. 6)
5	(0. 3 0. 6 0. 4 0. 8 0. 2 0. 4)	(0. 8 0. 5)
6	(0. 7 0. 9 0. 6 0. 1 0. 7 0. 4)	(0. 7 0. 6)
7	(0. 8 0. 1 0. 6 0. 5 0. 4 0. 3)	(0. 6 0. 4)

Tab. 2 Data2

k	A_k	B_k
1	(0. 1 0. 9 0. 5 0. 4 0. 5 0. 6)	(0. 8 0. 5)
2	(0. 3 0. 2 0. 4 0. 3 0. 6 0. 4)	(0. 6 0. 4)
3	(0. 9 0. 2 0. 2 0. 1 0. 3 0. 0)	(0. 3 0. 6)
4	(0. 4 0. 0 0. 4 0. 7 0. 4 0. 1)	(0. 7 0. 4)
5	(0. 8 0. 2 0. 2 0. 1 0. 0 0. 2)	(0. 2 0. 6)
6	(0. 4 0. 1 0. 5 0. 3 0. 2 0. 9)	(0. 5 0. 5)
7	(0. 5 0. 2 0. 0 0. 3 0. 2 0. 6)	(0. 4 0. 5)