An Encoding Rule of Fuzzy Associative Memories*

Fan Junbo, Jin Fan and Shi Yan
Dept. of Computer Science and Engineering
Southwest Jiaotong University
Chengdu, Sichuan 610031, P. R. China

Abstract

In this paper a new encoding rule of fuzzy associative memories (FAMs) with the Max-Min composition units is presented. Under some 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 an illustrative example is given.

1 Introduction

In recent years several models that combine the strength of neural networks and fuzzy set theory have been developed and widely applied to prospective fields such as fuzzy reasoning, pattern recognition, and fuzzy control[1-4]. The FAM introduced by Kosko, which involves the 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 Hebbian rule. The inability to reliably store more than one pair is its essential drawback.

This leads us to seek for some algorithm to overcome this drawback. Due to the nondifferentiability of the node functions, gradient descent technique fails to apply. Authors of this paper ever suggested an iterative learning algorithm to encode multiple pairs in a FAM, which employs a so-called hill-climbing search strategy to conduct connection weight adjustments[5]. However, the algorithm is still a greedy method and possibly trapped at local minima of a defined cost function although satisfactory performence of the FAM is usually achieved. We propose here an alternative encoding rule that employs a suitable fuzzy logical-implication operation different from the fuzzy Hebbian rule. The proposed rule is capable of encoding multiple 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 pairs.

* This work is partly supported by National Natural Science Foundation of China

2 FAMs and Fuzzy Hebbian Rule

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

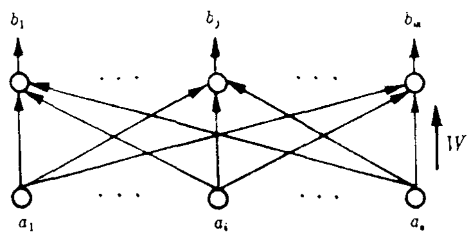

Figure 1: The topology of a FAM

In the FAM with Max-Min composition units, perfect association implies that the following equations should hold

$$A_k o W = B_k \qquad (k=1,2,\cdots,p) \tag{1}$$

where the symbol "o" denotes the Max-Min composition operation. Intuitively, the weight matrix W may be found using the equation

$$\begin{aligned} W &= \bigcap_{k=1}^{p} W_k \\ &= \bigcap_{k=1}^{p} (A_k \to B_k) \end{aligned} \tag{2}$$

where the symbol "$\to$" denotes some implication operation, and "$\bigcap$" denotes a generalized superimposition (e.g., maxima, minima etc). The intermediate matrix W_k is used to record the logical conditional(A_k, B_k).

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. In pointwise form, that is

$$\begin{aligned} w_{ij} &= \bigvee_{k=1}^{p} w_{ij}^k \\ &= \bigvee_{k=1}^{p} (a_i^k \xrightarrow{R_c} b_j^k) \\ &= \bigvee_{k=1}^{p} (a_i^k \wedge b_j^k) \end{aligned} \tag{3}$$

where the symbols "$\vee$" and "$\wedge$" stand for the maximum and minimum operations respectively, and w_{ij}s are components of the W. when an input pattern A is presented to the FAM, a pattern B is recalled using the Max-Min composition

$$b_j = \bigvee_{i=1}^{n} (a_i \wedge w_{ij}) \tag{4}$$

From equation (4), we notice that the recalled B is always a subset of the stored pattern $(\bigvee_{i=1}^{n} w_{i1}, \cdots, \bigvee_{i=1}^{n} w_{im})$ for arbitrary input pattern A. Therefore, multiple pairs can not be reliably stored in the FAM. An approach to overcoming the problem is that a bank of FAMs is employed, each of which seperately stores only one pair. But this results in consuming space of storage.

3 The Proposed Encoding 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 using the implication R_g, and we also replace the maximum superimposition by the minimum one as well. In this way, we obtain the following encoding rule

$$\begin{aligned} W &= \bigwedge_{k=1}^{p} W_k \\ &= \bigwedge_{k=1}^{p} (A_k \xrightarrow{R_g} B_k) \end{aligned} \tag{5}$$

In pointwise notation, that is

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

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

Obviously, the complexity of the proposed encoding rule is equivalent to that of the fuzzy Hebbian rule used by Kosko.

As we know, there doesn't always exist the weight matrix of a FAM, which forms the appropriate fuzzy relation for arbitrary pairs of input and output. In other words, these pairs have to satisfy certain condition for the purpose of perfect association. On the other hand, even if there exists some desired weight matrix, the fuzzy Hebbian rule is not guaranteed to yield the desired matrix, as shown (1,2). Based on analyzing the condition for existence of desired weight matrix, we show below that the proposed encoding rule can yield the desired matrix if it indeed exists.

Since the m units in the outut layer of the FAM are independent one another, we can consider only one output unit. So the equation (1) is simply rewritten as

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

We define the following sets and value

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

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

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

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

$$K_{ij} = G_{ij} \cup E_{ij} \tag{13}$$

$$m_{ij} = \begin{cases} \bigwedge\limits_{k \in G_{ij}} b_j^k & \text{if } G_{ij} \neq \Phi \\ 1 & \text{if } G_{ij} = \Phi \end{cases} \tag{14}$$

$$S_{ij} = \{k \in K_{ij} \mid b_j^k \leqslant m_{ij}\} \tag{15}$$

Then the necessary and sufficient condition for existence of the desired weight vector that enables the equation (8) to hold can be expressed as[6]

Theorem 1 There exists a desired weight vector, if and only if the following equation holds

$$\bigcup_{i \in I} S_{ij} = K \tag{16}$$

Proof If there exists some weight vector enabling the equation (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 $w_{ij} \leqslant m_{ij} < b_j^{k'}$.

Hence we have

$$b_j^{k'} = \bigvee_{i \in I} (a_i^{k'} \wedge w_{ij})$$

$$= \bigvee_{i \mid k' \notin S_{ij}} (a_i^{k'} \wedge w_{ij}) < b_j^{k'}$$

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 $w_{ij} \leqslant m_{ij} < b_j^k$ for all i as $k \in S_{ij}$; but $a_i^k = b_j^k$ and $m_{ij} \geqslant b_j^k$, 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

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

$$= \bigvee_{i \mid k \in S_{ij}} (a_i^k \wedge w_{ij})$$

$$= \bigvee_{i \mid k \in S_{ij}} (a_i^k \wedge m_{ij})$$

$$= b_j^k$$

This means that there exists a weight vector enabling the equation (8) to hold. Q. E. D.

So, there exists a weight matrix W enabling the equation (1) to hold, if and only if the equation (16) holds for any j. The proposed rule expressed by the equations (6) and (7) is in accordance with the equation (14) . From the proof of the above theorem, we can concluce that the proposed rule can yield the desired weight matrix if it exists.

4 Example

A set of fuzzy pattern pairs to be stored is shown in Table 1, where $n=6, m=2$ and $p=7$. It is easy to confirm that the equation (16) holds.

Table 1: The fuzzy pattern pairs to be stored

K	AK	BK
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)

Using the proposed rule, wd can obtain the weight matrix of the FAM

$$W = \begin{bmatrix} 0.6 & 0.4 \\ 0.5 & 0.5 \\ 1.0 & 0.4 \\ 0.9 & 0.4 \\ 1.0 & 0.6 \\ 0.5 & 0.4 \end{bmatrix} \tag{17}$$

The FAM with the above matrix can perfectly recall each output pattern B_k when presented with each input pattern A_k. But the fuzzy Hibbian Rule produces the following matrix

$$W=\begin{bmatrix} 0.9 & 0.6 \\ 0.7 & 0.6 \\ 0.6 & 0.6 \\ 0.9 & 0.6 \\ 0.7 & 0.6 \\ 0.7 & 0.6 \end{bmatrix} \tag{18}$$

With the matrix, the trained FAM can not perfectly recall these pairs.

Extensive simulation results show that the proposed rule usually produce a smaller system error than does the fuzzy Hebbian rule, even if there doesn't exist a desired weight matrix that enables the equation (1) to hold.

5 Conclusion

A new encoding rule for multiple fuzzy pattern pairs in FAMs with the Max-Min composition is presented. Under some condition, multiple pairs can be reliably encoded in a single FAM and perfect associations can be achieved. The correctness of the proposed rule is demonstrated. Because of the simple complexity ,the proposed rule may be much of use in the design of fuzzy systems where multiple 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, Volumes I-III, Prentece-Hall.

(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 Shi, Y.. A new Model of Fuzzy Associative Memories, Accepted for publication at IJCNN'92.

(5) Fan, J. B. , Jin, F. and Yuan, X.. A Learning Algorithm for Multiple Fuzzy Pattern pair Associations, Accepted for publication at IJCNN'92.

(6) Saito, T. and Mukaidono, M. (1991). A Learning Algorithm for Max-Min Network and Its Application to Solve Fuzzy Relation Equations, Proc. of the IFSA'92 Bussels.