

多模式对连接权矩阵的一种神经网络学习算法¹

杨群生 余英林

(华南理工大学电子与通信工程系 广州 510641)

摘 要 该文提出多模式对连接权矩阵的一种神经网络学习算法,并给出了严格的理论证明。该算法能够将多个模糊模式对可靠地编码存储到尽可能少的连接权矩阵中,从而大大地减少存储空间,而且容易实现,并举例验证了它的有效性。

关键词 模糊联想记忆, 连接权矩阵, 神经网络, 多模式, 学习算法

中图分类号 TN-052

1 引 言

Kosko^[1] 提出了 max-min 合成模糊联想记忆 (FAM) 网络, 首先将模糊理论和神经网络结合起来, 从而兼有两者信息处理的优点。随后许多学者如 Kosko^[2], Fan^[3], 范俊波等^[4], Pedrycz^[5], Blanco, Delgado and Requena^[6,7], 刘普寅^[8,9], 肖平^[10], Chung^[11], 李晓忠等^[12] 在 FAM 的性质及学习算法的研究领域取得了许多成果。Kosko 的 max-min 合成 FAM 模型采用模糊赫布 (Hebb) 规则学习, 它不能保证可靠地存储多个模糊模式对。近年来, FAM 已被成功地应用于函数估计、模糊控制、图像编码、模式识别等方面, 但这种模型是多数以低存储为代价, 即一个规则需要一个 FAM 连接权矩阵存储, 实现这种模型需要大量硬件和计算, 因而, 这就大大限制了它的应用。

如何克服模糊赫布规则学习的这种局限性, 文献 [4] 提出了一种有效的学习算法, 它利用最大解矩阵的方法改善了 Kosko 的算法, 并给出了一族给定的模糊模式对能完全被 max-min 神经网络所记忆的充要条件, 即定理 2。利用文献 [4] 的算法, 一个 max-min 神经网络能够存储更多的模糊模式对。文献 [8] 引进带阈值的 max-min 的神经网络对此作了进一步的推广。然而, 对于文献 [4] 所提出的定理 2 条件的验证很不容易, 而且该定理约束条件较多, 在现实应用中对于模糊模式对比较多时经常出现不符合该定理条件的情况, 如何克服这种限制, 使计算简单, 而且能以较少的连接权矩阵尽可能多地存储多个模糊模式对具有重要的意义。对这个问题, 本文提出一种不同于模糊赫布规则学习和文献 [4] 的学习算法, 文中提出了连接权矩阵的神经网络学习算法, 同时给出了严格的理论证明, 并且通过实例验证了该算法的正确性和有效性。

2 模糊联想记忆网络及其分析

Kosko 在文献 [1] 中提出的一般的 FAM 网络的数学模型为

$$A_k \circ W = B_k, \quad k = 1, 2, \dots, p \quad (1)$$

式中“ $\circ$ ”表示最大-最小合成运算, 用逐点表示为

$$\bigvee_{i=1}^n (a_{ki} \wedge w_{ij}) = b_{kj}, \quad j = 1, 2, \dots, m; \quad k = 1, 2, \dots, p \quad (2)$$

¹ 1999-04-29 收到, 1999-10-15 定稿

国家自然科学基金 (69772026) 及广东省自然科学基金资助

$A_k = (a_{k1}, a_{k2}, \dots, a_{kn}) \in [0, 1]^n$, $B_k = (b_{k1}, b_{k2}, \dots, b_{km}) \in [0, 1]^m$ 分别为 A -域和 B -域模糊模式向量; $W = (w_{ij})_{n \times m}$ 为网络的连接权矩阵, w_{ij} 为第 i 个输入神经元到第 j 个输出神经元之间连接权, 其值也在 $[0, 1]$ 范围内; n 和 m 分别为 A -域和 B -域神经元个数, p 为训练模式对数, $\vee$ 和 $\wedge$ 表示最大和最小运算。

对于这种模糊联想记忆网络, 是否存在完整的模式对联想意味着是否存在连接权矩阵 W 使得 (1) 或 (2) 式成立, 也就是能否求解模糊关系方程组 (1) 或 (2) 式。

Kosko 采用最小蕴涵算子将第 k 个模式对 (A_k, B_k) 编码到中间矩阵 W_k 中, 然后用最大叠加运算将 p 个 W_k 组合到连接权矩阵 W 中。逐点表示为

$$w_{ij} = \bigvee_{k=1}^p (a_{ki} \wedge b_{kj}), \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m \quad (3)$$

事实上, 多个模糊模式对并不能保证可靠地存储于 (3) 式所确定的连接权矩阵中, 解决这个问题的方法之一是对每个模糊模式对单独编码, 其缺点是耗费大量的存储空间, 增加硬件的复杂性。对此, 文献 [4] 提出了一个新的学习算法, 即用蕴涵 R_g 关联 A_k 和 B_k , 并用最小叠加运算组合各个 W_k , 即

$$W = \bigcap_{k=1}^p W_k = \bigcap_{k=1}^p (A_k \xrightarrow{R_g} B_k) \quad (4)$$

用逐点方式表示成

$$w_{ij} = \bigwedge_{k=1}^p w_{ij}^k = \bigwedge_{k=1}^p (a_{ki} \xrightarrow{R_g} b_{kj}), \quad i = 1, \dots, n; \quad j = 1, \dots, m, \quad (5)$$

式中

$$a_{ki} \xrightarrow{R_g} b_{kj} = \begin{cases} b_{kj}, & a_{ki} > b_{kj} \\ 1, & a_{ki} \leq b_{kj} \end{cases} \quad (6)$$

根据这种算法, 文献 [4] 中定理 2 给出了使 (1) 式成立的充要条件。对于一组给定的模糊输入输出模式对, 如果网络存在希望的连接权矩阵, 那么, 该算法必能求出一个希望的连接权矩阵。但是根据文献 [4] 中定理 2 来判断是否有解, 显得比较复杂, 而且在现实应用中, 经常遇到 (1) 式无解的情形, 在这种情况下, 企图找到一个连接权矩阵来可靠地存储多个模糊模式对是不现实的, 也是不可能的。因此, 如何应用尽可能少的连接权矩阵来可靠地存储给定的模糊模式对就显得十分有意义。

3 模糊联想记忆神经网络学习算法

3.1 定理及算法

引理 1^[2] 设 (A_k, B_k) 是第 k 个模式对, 如果 $A_k \circ W = B_k$ 有解, 那么 $H(A_k) \geq H(B_k)$ 。其中 $H(A_k) = \max_{i=1}^n a_{ki}$ 。

定理 1 设 $(A_k, B_k) (k = 1, 2, \dots, p)$ 是 p 个向量模式对, 且满足 $H(A_k) \geq H(B_k)$, 其中 $A_k = (a_{k1}, a_{k2}, \dots, a_{kn}) \in [0, 1]^n$, $B_k = (b_{k1}, b_{k2}, \dots, b_{km}) \in [0, 1]^m$, 则对于任意给定的

$k \in \{1, 2, \dots, p\}$, 模式向量 B_k 可由模式向量 A_k 一次联想出来的充分条件是对于任意的 $j \in \{1, 2, \dots, m\}$, 都至少存在一个 $i \in \{1, 2, \dots, n\}$ 使得

$$b_{kj} = a_{ki} \wedge w_{ij} \quad (7)$$

对任意 $l \neq i$ 时都有

$$b_{kj} \geq a_{kl} \wedge w_{lj}, \quad l \in \{1, 2, \dots, n\} \text{ 且 } l \neq i \quad (8)$$

证明 $\bigvee_{l=1, l \neq i}^n (a_{kl} \wedge w_{lj}) = \bigvee_{l=1}^n (a_{kl} \wedge w_{lj}) \vee (a_{ki} \wedge w_{ij}) = b_{kj}$, 对于任给 $k \in \{1, 2, \dots, p\}$

$j \in \{1, 2, \dots, m\}$ 都成立。

证毕

根据该定理, 我们提出用神经网络的方法来训练权矩阵。该网络结构简单, 容易实现。神经网络结构如图 1 所示:

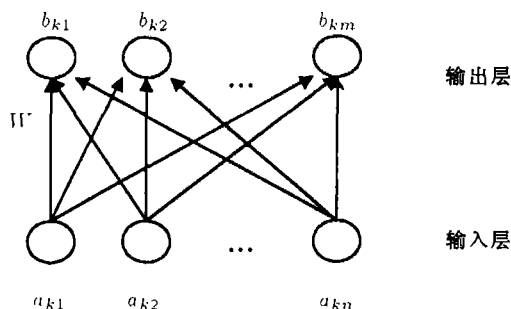


图 1 神经网络结构示意图

该算法流程如下:

(1) 初始化权值 $w_{ij} = 1, i = 1, 2, \dots, n; j = 1, 2, \dots, m;$

(2) 输入一对模糊模式对: 输入 $A_k = (a_{k1}, a_{k2}, \dots, a_{kn})$ 和输出 $B_k = (b_{k1}, b_{k2}, \dots, b_{km})$ $k = 1, 2, \dots, p;$

(3) 调节权值

$$\left. \begin{aligned} w_{ij}(t+1) &= w_{ij}(t) \wedge b_{kj}, & a_{ki} \wedge w_{ij}(t) &> b_{kj} \\ w_{ij}(t+1) &= w_{ij}(t), & \text{其它} \end{aligned} \right\} \quad (9)$$

所得权值作为下一次的初始权值;

(4) 返回第 (2) 步, 直到所有模糊模式对都输完。

显然, 这个算法反映了定理 1 的思想。一般的神经网络算法采用任意初始化权值, 我们的算法根据定理特点, 初始化权值为 1。

3.2 理论结果与证明

引理 2 如果方程

$$A \circ w^T = b \quad (10)$$

有解, 其中 $A = (a_1, a_2, \dots, a_n)$, $w = (w_1, w_2, \dots, w_n)$, w^T 为 w 的转置, b 为一个常数, 那么应用 3.1 节的算法所得到的解 $\bar{w}$ 就是该方程的最大解。

证明 (1) 若 $a_i \leq b, i \in \{1, 2, \dots, n\}$, 则 $\bar{w} = (1, 1, \dots, 1)$, 显然这是方程 (10) 的最大解。

(2) 设 $a_1, a_2, \dots, a_n$ 中存在 k 个数大于 b , $1 \leq k \leq n$, 不妨设 $a_1, a_2, \dots, a_k$ 大于 b , 则经过 3.1 节算法之后即得 $\bar{w} = (\overbrace{b, b, \dots, b}^{k \text{ 个 } b}, 1, \dots, 1)$, 显然, 这是方程 (10) 的最大解。证毕

引理 3 如果方程组

$$A \circ W = B \quad (11)$$

有解, 其中 $A = (a_1, a_2, \dots, a_n)$, $W = (w_{ij})_{n \times m}$, $i \in \{1, 2, \dots, n\}$, $j \in \{1, 2, \dots, m\}$, $B = (b_1, b_2, \dots, b_m)$, 那么应用 3.1 节的算法所得到的解矩阵 $\bar{W}$ 就是该方程组的最大解。

证明仿引理 2 即得。

由上述两个引理立即可得下面定理:

定理 2 假设方程组 (1) 式有解, 那么应用 3.1 节的算法所得到的权矩阵一定是方程组 (1) 式的最大解。

3.3 模拟结果

设

$$\begin{aligned} A_1 &= \{0.6, 0.5, 1.0, 0.5\}, B_1 = \{0.7, 0.5, 0.6\}; & A_2 &= \{0.4, 0.2, 0.7, 0.8\}, B_2 = \{0.8, 0.4, 0.5\}; \\ A_3 &= \{0.2, 0.4, 0.5, 0.6\}, B_3 = \{0.6, 0.4, 0.5\}; & A_4 &= \{0.7, 0.3, 0.9, 0.1\}, B_4 = \{0.7, 0.4, 0.6\}; \\ A_5 &= \{0.2, 0.1, 0.2, 0.3\}, B_5 = \{0.3, 0.3, 0.3\}. \end{aligned}$$

采用上述算法, 很容易得出权矩阵为

$$W = \begin{pmatrix} 1.0 & 0.4 & 0.6 \\ 1.0 & 1.0 & 1.0 \\ 0.7 & 0.4 & 0.5 \\ 1.0 & 0.4 & 0.5 \end{pmatrix}$$

容易验证 W 是 (1) 式的解。

必须注意到, 不管 (1) 式是否有解, 采用 3.1 节的算法都能收敛到某个权矩阵, 因此这种算法所得的结果并不一定是 (1) 式的解, 因为在 (1) 中方程的个数较多时, (1) 式很可能没有解。在方程组 (1) 式没有解的情况下, 如何以尽可能少的权矩阵来存储尽可能多的模糊模式对就变得十分有意义, 下面我们就提出一种解决这个问题的有效算法。

4 多模式对联想记忆学习算法

为了解决上面所提出的问题, 我们提出下面的算法, 算法流程如下:

- (1) 初始化权矩阵 $w_{ij} = 1$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$;
- (2) 输入一对模糊模式对: 输入 $A_k = (a_{k1}, a_{k2}, \dots, a_{kn})$ 和输出 $B_k = (b_{k1}, b_{k2}, \dots, b_{km})$, $k = 1, 2, \dots, p$;
- (3) 调节权值

$$\left. \begin{aligned} w_{ij}(t+1) &= w_{ij}(t) \wedge b_{kj}, & a_{ki} \wedge w_{ij}(t) &> b_{kj} \\ w_{ij}(t+1) &= w_{ij}(t), & \text{其它} \end{aligned} \right\} \quad (12)$$

检验所得权矩阵是否是前面所输模式对的解。

不妨设已输入 l 对模式 $(A_1, B_1), (A_2, B_2), \dots, (A_l, B_l)$, 最近输入模式对为 (A_l, B_l) , 而此前所得到的权矩阵是前 $l-1$ 对模式对的解, 令

$$b'_{kj} = \bigvee_{i=1}^n (a_{ki} \wedge w_{ij}), \quad k = 1, 2, \dots, l, \quad j = 1, 2, \dots, m \quad (13)$$

若 $b'_{kj} = b_{kj}$, $k = 1, 2, \dots, l, j = 1, 2, \dots, m$, 则将所得权值作为下一次的初始权值; 若存在某个 $b'_{kj} \neq b_{kj}$, $k \in \{1, 2, \dots, l\}, j \in \{1, 2, \dots, m\}$, 则将这次权值取消, 将上次权值作为下次输入时的初始值, 并将 (A_l, B_l) 作为下一轮的输入模式对;

(4) 继续输入下一模式对, 返回 (2), 直到这一轮输完;

(5) 下一轮, 重复 (1), 直到所有模式对都找到相应的连接权矩阵。

这个算法很容易地解决了上一节提出的问题。

5 实 例

下面我们通过一个例子来说明这个算法的有效性。

$A_1 = \{0.6, 0.5, 1.0, 0.5\}, B_1 = \{0.7, 0.5, 0.6\}; A_2 = \{0.4, 0.3, 0.9, 0.8\}, B_2 = \{0.8, 0.4, 0.5\};$
 $A_3 = \{0.4, 0.7, 0.2, 0.4\}, B_3 = \{0.5, 0.5, 0.2\}; A_4 = \{0.3, 0.4, 0.5, 0.6\}, B_4 = \{0.6, 0.4, 0.5\};$
 $A_5 = \{0.7, 0.2, 1.0, 0.1\}, B_5 = \{0.7, 0.4, 0.6\}; A_6 = \{0.9, 0.5, 0.7, 0.7\}, B_6 = \{0.8, 0.4, 0.7\};$
 $A_7 = \{0.6, 0.4, 0.7, 0.2\}, B_7 = \{0.4, 0.7, 0.7\}; A_8 = \{0.7, 0.7, 0.3, 0.6\}, B_8 = \{0.7, 0.4, 0.6\};$
 $A_9 = \{0.8, 0.9, 0.3, 0.2\}, B_9 = \{0.9, 0.4, 0.6\}; A_{10} = \{0.2, 0.1, 0.2, 0.3\}, B_{10} = \{0.3, 0.3, 0.3\}.$

若按每对模式单独编码, 则需要 10 个权矩阵来存储它们, 根据我们的算法, 很容易得出可靠地存储这 10 对模糊模式对只需要下面 3 个权矩阵, 而且不会对判断哪些模糊模式对组成的方程组是否有解感到为难。

存储上面 10 对模糊模式对所需的权矩阵为

$$W_1 = \begin{pmatrix} 1.0 & 0.4 & 0.6 \\ 1.0 & 1.0 & 1.0 \\ 0.7 & 0.4 & 0.5 \\ 1.0 & 0.4 & 0.5 \end{pmatrix}, W_2 = \begin{pmatrix} 0.4 & 1.0 & 0.2 \\ 0.5 & 0.5 & 0.2 \\ 0.4 & 1.0 & 1.0 \\ 1.0 & 1.0 & 0.2 \end{pmatrix}, W_3 = \begin{pmatrix} 0.8 & 0.4 & 0.6 \\ 1.0 & 0.4 & 0.6 \\ 1.0 & 0.4 & 1.0 \\ 1.0 & 0.4 & 1.0 \end{pmatrix}$$

其中 W_1 可存储模式对 $(A_1, B_1), (A_2, B_2), (A_4, B_4), (A_5, B_5), (A_{10}, B_{10})$; W_2 可存储模式对 $(A_3, B_3), (A_7, B_7)$; W_3 可存储模式对 $(A_6, B_6), (A_8, B_8), (A_9, B_9), (A_{10}, B_{10})$ 。其中 (A_{10}, B_{10}) 同时可被二个权矩阵存储, 说明有些模糊模式对可以同时通过不同权矩阵来编码存储。

6 结 论

本文提出了使用神经网络的方法求解模糊联想记忆连接权矩阵的算法, 对于给定的模糊输入输出模式对, 若这些模式对存在连接权矩阵, 则该算法很容易求出它们的最大权矩阵, 如果这些模式对不能够用一个连接权矩阵来存储, 应用我们的算法, 可以应用尽可能少的连接权矩阵来存储, 并且容易求出这些权矩阵。这种算法简单, 容易实现, 可以有效地减少存储空间, 从而大大减少硬件的复杂性。该算法在有关多规则模糊系统的设计和模式识别等方面应该是十分有用的。

参 考 文 献

- [1] B. Kosko, Fuzzy Associative Memory Systems, Fuzzy Expert Systems Reading, MA, Addison-Weley, 1987.
- [2] B. Kosko, Neural Networks and Fuzzy Systems, New Jersey, Prentice-Hall, Inc., 1992, 299-339.
- [3] J. b. Fan, F. Jin F, Y. Shi, A class of fuzzy associative memories, Chinese Journal of Electronics, 1993, 2(1): 26-30.
- [4] 范俊波, 靳蕃, 史燕, 模糊联想记忆网络的一种有效学习算法, 电子学报, 1996, 24(1), 112-114.
- [5] W. Pedrycz, Processing in relational structures, Fuzzy relational equations, Fuzzy Sets and Systems, 1991, 40, 77-106.
- [6] A. Blanco, M. Delgado, I. Requena, Identification of relational equations by fuzzy neural networks, Fuzzy Sets and Systems, 1995, 71, 215-226.
- [7] A. Blanco, M. Delgado, I. Requena, Improved fuzzy neural networks for solving relational equations, Fuzzy Sets and Systems, 1995, 72, 311-322.
- [8] 刘普寅, 基于一类模运算的模糊神经网络的模糊联想记忆, 系统工程与电子技术, 1997, 11, 54-58.
- [9] 刘普寅, 马超群, 张汉江, 一类广义模糊双向联想记忆的稳定性, 模糊系统与数学, 1998, 12(4), 40-46.
- [10] 肖平, 最大-乘积型模糊联想记忆网络的最大最小编码学习算法, 通信学报, 1999, 20(1), 7-22.
- [11] F. L. Chung, T. Lee, On fuzzy associative memory with multiple-rule storage capacity, IEEE Trans. on Fuzzy Systems, 1996, 4, 375-384.
- [12] 李晓忠, 汪培庄, 罗承忠, 模糊神经网络, 贵阳, 贵州科技出版社, 1994, 211-232.

A KIND OF NEURAL NETWORKS LEARNING ALGORITHMS
FOR MULTIPLE-PATTERN PAIRS WEIGHTED MATRIX

Yang Qunsheng Yu Yinglin

(Dept. of Electron. and Comm. Eng., South China Univ. of Tech., Guangzhou 510641, China)

Abstract A kind of neural networks learning algorithms for multiple-pattern pairs weighted matrix of fuzzy associative memories(FAMs) and its strict theoretic proofs are presented in this paper. Multiple fuzzy pattern pairs can be encoded to store in FAM connection weight matrixes as few as possible by the algorithm, so it can cut down storage space greatly and this algorithm can easily be implimented. Its effectiveness is testified by an example.

Key words Fuzzy associative memory, Weighted matrix, Neural network, Multiple-pattern, Learning algorithm

杨群生: 男, 1965 年生, 博士生, 从事神经网络、模糊逻辑、图像处理方面的研究.

余英林: 男, 1932 年生, 教授, 博士生导师, 从事神经网络和图像处理等方面的研究.