

神经网络在故障诊断中的双向联想记忆法

谭民 疏松桂

(中国科学院自动化研究所)

摘 要

本文介绍了神经网络用于控制系统故障诊断的双向联想记忆法。该方法是根据神经网络的特点和功能以及控制系统故障诊断的要求而提出。文中首先介绍了双向联想记忆模型和算法,然后给出一个简单的应用例子。

关键词: 神经网络,双向联想记忆法,故障诊断,贴近度。

一、引 言

人工神经网络是大规模的分布式并行处理系统,它具有自组织性、自学习性。人脑善于概括、类比和推广,它可以将各种知识结合起来,灵活运用。比如,在设备的故障诊断中,有经验的修理工根据以往故障的模式,通过概括、类比和推广,能很快找到设备的故障,这就是靠人的联想记忆。

联想记忆在神经网络的学习算法中是很重要的^[1]。双向联想记忆(Bidirectional Associative Memory)是它的一种,通过双向联想记忆矩阵,把事先概括好的样本数据对存贮起来。当有新的信息输入时,双向联想记忆(BAM)通过前向信息流、后向信息流,对输入信息进行处理,当BAM达到稳定时,可以得到满意的输出。

二、双向联想记忆算法

1. 算法及双向稳定性

双向联想记忆是一种两层的反馈型神经网络,如图1所示。输入层 F_A 包括 n 个神经元 (a_1, a_2, \dots, a_n) , $a_i = 1$ 表示第 i 个神经元兴奋, $a_i = 0$ 表示第 i 个神经元抑制, $F_A = \{0, 1\}^n$; 输出层 F_B 包括 m 个神经元 $\{b_1, b_2, \dots, b_m\}$, $b_j = 1$ 表示第 j 个神经元兴奋, $b_j = 0$ 表示第 j 个神经元抑制, $F_B = \{0, 1\}^m$ 。联想记忆就是一个向量空间的变换 $M: \mathbb{R}^n \rightarrow \mathbb{R}^m$, 假若映射是线性的,那么输入向量 A 时,经过变换,输出向量 $B = AM$ 。

这样, BAM 就是二值乘积空间 $F_A \times F_B$ 上的一个点 (A, B) 。

把样本数据对用矩阵的方式存贮如下:

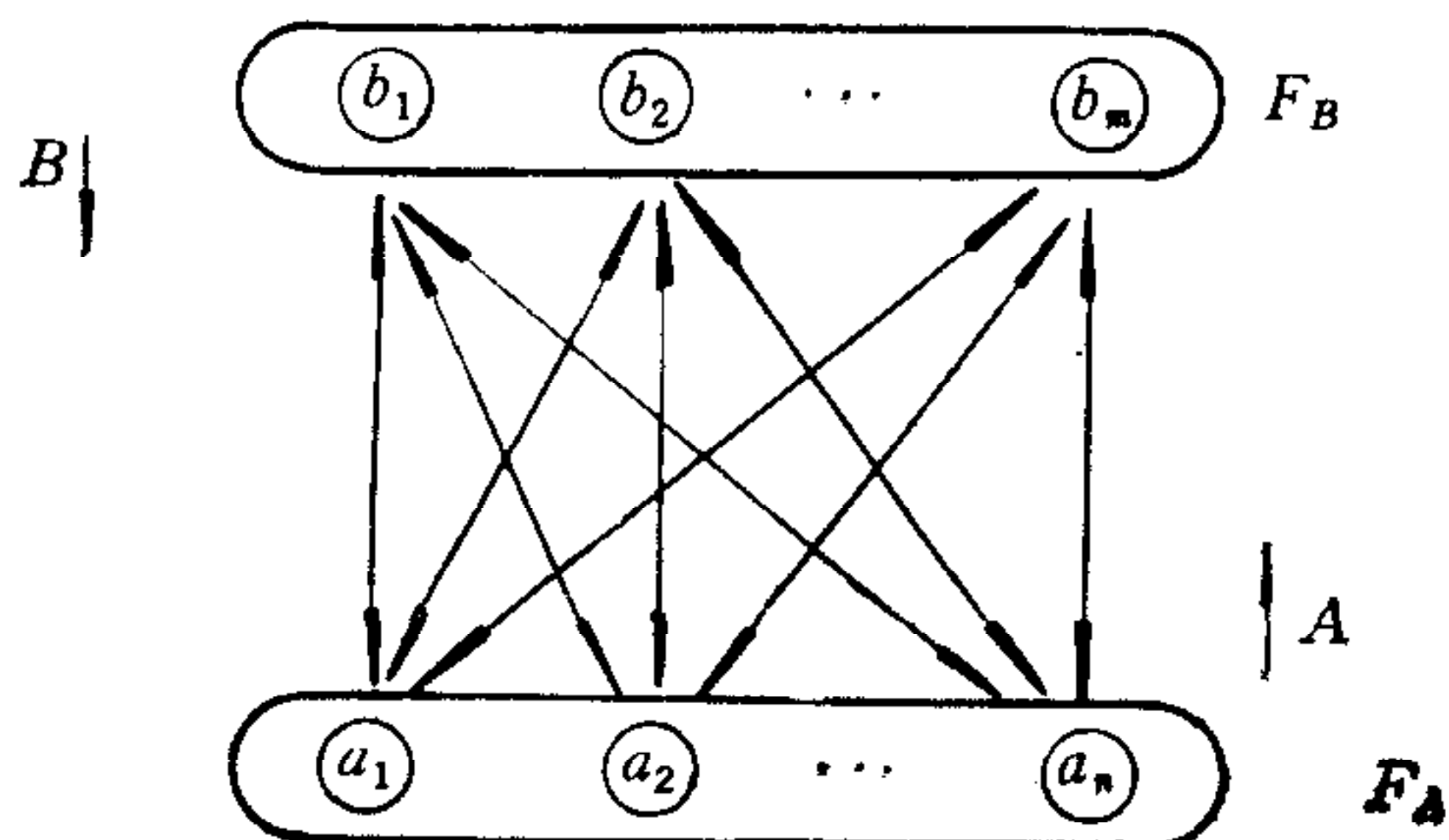


图 1 双向联想记忆结构图

$$M = \sum_{i=1}^p M_i = \sum_{i=1}^p A_i^T \cdot B_i = A_1^T B_1 + A_2^T B_2 + \cdots + A_p^T B_p. \quad (1)$$

如果输入 A_1, A_2, \cdots, A_p 是正交的, 即

$$A_i \cdot A_j^T = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

那么, $A_i M = A_i A_i^T B_i + \sum_{j \neq i} (A_i A_j^T) B_j = B_i$.

为了提高联想记忆的精度, 可以把输出得到的 B , 反馈回到 BAM 中得到 A' , 再把 A' 送入 BAM 中得到 B' , B' 再反馈得到 $A'' \cdots \cdots$ 重复进行, 最后会收敛到 $(A_j, B_j)^{[2]}$.

$$\begin{aligned} A &\rightarrow M \rightarrow B, \\ A' &\leftarrow M^T \leftarrow B, \\ A' &\rightarrow M \rightarrow B', \\ A'' &\leftarrow M^T \leftarrow B', \\ &\cdots \cdots \\ A_j &\rightarrow M \rightarrow B_j, \\ A_j &\leftarrow M^T \leftarrow B_j. \end{aligned}$$

另外, b_j 和 a_i 处的状态规定为

$$b_j = \begin{cases} 1, & AM_j > 0, \\ 0, & AM_j < 0, \end{cases} \quad a_i = \begin{cases} 1, & BM_i^T > 0, \\ 0, & BM_i^T < 0. \end{cases} \quad (2)$$

若 BAM 对每一输入对 (A, B) 都收敛, 那么 M 就是双向稳定的^[3].

2. 贴近度

假若已经存好了两个样本 $A_1 = (1, 1, 0, 0)$ 和 $A_2 = (1, 0, 1, 1)$, 当一个输入 $A = (1, 0, 0, 0)$ 时, 它是靠近 A_1 还是 A_2 呢? 为了解决这个问题, 规定一个 A_i 与 A_j 的贴近度 $\rho(A_i, A_j)$.

令 $\rho(A_i, A_j)$ 是 l^1 空间上的一个度量

$$\rho(A_i, A_j) = \|A_i - A_j\| = \sum_{k=1}^n |a_{ik} - a_{jk}|. \quad (3)$$

ρ 越小表示 A_i 与 A_j 越贴近, ρ 越大表示 A_i 与 A_j 的差异越大. 当 $\rho = 0$ 时, A_i 与 A_j 完全贴近, 即 $A_i = A_j$. 当 $\rho = n$ 时, A_i 与 A_j 差异最大, $A_i^c = A_j$ (A_i^c 是 A_i 的补集).

上面的情况 $\rho(A, A_1) = 1 < 2 = \rho(A, A_2)$, 说明 A 更接近 A_1 .

3. 算法具体应用中的一些变化

因为 A_i 和 B_i 均为二值向量, $A_i = (a_{i1}, a_{i2}, \dots, a_{in})$ $a_{ij} \in \{0, 1\}^n, (j = 1, 2, \dots, n)$, 所以 AM_i 和 BM_i^T 永远不会为负值, 状态转移规则(2)式就有 $a_i = b_i = 1$, M 阵将不包含任何抑制的信息, 这就不能正确地工作. 因此, 用两极状态向量代替二值状态向量, 以解决这个问题. 二值向量中的零由 -1 代替, 组成两极对 (X_i, Y_i) .

$$X_i = 2A_i - I, Y_i = 2B_i - I, (I \text{ 为单位向量}).$$

那么 $M = X_1^T Y_1 + X_2^T Y_2 + \dots + X_p^T Y_p$.

三、算法在控制系统故障诊断中的应用

首先分析一个生产过程, 如图 2. 系统中每个部分出现故障, 都将造成整个系统的故障, 或者使系统异常.

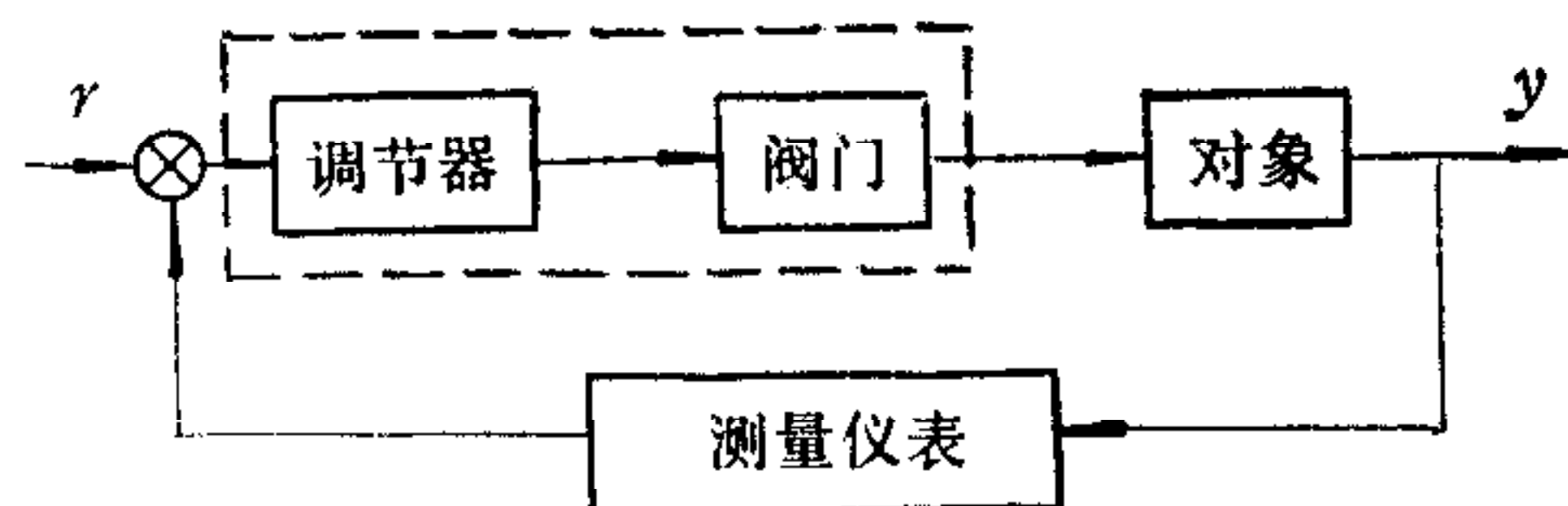


图 2 生产过程控制系统

下面为了方便, 仅讨论调节器与调节阀两个部分故障的情形, 即图 2 中的虚线部分.

它们出现故障的形式有调节器误指令, 调节器不工作, 阀门误操作, 阀门不工作.

采用两层神经网络, 输入层有四个输入神经元, 令 $A_i = (a_{i1}, a_{i2}, a_{i3}, a_{i4})$ 为输入向量, 表示输入的故障现象; 输出层也有四个神经元, 令 $B_i = (b_{i1}, b_{i2}, b_{i3}, b_{i4})$ 为输出向量, 表示解决故障的办法.

四个输入神经元的定义如下:

$a_{i1} = 1$ 是特征量位, 表示调节器故障;

$a_{i2} = 1$ 表示误指令或误操作;

$a_{i3} = 1$ 表示不工作;

$a_{i4} = 1$ 是阀门故障的特征量位.

根据这个规定, 先存入一些样本, 令 $A_1 = (1, 1, 0, 0)$ 表示调节器误指令, $B_1 = (1, 0, 0, 0)$ 表示检查调节器的传动机构, $A_1 \rightarrow B_1$; $A_2 = (1, 0, 1, 0)$ 表示调节器不工作, $B_2 = (0, 1, 0, 0)$ 表示检修整个调节器, $A_2 \rightarrow B_2$; $A_3 = (0, 1, 0, 1)$ 表示阀门误操作, $B_3 = (0, 0, 1, 0)$ 表示检查阀门的传动机构, $A_3 \rightarrow B_3$; $A_4 = (0, 0, 1, 1)$ 表示阀门不工作, $B_4 = (0, 0, 0, 1)$ 表示检修阀门, $A_4 \rightarrow B_4$. 这样, $A_i \rightarrow B_i$ 的映射就是故障信息到解决办法的对应关系.

把二元对化为二极对

$$\begin{aligned} X_1 &= (1, 1, -1, -1) \rightarrow Y_1 = (1, -1, -1, -1), \\ X_2 &= (1, -1, 1, -1) \rightarrow Y_2 = (-1, 1, -1, -1), \\ X_3 &= (-1, 1, -1, 1) \rightarrow Y_3 = (-1, -1, 1, -1), \\ X_4 &= (-1, -1, 1, 1) \rightarrow Y_4 = (-1, -1, -1, 1). \end{aligned}$$

那么它的双向联想记忆矩阵 M 为

$$M = \sum_{i=1}^4 X_i^T Y_i = \begin{bmatrix} 2 & 2 & -2 & -2 \\ 2 & -2 & 2 & -2 \\ -2 & 2 & -2 & 2 \\ -2 & -2 & 2 & 2 \end{bmatrix}.$$

所以

$$\begin{aligned} A_1 M &= (4, 0, 0, -4) \rightarrow (1, 0, 0, 0) = B_1, \\ A_2 M &= (0, 4, -4, 0) \rightarrow (0, 1, 0, 0) = B_2, \\ A_3 M &= (0, -4, 4, 0) \rightarrow (0, 0, 1, 0) = B_3, \\ A_4 M &= (-4, 0, 0, 4) \rightarrow (0, 0, 0, 1) = B_4. \end{aligned}$$

同样, 后向信息流有

$$\begin{aligned} B_1 M^T &= (2, 2, -2, -2) \rightarrow (1, 1, 0, 0) = A_1, \\ B_2 M^T &= (2, -2, 2, -2) \rightarrow (1, 0, 1, 0) = A_2, \\ B_3 M^T &= (-2, 2, -2, 2) \rightarrow (0, 1, 0, 1) = A_3, \\ B_4 M^T &= (-2, -2, 2, 2) \rightarrow (0, 0, 1, 1) = A_4. \end{aligned}$$

这就说明存入的 A_i 与 B_i 是一一对应的, 当输入 A 与 A_i 中的任一个相同时, BAM 很快就会找到相应的 B_i , 得到解决办法. 当输入 A 与 A_i 中的任一个都不相同时, BAM 同样会给出满意的结果.

当输入 $A = (1, 0, 0, 0)$ 时, 按上面的规则, 只知道这是调节器故障, 但究竟是哪一类还不清楚, 需要计算它的贴进度

$$\rho(A, A_1) = \rho(A, A_2) = 1, \quad \rho(A, A_3) = \rho(A, A_4) = 3.$$

这说明输入的信息 A 接近于 A_1 和 A_2 , 这正好跟人们的直观理解是一致的. 事实上,

$$\begin{aligned} AM &= (2, 2, -2, -2) \rightarrow (1, 1, 0, 0) = B = B_1 + B_2, \\ BM^T &= (4, 0, 0, -4) \rightarrow (1, 0, 0, 0) = A. \end{aligned}$$

收敛到 $B = (1, 1, 0, 0)$, 说明解决办法 B_1 和 B_2 都要考虑, 这为进行正确的故障诊断提供了依据.

上面仅仅用一个简单的例子说明了这种诊断的方法, 其实对一个实际的系统来说, 还要考虑很多问题, 特别是两组对应样本所组成的数据对的选择.

笔者把这个诊断方法用在一个“锅炉给水控制”的装置上. 这个装置共有 20 个故障模式, 对每个模式找到一种解决的办法, 组成了 20 个样本数据对, 计算出它的双向联想记忆矩阵 M , 形成了一个故障诊断的双向联想记忆的神经元网络. 该算法在 IBM-PC/XT 上进行了仿真, 得到了满意的结果.

四、结 束 语

控制系统的故障诊断是自动控制发展的一个新领域。本文用神经网络双向联想记忆的算法,进行故障诊断,首先是根据各种故障的模式,找到解决各类故障的办法,组成数据对,以 $M = A^T B$ 的方式存贮起来。当遇到新的信息输入时,它可以找到适当的解决办法。把神经网络的方法用于控制系统的故障诊断是一个崭新的课题^[4], 这项研究一定会深入发展。

参 考 文 献

- [1] 张承福,神经网络系统,力学进展,(1988),No. 2.
- [2] Kosko, B., Adaptive Bidirectional Associative Memory, *Applied Optics*, Dec. 1987.
- [3] Kosko, B., Fuzzy Associative Memory, *Fuzzy Expert System*, Addison Wesley Reading MA, 1987.
- [4] Bavarian, B., Introduction To Neural Networks For Intelligent Control, *IEEE Magazine Control System*, Apr. 1988.

THE NEURAL NETWORKS BIDIRECTIONAL ASSOCIATIVE MEMORY USED IN FAULT DIAGNOSIS

TAN MIN SHU SONGGUI

(Institute of Automation, Academia Sinica)

ABSTRACT

In this paper, the bidirectional associative memory algorithm and its application to fault diagnosis are presented. First the algorithm of BAM and its bidirectional stable is introduced, then an example is given to illustrate the process.

Key words: Neural network; bidirectional associative memory; fault diagnosis.