第 17 卷 第 4 期　　自 动 化 学 报　　Vol. 17, No. 4
1991 年 7 月　　ACTA AUTOMATICA SINICA　　July, 1991

递归联想记忆及在故障诊断中的应用

谭 民　疏松桂
（中国科学院自动化所）

摘　要

本文介绍了神经元网络递归联想记忆模型，并应用于控制系统的故障诊断中。

关键词： 递归联想记忆，广义逆，神经元网络，故障诊断。

一、引　言

联想记忆是神经元网络模拟人脑的一种重要的记忆方式，它具有容错性及抗干扰性[1]。

联想记忆的过程分为两个阶段：第一阶段是样本的存贮。首先把实际问题进行抽象、概括、总结出样本数据向量，设 $\boldsymbol{x}^{(p)}=(x_1^{(p)},x_2^{(p)},\cdots,x_n^{(p)})^T$ 为样本输入向量，$\boldsymbol{y}^{(p)}=(y_1^{(p)},y_2^{(p)},\cdots,y_m^{(p)})^T$ 为样本输出向量，$p=1,2,\cdots,l$ 且 $\boldsymbol{x}^{(p)}\in\mathbb{R}^n$，$\boldsymbol{y}^{(p)}\in\mathbb{R}^m$，把 $(\boldsymbol{x}^{(p)},\boldsymbol{y}^{(p)})$ 称为样本向量对；然后把 $(\boldsymbol{x}^{(p)},\boldsymbol{y}^{(p)})$ 以一定的方式存贮到联想矩阵M中[2]。第二阶段是联想回忆。当输入一信息 $\boldsymbol{x}^{(p)}$ 时，经过联想矩阵M，得到 $\boldsymbol{y}^{(p)}$，即 $\boldsymbol{y}^{(p)}=M\boldsymbol{x}^{(p)}$。

递归联想记忆模型是在已经存贮$K-1$对样本向量之后，对增加的新样本采用递归方式，获得K对样本向量的联想记忆矩阵。

把这种方法应用于控制系统的故障诊断，$\boldsymbol{x}^{(p)}$ 存入故障的信息，$\boldsymbol{y}^{(p)}$ 存入解决故障的办法，通过递归得到联想记忆阵 M。当一个故障信息输入时，即可进行诊断。

二、递归联想记忆模型

1. 样本的存贮方式

对样本数据对 $\boldsymbol{x}^{(p)}\rightarrow\boldsymbol{y}^{(p)}$，经过联想矩阵$M$的作用，得到 $\boldsymbol{y}^{(p)}=M\boldsymbol{x}^{(p)}$，$p=1,2,\cdots,l$，对所有的样本可得

$$Y=MX. \tag{1}$$

其中X是 $n\times l$ 阵，Y是$m\times l$ 阵，M是$m\times n$ 阵。

若X是 $n\times n$ 的满秩阵，X^{-1} 存在，那么可以得到 $M=YX^{-1}$。但是实际问题中X经常是退化的或者不是方阵，X^{-1} 一般是不存在的。为此，引入广义逆的概念，设A是秩

本文于 1989 年 12 月 20 日收到。

为 r 的一个 $m\times n$ 阵，A 的一个广义逆是一个 $n\times m$ 阵 G，使得 $X=GY$ 是方程 $AX=Y$ 的一个解，令 $G=A^+$，A^+ 表示 A 的广义逆.

广义逆有一个重要的性质. 设 A 是 $m\times n$ 的矩阵，B 是 $m\times r$ 阵，C 是 $r\times n$ 阵，若 $A=BC$，那么广义逆

$$A^+=C^T(CC^T)^{-1}\cdot(B^TB)^{-1}B^T,$$

其中 T 表示转置. 特别地当 A 为列满秩时，

$$A^+=(A^TA)^{-1}A^T.$$

对于(1)式，方程的解为

$$M=YX^+. \tag{2}$$

可以证明在最小均方差（MSE）准则的条件下，式(2)中 M 是式(1)的最优解，即当 $\boldsymbol{x}^{(p)}$ 输入时，$\hat{\boldsymbol{y}}^{(p)}=M\boldsymbol{x}^{(p)}$ 在 MSE 准则下最接近 $\boldsymbol{y}^{(p)}$[3].

2. 递归联想记忆模型

设 $Y_k=(\boldsymbol{y}^{(1)},\boldsymbol{y}^{(2)},\cdots,\boldsymbol{y}^{(k)})$ 是 $m\times k$ 阵，$\boldsymbol{y}^{(i)}$ 是列向量，表示第 i 个输出样本，$X_k=(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(k)})$ 是 $n\times k$ 阵. 假若对于已经存好 $k-1$ 个样本向量，得到了 $M_{k-1}=Y_{k-1}X_{k-1}^+$，当增加一个新样本时，可以重新计算 $M_k=Y_kX_k^+$，但这样做是很复杂的，而且大量的是重复性工作，所以采用递归方式从 M_{k-1} 得到 M_k.

对于 $X_k=(X_{k-1}|\boldsymbol{x}^{(k)})$，它的广义逆的形式一定为[4]

$$X_k^+=\begin{pmatrix}Z_k\\ \hline \boldsymbol{b}_k\end{pmatrix}\begin{matrix}(k-1)\times n\\ 1\times n\end{matrix}.$$

计算 $X_kX_k^+$ 并利用广义逆的性质 $X_{k-1}^+X_kX_k^+=X_{k-1}^+$，可以得到 $Z_k=X_{k-1}^+-X_{k-1}^+\boldsymbol{x}^{(k)}\boldsymbol{b}_k$，所以

$$X_k^+=\begin{pmatrix}X_{k-1}^+-X_{k-1}^+\boldsymbol{x}^{(k)}\boldsymbol{b}_k\\ \hline \boldsymbol{b}_k\end{pmatrix}. \tag{3}$$

那么

$$\begin{aligned}M_k&=Y_kX_k^+=(Y_{k-1}|\boldsymbol{y}^{(k)})\begin{pmatrix}X_{k-1}^+-X_{k-1}^+\boldsymbol{x}^{(k)}\boldsymbol{b}_k\\ \hline \boldsymbol{b}_k\end{pmatrix}\\&=Y_{k-1}X_{k-1}^+-Y_{k-1}X_{k-1}^+\boldsymbol{x}^{(k)}\boldsymbol{b}_k+\boldsymbol{y}^{(k)}\boldsymbol{b}_k\\&=M_{k-1}-M_{k-1}\boldsymbol{x}^{(k)}\boldsymbol{b}_k+\boldsymbol{y}^{(k)}\boldsymbol{b}_k.\end{aligned} \tag{4}$$

其中

$$\boldsymbol{b}_k=\begin{cases}(\boldsymbol{c}_k^T\boldsymbol{c}_k)^{-1}\boldsymbol{c}_k^T, & 若\ \boldsymbol{c}_k\neq 0;\\ (1+\boldsymbol{d}_k^T\boldsymbol{d}_k)^{-1}\boldsymbol{d}_k^TX_{k-1}^+, & 若\ \boldsymbol{c}_k=0.\end{cases}$$

把它合并为

$$\boldsymbol{b}_k=\boldsymbol{c}_k^++(1-\boldsymbol{c}_k^+\boldsymbol{c}_k)(1+\boldsymbol{d}_k^T\boldsymbol{d}_k)^{-1}\boldsymbol{d}_k^TX_{k-1}^+. \tag{5}$$

其中

$$\boldsymbol{c}_k=\boldsymbol{x}^{(k)}-X_{k-1}\boldsymbol{d}_k, \tag{6}$$

$$\boldsymbol{d}_k=X_{k-1}^+\boldsymbol{x}^{(k)}. \tag{7}$$

由式(3)—式(7)就构成了联想矩阵 $M_k(k=2,3,\cdots,l)$ 的递归算式. 对于初始值，在存贮第一对样本向量 $(\boldsymbol{x}^{(1)},\boldsymbol{y}^{(1)})$ 时，

$$X_1^+ = (\boldsymbol{x}^{(1)T}\boldsymbol{x}^{(1)})^{-1}\boldsymbol{x}^{(1)T},$$
$$M_1 = \boldsymbol{y}^{(1)}(\boldsymbol{x}^{(1)T}\boldsymbol{x}^{(1)})^{-1}\boldsymbol{x}^{(1)T}.$$

由此就可以进行递归联想存贮,至全部样本向量存完为止,得到联想矩阵 M_l.

3. 数值例子

设有两组样本 $\boldsymbol{x}^{(1)} = (1, 1, 1, 0)^T$ 表示一种故障现象, $\boldsymbol{y}^{(1)} = (1, 0, 0, 0)^T$ 表示解决 $\boldsymbol{x}^{(1)}$ 的策略, $\boldsymbol{x}^{(1)} \to \boldsymbol{y}^{(1)}$; $\boldsymbol{x}^{(2)} = (0, 0, 1, 1)^T$ 表示另一种故障, $\boldsymbol{y}^{(2)} = (0, 1, 0, 0)^T$ 表示解决 $\boldsymbol{x}^{(2)}$ 的办法, $\boldsymbol{x}^{(2)} \to \boldsymbol{y}^{(2)}$. 用递归算法把这两对样本向量存贮起来.

第一对样本 $(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)})$

$$\boldsymbol{x}_1^+ = (\boldsymbol{x}^{(1)T}\boldsymbol{x}^{(1)})^{-1}\boldsymbol{x}^{(1)T} = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0\right);$$

$$M_1 = \boldsymbol{y}_1\boldsymbol{x}_1^+ = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

增加新样本 $(\boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)})$

$$d_2 = \boldsymbol{x}_1^+\boldsymbol{x}^{(2)} = \frac{1}{3}(1, 1, 1, 0)(0, 0, 1, 1)^T = \frac{1}{3};$$

$$\boldsymbol{c}_2 = \boldsymbol{x}^{(2)} - \boldsymbol{x}_1 d_2 = \left(-\frac{1}{3}, -\frac{1}{3}, \frac{2}{3}, 1\right)^T;$$

$$\boldsymbol{b}_2 = \boldsymbol{c}_2^+ = (\boldsymbol{c}_2^T\boldsymbol{c}_2)^{-1}\boldsymbol{c}_2^T = \left(-\frac{1}{5}, -\frac{1}{5}, \frac{2}{5}, \frac{3}{5}\right);$$

$$X_2^+ = \left(\frac{\boldsymbol{x}_1^+ - d_2\boldsymbol{b}_2}{\boldsymbol{b}_2}\right) = \begin{pmatrix} \frac{2}{5} & \frac{2}{5} & \frac{1}{5} & -\frac{1}{5} \\ -\frac{1}{5} & -\frac{1}{5} & \frac{2}{5} & \frac{3}{5} \end{pmatrix};$$

$$M_2 = \begin{pmatrix} \frac{2}{5} & \frac{2}{5} & \frac{1}{5} & -\frac{1}{5} \\ -\frac{1}{5} & -\frac{1}{5} & \frac{2}{5} & \frac{3}{5} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

这样得到了递归联想记忆矩阵.

当原存贮的样本 $\boldsymbol{x}^{(1)} = (1, 1, 1, 0)^T$ 输入时,

$$\hat{\boldsymbol{y}}^{(1)} = M_2\boldsymbol{x}^{(1)} = (1, 0, 0, 0)^T = \boldsymbol{y}^{(1)}.$$

当原存贮的样本 $\boldsymbol{x}^{(2)} = (0, 0, 1, 1)^T$ 输入时,

$$\hat{\boldsymbol{y}}^{(2)} = M_2\boldsymbol{x}^{(2)} = (0, 1, 0, 0)^T = \boldsymbol{y}^{(2)}.$$

假若一个新的信息 $\boldsymbol{x} = (1, 0, 1, 0)^T$ (它可以看作是样本 $\boldsymbol{x}^{(1)}$ 第二项信息丢失后

的向量)作为输入，

$$\hat{\boldsymbol{y}} = M_2\boldsymbol{x} = \left(\frac{3}{5}, \frac{1}{5}, 0, 0\right)^T$$

$$= \frac{3}{5}\boldsymbol{y}^{(1)} + \frac{1}{5}\boldsymbol{y}^{(2)}.$$

$\hat{\boldsymbol{y}}$ 是输出样本 $\boldsymbol{y}^{(1)}$ 和 $\boldsymbol{y}^{(2)}$ 的线性组合,但权重倾向于 $\boldsymbol{y}^{(1)}$ 一边.

三、递归联想记忆在故障诊断中的应用

上面的简单数值例子解释了递归联想记忆算法的过程及应用. 应当说明在实际控制系统的故障诊断中会遇到很多复杂的问题,对具体系统的故障诊断还要作具体分析.

把这种方法应用到“锅炉给水装置”的控制系统故障诊断上. 对这套装置，总结出 20 个故障模式,针对每个故障模式找到相应的解决办法,然后进行数据处理，得到样本数据对 $(\boldsymbol{x}^{(p)}, \boldsymbol{y}^{(p)})$，$(p = 1, 2 \cdots 20)$,把这些样本经过递归运算,存贮到联想记忆矩阵$M$中，在 IBM—PC/XT 上进行了仿真，结果表明，这种方法用于控制系统故障诊断，效果较好.

四、结　束　语

联想记忆是神经元网络的重要特征之一. 本文给出了递归联想记忆模型，其中的M是在最小均方差准则的条件下得到的,其精度较高,另外,当一个新样本向量增加时,这个模型是在原有的基础上,对样本向量进行递归运算,因而更合乎人的记忆方式.

参　考　文　献

[1] Kohonen, T., Self-Organization and Association Memory, Springer-Verlab, Berlin, 1984.

[2] Kohonen, T., Correlation Matrix Memories, *IEEE Trans. Computer*, April 1972.

[3] Kohonen T. and Ruohonen, M., Representation of Associated Data by Matrix Operators, *IEEE Trans. Computer*, July 1973.

[4] Greville, T. Some Application of the Pseudoinverse of A Matrix, *SIAM Rev.*, 2(1960), 1.

A MODEL OF RECURSIVE ASSOCIATIVE MEMORY AND ITS APPLICATION TO FAULT DIAGNOSIS

TAN MIN SHU SONGGUI
(Institute of Automation, Academia Sinica)

ABSTRACT

In this paper, a model of recursive associaive memory is introduced and its application to fault diagnosis is presented.

Key words: Recursive associative memory; pseudoinverse; neural network; fault diagnosis.