

小脑模型神经网络改进算法的研究¹⁾

刘 慧 许晓鸣 张钟俊

(上海交通大学自动化系 上海 200030)

摘 要 该文介绍了小脑模型神经网络的基本原理,在分析 Albus^[1]算法的基础上,指出了该算法在批量学习时的缺陷.针对批量学习提出了相应的改进算法,并证明了该算法的收敛性,仿真结果表明了该改进算法具有收敛速度快的特点.

关键词 神经网络, 杂凑编码, 联想网络, CMAC.

1 引言

自 1982 年 Hopfield 发表了关于反馈神经网络的文章以及 Rumelhart 等人发表了专著 PDP 以来,在世界范围内掀起了研究神经网络的热潮.近年来,人工神经网络和人工智能在控制界中的应用研究正在兴起和蓬勃发展,特别是神经网络具有充分逼近任意复杂非线性函数的能力,为解决复杂的非线性问题开辟了一条控制的具有特殊联想功能的神经网络 CMAC(Cerebellar Model Articulation Controller).

CMAC 神经网络是由 J. S. Albus^[2]在 1975 年提出的.它与 Perceptron 网相似,虽然从每个神经元看其关系是一种线性关系,但从结果总体看 CMAC 模型适合于非线性的映射关系.同时它的算法是十分简单的 δ 算法,所以速度很快.它把输入在一个多维状态空间中的量,映射到一个比较小的有限区域.只要对多维状态空间中部分样本进行学习,就可达到轨迹学习和控制的解,因此特别适合于机器人的轨变学习控制,实时学习控制,非线性函数映射,以及模式识别等领域. CMAC 具有自适应的作用,并且易于硬件化实现.

2 CMAC 的基本原理

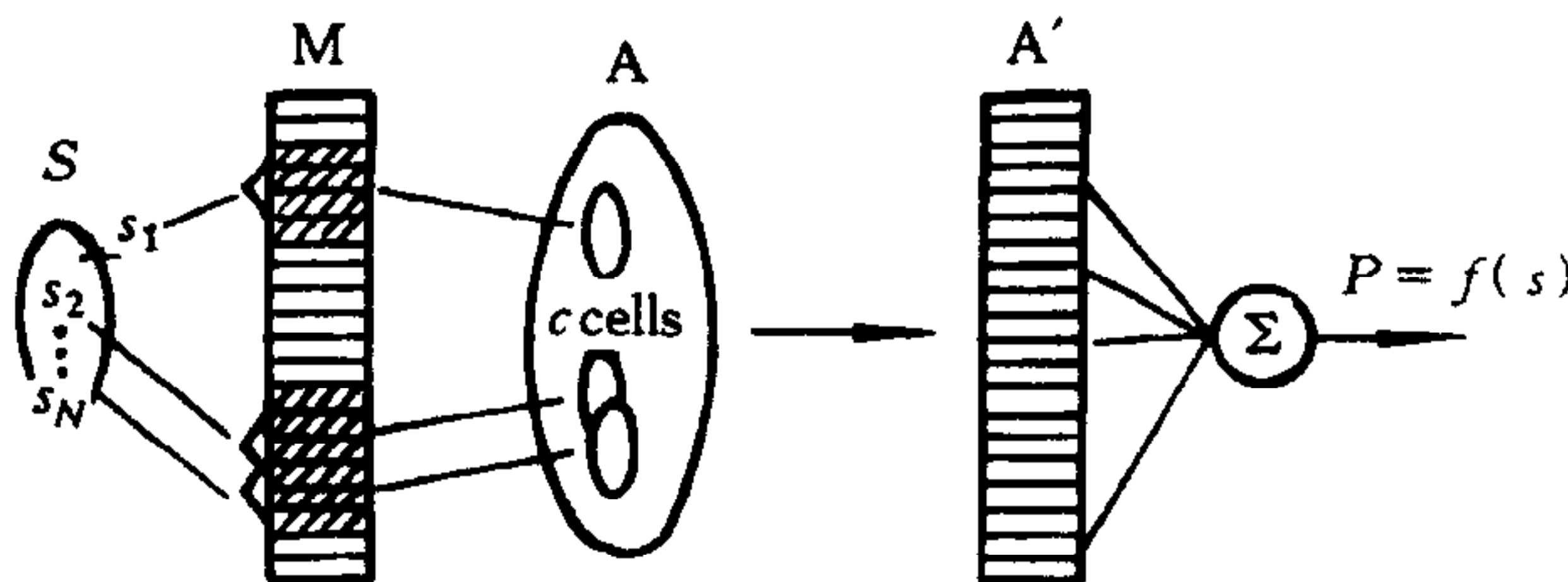


图 1 CMAC 网络的模型结构

CMAC 的简单结构模型如图 1 所示,输入空间 S 由所有可能的输入向量 S_i 组成,CMAC 网络将其接受到的任何输入,通过感知器 M 映射到一个很大的联想存储器 A 中的 c 个单元.输入空间邻近的两个输入向量在存储器 A 中有部分重叠的单元;距离越近,重叠越多.

1) 留学回国人员科研基金资助课题.
收稿日期 1995-04-10

反之,若输入空间远离的两个输入向量在 A 中并不重叠. 对一个实际系统,输入空间中的向量数是很大的,例如,某系统有 10 个输入,而每个输入可取 100 个值,则在输入空间有 100^{10} 个向量, A 的存储量需要 100^{10} 个单元. 由于绝大多数学习问题并不包括所有输入空间中的状态,故所需的存储空间可通过常用的计算机存储压缩技术——随机杂凑编码 (Hash-coding) 技术映射到一个小得多的物理可实现的存储器 A' . 任何 CMAC 网络的输入激活 c 个真实存储位置,而这些位置的值被相加得到输出向量 P .

3 Albus 的 CMAC 算法

设共有 P 个训练样本,联想向量 \mathbf{a}_i 为 $M \rightarrow A$ 的映射,CMAC 网络对于第 i 个训练样本的输出为 $f(S_i) = \mathbf{w}^T \mathbf{a}_i$,理想输出为 d_i ,误差信号 $\delta_i = d_i - f(s_i)$,CMAC 是按 δ 学习律调整网络的权值,是在梯度法的基础上采用 LMS 算法得到的^[2]

$$\Delta \mathbf{w}_i = \frac{\beta}{c} \delta_i \mathbf{a}_i \quad (1)$$

其中 β 为学习步长.

当我们在第 l 次循环时对第 k 个样本进行训练时,产生的修整量为 $\beta \delta_k^{(l)} / c$,其中 $\delta_k^{(l)}$ 为第 k 个样本的输出误差. 因 CMAC 网络内在的泛化能力,在输入空间相近的向量在实际存储器 A' 中有重叠,故该修正量势必影响其它样本的输出,如它使第 i 个样本的新输出为

$$f(s_i) = \mathbf{w}_i^{(l)T} \mathbf{a}_i^{(l)} + c_{ik} \frac{\beta}{c} \delta_k^{(l)} \quad (2)$$

其中 c_{ik} 为第 i, k 个样本在 A' 中的重叠单元数,显然 $c_{ik} = c_{ki}$. 经 l 次循环后,输入样本 k 产生的修正量对其本身的累积输出误差为 $E_k = \sum_l \beta \delta_k^{(l)}$,与之相关的每个权的修正量为 $\Delta_k = E_k / c$, Δ_k 称为累积权误差. 累积权误差对第 i 个样本的输出贡献为 $c_{ik} \Delta_k$. 当 CMAC 算法收敛, $\delta_k^{(l)} \rightarrow 0$,则 $\Delta_k \rightarrow \text{const}$. 若我们由 Δ_i 得 \mathbf{w}_i ,则在网络算法中可换一个角度将累积权误差 Δ_i 而不是权 \mathbf{w}_i 视为需要学习的变量,这样 CMAC 的收敛也可转化为 Δ_i 的收敛问题. 设网络初始权为零,则对每个输入向量 s_i ,网络的输出为

$$f(s_i) = \sum_{j=1}^P c_{ij} \Delta_j \quad (3)$$

学习的目的是使 $f(s_i) = d_i$,即 $\sum_{j=1}^P c_{ij} \Delta_j = d_i$,写成矩阵形式

$$C \Delta = D \quad (4)$$

令 $A = [\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_p]^T$,则

$$A W = D \quad (5)$$

通常神经网络权的学习有逐一和批量两种方法,设训练样本集为 $\{(s_1, d_1), (s_2, d_2), \dots, (s_p, d_p)\}$. 逐一学习是每输入一个样本便对整个网络的权进行修正. 即先对 (s_1, d_1) 进行学习,利用输出误差来对网络的权进行修正,使网络的对应关系满足 $f(s_1) = d_1$,设此时网络的权为 \mathbf{w}_1 . 然后以 \mathbf{w}_1 为基础,对样本 (s_2, d_2) 进行学习,直到权值 \mathbf{w}_2 满足 $f(s_2) = d_2$. 但是一般情况下, $\mathbf{w}_1 \neq \mathbf{w}_2$,故此时未必有 $f(s_1) = d_1$ 成立. 所以逐一学习的方法会出现学了

新的,忘了旧的“遗忘”现象.为了克服这个缺点需采用反复循环学习,但这样又会带来收敛慢的问题.我们建议用批量学习的方法来对权进行更新,将样本一一输入,为一批样本输入后用总的误差来修整权,这种方法可以克服“遗忘”的缺点,又有较快的收敛速度.

Albus 本人并未给出 CMAC 算法收敛性的证明,Wong^[3]和 Parks^[4]从线性方程组迭代解的角度,Wong^[5]还从频域的角度阐述了 Albus 算法的收敛性问题.他们的研究都是针对逐一学习的,对批量学习的情况并未进行研究.作者针对批量学习情况进行了一些探索,指出了 Albus 算法在逐一学习时适用,而在批量学习时易发散的缺陷,并提出了相应的改进算法.

4 批量学习时 Albus 算法的缺陷及改进算法

定理 1. 对于多输入输出目标中的一组训练样本,如果输入空间被量化后不存在两个不同训练样本激活相同一组神经元的情况,则 Albus 算法采用批量学习方法时的收敛条件为 $\frac{\pi}{2} < \omega < \frac{3\pi}{2}$.

证明. 当 Albus 算法采用批量学习时,累积权误差的修正公式为

$$\Delta_i^{(l+1)} = \frac{1}{c} (d_i - \sum_{j \neq i} c_{ij} \Delta_j^{(l)}), \quad (6)$$

其中

$$c_{ij} = \begin{cases} c - |i - j| & \text{if } |i - j| < c, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

为方便起见,设输入输出空间均为一维,所有训练样本的量化值都在 $[0, R]$ 上,则所需神经元为 $R + c - 1$ 个.

当收敛到 Δ^* 时,由式(6)得

$$\Delta_i^* = \frac{1}{c} (d_i - \sum_{j \neq i} c_{ij} \Delta_j^*). \quad (8)$$

令 $e_i^{(l)} = \Delta_i^{(l)} - \Delta_i^*$, 得

$$e_i^{(l+1)} = -\frac{1}{c} \sum_{j \neq i} c_{ij} e_j^{(l)}. \quad (9)$$

e 为一有限长的 N_1 点非周期序列, $N_1 = R + 2c - 2$. 将其作周期延拓, e 可写成 Fourier 级数形式

$$e_n = \sum_{k=0}^{N_1-1} \tilde{e}_k \cdot \exp(j \frac{2\pi}{N_1} kn). \quad (10)$$

将(10)代入(9)并考虑(7)可得

$$\tilde{e}_k^{(l+1)} = H(e^{j\omega}) \tilde{e}_k^{(l)}, \quad (11)$$

其中

$$\begin{aligned} H(e^{j\omega}) &= -\frac{1}{c} \left(\sum_{n=1}^c (c-n) e^{-j\omega n} + \sum_{n=1}^c (c-n) e^{j\omega n} \right) \\ &= -j \frac{\sin \omega - \sin(c\omega)/c}{2\sin^2(\omega/2)}. \end{aligned} \quad (12)$$

c 一般取为 $32 \sim 256$, 故 $\sin(c\omega)/c$ 可忽略. 若 $\omega \neq 2\pi$, 则

$$|H(e^{j\omega})| \approx \left| \text{ctg} \frac{\omega}{2} \right|. \quad (13)$$

当且仅当 $|H(e^{j\omega})| < 1$ 时算法收敛, 在一个数字频率周期 (2π) 内 $|H(e^{j\omega})| < 1$, 即

$$\frac{\pi}{2} < \omega < \frac{3\pi}{2}. \quad (14)$$

当 $\omega = 2\pi$ 时, $\omega \in \text{Ker}(A)$, 不影响 w 和 Δ 的收敛. 所以 Albus 算法采用批量学习时的收敛条件为 $\frac{\pi}{2} < \omega < \frac{3\pi}{2}$.

推论 1. 对于多输入输出目标中的一组训练样本, 设所有训练样本的量化值都在 $[0, R]$ 上, 感受野宽度为 c . 如果输入空间被量化后不存在两个不同训练样本激活相同一组神经元的情况, 则 Albus 算法采用批量学习方法时的收敛条件为

$$\frac{N_1}{4} < k < \frac{3N_1}{4}, \quad N_1 = R + 2c - 2.$$

证明. 将 $\omega = \frac{2\pi k}{N_1}$ 代入式(14), 得

$$\frac{\pi}{2} < \frac{2\pi k}{N_1} < \frac{3\pi}{2}. \quad (15)$$

$$\therefore \frac{N_1}{4} < k < \frac{3N_1}{4}.$$

由 Fourier 级数(10)知, k 的取值为 $[0, N_1 - 1]$ 间的连续整数, 而采用 Albus 算法批量学习时的收敛条件为 $\frac{N_1}{4} < k < \frac{3N_1}{4}$, k 在 $[0, \frac{N_1}{4}]$ 和 $[\frac{3N_1}{4}, N_1 - 1]$ 两段取值时算法不收敛. N_1 越大, 即量化取值数 R 和感受野宽度 c 越大, 不收敛的点越多.

为了克服 Albus 算法对批量学习不太适合的缺陷, 提出了一种改进的 CMAC 算法把式(1)修改为

$$\Delta w_i = \frac{\beta}{c} \cdot \frac{\delta_i}{q} \cdot a_i, \quad (16)$$

其中 q 为批量学习一次每个权平均被更新的次数.

定理 2. 对于多输入输出目标中的一组训练样本, 如果输入空间被量化后不存在两个不同训练样本激活相同一组神经元的情况, 改进 CMAC 算法采用批量学习方法时的收敛条件为 $2\text{ctg}^{-1}q < \omega < 2(\pi - \text{ctg}^{-1}q)$.

证明. 改进 CMAC 算法采用批量方法对权进行更新时, 相应的算法收敛证明中的公式(13)变为

$$|H(e^{j\omega})| \approx \frac{1}{q} \left| \text{ctg} \frac{\omega}{2} \right|. \quad (17)$$

当且仅当 $|H(e^{j\omega})| < 1$ 时算法收敛, 在一个数字频率周期 (2π) 内 $|H(e^{j\omega})| < 1$, 即

$$2\text{ctg}^{-1}q < \omega < 2(\pi - \text{ctg}^{-1}q). \quad (18)$$

当 $\omega = 2\pi$ 时, $\omega \in \text{Ker}(A)$, 不影响 w 和 Δ 的收敛. 所以改进 CMAC 算法采用批量学习时的收敛条件为 $2\text{ctg}^{-1}q < \omega < 2(\pi - \text{ctg}^{-1}q)$.

讨论. 由于 CMAC 网络所固有的局域泛化能力, 每个输入向量使其对应量化值周围

c 个感知器同时被激励;相应地,当一批样本被学习一遍后,每个网络的权也不止一次被更新.所以 $\min q=2$. 当 q 取最小值 2 时,由(18)得改进算法的收敛条件为 $0.93 < \omega < 5.35$ 弧度,改进算法的收敛范围比 Albus 算法(14)宽, q 越大改进算法的收敛范围将越宽.

推论 2. 对于多输入输出目标中的一组训练样本,设所有训练样本的量化值都在 $[0, R]$ 上,感受野宽度为 c . 如果输入空间被量化后不存在两个不同训练样本激活相同一组神经元的情况,改进 CMAC 算法采用批量学习方法时的收敛条件为 $\frac{(\text{ctg}^{-1}q) \cdot N_1}{\pi} < k < \frac{(\pi - \text{ctg}^{-1}q) \cdot N_1}{\pi}$, $N_1 = R + 2c - 2$.

证明. 将 $\omega = \frac{2\pi k}{N_1}$ 代入式(18)即可得改进算法采用批量学习时的收敛条件为

$$\frac{(\text{ctg}^{-1}q) \cdot N_1}{\pi} < k < \frac{(\pi - \text{ctg}^{-1}q) \cdot N_1}{\pi}. \quad (19)$$

讨论. 将(19)进一步展开为

$$\frac{N_1}{4} - b_1 < k < \frac{3N_1}{4} + b_2, \quad (20)$$

其中

$$\begin{aligned} b_1 &= \frac{N_1}{4} - \frac{N_1 \cdot \text{ctg}^{-1}q}{\pi} = \frac{N_1 \cdot (\pi - \text{ctg}^{-1}q)}{4\pi}, \\ b_2 &= \frac{N_1}{4} - \frac{N_1 \cdot \text{ctg}^{-1}q}{\pi} = \frac{N_1 \cdot (\pi - 4\text{ctg}^{-1}q)}{4\pi}. \end{aligned} \quad (21)$$

当 q 取最小值 2 时, $\text{ctg}^{-1}q = 0.46$ 弧度,而反余切函数在区间 $[0, \pi]$ 内是递减的,所以 $\text{ctg}^{-1}q < 0.46$, 从而 $b_1 > 0, b_2 > 0$. 可见改进算法的收敛范围比 Albus 算法宽. q 越大则 b_1 和 b_2 越大,改进算法的收敛范围也将越宽.

改进 CMAC 算法对逐一学习也是适用的,其收敛性证明可参见文献[4],其中累积权误差的修正类似 Jacobi 迭代法解线性方程组.

5 仿真结果

设某 CMAC 网络用来实现下列一维非线性函数的映射,

$$f(s) = \sin\left(\frac{2\pi s}{360}\right). \quad (22)$$

其中输入变量 $s_i \in [0, 180]$, 分辨率为 1, 所有训练样本的量化值都在 $[0, R]$ 上, $R = 180$. CMAC 网络的感受野宽度 $c = 32$, 批量学习一次每个权平均被更新的次数 q 取为 6. 则 $N_1 = 242, \text{ctg}^{-1}q = 0.17$. 按 $\Delta s = 5$ 的间隔均匀取点, 由式(22)进行理论计算, 可得 37 组输入输出的样本数据, 运用这些样本数据分别按 Albus 算法和本文的改进算法对 CMAC 神经网络进行训练. 为了显示网络的泛化能力, 计算网络输出时采用了不同于训练时的数据 $\Delta s = 6$. 由式(15)计算得 Albus 算法批量训练时的收敛条件为 $60 < k < 182$; 而由式(19)知 $12 < k < 230$ 时本文的改进算法批量训练时收敛, 易见改进后的算法比 Albus 算法收敛条件宽.

图 2 所示为逐一学习时的误差曲线,两种算法都可取得较好的收敛效果,而本文提出的改进方法收敛速度更快一些.图 3 是在批量学习时的误差曲线,由该图可见,Albus 算法易导致发散,而采用本文的改进算法可以获得收敛较快的效果.

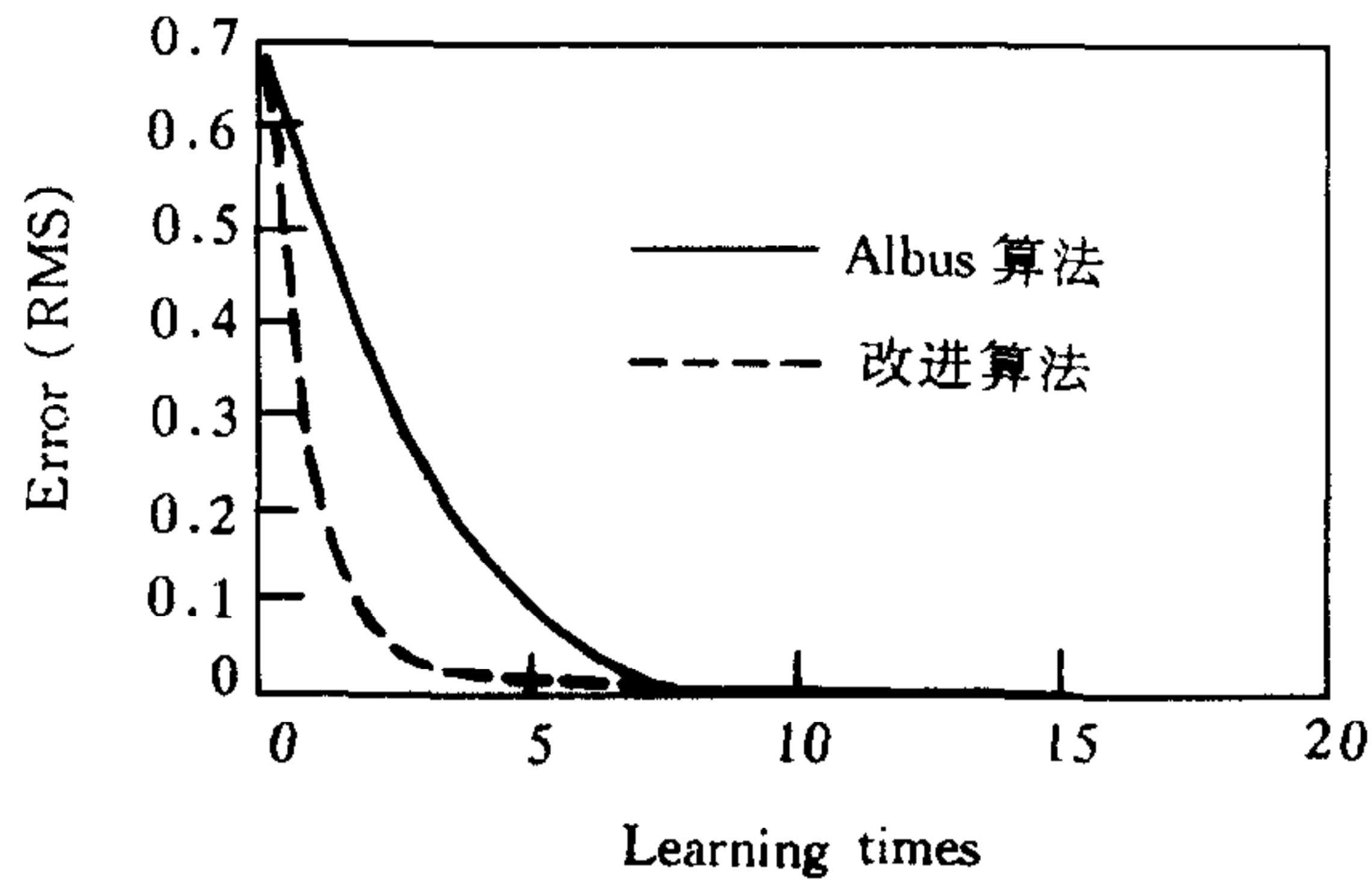


图 2 逐一学习误差曲线

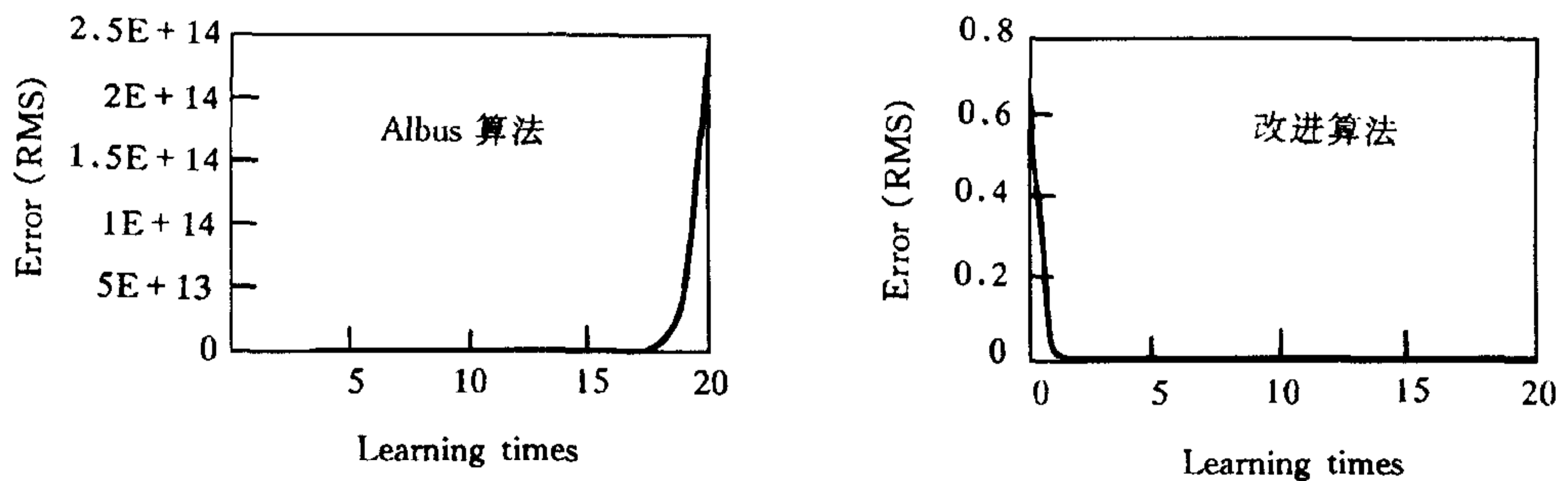


图 3 批量学习误差曲线

6 结论

本文在对 Albus 算法研究的基础上,提出了适合于批量学习的改进算法,并在理论上证明了该算法的收敛性,仿真结果表明了该算法在逐一和批量学习时都有较好的收敛性和快速性.

参 考 文 献

- [1] Albus J S. Data storage in the cerebellar model articulation controller (CM • AC). *Trans. ASME, J. Dyn. Syst. Meas. Contr.*, 1975, **97**:228—233.
- [2] Albus J S. A new approach to manipulator control: The cerebellar model articulation Controller (CMAC). *Trans. ASME, J. Dyn. Syst. Meas. Contr.*, 1975, **97**:220—227.
- [3] Wong Y F, Sideris A. Learning convergence in the cerebellar model articulation controller. *IEEE Trans. on Neural Networks*, 1992, **3**(1):115—121.
- [4] Parks P C, Militzer J. Convergence properties of associative memory storage for learning control system. *Automation and Remote Control*, 1989, **50**(2):254—286.
- [5] Wong Y F. CMAC learning is governed by a single parameter. *IEEE Int. Conf. on NN*, 1993, 1439—1443.

AN IMPROVED CMAC NEURAL NETWORK ALGORITHM

LIU HUI XU XIAOMING ZHANG ZHONGJUN

(Department of Automation, Shanghai Jiaotong University, Shanghai 200030)

Abstract The basic principle of CMAC (Cerebellar Model Articulation Controller) is introduced. Based on a detail analysis of Albus algorithm^[1], the paper points out its drawback in batch learning. An improved algorithm is proposed and theoretical proofs are also given. Simulation results show that the improved method has higher speed and better convergence than the original.

Key words Neural network, hash-coding, associative network, CMAC.

刘 慧 1968年生,1996年获上海交通大学自动控制理论及应用专业博士学位.现为上海交通大学自动化系讲师.主要研究兴趣是智能控制,学习控制及神经网络在控制中的应用.

许晓鸣 1957年生,现任上海交通大学自动化系教授、博士生导师.主要研究兴趣是智能控制和复杂工业系统预测控制等方面的研究.