

文章编号: 1001-0920(2001)05-0523-08

CMAC 算法收敛性分析及泛化能力研究

何超, 徐立新, 张宇河

(北京理工大学 自动控制系, 北京 100081)

摘要: 利用矩阵理论和线性方程组迭代收敛的一般性原理, 在不附加特殊条件的情况下, 证明了 CMAC 算法在批量和增量两种学习方式下的收敛定理, 对在关联矩阵正定条件下得出的结论进行推广和改进。在此基础上提出一种学习率自寻优的 CMAC 改进算法, 并提出一种简单可行的评价 CMAC 网络整体泛化性能的指标。通过计算机仿真验证了收敛定理的正确性和改进算法的优越性, 并研究得出了 CMAC 网络各个参数对其泛化性能影响的相关结论。

关键词: CMAC; 神经网络算法; 批量学习; 增量学习; 收敛性; 泛化能力

中图分类号: TP 18 **文献标识码:** A

Convergence and Generalization Ability of CMAC

H E Chao, XU L i-x in, ZHAN G Yu-he

(Department of Automatic Control, Beijing Institute of Technology, Beijing 100081, China)

Abstract: Using the matrix theory and general principle of the iterative convergence of the linear coupled equations, the convergent theorems of the CMAC algorithm are proved both in the batch and the incremental learning styles without any special conditions attached. Some existing conclusions under the condition that the articulation matrix is positive definite are improved. An improved CMAC algorithm of self-optimizing learning rate is presented. Moreover, a simple and feasible criterion is presented to evaluate the generalization ability of the whole CMAC network. Simulation results show the correctness of the convergent theorems and the advantages of improved algorithm.

Key words: CMAC; neural network algorithm; batch learning; incremental learning; convergence; generalization ability

1 引言

小脑模型关联控制器(CMAC), 是 Albus 根据小脑的生物模型提出的一种人工神经网络^[1,2]。它学习速度快, 具有局域泛化能力, 能够克服 BP 网络的局部最优问题, 且易于硬件实现。近年来, CMAC 广泛应用于实时控制^[3~7], 模式识别^[3], 非线性时间序列分析^[8]等领域。但相对应用而言, CMAC 的理论

体系还远未成熟, 并且在具体应用中也存在一些问题。

文献[9~11]用不同方法对 CMAC 的收敛性进行了分析, 但结论都是在关联矩阵正定的特殊条件下得到的。本文利用矩阵理论和线性方程组迭代收敛的一般性原理, 在不附加任何特殊条件的情况下, 证明了 CMAC 算法在批量和增量两种学习方式下的收敛定理, 并以此为理论依据, 提出一种学习率自

收稿日期: 2000-05-08; 修回日期: 2000-08-20

作者简介: 何超(1975—), 男, 甘肃庆阳人, 博士生, 从事神经网络、模式识别等研究; 张宇河(1940—), 男, 福建漳州人, 教授, 博士生导师, 从事运动控制等研究。

寻优的CMAC改进算法,以解决传统CMAC算法中学习率难以确定的问题。同时,提出一种简单可行的评价CMAC网络整体泛化性能的指标,研究网络各个参数对其泛化能力的影响。

CMAC 其本质是一种用于表示复杂非线性函数的查表技术,数学上可将其看成由 $S \rightarrow A$ 和 $A \rightarrow Y$ 两个映射组成,其中 S, A 和 A 分别表示输入空间、虚拟联想空间和物理存储空间。CMAC 具体的实现思想是将输入空间 S 划分为许多子空间,每相邻的子空间互相有重叠。 S 中的各个输入向量在接受场函数的作用下落入相应的子空间,每个子空间通过上述两个映射激活 g 个存储单元,这些单元存储的权值叠加产生CMAC的输出。设输入向量为 s ,物理空间中存储单元的个数为 H ,定义列向量 $x = [x_1, x_2, \dots, x_H]^T$ 的每个元素

$$x_i = \begin{cases} 1, & A \text{ 中第 } i \text{ 个存储单元被 } s \text{ 激活} \\ 0, & \text{其它} \end{cases} \quad (1)$$

称 x 为 s 对应的地址向量。

用 $w = [w_1, w_2, \dots, w_H]^T$ 表示 A 中存储的权值向量,则CMAC网络的输出为

$$y = x^T w \quad (2)$$

CMAC网络权值的调整一般采用Widrow-Hoff规则,即如下LMS方法

$$w_{l+1} = w_l + \beta x_l^T (y_l - x_l^T w_l) / g \quad (3)$$

其中, x_l 是第 l 次学习时的输入对应的地址向量, y_l 是期望输出, w_{l+1} 和 w_l 分别是第 l 次和第 $l-1$ 次学习后的权值向量, g 是泛化常数, β 是学习率。

2 CMAC 算法的收敛性

Wong 和 Sideris 对CMAC网络的收敛性进行分析,得出当采用增量学习法且学习率为 1 时,CMAC 算法本身可保证其收敛性的结论^[9]。文献 [10, 11] 在 Wong 和 Sideris 所做研究的基础上,对CMAC的收敛性进行进一步研究,但其结论也都是在关联矩阵正定的特殊条件下得到的。本节将从 Wong 和 Sideris 提出的算法表述形式出发,分析CMAC算法在批量和增量两种学习方式下的收敛性,给出CMAC算法收敛的一般性定理(不要求关联矩阵正定)。同文献 [9~11], 在下面的分析中假设 $S \rightarrow A$ 的映射对于每个样本是唯一的。事实上,Albus 证明了只要取 $H > 100g$, $S \rightarrow A$ 的映射出现碰撞的概率是微乎其微的^[11]。

2.1 CMAC 学习算法描述

设训练样本为 $(s_j, y_j), j = 1, 2, \dots, P$, 输入向量 s_j 对应的地址向量为 x_j, y_j 为期望输出。则第 j 个样本第 1 次学习时,CMAC 的输出误差为 $\delta_j^{(1)} = y_j - x_j^T w_{j0}$ 。由式(3)可得,每个被第 j 个样本激活的存储单元中权值的修正量为 $\Delta w_j^{(1)} = \beta \delta_j^{(1)} / g$ 。因为CMAC具有的局域泛化能力,这种修正必将对其它样本的输出产生影响。此时,第 k 个样本的输出变为 $x_k^T w_k + c_{kj} \Delta w_j^{(1)}$, 其中 c_{kj} 是第 k, j 个样本在物理空间 A 中的重叠单元数,显然 $c_{kj} = c_{jk}$ 。经过 l 次学习后,第 j 个样本产生的修正量对其本身的累积输出误差为 $E_j = \beta \delta_j^{(l)}$, 与之相关的每个权值的修正量为 $\Delta_j = E_j / g$, 称为累积权误差。累积权误差对第 k 个样本的输出贡献为 $c_{kj} \Delta_j$ 。

当CMAC算法收敛时,随着 $\delta_j^{(l)} \rightarrow 0, \Delta_j \rightarrow \text{const}$ 。若可由 Δ_j 计算 w_j , 则在网络算法中可以换一个角度将累积权误差 Δ_j 而不是权 w_j 视为需要学习的变量,这样CMAC的收敛问题便可转化为 Δ_j 的收敛问题。设网络初始权值为 0, 则对任意输入向量 s_k , 网络的输出可表示为 $f(s_k) = \sum_{j=1}^P c_{kj} \Delta_j, k = 1, 2, \dots, P$ 。学习的目的是使 $f(s_k) = y_k$, 即 $\sum_{j=1}^P c_{kj} \Delta_j = y_k$ 。写成矩阵形式,有

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

其中, $C = (c_{ij}), i, j = 1, 2, \dots, P, \Delta = [\Delta_1, \Delta_2, \dots, \Delta_P]^T, y = [y_1, y_2, \dots, y_P]^T$ 。

矩阵 C 称为关联矩阵。一旦给定了CMAC的训练样本集和泛化常数,矩阵 C 也就确定。如果定义 $X = [x_1, x_2, \dots, x_P]$, 其中 x_j 为输入向量 s_j 对应的地址向量,则根据 c_{kj} 的定义很容易得出 $C = X^T X$ 。矩阵 C 的性质可表述为如下定理:

定理 1 设CMAC网络输入向量 $s_j \in R^n, j = 1, 2, \dots, P, R^n$ 表示 n 维实空间,关联矩阵 C 有如下性质:

- 1) C 为实对称矩阵,所有元素均为非负整数,对角线元素为 g ;
- 2) C 是半正定矩阵;特别地,当 $n = 1$ 时, C 是正定矩阵。

证明 性质 1) 很容易由 C 的定义得出。现在证明性质 2), 任给 $P > 0$, 则有

$$P^T C P = P^T X^T X P = (X P)^T X P = X P^2 \geq 0$$



因此 C 是半正定的。(证毕)

Wong 在文献[9]中给出了 $n = 1$ 时 C 正定的证明,但他错误地将结论推广到 $n > 1$ 的情形。文献[10]给出了 $n = 2$ 时 C 不正定的反例。

2.2 CMAC 批量学习算法的收敛性

批量学习法是指在每一步学习中,依次使用所有的样本进行训练,但每次训练后并不立即修改权值,而是等到这一步所有样本都训练完毕后,再一并对权值进行修改。

批量学习时,由式(4)可得CMAC累积权误差的修正公式为

$$\Delta_j^{(l+1)} = \Delta_j^{(l)} + \frac{\beta}{g} \left(y_j - \sum_{k=1}^P c_{kj} \Delta_k^{(l)} \right) \quad (5)$$

其中, $\Delta_j^{(l)}$ 表示第 j 个样本第 l 步学习后的累积权误差,其它参数定义如前。写成矩阵形式为

$$\Delta^{(l+1)} = \Delta^{(l)} + \frac{\beta}{g} (y - C\Delta^{(l)}) = \left(I - \frac{\beta}{g} C \right) \Delta^{(l)} + \frac{\beta}{g} y \quad (6)$$

矩阵 C , 向量 Δ 和 y 定义如上节。由于式(6)表述的学习算法实质上等同于解线性方程组 $C\Delta = y$, 因此它的收敛性也满足线性方程组迭代的收敛定理。现引用线性方程组迭代法的收敛条件如下^[12]:

引理 1 对任意初始向量 $x^{(0)}$ 和右端项 g , 由迭代格式 $x^{(k+1)} = Mx^{(k)} + g$ ($k = 0, 1, \dots$) 产生的向量序列 $\{x^{(k)}\}$ 收敛的充要条件, 是迭代矩阵 M 的谱半径小于 1, 即 $\rho(M) < 1$ 。

根据矩阵理论^[13], 有如下定义:

定义 1 设 A 是 Hermitian 矩阵, 称实数 $R(x) = x^H A x / x^H x$, ($x \in C^n, x \neq 0$) 为 A 的 Rayleigh 商, 其中 C^n 表示 n 维复向量集合。

引理 2 设 Hermitian 阵 A 的最大、最小特征值分别为 λ_{\max} 和 λ_{\min} , 则 A 的 Rayleigh 商具有如下性质: $\lambda_{\min} \leq R(x) \leq \lambda_{\max}$ 。

由引理 1 和引理 2, 给出 CMAC 批量学习算法的收敛条件如下:

定理 2 CMAC 批量学习算法收敛的充要条件是学习率 β 满足 $0 < \beta < 2g/\lambda_{\max}$, 其中 $\lambda_{\max} > 0$ 是关联矩阵 C 的最大特征值。

证明 由引理 1 知, 式(6)表述的 CMAC 批量学习算法收敛的充要条件是

$$\rho \left(I - \frac{\beta}{g} C \right) < 1 \quad (7)$$

设 λ 为迭代矩阵 $I - \frac{\beta}{g} C$ 的任一特征值, x 为其

特征向量, 即

$$\left(I - \frac{\beta}{g} C \right) x = \lambda x \quad (8)$$

上式两边左乘以 x 的共轭转置向量 x^H , 有

$$x^H \left(I - \frac{\beta}{g} C \right) x = x^H \lambda x \quad (9)$$

$$\lambda = 1 - \frac{\beta}{g} \frac{x^H C x}{x^H x} \quad (10)$$

由于 C 为实对称矩阵, 且 $x \neq 0$, 根据 Rayleigh 商的定义, 可得 $x^H C x / x^H x$ 为 C 的 Rayleigh 商, 记为 $R(x)$ 。由 C 的半正定性和引理 2 得 $0 \leq \lambda_{\min} \leq R(x) \leq \lambda_{\max}$, 其中 $\lambda_{\min}, \lambda_{\max}$ 分别为 C 的最小、最大特征值。由定理 1, C 不可能相似于零矩阵, 所以 $\lambda_{\max} > 0$ 。于是式(10)变为 $\lambda = 1 - \frac{\beta}{g} R(x)$, 则式(7)即为

$$\max \left| 1 - \frac{\beta}{g} R(x) \right| < 1 \Leftrightarrow \quad (11)$$

$$\left| 1 - \frac{\beta}{g} R(x) \right| < 1, \quad \forall R(x) \Leftrightarrow \quad (12)$$

$$0 < \beta < \frac{2g}{R(x)}, \quad \forall R(x) \quad (13)$$

式(13)表示 CMAC 批量学习算法收敛的充要条件是学习率 β 同时满足 $R(x)$ 取任意值时的要求, 即 $\beta \in \bigcap_{\forall R(x)} \left\{ \beta \mid 0 < \beta < \frac{2g}{R(x)} \right\}$, 亦即 $0 < \beta < \frac{2g}{\lambda_{\max}}$ 。(证毕)

2.3 CMAC 增量学习算法的收敛性

增量学习是指在每一步学习时,依次使用所有样本进行训练,且每次训练后立即修改权值。

增量学习时,由式(4)可得CMAC累积权误差的修正公式为

$$\Delta_j^{(l+1)} = \Delta_j^{(l)} + \frac{\beta}{g} \left(y_j - \sum_{k=1}^{j-1} c_{kj} \Delta_k^{(l+1)} - \sum_{k=j}^P c_{kj} \Delta_k^{(l)} \right) \quad (14)$$

令 $C = L + D + U$, 其中 L, U 分别是 C 不含对角线元素的下三角和上三角阵, D 为 C 对角线元素构成的对角阵, $D = gI$ 。则式(14)写成矩阵形式为

$$\Delta^{(l+1)} = \Delta^{(l)} + \frac{\beta}{g} (y - L\Delta^{(l+1)} - (D + U)\Delta^{(l)}) \quad (15)$$

即

$$\Delta^{(l+1)} = (D + \beta I)^{-1} [(1 - \beta)D - \beta U] \Delta^{(l)} + \beta (D + \beta I)^{-1} y \quad (16)$$

式(16)恰好是方程组(4)的松弛迭代法的公式, β 是松弛因子。根据线性方程组迭代理论^[14], 有如下引理:

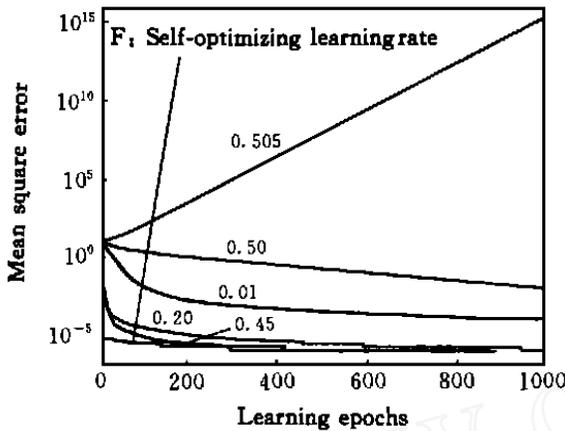


图1 CMAC 批量学习仿真结果

引理3 松弛迭代收敛的必要条件是松弛因子 ω 满足 $0 < \omega < 2$ 。

引理4 若 A 为正定矩阵, 则当松弛因子 ω 满足 $0 < \omega < 2$ 时, 线性方程组 $Ax = b$ 的松弛迭代恒收敛。

给出CMAC 增量学习算法的收敛条件如下:

定理3 CMAC 增量学习算法收敛的必要条件是学习率满足 $0 < \beta < 2$ 。特别地, 当CMAC 输入向量是一维时, $0 < \beta < 2$ 成为CMAC 增量学习算法收敛的充要条件。

由定理1、引理3和引理4可方便地推出该定理。

文献[9~11]给出的CMAC 算法的收敛定理都是在关联矩阵 C 正定的条件下推出的, 而定理2和定理3是在 C 半正定的条件下推出的。由定理1, CMAC 只有在—维输入时才可确保 C 的正定性, 而CMAC 算法本身就能确保 C 的半正定性, 因此定理2和定理3的结论更具普遍性。定理中给出的学习率取值范围表达式也更简单, 意义更明确, 对于实际应用CMAC 时学习率的选择具有更强的指导意义。

2.4 仿真实验

本节以CMAC 逼近复杂函数

$$f(x) = \sin(x) + 2\cos(2x) + e^{-x} + 4\sin(0.5x) + 0.5\cos(8x), \quad x \in [0, 9] \quad (17)$$

为例进行仿真, 验证定理2和定理3的正确性。仿真参数为: 样本采样间隔和网络量化间距都取 $0.1, g = 4, H = 400$ 。

由定理2, CMAC 批量算法收敛时学习率 β 的取值范围由 C 的最大特征值 λ_{\max} 决定。在上述条件下, 可计算出 $\lambda_{\max} = 15.977$, 则理论收敛范围为 0

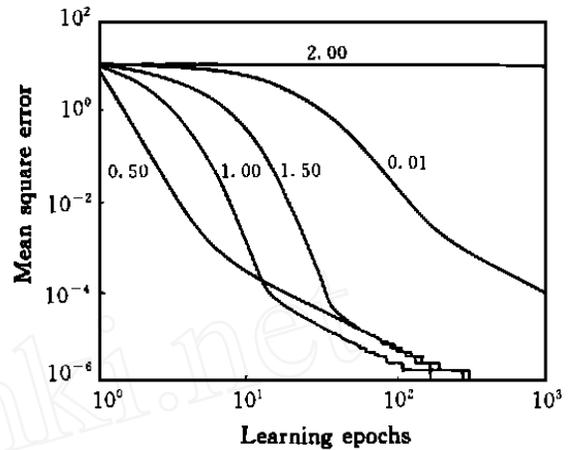


图2 CMAC 增量学习仿真结果

$< \beta < \frac{2 \times 4}{15.977} = 0.501$ 。 β 取不同值时, CMAC 批量学习的仿真结果如图1(纵坐标取对数)所示, 当 β 取 $0.01, 0.20, 0.45, 0.50$ 时, 算法均收敛; 而当 β 取 0.505 时, 算法发散。实际和理论的收敛范围一致, 从而验证了定理2的正确性。

由定理3, CMAC 增量学习算法收敛时学习率 β 应满足 $0 < \beta < 2$ 。 β 取不同值时, CMAC 增量学习的仿真结果如图2(双对数坐标)所示, 当 β 取 $0.01, 0.50, 1.00, 1.50$ 时, 算法均收敛; 而当 β 取 2.00 时, 算法发散, 从而也验证了定理3的正确性。

3 学习率自寻优的CMAC 改进算法

3.1 改进算法描述

第1步: 探求学习率收敛界, 得到算法收敛可采用的最大学习率 β_{\max} 。

1) 设定最大搜索步长 $\max\text{-epoch}$ 和学习率初值 β_0, β , 计算初始误差 preMse_1 , 设标志 $\text{flag} = 0$, 当前步长 $\text{epoch} = 1$;

2) 采用批量算法计算CMAC 输出, 得到当前误差 $\text{Mse}_1, \beta_0, \beta$;

3) 判断: 如果 $\text{Mse}_1 < \text{preMse}_1$, 继续; 否则转5);

4) 判断: 如果 $\text{flag} = 0$, 则 $\beta = 0.7\beta$; 如果 $\text{flag} = 1$, 则 $\beta = 0.95\beta$, 并转6);

5) 判断: 如果 $\text{flag} = 0$, 则 $\beta = (\beta + \beta_0)/2$, $\text{flag} = 1$, 转6); 如果 $\text{flag} = 1$, 则 $\beta_{\max} = \beta$, 输出学习率收敛界 β_{\max} , 结束第1步转第2步;

6) $\text{epoch} = \text{epoch} + 1$, 判断: 如果 $\text{epoch} > \max\text{-epoch}$, 返回2); 否则结束程序, 输出错误信息, 并提示应加大搜索步长。

第 2 步: 在 $(0, \beta_{\max})$ 区域内搜索最优学习率

β_{best}

借鉴混沌的思想, 利用 Logistic 映射

$$x_{n+1} = 4x_n(1 - x_n), \quad n = 0, 1, \dots \quad (18)$$

当 $0 < x_0 < 1$ 时, 产生的序列 $\{x_i\} (i = 0, 1, \dots)$ 在 $[0, 1]$ 之间具有随机性和遍历性的特点^[14]。设定 h , 使用式 (18) 以初值 $x_0 = 0.1$ 产生序列 $\{x_i\}, i = 0, 1, \dots, h - 1$, 然后做变换 $x_i = \beta_{\max} x_i$, 则变换后的序列随机地覆盖分布在学习率收敛区域 $(0, \beta_{\max})$ 内。

设定参数 max-epoch2 , 以 $x_i (i = 0, 1, \dots, h - 1)$ 分别作为学习率, 采用 CMAC 算法探索 max-epoch2 步, 比较各 x_i 对应的网络输出误差, 挑选误差最小的 x_i 近似作为最优学习率 β_{best} 。

第 3 步: 以 β_{best} 作为学习率, 在第 2 步已学习 max-epoch2 步的基础上, 采用批量 CMAC 算法继续学习, 直到达到误差限或最大步长为止, 结束。

给出上述算法中各参数经验值: $\beta_0 = 2, \text{max-epoch1} = 100, h = 20, \text{max-epoch2} = 100$ 。

3.2 改进算法的仿真

逼近 1 维输入函数: 采用学习率自寻优的 CMAC 改进算法逼近式 (17) 所示的函数, 仿真参数同 2.4 节, 仿真结果如图 1 中的曲线 F 所示。由图可见, 学习率自寻优的 CMAC 改进算法收敛速度最快, 改进算法得到的学习率收敛范围是 $0 < \beta < 0.498$, 与理论推出的 $0 < \beta < 0.501$ 很接近, 得出的最优学习率为 0.461。

逼近多维输入函数: CMAC 增量算法在 1 维输入时, 学习率一般取 1 即可保证接近最优的收敛速度, 但当多维输入时, 学习率取 1 不一定收敛, 而采用学习率自寻优的 CMAC 改进算法可保证以接近最优的学习速率收敛。

以 Sombrero 函数

$$f(x, y) = \sin(\sqrt{x^2 + y^2}) / \sqrt{x^2 + y^2} - 8x/8, -8y/8 \quad (19)$$

为例, 分别采用增量 CMAC ($\beta = 1$) 和学习率自寻优 CMAC 算法进行逼近, 样本采样间隔和网络量化间距都取 0.5, 泛化常数 g 取 3, 物理存储单元数取 400, 最大步长设为 500。仿真结果分别如图 3 和图 4 所示。

由图可见, 采用学习率自寻优 CMAC 算法逼近的效果比采用增量 CMAC 算法明显要好。学习率自寻优 CMAC 算法得到算法收敛的学习率取值范围是 $0 < \beta < 0.116$, 与理论推出的 $0 < \beta < 0.121$ 很

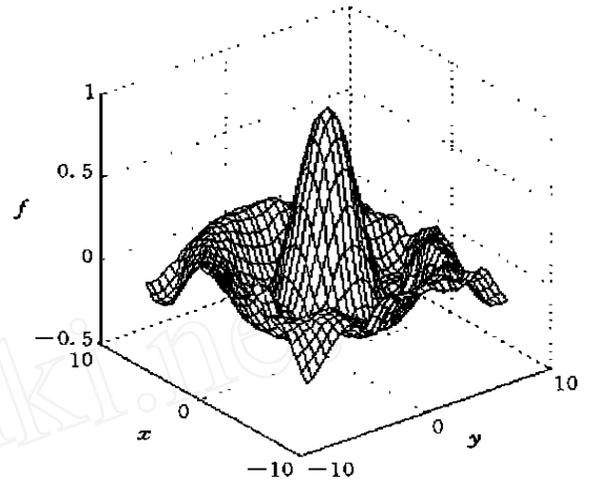


图 3 增量 CMAC 算法逼近结果

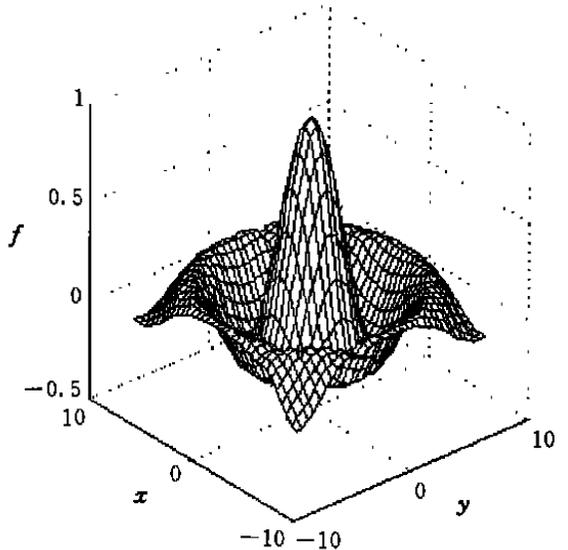


图 4 学习率自寻优 CMAC 算法逼近结果

接近, 得出的最优学习率是 0.0956, 学习 500 步时的网络输出均方差为 $2.234e-4$ 。如果加大学习步长, 误差还将继续收敛。而增量 CMAC 算法 ($\beta = 1$) 学习约 100 步时, 网络输出均方差达到 $3.256e-3$, 此后误差便不再收敛。

仿真结果表明, 学习率自寻优的 CMAC 改进算法在一维和多维输入条件下, 均比一般的 CMAC 批量算法和增量算法收敛速度快, 而且克服了传统算法需事先试凑学习率的困难, 可以自动地在学习过程中搜索出收敛所需的学习率取值范围和近似最优学习率。

4 CMAC 泛化能力的研究

泛化能力是指将输入集中样本点的给定邻域映射到输出集中映射点 (与样本点对应) 的某一邻

域的能力^[15]。在CMAC网络中,若两个输入向量相距较近,则它们所触发的神经元有重叠,距离越近,重叠越多。若两个输入向量相距较远,则它们触发的神经元没有重叠。因此CMAC网络具有局域泛化能力,它的泛化能力源于网络结构本身。

对于一个神经网络而言,泛化能力越强,意味着经样本点训练后,对于样本点附近非样本点(即测试点)的输入,网络输出与期望输出间的误差越小。设测试点序列 t_i 对应的期望输出为 y_i ,CMAC输出为 $f(t_i)$, $i=1,2,\dots,n$,则可用测试点的误差平方和

$$SSE = \sum_{i=1}^n (f(t_i) - y_i)^2 \quad (20)$$

作为指标,来衡量网络整体泛化性能的优越,SSE越小,网络泛化能力越强。

设CMAC网络量化间距为 q_{ti} ,训练样本采样间隔为 $sp1$,泛化常数为 g ,学习最大步长为 $max\text{-epoch}$,测试点采样间隔为 $test\text{-}sp1$,物理存储单元数为 $100g$ 。下面分3种情况用CMAC逼近式(17)所示函数,研究各个参数变化对其泛化性能的影响。

4.1 量化精度等于样本采样精度

参数取值: $q_{ti}=sp1=0.10$, $test\text{-}sp1$ 分别取 0.01 和 $sp1$ (分别偏移20%,50%,80%), $max\text{-epoch}=1000$,仿真结果如表1所示。

由表1可以看出,当 $q_{ti}=sp1$ 时, g 的变化不影响测试点SSE,只影响样本点SSE;在样本点附近

(偏移20%和80%)的测试点SSE明显小于距样本点较远(偏移50%)的测试点SSE。说明当 $q_{ti}=sp1$ 时, g 的增大并不能提高网络的泛化性能,而CMAC网络固有的结构特点保证了其在样本点邻域内具有一定的泛化能力。

4.2 量化精度小于样本采样精度

这种情况下,当采用样本点训练时,CMAC网络收敛速度变得很慢,并且误差收敛到一定范围后便不再收敛。因此这种情况下讨论 g 对网络泛化性能的影响没有任何意义。在训练CMAC时也要避免采用这种参数设置。

4.3 量化精度大于样本采样精度

参数取值: $q_{ti}=0.05$, $sp1=0.1$, $test\text{-}sp1$ 分别取 0.01 和 $sp1$ (分别偏移20%,25%,50%,75%,80%), $max\text{-epoch}=1000$,仿真结果如表2所示。

由表2可以看出,当 $q_{ti}<sp1$ 时:1) g 只有大于某一数值时网络才收敛,收敛后 g 的增大不能提高网络的泛化性能;2)与非样本量化节点(测试采样间隔取 $sp1$ 并偏移50%)重合的测试点SSE明显小于其它测试点SSE;3)表1和表2在样本点邻域(偏移20%和80%)的测试点SSE数值几乎完全相同,说明在样本点邻域内具有的泛化能力的确是CMAC网络结构本身赋予的;4)同表1相比,表2在 $test\text{-}sp1=0.01$ 的测试点SSE小,说明量化精度比样本采样精度大时,CMAC网络整体泛化性能好。

表1 $q_{ti}=sp1$ 时 g 的变化对CMAC泛化性能的影响

g	样本点 SSE	测试点 SSE			
		0.01	sp1(偏移 20%)	sp1(偏移 50%)	sp1(偏移 80%)
1	7.392 1e-12	12.136 3	0.579 5	3.578 6	0.578 5
4	2.885 3e-6	12.136 5	0.579 5	3.579 2	0.578 5
10	5.398 7e-7	12.136 5	0.579 5	3.578 8	0.578 5
50	6.768 3e-8	12.136 3	0.579 5	3.578 6	0.578 4
100	1.960 8e-8	12.136 3	0.579 6	3.578 5	0.578 4

表2 $q_{ti}<sp1$, $q_{ti}=0.05$ 时 g 的变化对CMAC泛化性能的影响

g	样本点 SSE	测试点 SSE					
		0.01	sp1(+ 20%)	sp1(+ 25%)	sp1(+ 50%)	sp1(+ 75%)	sp1(+ 80%)
1		训练时未收敛,误差很大					
2	3.277 8e-11	3.204 5	0.579 5	0.936 3	0.074 06	0.940 4	0.578 5
4	6.170 1e-6	3.204 5	0.579 5	0.936 3	0.074 07	0.940 4	0.578 5
10	5.353 0e-7	3.204 5	0.579 5	0.936 3	0.074 06	0.940 5	0.578 5
50	1.979 9e-8	3.204 4	0.579 6	0.936 2	0.074 05	0.940 4	0.578 4
100	1.292 1e-9	3.204 5	0.579 5	0.936 3	0.074 06	0.940 4	0.578 4

表 3 $qti < spl, qti = 0.03$ 时 CMAC 的泛化性能

g	样本点 SSE	测试点 SSE				
		0.01	sp1(+ 20%)	sp1(+ 30%)	sp1(+ 50%)	sp1(+ 80%)
1						
2		训练时未收敛				
3						
4	5.429 6e-11	102.772 0				
6	1.260 7e-10	24.480 4				
9	1.776 2e-10	11.154 3				
10	8.951 6e-5	1.832 6	0.312 3	0.074 85	0.332 5	0.310 1
50	5.689 1e-6	1.832 6	0.312 0	0.074 08	0.332 8	0.311 4
100	9.890 3e-9	1.834 8	0.312 2	0.074 46	0.333 1	0.311 5

表 4 CMAC 增量算法在 $\beta = 1$, 误差目标为 10^{-6} 时的运算次数

泛化常数 g	1	2	4	10	50	100
收敛步数	1	588	1 238	938	674	798
浮点运算次数	1.72e4	7.79e5	2.75e6	4.71e6	1.60e7	3.75e7

是否量化精度越大, 网络泛化性能越好呢? 参数取值: $qti = 0.03$, $sp1 = 0.1$, $test-sp1$ 分别取 0.01 和 $sp1$ (分别偏移 20% , 30% , 50% , 80%), $max-epoch = 1000$, 仿真结果如表 3 所示。

由表 3 可以看出, 当 $qti < spl$ 时, 量化间隔 qti 小到一定程度, 网络收敛要求的泛化常数 g 相应地增大, 随着 g 增加, 泛化性能在一定范围内变好, 但当 g 达到一定程度后 (本例中等于 10), g 的增加不再影响网络泛化性能。

表 1、表 2 和表 3 在收敛时都是 g 取某一个较小值, CMAC 样本点的 SSE 最小, g 一旦大于这个数值, 样本点 SSE 便增大几个数量级, 此后随着 g 的增大, 样本点 SSE 再逐渐减小, 但对网络泛化性能没有任何影响。而 g 的增大, 意味着运算量成倍增加 (如表 4 所示)。因此, 对于给定的误差精度, g 的选取有一个相对最优值, 表 1 的情况下可取 1, 表 2 的情况下可取 2, 表 3 的情况下可取 10。

综上所述, 关于 CMAC 各参数对网络泛化性能的影响可得以下结论:

1) CMAC 网络的结构本身保证了对于训练样本点邻域内的测试点具有一定的泛化能力;

2) 在训练样本采样精度不变的条件下, 量化精度越高, CMAC 网络整体泛化性能越好, 但随着量化精度的提高, CMAC 学习收敛要求的最小泛化常数随之增大;

3) 在训练样本采样精度和网络量化精度保持不变的条件下, 泛化常数的增加可在一定范围内提

高 CMAC 网络整体的泛化性能, 但当泛化常数增大到一定程度后, 它的变化不再影响网络泛化性能;

4) 在训练样本采样精度和网络量化精度保持不变的条件下, CMAC 泛化常数存在一个相对最优值, 可保证样本点和测试点误差都较小。

5) 为保证 CMAC 收敛, 应避免采用网络量化精度小于训练样本采样精度的参数配置训练网络;

6) 提高训练样本采样精度, 可提高 CMAC 网络整体泛化性能, 但同时必须保证网络量化精度等于或大于样本采样精度。

5 结 论

CMAC 网络具有学习算法简单, 收敛速度快和局域泛化等优点, 但对其收敛性的结论都是在关联矩阵正定的特殊条件下得出的。传统 CMAC 算法在应用中存在以下问题: 一是学习率的确定非常麻烦; 二是如何提高网络泛化能力缺乏有力的理论依据。为解决上述问题, 本文在不附加任何特殊条件的情况下, 利用矩阵理论和线性方程组迭代收敛的一般性原理, 分析并证明了 CMAC 算法在批量和增量两种学习方式下的收敛定理。以此作为理论依据, 提出一种学习率自寻优的 CMAC 改进算法, 同时提出一种简单可行的评价 CMAC 网络整体泛化性能的指标。通过计算机仿真实验, 验证了收敛定理的正确性和改进算法的优越性, 并研究得出了 CMAC 网络各个参数对其泛化能力影响的相关结论。

(下转第 534 页)

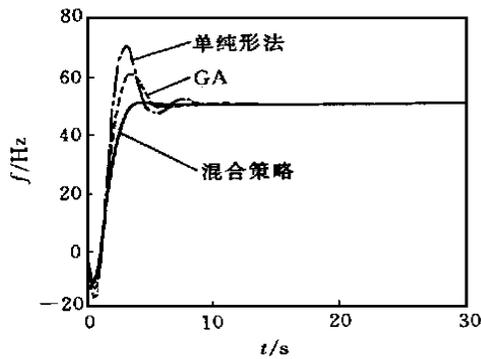


图4 对水轮机系统的整定结果比较

6 结 论

本文通过合理地融合 SA 和 GA, 构造出一类高效混合策略, 解决了参数估计和控制器参数的整定问题, 仿真结果验证了混合策略的优越性, 表现为对模型类型具有适应性且估计效果较正确, 控制品质超调小, 调整时间短, 误差指标小。由于所研究问题的重要性和 PD 应用的广泛性, 本文方法具有良好

的应用前景。进一步的研究目标将是在线估计和整定问题。

参考文献:

- [1] Michalewicz Z. Genetic algorithms+ data structures= evolution programs[M]. Berlin: Springer-Verlag, 1994
- [2] 姜波, 汪秉文. 基于遗传算法的非线性系统模型参数估计[J]. 控制理论与应用, 2000, 17(1): 150-152
- [3] 黄炯, 邬永革. 基于遗传算法的系统在线辨识[J]. 信息与控制, 1996, 25(3): 171-176
- [4] Varsek A, Urbancic T, Filipic B. Genetic algorithms in controller design and tuning[J]. IEEE Trans on Systems, Man and Cybernetics, 1993, 23(5): 1330-1339
- [5] Visioli A. Fuzzy logic based setpoint weight tuning of PD controllers[J]. IEEE Trans on Systems, Man and Cybernetics, 1999, 29(6): 587-592
- [6] Kirkpatrick S, Gelatt C, Vecchi C. Optimization by simulated annealing[J]. Science, 1983, 220(4598): 671-680
- [7] Liu L, Mao Z. Water turbines PD controller based on genetic algorithm[J]. Automation of Electric Power Systems, 1997, 21(12): 41-43

(上接第 529 页)

参考文献:

- [1] J S Albus. A new approach to manipulator control: The cerebellar model articulation controller (CMAC) [J]. J of Dyn Syst, Meas and Contr, 1975, 97(3): 220-227.
- [2] J S Albus. Data storage in the cerebellar model articulation controller (CMAC) [J]. J of Dyn Syst, Meas and Contr, 1975, 97(3): 228-233
- [3] W Thomas Miller, Filson H Glanz, L Gordon Kraft. CMAC: An associative neural network alternative to backpropagation[J]. Proc of the IEEE, 1990, 78(10): 1561-1567.
- [4] S Cetinkunt, A Donmez. CMAC learning controller for servo control of high precision machine tools[A]. Proc of American Contr Conf[C]. San Francisco, 1993. 1976-1980
- [5] Gary A Larsen, Sang S Ku, Sabri Cetinkunt. Low speed motion control experiments on a single point diamond turning machine using CMAC learning control algorithm [J]. Proc of ASME Dyn Syst and Contr Div, 1995, 57(1): 497-501.
- [6] G A Larsen, S Cetinkunt, A Donmez. CMAC neural network control for high precision motion control in the presence of large friction[J]. J of Dyn Syst, Meas and Contr, 1995, 117: 415-420
- [7] Sang-soon Ku, Gary Larsen, Sabri Cetinkunt. Fast tool servo control for ultra-precision machining at extremely low feed rates[J]. Mechatronics, 1998, 8(4): 381-393
- [8] Touretzky D S. Neural information processing systems (1) [M]. Los Altos: Morgan Kaufmann, 1989. 29-39
- [9] Yiu-fai Wong, Athanasios Sideris. Learning convergence in the cerebellar model articulation controller [J]. IEEE Trans on Neural Networks, 1992, 3(1): 115-121.
- [10] 张铃, 张钊. 神经网络理论及应用[M]. 杭州: 浙江科学技术出版社, 1997. 214-261.
- [11] 罗忠, 谢永斌, 朱重光. CMAC 学习过程收敛性的研究 [J]. 自动化学报, 1997, 23(4): 455-461.
- [12] 丁丽娟. 数值计算方法[M]. 北京: 北京理工大学出版社, 1997. 54-59
- [13] 史荣昌. 矩阵分析[M]. 北京: 北京理工大学出版社, 1996. 134-138
- [14] 李兵. 混沌搜索与随机搜索相结合的优化方法[A]. 1999 年中国智能自动化学术会议论文集[C]. 北京: 清华大学出版社, 1999. 823-826
- [15] 欧阳楷, 陈卉, 周萍, 等. 神经计算中坐标变换的网络模型(CMAC)的泛化特性[J]. 自动化学报, 1997, 23(4): 475-481.