

直接映射低维小脑模型神经网络 及在机器人传感器中的应用

朱庆保

(南京师范大学计算机科学系 南京 210097)

摘 要 提出了一种能高速度、高精度学习的低维小脑模型神经网络. 模型算法采用直接权地址映射技术, 将训练样本的输入量化后直接作为联想存储器中 C 个权的权首地址, 建立起输入与权的关系. 经样本训练后, 任意输入作为相近的两个样本间的权首地址, 经过输出映射算法即可得到较精确的输出. 实验表明, 它学习非线性函数的精度比最新改进的 CMAC 高十倍以上, 收敛速度则快五十倍以上, 且算法简单, 不会发散, 学习过程要求的存储器很小, 实现容易. 此算法已用于机器人传感器的信号细分, 取得了非常好的效果.

关键词 CMAC 神经网络; 直接映射; 机器人; 传感器

中图法分类号 TP18

A Direct Weight Address Mapping LCMAC Neural Network and Its Application in Robot Sensor

ZHU Qing-Bao

(Department of Computer Science, Nanjing Normal University, Nanjing 210097)

Abstract A novel computational approach of the low-dimensional cerebella model articulation controller(LCMAC), with the features of significantly improved processing precision and speed of convergence on learning low-dimensional nonlinear functions, is proposed in this paper. The direct weight address mapping techniques are employed in the underlying algorithm of the model, and the relationship between the inputs and weights is established by taking the scaled inputs of training samples as the head address of C weights unit in the associative memory. After training with the samples, any input is taken as the head address of C weights between the two similar samples. The subsequent accurate outputs are obtained by output mapping algorithm. Simulation experiments show that the accuracy of learning and approximating nonlinear functions is over ten times higher, and the speed of convergence is over fifty times higher than the latest improved CMAC. The algorithm, which requires a small memory storage in learning process, is very simple and also easy to be implemented. It has been applied successfully for subdividing signals of robot sensor and the result is very satisfying.

Keywords CMAC neural network; direct weight address mapping; robot; sensor

1 引 言

20 世纪 70 年代 Albus 提出的 CMAC (Cere-

bellar Model Articulation Controller) 神经网络^[1], 至今已得到了广泛的应用. 由于它是基于局部学习的网络, 因此学习速度比 BP 网络快得多, 特别适于实时学习控制、非线性函数映射等领域. 但是当用 CMAC

学习非线性函数时,CMAC 是用 Hash 函数实现从概念存储器到物理存储器的映射.使用 Hash 映射存在一些问题,一方面构造一个好的 Hash 函数往往比较困难.另一方面,也正是由于 Hash 映射有可能使得 CMAC 在某些学习过程中不收敛或使收敛性态变坏,至少使收敛过程变慢^[2,3].若不用 Hash 映射,虽保证了学习过程的收敛性,但学习过程所占用的容量与学习函数的维数成指数关系迅速膨胀,存储空间利用率极低,实现困难^[4].此外,由于不同的输入有可能激励相同的权单元,因此 Albus 的 CMAC 存在着学了新的忘了旧的遗忘现象^[5].如何解决这些问题呢?在实际应用中,许多应用问题都是求解一维函数问题,许多多维问题可以分解成一维问题来解决.例如目前大多数传感器都是一维的;一些经济型数控机床的多维坐标进给就是分维分时控制的,可以将其分解成几个一维控制.因此,研究一种能高精度映射一维非线性函数的 CMAC 是十分必要的,根据 CMAC 的基本原理,作者研究了一

种用直接权地址映射技术实现的低维 CMAC,称它为 LCMAC(低维小脑模型关节控制控制器).LCMAC 具有学习样本少、收敛快、学习精度高、占用内存少、实现容易(即使用单片机也容易实现)等特点.作者用 LCMC 实现机器人速度、位置传感器的信号细分,取得了非常好的效果.

2 LCMAC 的基本原理结构及基本算法

2.1 基本原理结构

LCMAC 的基本原理结构如图 1(a)示, \mathbf{X} 为输入状态空间,由 N 个低维的输入向量组成,每个输入向量映射到 A 中的不同区域(在图 1(a)中仅示出了 \mathbf{X}_1 和 \mathbf{X}_2 两个输入向量). A 和 CMAC 的 A 相同,是联想存储器,是可实现的物理存储器.因为每个向量映射到 A 中不同的区域,其映射方法完全相同.为了说明原理,先讨论一个一维输入向量的映射算法.

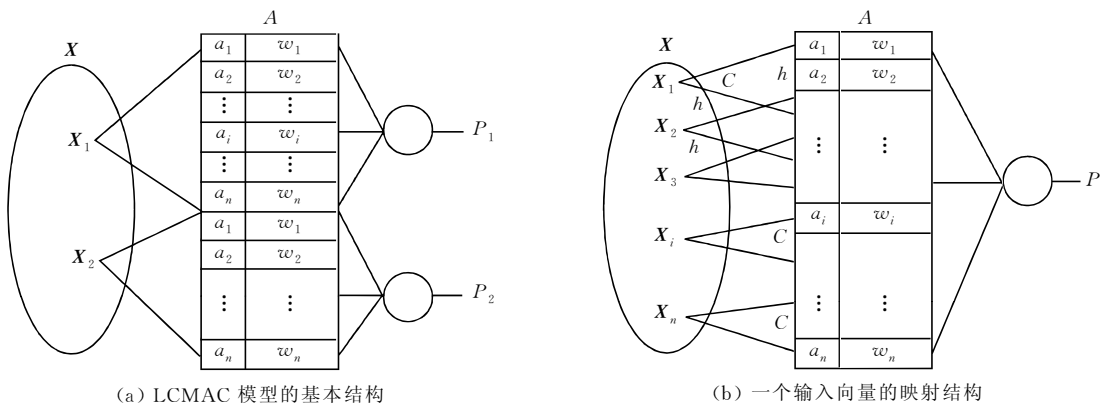


图 1

一个一维输入向量的映射结构如图 1(b)所示, x_1, x_2, x_i, x_n 为输入向量 \mathbf{X} 的样本输入.一个训练样本的输入映射到存储器 A 中 C 个单元(a_i 是权地址), C 为 LCMAC 视野参数.这样,每一个单元有一个权值.对于一个给定的输入 x_j ,产生一个输出 $p_j = F(x_j)$.为了对给定的输入 x_j 计算输出 p_j ,也进行了二次映射: $f: \mathbf{X} \rightarrow A, g: A \rightarrow P$.

2.2 $\mathbf{X} \rightarrow A$ 的映射

设训练样本为 $(x_i, y_i) (i = 1, 2, \dots, n)$, 为 n 个离散点的集合,且有 $x_1 < x_2 < \dots < x_n$, 其中 x_i 为给定的输入, y_i 为期望的输出.为了把输入映射到 A 中,令 x_i 作为 A 中 C 个权单元的逻辑首地址, $x_i \in [0, R]$ 是整型值,从而激活从地址 x_i 开始的 C 个权.对于非样本的任意输入 x_k ,则作为相近的两个样本间的权单元首地址,也激活从地址 x_k 开始的 C 个权.

实际的输入 \mathbf{X} 多为模拟量,每个输入值可以是小数、负数等在某一区间上的任意数.根据上述模型原理,必须将任意输入值进行量化、坐标平移,变换成 $[0, R]$ 区间的整型值.对于每一个输入元素 x_j ,将其量化为 $0 \sim R$ 范围内的一个整数,算法如下:

$$\Delta x_j = \frac{x_{j\max} - x_{j\min}}{R}, s = \text{int} \left(\frac{x_j - x_{j\min}}{\Delta x_j} \right) \quad (1)$$

式中, $x_{j\max}$ 为 x_j 的最大值; $x_{j\min}$ 为 x_j 的最小值, Δx_j 为量化分辨率; s 为输入元素量化后的整型值.

2.3 训练算法

设经上述量化后的训练样本的输入为 $s_1, s_2, \dots, s_i, \dots, s_n, s_{i+1} = s_i + h, h$ 为样本间隔;期望的输出为 $\mathbf{Y} = \{y_1, y_2, \dots, y_j, \dots, y_n\}$.对于输入 s_i ,LCMAC 的实际输出 p_i 由式(2)给出:

$$p_i = \sum_{j=i}^{i+C} \omega_{ij} \quad (2)$$

式中, w_{ij} 为第 i 个样本的第 j 个权值; C 为视野参数.

如果 $y_i - p_i > \epsilon$, 则按式(3)调整权值, 直到 $y_i - p_i \leq \epsilon$ 为止. 经过反复若干次训练, 直到所有样本的输出与其期望输出之差 $\leq \epsilon$ 时训练结束.

$$\Delta w_{ij} = \beta \frac{y_i - p_i}{C} \quad (3)$$

式中, ϵ 为给定的学习误差, β 为学习步长. C 的值必须大于等于 h , 才能实现泛化功能, 但 C 太大, 会使学习速度降低. C 值还会影响到学习精度. 由于 LCMAC 训练速度非常快, 因而可用实验法确定最佳 C 值. 此外, 可采用自适应算法确定 ϵ 和 β 的值, 以达到动态的最佳值.

2.4 A→P 的映射

训练完毕后, 对于任意输入 x_k , 经式(1)量化后得到 s_k , 令 $s_i - s_k = m, k = i + m$, 则有

$$p_k = \sum_{j=k}^{k+C} w_{kj} \quad (4)$$

其中, s_i 是与 s_k 相近且小于 s_k 的输入样本; p_k 即为任意输入 x_k 对应的输出. 也就是让 s_k 作为相近样本 $s_i + m$ 处的权地址, 从此处激活 C 个权, C 个权的响应即为输出. 很显然, 当 $m=0$ 时, $s_k = s_i$ 即为样本点的输出.

2.5 收敛性分析

Albus 提出 CMAC 时没有给出收敛性证明, Wong^[6] 等人从线性方程组迭代的角度研究了 CMAC 算法的收敛性. 在此基础上, 文献[2, 3, 7]的研究结论指出: 如果不使用 Hash 编码, CMAC 一定是收敛的. LCMAC 与 CMAC 的最大差别就是 LCMAC 不使用 Hash 编码, 而是采用了直接权地址映射技术, 且仅限于低维应用, 由文献[2, 3, 7]的研究成果推知, LCMAC 的学习过程一定是收敛的.

3 仿真实验

为了在脱离硬件的情况下即能检验本算法的正确性及精度, 用仿真方法进行实验. 因为本课题的应用对象是机器人速度、位置传感系统, 其输出为正弦信号, 因此, 通过让 LCMAC 学习一个正弦函数(式(5))来检验本文算法的精度.

$$u = \sin\theta, \quad \theta \in [0^\circ, 360^\circ] \quad (5)$$

3.1 学习训练

在区间 $[0^\circ, 360^\circ]$ 上取 n 个训练样本 (θ_i, u_i) , 且有 $\theta_{i+1} = \theta_i + h, i = 1, 2, \dots, n, \theta_i$ 作为网络的输入, u_i 为期望的输出作为教师信号, 当样本点的间隔 h 分别取 10, 20, 30, 60 时, 训练样本数 $n = 360/h + 1$, 分别为 37, 19, 13, 7. 相应的教师信号 u_i 由式(5)给出.

用这些训练样本逐一地对网络进行训练, 其中, 用式(2)计算每个样本实际的输出, 用式(3)调权, 直至所有样本的输出误差小于给定误差时训练结束.

3.2 仿真实验结果

仿真结果见表 1. 其中的数值来源及算法如下: 令 $\theta_i = \{0^\circ, 1^\circ, 2^\circ, \dots, i^\circ, \dots, 360^\circ\}$ 作为检验网络精度的数据, 则有

$$e_i = \text{fabs}(u_i - p_i)$$

$$\text{aver} = Q/n = \sum e_i/n, \text{max} = \max\{e_i\},$$

$$\text{min} = \min\{e_i\}, i \in [0^\circ, 360^\circ].$$

式中, u_i 为用式(5)计算出的对应每个输入的理论输出值; p_i 为网络的实际输出; e_i 为误差的绝对值; n 为误差的个数; aver 为平均误差.

由于数据处理量较大, 为此, 我们用 C 语言编写了一个检验函数来完成这一工作. 表 1 的数据由该函数完成. 表 1 中, $\text{max}, \text{aver}, \text{min}$ 分别为最大误差、平均误差和最小误差.

表 1 仿真学习结果 (单位: °)

h	样本数	max	aver	min
10°	37	0.0015	0.0001	0.0000
20°	19	0.0032	0.0002	0.0000
30°	13	0.0143	0.0026	0.0000
60°	7	0.2361	0.0833	0.0000

表中数据表明, 当样本间隔为 30 时, 在区间 $[0^\circ, 360^\circ]$ 内仅用 13 个样本数据对网络进行训练, 训练后, 在区间内任意输入的最大输出误差只有 0.0143, 说明了本算法具有极强的泛化能力.

4 LCMAC 在机器人速度、位置传感器中的应用

LCMAC 可以学习种类广泛的一维非线性函数, 多维问题只要能分解成一维问题, 也可用 LCMAC 映射. 其学习速度比 CMAC 快得多, 可广泛应用于实时学习控制、非线性函数映射等. 在实际应用中, 要 LCMAC 学习的函数不一定有明确的解析式子, 但可以通过给定一组输入信号, 根据题目要求, 设计出一组期望的输出作为教师信号, 经 LCMAC 学习后, 即可以由 LCMAC 根据任意输入求解输出. 其中, 输入可以是由传感器获取的信号, 也可是输入等其它来源. 作者已用 LCMAC 实现机器人速度、位置传感器的信号细分, 取得了非常好的效果.

4.1 用 LCMAC 解决机器人传感器信号细分问题的提出

在很多应用场合, 需要研制一些能高精度运行的小型或微型机器人, 由于传统的机器人位置、速度

传感器的体积都较大,精度都不是很高,它成为研制小型机器人的瓶颈之一.因此,在国家自然科学基金的资助下,我国研制成一种新颖的机器人位置、速度控制传感器测量系统,该成果在传感器制造工艺上有重大突破,即解决了计量光栅的高密度刻线问题,使得刻线分辨率达到 250 线对/mm 以上^[8].这种方法确实在制造工艺和技术水平上都是很高的,系创新性优秀成果.然而,提高单位刻线数的主要目的是提高其分辨率,这种方法制造工艺相对较复杂,刻线难度很大,一般单位难以生产该产品,成品率相对较低,要想进一步提高其刻线数或进一步减小传感器的体积则十分困难.为此,作者从另一个角度研究了一种不用提高刻线数,仅用最普通的刻线工艺、很少的刻线数(整个圆光栅仅用 360 条刻线)即能达到高分辨率且具有智能的机器人速度、位置的传感系统,其最基本的原理是利用人工神经网络的泛化能力实现光栅信号的高精度连续细分,只要给出极少量的学习样本让神经网络进行学习后,对任意的输入值,经神经网络泛化后,都可以高精度输出.由于这种智能传感器的刻线数很少、使用新型集成光电转换器件进行光电转换、用人工神经网络进行连续细分、取消了传统的硬件细分电路,因此,不仅可使分辨率做得很高、硬件电路大大减化、可靠性大大提高、体积可做得很小,且制造容易、造价较低.随着集成光电转换器件体积的不断减小,它的体积还存在不断减小的可能.从另一方面,使机器人传感系统具有学习能力等智能,符合机器人的特征,是未来型机器人的发展趋势.查新表明,用神经网络实现光栅信号细分尚未见报道.

作者用 LCMAC 神经网络实现这种机器人小型传感系统的信号细分,取得了非常好的效果,已申请一项发明专利(已公告)和一项实用新型专利(已授权).

4.2 用 LCMAC 细分机器人传感系统的信号的基本原理

机器人位置、速度传感系统常用光栅及光电转换元件为检测部件,其输出信号可表示为

$$u = U_0 + U_m \sin(2\pi x/w) \quad (6)$$

式中, u 为光栅传感器的输出信号; U_0 为输出信号中的直流分量; U_m 为输出信号的幅度; x 为光栅的瞬时角度; w 为光栅的栅距.可见, u 反映了瞬时角度 x 的大小.当 x 变化了一个栅距时,则相当于电角度变化了 2π .如采用 360 线/周的圆光栅,当主光栅转动了 x 时,也就是转动了 x 度.通过计数输出信号的周期数,即可测得 x 值(角度值).

为了判别光栅移动的方向及便于细分,通常安

装 4 个光电接收元件,得到四路输出信号,一路近似为正弦信号,其余三路依次相位相差 90° ,图 2 示出了其中两路相位相反的信号.

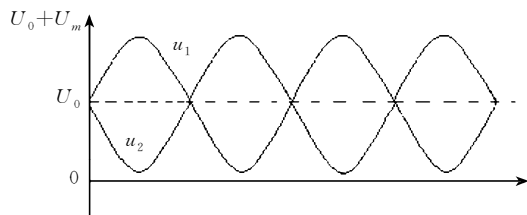


图 2 光栅传感器的输出信号

如果仅通过计数输出信号的周期数来测量光栅转动的角度,其分辨率仅为光栅的栅距,若采用 360 线/周的光栅,其分辨率仅为 1° .这一分辨率是不能满足大多数应用要求的.因此,为了得到比栅距更小的角度值,必须对输出信号进行细分,通常采用内插法进行细分,即在信号变化的一个周期内,给出 N 个计数脉冲,以减小脉冲当量(即每个脉冲所相当的角度值减小到原来的 $1/N$,则分辨率提高 N 倍).传统的细分方法用硬件实现,存在硬件复杂、细分数不高等若干缺点,因此,作者研究了用 LCMAC 进行细分的方法,可实现连续细分,使分辨率得到了很大提高,大大简化了硬件设计,从而提高了系统的可靠性.

由图 2 可见, $\sin\theta$ 在区间 $(0^\circ, 180^\circ)$ 上为正值,在 $(180^\circ, 360^\circ)$ 上为负值, $0^\circ, 180^\circ$ 和 360° 时为 0 值,根据采样值的正负变化即可判定一个信号周期.测量时,微机记忆信号周期起始值,用软件累加信号周期数得到光栅移动 k 个栅距对应的角度值.如何在一个周期内进一步细分呢?用 LCMAC 神经网络细分的基本方法如下:

取 n 个训练样本 (u_i, θ_i) 对网络进行训练,且有 $\theta_i = 360x_i/2\pi, \theta_i \in [0^\circ, 360^\circ], u_{i+1} = u_i + h$,其中, $i = 1, 2, \dots, n, h$ 为样本间隔, u_i 是传感器输出信号经 A/D 转换得到的值; θ_i 为期望的输出作为教师信号,可用高精度标准角度装置给出.改变角度标准器,使 $u_i = u_{i-1} + h$ 时,由微机记忆相应的采样值 u_i ,并将标准器给出的 θ_i 键入微机,即可用 (u_i, θ_i) 作为训练样本对网络进行逐一训练.其中,用式(2)计算每个样本实际的输出,用式(3)调权,直至所有样本的输出误差小于给定误差时训练结束.训练完毕后,A/D 采样后的任意值,作为网络的输入,网络的输出即为细分值,也就是得到在最后一个栅距内的微角度.光栅转动的总角度 β 可表示为

$$\beta = kw' + d \quad (6)$$

式中, w' 为度/栅距, k 为信号整数周期数; d 为神经网络的输出,是当前一个栅距内的微角度.这样用少

量的训练样本即可得到一个高精度的连续的细分值. 由于这一过程是通过学习实现的, 在更换传感器或元器件时只要重新学习一遍即可, 具有较好的可维护性.

4.3 实际应用情况

在实用中, 首先用硬件去掉式(6)中的直流分量, 即成为一个不含直流分量的正弦函数, 可表示为

$$u = U_m \sin(\theta), \text{ 其中 } \theta = 2\pi x/w \quad (7)$$

按 4.2 节方法取 n 个训练样本 (u_i, θ_i) 对网络进行训练. 训练完毕后, 将权值存入数据文件或写入非易失 RAM 中, 即可应用该训练过的网络完成对光栅信号的连续细分, 亦即光栅信号经 A/D 采样后作为网络的输入, 网络的输出即为细分值, 也就得到了最后一个栅距内的微角度值. 为了使算法更简单及得到一个单值输入, 根据输入信号特点分为四个区间进行学习和应用, 并根据光栅的输出是四路互差 90° 的信号这一特点, 在端点处通过学习光栅的另两路信号进行补偿, 得到了更高精度的细分. 其结果见表 2.

表 2 实际应用结果

样本数	max	aver	min
18	0.00065	0.000021	0.0000
9	0.00042	0.000035	0.0000
7	0.00252	0.000382	0.0000

表中数据表明, 当仅用 7 个样本对网络进行训练时, 训练后, 在区间内任意输入的最大输出误差只有 0.0052, 说明了本算法具有较强的泛化能力. 用这种方法仅使用 360 刻线的光栅, 即可实现 0.0052° 的分辨率. 用 LCMAC 神经网络细分, 去掉了原来的细分电路, 使硬件电路大大简化, 从而提高了系统的可靠性. 并且, 由于刻线少, 可以把光栅做得很小, 有利于实现微型化.

5 结 语

本文提出的 LCMAC 神经网络可用于解决广泛存在的低维问题, 仿真实验和实际应用都证明, LCMAC 具有算法简单、实现容量(用单片机也容易

实现)、学习速度快、精度高等特点. 然而, LCMAC 并不是万能的, 当学习信号的变化率很大时, 学习精度会受到一定影响, 这时可适当增多训练样本, 并根据信号特点采用非均匀间隔样本对 LCMAC 进行学习训练, 可满足大部分应用要求.

LCMAC 已在机器人传感器的信号细分中取得了非常成功的应用, 并且已获得一项实用新型专利、一项发明专利(已公告). LCMAC 具有一定的推广应用价值.

参 考 文 献

- 1 Albus J S. A new approach to manipulator control: The cerebellar model articulation controller (CMAC). Transactions of the ASME, Journal of Dynamic System, Measurement and Control, 1975, 97(9): 220~227
- 2 Lin Chun-Shin, Chiang Ching-Tsan. Learning convergence of CMAC technique. IEEE Transactions on Neural Networks, 1997, 8(6): 1281~1292
- 3 Luo Zhong, Zhao Zhong-Ming, Zhu Chongguangle. The unfavorable effects of Hash coding on CMAC convergence and compensatory measure. In: Proceedings of IEEE International Conference on Intelligent Processing Systems, New York, USA, 1997, 1: 419~422
- 4 Jar-Shone Ker, Yau-Hwang Kuo. Hardware implementation of CMAC neural network with reduced storage requirement. IEEE Transactions on Neural Networks, 1997, 8(6): 1545~1556
- 5 Liu Hui, Xu Xiao-Ming. An improved CMAC neural network algorithm. Acta Automatica Sinica, 1997, 23(4): 482~487 (in Chinese)
(刘 慧, 许晓鸣. 小脑模型改进算法的研究. 自动化学报, 1997, 23(4): 483~487)
- 6 Wong Y F, Sideris A. Learning convergence in the cerebellar model articulation controller. IEEE Transactions on Neural Networks, 1992, 3(1): 115~121
- 7 Luo Zhong, Xie Yong-Bin, Zhu Chong-Guang. A study of the convergence of the CMAC learning process. Acta Automatica Sinica, 1997, 23(4): 455~461 (in Chinese)
(罗 忠, 谢永斌, 朱重光. CMAC 学习过程收敛性研究. 自动化学报, 1997, 23(4): 455~461)
- 8 Yu Hong-Lin, He An-Guo, Liu Sheng *et al.* Position and speed control sensor of microrobot. In: Proceedings of International Conference on Sensors and Control Techniques (ICSC 2000), Wuhan, China, 2000. 418~419



ZHU Qing-Bao, born in 1955, professor. His research interests include AI and Intelligent control.