

DOI: 10.7652/xjtuxb201801019

双关节刚性机器人自适应 BP 神经网络算法

杨航, 刘凌, 倪骏康, 张诚
(西安交通大学电气工程学院, 710049, 西安)

摘要: 为解决反向传播(BP)神经网络学习速度慢、泛化能力弱以及易陷入局部极小值等问题,提出了一种双关节刚性机器人自适应 BP 神经网络算法。首先,设计了一种新结构的参数可调激活函数,其映射范围、斜率因子、水平和竖直位置等参数皆可自适应调整,以使 BP 神经网络具有更强的非线性映射能力;然后,设计了一个模糊推理器来整定激活函数的斜率因子,以使斜率因子保持参数最优;最后,设计了一套控制系统并应用于双关节刚性机器人系统的位置跟踪控制,采用自适应 BP 神经网络算法对该系统位置控制器的比例增益、积分增益和微分增益进行了实时整定。仿真结果表明:与使用经典固定参数 S 型激活函数的 BP 神经网络算法相比,所提出的自适应 BP 神经网络算法考虑了激活函数的自适应性,提高了 BP 神经网络的学习速度和泛化能力,并抑制了假饱和现象,对位置误差的收敛速度可提高 20 倍以上,且可使位置和速度误差减小并接近于 0。

关键词: 刚性机器人;反向传播神经网络;自适应;激活函数;模糊推理

中图分类号: TP183 **文献标志码:** A **文章编号:** 0253-987X(2018)01-0129-07

An Adaptive BP Neural Network Algorithm for 2-Joint Rigid Robots

YANG Hang, LIU Ling, NI Junkang, ZHANG Cheng
(School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: According to the existing problems that back-propagation (BP) neural network with fixed activation function parameters has the shortcomings of slow learning speed, weak generalization ability and ease of falling into local minimum, an adaptive BP neural network algorithm for 2-joint rigid robot is proposed. Firstly, a novel structure of activation function with adjustable parameters is designed. In order to make the BP neural network have better nonlinear mapping capability, its mapping range, steep degree, and horizontal and vertical position parameters can be adjusted adaptively. Then, a fuzzy controller is designed to adjust the slope factor. Thus, the optimality of the slope factor can be guaranteed. Finally, a control system is designed and applied to the tracking control of a 2-joint rigid robotic system. The adaptive BP neural network algorithm is adopted to tune the proportional, integral and differential gains of the position controller. Simulation results show that in comparison with the classical BP neural network algorithm based on Sigmoid activation function with fixed parameters, the adaptive BP neural network algorithm takes the adaptability of activation function into account, improves the learning speed and generalization ability, and restrains false saturation. The convergence rate of position errors can be increased by 20 times, and the position and velocity tracking errors can be

收稿日期: 2017-04-16。 作者简介: 杨航(1992—),男,硕士生;刘凌(通信作者),男,副教授。 基金项目: 国家自然科学基金资助项目(51307130);中央高校基本科研业务费专项资金资助项目(1191320056)。

网络出版时间: 2017-10-20

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1069.T.20171020.1622.008.html>

reduced to a small value close to zero with the proposed algorithm.

Keywords: rigid robot; back propagation neural network; adaptability; activation function; fuzzy inference

近年来,人们越来越关注机器人的跟踪控制问题^[1],所要实现的目标是减小机械臂实际移动轨迹和设定轨迹之间的偏差,并达到所期望的精度。然而,机器人系统是一个非线性、强耦合、高阶和多变量的系统,其数学模型是一个具有时变结构的非线性模型^[2]。由于机器人的动作常常可通过不同的方式和路径来完成,手臂运动解不唯一,故需要在一定约束条件下考虑优化决策与控制问题,但是基于传统算法的控制器已无法满足控制要求。随着机器人技术的发展,与其相关的控制算法逐渐丰富起来,如神经网络控制、自适应控制、自校正控制、鲁棒控制、变结构控制和预测控制等。由于神经网络需要相对较少的系统动力学信息,且反向传播(BP)神经网络已被证明能够在一定条件下以任何所需的精度逼近各种非线性函数,所以神经网络算法已广泛应用于机器人系统的控制^[3]。

文献[3-6]提出了一些使用神经网络技术解决机器人系统控制问题的方案。虽然大多数神经网络采用BP神经网络或其改进形式,但经典的BP神经网络有缺陷,这使它并不总能满足实际应用的需求。激活函数是神经网络解决非线性问题的关键,而经典的S型激活函数的映射范围、斜率因子及位置参数是固定不变的,故其在学习及训练过程中无法自动改变函数结构和寻找最佳参数,由此会导致BP神经网络易陷入局部极小值、收敛速度慢和泛化能力弱等问题的出现。

BP神经网络的学习能力与网络复杂度之间的矛盾制约了其非线性映射能力,导致学习速度和收敛精度不理想。近十几年来,针对BP神经网络算法的不足,研究人员做了大量深入的研究,并提出了许多改进算法,如使用动量项加快离线训练速度^[7]、快速传播算法^[8]、二阶优化^[9]以及最优滤波法^[10]等。文献[11]提出了参数可调的激活函数模型;文献[12]采用免疫遗传算法离线优化和神经网络自学习在线调整隶属函数参数的方法,增强了水下机器人对环境变化的反应能力;文献[13]提出了一种自适应T-S(Takagi-Sugeno)模糊神经网络控制器,以实现欠驱动无人艇直线航迹的跟踪控制。这些改进虽然改善了神经网络算法的性能,但只是简单调节了激活函数的陡峭程度和映射范围,而映射区间的

长度并不能改变,无法使神经网络达到最佳的非线性映射能力。

为了克服由于固定的S型激活函数参数所导致的问题,本文提出了一种基于新型参数可调整激活函数的自适应BP神经网络算法,以提高闭环系统的鲁棒性;随后,提出了一种新结构的激活函数,其映射范围、斜率因子、水平位置和竖直位置等参数皆可自适应调整,以使神经网络具有更强的非线性映射能力,并提高学习速度和泛化能力;最后,通过设计的模糊推理器来整定激活函数的斜率因子。仿真结果证实了本文算法的有效性。

1 刚性机器人系统的数学模型

对于一个 n 关节刚性机器人系统,考虑摩擦力的动态方程^[14]可以表示为

$$\mathbf{M}(q)\ddot{q} + \mathbf{V}(q, \dot{q})\dot{q} + \mathbf{F}(\dot{q}) + \mathbf{G}(q) = \boldsymbol{\tau} \quad (1)$$

式中: $\mathbf{M}(q) \in \mathbb{R}^{n \times n}$ 为对称正定的惯性矩阵; $\mathbf{V}(q, \dot{q}) \in \mathbb{R}^{n \times n}$ 为向心力和哥氏力矩阵; $\mathbf{F}(\dot{q}) \in \mathbb{R}^n$ 为摩擦力矢量; $\mathbf{G}(q) \in \mathbb{R}^n$ 为重力矢量; $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ 分别为机械臂的位置、速度和加速度; $\boldsymbol{\tau} \in \mathbb{R}^n$ 为力矩矢量。

双关节刚性机器人机械臂示意图如图1所示,其中: m_i, l_i 是第 i 个支架的质量和长度; q_1, q_2 为关节1和关节2的位置角(rad); l_{ci} 是从第 $i-1$ 个关节点到第 i 个支架质心的长度,假设连杆材质均匀,则 $l_{c1} = 0.5l_1, l_{c2} = 0.5l_2$; I_i 为第 i 个支架的转动惯量。

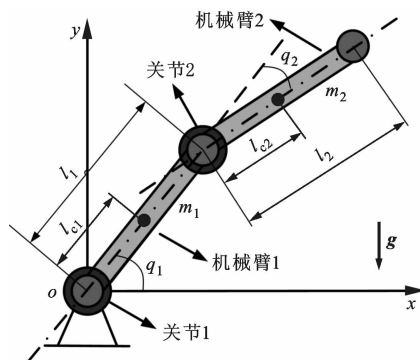


图1 双关节刚性机器人机械臂示意图

双关节刚性机器人的2个机械臂由2个永磁无刷电机驱动,而电机由电流伺服放大器驱动。伺服放大器的增益矩阵 $\mathbf{K}_{sa} = k\mathbf{I}$ ($k > 0$),其中 $\mathbf{K}_{sa} \in \mathbb{R}^{n \times n}$, $\mathbf{I}$ 是一个具有相应维数的单位矩阵, k 为正的

常数。永磁无刷电机的输出转矩^[14]为

$$\tau(t) = \mathbf{K}_m \mathbf{i}_m(t) = \mathbf{K}_m \mathbf{i}_d(t) = \mathbf{K}_m \mathbf{K}_{sa} \mathbf{u}(t) \quad (2)$$

式中: $\mathbf{K}_m \in \mathbb{R}^{n \times n}$ 为电机转矩常数矩阵; $\mathbf{i}_m \in \mathbb{R}^n$ 为驱动电机的电流矢量; $\mathbf{i}_d \in \mathbb{R}^n$ 为电机的理想电流矢量; $\mathbf{u} \in \mathbb{R}^n$ 为伺服放大器的输入电压矢量。

本文的控制目标是驱动双关节刚性机器人系统的机械臂以参考速度跟踪参考位置轨迹,并实现位置跟踪误差和速度跟踪误差的最小化。

2 自适应 BP 神经网络算法

2.1 参数可调的激活函数

激活函数是神经网络解决非线性问题的关键。激活函数的种类繁多,如线性函数、阈值函数和 S 型函数。这里将基于经典固定参数 S 型激活函数的 BP 神经网络定义为 FPBP 神经网络,相应的算法称为 FPBP 神经网络算法。本文提出一种新型的参数可调激活函数,结构为

$$f = f(x, a, b, c, d) = \begin{cases} d - a, & x \in (-\infty, -\frac{\pi}{2b} - c) \\ a \sin[b(x + c)] + d, & x \in (-\frac{\pi}{2b} - c, \frac{\pi}{2b} - c) \\ d + a, & x \in (\frac{\pi}{2b} - c, +\infty) \end{cases} \quad (3)$$

式中: a 为区间因子(正实数); b 为斜率因子(正实数); c 和 d 为位置参数。

显然,只要给出相应的参数,传统的 S 型激活函数 $(1 + e^{-x})^{-1}$ 只是式(3)的一个特例。因此,本文提出的激活函数具有更丰富的结构形式和非线性表达能力。

2.2 参数更新算法

相关文献已经证明,只要隐藏层中节点的数量足够,3 层 BP 神经网络具有模拟任何复杂非线性映射的能力^[15]。本文提出的自适应 BP 神经网络拓扑图如图 2 所示。

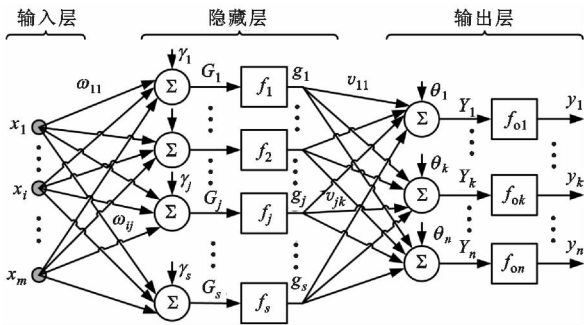


图 2 自适应 BP 神经网络拓扑图

在图 2 中:自适应 BP 神经网络的输入层有 m 个输入量 $x_i (i=1, 2, \dots, m)$; 隐藏层有 s 个神经元, 神经元阈值为 $\gamma_j (j=1, 2, \dots, s)$, 激活函数为本文提出的参数可调激活函数 $f_j (j=1, 2, \dots, s)$, 其输入量为 $G_j (j=1, 2, \dots, s)$, 输出量为 $g_j (j=1, 2, \dots, s)$; 自适应 BP 神经网络的输出层有 n 个神经元, 层神经元阈值为 $\theta_k (k=1, 2, \dots, n)$, 激活函数为本文的参数可调激活函数 $f_{ok} (k=1, 2, \dots, n)$, 其输入量为 $Y_k (k=1, 2, \dots, n)$, 输出量为 $y_k (k=1, 2, \dots, n)$; ω_{ij} 表示输入层第 i 个输入量与隐藏层第 j 个神经元连接的权值; v_{jk} 表示隐藏层第 j 个输入量与输出层第 k 个神经元连接的权值。于是,有

$$G_j = \sum_{i=1}^m \omega_{ij} x_i - \gamma_j \quad (4)$$

$$g_j = f_j(G_j) \quad (5)$$

$$Y_k = \sum_{j=1}^s v_{jk} g_j - \theta_k \quad (6)$$

$$y_k = f_{ok}(Y_k) \quad (7)$$

方程(4)~(7)完成了自适应 BP 神经网络从输入到输出的映射。将 $f_j (j=1, 2, \dots, s)$ 和 $f_{ok} (k=1, 2, \dots, n)$ 简写为 f , 并对参数 x, a, b, c 和 d 分别求偏导, 可得

$$\frac{\partial f}{\partial x} = ab \left[1 - \left(\frac{f-d}{a} \right)^2 \right]^{1/2} \quad (8)$$

$$\frac{\partial f}{\partial a} = \frac{f-d}{a} \quad (9)$$

$$\frac{\partial f}{\partial b} = a(x+c) \left[1 - \left(\frac{f-d}{a} \right)^2 \right]^{1/2} \quad (10)$$

$$\frac{\partial f}{\partial c} = ab \left[1 - \left(\frac{f-d}{a} \right)^2 \right]^{1/2} \quad (11)$$

$$\frac{\partial f}{\partial d} = 1 \quad (12)$$

显然, 由式(3)可得 $\left(\frac{f-d}{a} \right)^2 \leq 1$, 因此保证了式(8)、式(10)和式(11)中平方根运算的可行性, 且为神经网络权值和激活函数参数的更新计算提供了很大方便。由式(3)还可知, 本文的激活函数连续且单调递增, 故式(8)、式(10)和式(11)中的求平方根运算不考虑结果为负的情况。

下面, 以批处理训练模式来说明参数的调整过程。为了简化推导, 把各神经元的阈值并入权矢量。设有 P 个学习样本矢量, 对应的期望输出是 $T_k^{(1)}, T_k^{(2)}, \dots, T_k^{(P)}$ 。基于最小均方误差准则, 当第 p 个样本输入网络并产生输出时, 定义神经网络的性能指标函数为

$$E^{(p)} = \frac{1}{2} \sum_{k=1}^n (y_k^{(p)} - T_k^{(p)})^2 \quad (13)$$

则总误差

$$E = \sum_{p=1}^P E^{(p)} = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^n (y_k^{(p)} - T_k^{(p)})^2 \quad (14)$$

2.2.1 神经网络权值更新 输出层权值 v_{jk} 的更新公式为

$$\frac{\partial E}{\partial v_{jk}} = \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial v_{jk}} = \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial y_k^{(p)}} \frac{\partial y_k^{(p)}}{\partial Y_k^{(p)}} \frac{\partial Y_k^{(p)}}{\partial v_{jk}} = \sum_{p=1}^P \delta_k^{(p)} g_j^{(p)} \quad (15)$$

式中: $\delta_k^{(p)} = (y_k^{(p)} - T_k^{(p)}) a_k b_k \left[1 - \left(\frac{y_k^{(p)} - d_k}{a_k} \right)^2 \right]^{1/2}$ 。

考虑学习率和动量因子,输出层权值更新为

$$v(t+1) = v(t) + \eta_v \left[-\frac{\partial E(t)}{\partial v} \right] + \alpha_v \Delta v(t) \quad (16)$$

$$\Delta v(t) = v(t) - v(t-1) = \eta_v \left[-\frac{\partial E(t-1)}{\partial v} \right] \quad (17)$$

式中: η_v 为 v 的学习率; α_v 为 v 的动量因子。

隐藏层权值 ω_{ij} 的更新公式为

$$\frac{\partial E}{\partial \omega_{ij}} = \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial \omega_{ij}} = \sum_{p=1}^P \sum_{k=1}^n \frac{\partial E^{(p)}}{\partial y_k^{(p)}} \frac{\partial y_k^{(p)}}{\partial Y_k^{(p)}} \frac{\partial Y_k^{(p)}}{\partial g_k^{(p)}} \frac{\partial g_k^{(p)}}{\partial G_k^{(p)}} \frac{\partial G_k^{(p)}}{\partial \omega_{ij}} = \sum_{p=1}^P \sum_{k=1}^n \delta_k^{(p)} v_{jk}^{(p)} \delta_j^{(p)} x_i^{(p)} \quad (18)$$

式中: $\delta_j^{(p)} = a_j b_j \left[1 - \left(\frac{g_j^{(p)} - d_j}{a_j} \right)^2 \right]^{1/2}$ 。

考虑学习率和动量因子,隐藏层权值更新为

$$\omega(t+1) = \omega(t) + \eta_\omega \left[-\frac{\partial E(t)}{\partial \omega} \right] + \alpha_\omega \Delta \omega(t) \quad (19)$$

$$\Delta \omega(t) = \omega(t) - \omega(t-1) = \eta_\omega \left[-\frac{\partial E(t-1)}{\partial \omega} \right] \quad (20)$$

式中: η_ω 为 ω 的学习率; α_ω 为 ω 的动量因子。

2.2.2 激活函数参数更新

(1) 区间因子 a 的更新。输出层区间因子 a_k 的更新为

$$\frac{\partial E}{\partial a_k} = \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial a_k} = \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial y_k^{(p)}} \frac{\partial y_k^{(p)}}{\partial a_k} = \sum_{p=1}^P (y_k^{(p)} - T_k^{(p)}) \frac{y_k^{(p)} - d_k}{a_k} \quad (21)$$

隐藏层区间因子 a_j 的更新为

$$\frac{\partial E}{\partial a_j} = \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial a_j} = \sum_{p=1}^P \sum_{k=1}^n \left(\frac{\partial E^{(p)}}{\partial y_k^{(p)}} \frac{\partial y_k^{(p)}}{\partial Y_k^{(p)}} \frac{\partial Y_k^{(p)}}{\partial g_j^{(p)}} \cdot \frac{\partial g_j^{(p)}}{\partial a_j} \right) = \sum_{p=1}^P \sum_{k=1}^n \delta_k^{(p)} v_{jk}^{(p)} \frac{g_j^{(p)} - d_j}{a_j} \quad (22)$$

考虑学习率和动量因子,区间因子更新为

$$a(t+1) = a(t) + \eta_a \left[-\frac{\partial E(t)}{\partial a} \right] + \alpha_a \Delta a(t) \quad (23)$$

$$\Delta a(t) = a(t) - a(t-1) = \eta_a \left[-\frac{\partial E(t-1)}{\partial a} \right] \quad (24)$$

式中: η_a 为 a 的学习率; α_a 为 a 的动量因子。

(2) 斜率因子 b 的更新。模糊推理不依赖于系统的数学模型,能获得更好的动态控制性能,实现高精度、强鲁棒性控制,提高系统的智能化程度,因此采用模糊推理算法设计斜率因子 b 的更新算法。将模糊子集划分为 $\{Z, PS, PM, PB\}$, 分别代表 $\{\text{零, 正小, 正中, 正大}\}$ 。选取高斯函数和三角形函数相结合的隶属度函数。

首先,当位置误差较大时,选用高斯隶属度函数,且以快速减小误差为主要目的。此时应选用较大的斜率因子,但当误差迅速减小时,为防止神经网络发生饱和,斜率因子可保持不变。

其次,当位置误差减小到设定的区段之后,选用三角形隶属度函数,且以防止 BP 神经网络出现假饱和现象为主要目的。此时若误差缓慢地减小,应选用较大的斜率因子,使神经网络跳出平坦区域。

最后,当位置误差继续减小到模糊子集为 Z 的区间时,应选用较小的斜率因子,以防止 BP 神经网络过饱和,提高系统的稳定性。根据大量的实验,制定了斜率因子变化量 Δb 的模糊规则表,见表 1。

表 1 Δb 的模糊规则表

$ e_i $	$ \dot{e}_i $			
	Z	PS	PM	PB
Z	PS	PS	PS	Z
PS	PM	PM	PS	Z
PM	PB	PM	PS	Z
PB	PB	PM	PM	Z

注: $|e_i| = |q_{di} - q_i|$ ($i=1,2$) 为机械臂关节角位置误差的绝对值,其中 q_{di} ($i=1,2$) 为机械臂关节的给定角位置, q_i ($i=1,2$) 为机械臂关节的实际角位置; $|\dot{e}_i| = |de_i/dt|$ ($i=1,2$) 为机械臂关节角位置误差变化率的绝对值。

模糊推理选用马丹尼(Mardani)型,整定好的斜率因子 b 表示为

$$b(t+1) = b(0) + \Delta b(t) \quad (25)$$

式中: $b(0) > 0$ 为斜率因子初值; $\Delta b(t)$ 为斜率因子变化量。为避免依赖精确的数学模型和长时间参数整定的困难,选用重心法作为解模糊的方法,并导出斜率因子变化量的查询表,以实现斜率因子的在线整定。

(3) 位置参数 c 的更新。输出层位置参数 c_k 的更新为

$$\frac{\partial E}{\partial c_k} = \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial c_k} = \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial y_k^{(p)}} \frac{\partial y_k^{(p)}}{\partial c_k} = \sum_{p=1}^P \delta_k^{(p)} \quad (26)$$

隐藏层位置参数 c_j 的更新为

$$\begin{aligned} \frac{\partial E}{\partial c_j} &= \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial c_j} = \\ &\sum_{p=1}^P \sum_{k=1}^n \left(\frac{\partial E^{(p)}}{\partial y_k^{(p)}} \frac{\partial y_k^{(p)}}{\partial Y_k^{(p)}} \frac{\partial Y_k^{(p)}}{\partial g_j^{(p)}} \frac{\partial g_j^{(p)}}{\partial c_j} \right) = \\ &\sum_{p=1}^P \sum_{k=1}^n \delta_k^{(p)} v_{jk}^{(p)} \delta_j^{(p)} \end{aligned} \quad (27)$$

考虑学习率和动量因子,位置参数更新为

$$c(t+1) = c(t) + \eta_c \left[-\frac{\partial E(t)}{\partial c} \right] + \alpha_c \Delta c(t) \quad (28)$$

$$\Delta c(t) = c(t) - c(t-1) = \eta_c \left[-\frac{\partial E(t-1)}{\partial c} \right] \quad (29)$$

式中: η_c 为 c 的学习率; α_c 为 c 的动量因子。

(4) 位置参数 d 的更新。输出层位置参数 d_k 的更新为

$$\begin{aligned} \frac{\partial E}{\partial d_k} &= \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial d_k} = \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial y_k^{(p)}} \frac{\partial y_k^{(p)}}{\partial d_k} = \\ &\sum_{p=1}^P (y_k^{(p)} - T_k^{(p)}) \end{aligned} \quad (30)$$

隐藏层位置参数 d_j 的更新为

$$\begin{aligned} \frac{\partial E}{\partial d_j} &= \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial d_j} = \\ &\sum_{p=1}^P \sum_{k=1}^n \left(\frac{\partial E^{(p)}}{\partial y_k^{(p)}} \frac{\partial y_k^{(p)}}{\partial Y_k^{(p)}} \frac{\partial Y_k^{(p)}}{\partial g_j^{(p)}} \frac{\partial g_j^{(p)}}{\partial d_j} \right) = \\ &\sum_{p=1}^P \sum_{k=1}^n \delta_k^{(p)} v_{jk}^{(p)} \end{aligned} \quad (31)$$

考虑学习率和动量因子,位置参数更新为

$$d(t+1) = d(t) + \eta_d \left[-\frac{\partial E(t)}{\partial d} \right] + \alpha_d \Delta d(t) \quad (32)$$

$$\Delta d(t) = d(t) - d(t-1) = \eta_d \left[-\frac{\partial E(t-1)}{\partial d} \right] \quad (33)$$

式中: η_d 为 d 的学习率; α_d 为 d 的动量因子。

学习率决定了参数移动到最优值的速度快慢,所以学习率对于算法的性能至关重要。附加动量因子使网络在修正其权值时,不仅考虑误差在梯度上的作用,而且考虑在误差曲面上变化趋势的影响。在没有附加动量的作用时,网络可能陷入浅的局部极小值,而利用附加动量的作用则有可能滑过这些极小值。

对于不同的输入样本,最佳的激活函数参数可能是不一致的。然而,由上述分析可以得出如下结论:本文所提出的新激活函数的参数可以自适应调整,使其具有更丰富的非线性表达能力,特别地,斜率因子可以根据位置误差和它们的变化率,通过模糊推理器进行推理,因此斜率因子也具有自适应性。

3 仿真研究

3.1 系统模型

考虑图 1 所示的双关节刚性机器人系统机械臂。机器人系统的参数为: $m_1 = 2.0 \text{ kg}$, $m_2 = 1.0 \text{ kg}$, $l_1 = 0.2 \text{ m}$, $l_2 = 0.15 \text{ m}$, $I_1 = 61.25 \text{ g} \cdot \text{m}^2$, $I_2 = 20.42 \text{ g} \cdot \text{m}^2$, $g = 9.8 \text{ m/s}^2$ 。为了便于比较 FPBP 神经网络算法和本文提出的自适应 BP 神经网络算法,除了激活函数的结构和激活函数的参数更新方法外,其余所有的条件和参数设置相同。为了更加客观地进行对比,2 种 BP 神经网络算法的学习率和动量因子设置相同的值,即 $\eta = 0.03$, $\alpha = 0.05$ 。

对于式(1)所表示的 2 自由度机器人系统动态方程,各参数^[16]为

$$\mathbf{M}(q) = \begin{bmatrix} p_1 + p_2 + 2p_3 \cos q_2 & p_2 + p_3 \cos q_2 \\ p_2 + p_3 \cos q_2 & p_2 \end{bmatrix} \quad (34)$$

$$\mathbf{V}(q, \dot{q}) = \begin{bmatrix} -p_3 \dot{q}_2 \sin q_2 & -p_3 (\dot{q}_1 + \dot{q}_2) \sin q_2 \\ p_3 \dot{q}_1 \sin q_2 & 0 \end{bmatrix} \quad (35)$$

$$\mathbf{F}(\dot{q}) = \begin{bmatrix} 0.4 \dot{q}_1 \\ 0.4 \dot{q}_2 \end{bmatrix} \quad (36)$$

$$\mathbf{G}(q) = \begin{bmatrix} p_4 g \cos q_1 + p_5 g \cos(q_1 + q_2) \\ p_5 g \cos(q_1 + q_2) \end{bmatrix} \quad (37)$$

式中: $p_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1$; $p_2 = m_2 l_{c2}^2 + I_2$; $p_3 = m_2 l_1 l_{c2}$; $p_4 = m_1 l_{c1} + m_2 l_1$; $p_5 = m_2 l_{c2}$ 。

设定机器人的初始位置为 $q_1(0) = 1 \text{ rad}$, $q_2(0) = 0$; 初始速度为 $\dot{q}_1(0) = \dot{q}_2(0) = 0$; 位置跟踪轨迹为

$$\mathbf{q}_d = [q_{d1}, q_{d2}]^T = [\sin(\pi t), \cos(\pi t)]^T \quad (38)$$

其中 $t \in [0, t_f]$ 且 $t_f = 20 \text{ s}$ 。

对2种神经网络算法进行仿真分析。它们的输入为 $\mathbf{x}=[e_i, \dot{e}_i, q_{di}, \dot{q}_{di}, \ddot{q}_{di}] (i=1,2)$, 隐藏层的层数为1, 神经元数目 $s=15$, 输出 $\mathbf{y}=[K_{pj}, K_{ij}, K_{dj}] (j=1,2)$, 初始权值设置为 $\omega(0)=v(0)=0$ 。

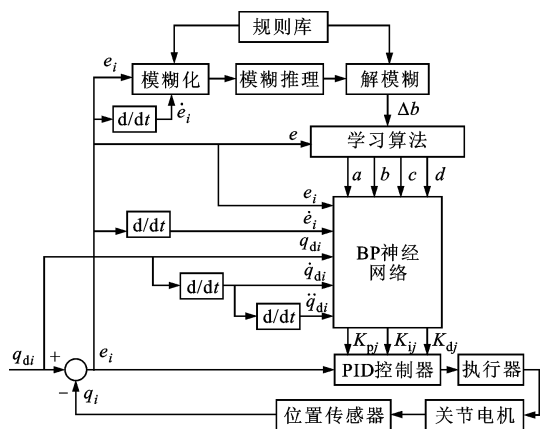
FPBP神经网络算法使用经典固定参数的S型激活函数, 即

$$f_1(x) = \frac{1}{1 + e^{-x}} \quad (39)$$

自适应BP神经网络算法使用新型激活函数, 初始值设定为: $a(0)=b(0)=d(0)=0.5 \text{ rad}, c(0)=0$ 。

$$f_{2, \text{initial}}(x) = \begin{cases} 0, & x \in (-\infty, -\pi) \\ 0.5 \sin(0.5x) + 0.5, & x \in [-\pi, \pi] \\ 1, & x \in (\pi, +\infty) \end{cases} \quad (40)$$

自适应模糊神经网络控制系统框图见图3。



$q_{di}, \dot{q}_{di}, \ddot{q}_{di} (i=1,2)$ 分别为机械臂的给定角位置、给定角速度和给定角加速度; $q_i (i=1,2)$ 为关节的实际位置角; $e_i = q_{di} - q_i (i=1,2)$ 为位置角误差; $\dot{e}_i = de_i/dt (i=1,2)$ 为位置角误差的变化率。

图3 自适应模糊神经网络控制系统框图

设 $u(k)$ 为第 k 次采样时刻控制器的输出值, 可得离散的位置控制器输出

$$u(k) = u(k-1) + K_p \Delta e(k) + K_i e(k) + K_d [\Delta e(k) - \Delta e(k-1)] \quad (41)$$

式中: K_p, K_i, K_d 分别为位置控制器的比例增益、积分增益和微分增益; $\Delta e(k) = e(k) - e(k-1)$ 。

3.2 结果及分析

双关节刚性机器人系统2个关节的位置跟踪轨迹如图4和图5所示, 从中可以看出: 机器人系统的所有机械臂关节都可以通过自适应BP神经网络算法成功地跟踪所设定的轨迹, 且位置误差可以收敛并接近于0, 而使用FPBP神经网络算法不能准确地跟踪期望轨迹, 且稳态位置角误差较大。

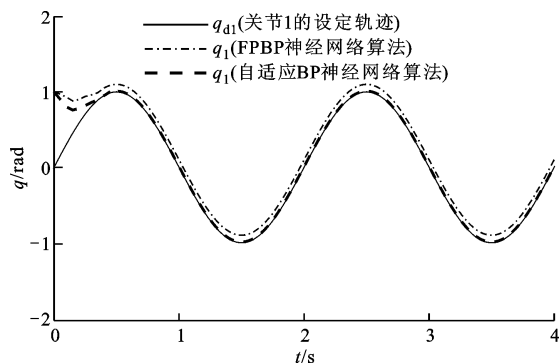


图4 关节1的位置跟踪轨迹

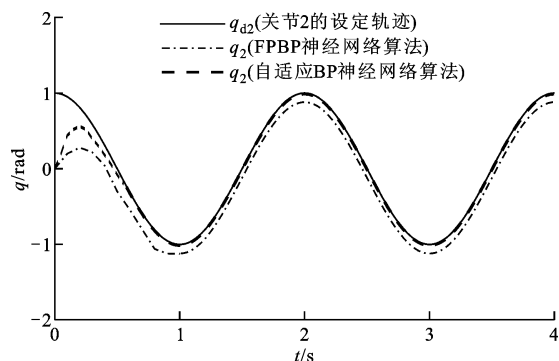


图5 关节2的位置跟踪轨迹

从图4和图5还可以看出, 在机器人系统的参数相同的前提下, 采用FPBP神经网络算法的位置误差比采用自适应BP神经网络算法的位置误差大, 即FPBP神经网络算法的跟踪性能不如自适应BP神经网络算法好。为了更清晰地比较2种算法的性能, 对每个控制周期内采样的一组位置跟踪误差求取均方根误差 (E_{rms}), 结果如图6、图7所示。

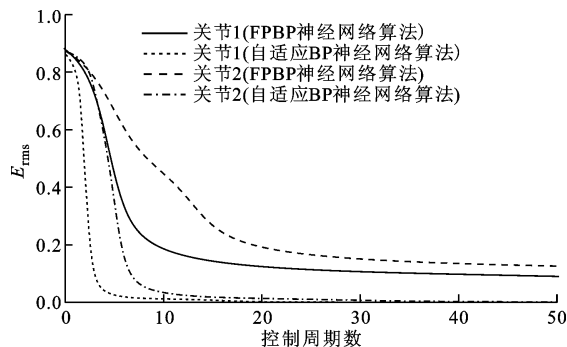


图6 位置跟踪误差的均方根误差曲线

由图6可以看出: 采用FPBP神经网络算法的机器人机械臂关节不能成功地跟踪期望轨迹, 位置跟踪误差下降缓慢; 相反, 采用自适应BP神经网络算法的机器人机械臂关节的位置跟踪误差能在很短的时间内收敛并接近于0。从图7可知, 采用自适应BP神经网络算法比采用FPBP神经网络算法的

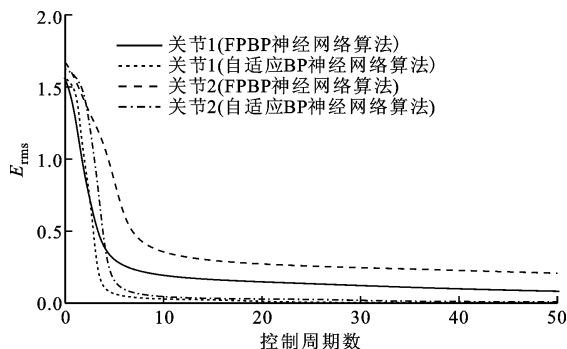


图 7 速度跟踪误差的均方根误差曲线

速度跟踪误差小,说明自适应 BP 神经网络算法极大地提高了 BP 神经网络权值的收敛速度。自适应 BP 神经网络算法的上述优势的体现正是由于它能对激活函数的映射范围、斜率及位置等参数进行调整,特别是能通过模糊推理器对激活函数的斜率因子进行自适应调整,从而保证了机械臂关节的高精度轨迹跟踪性能。

4 结束语

本文针对 BP 神经网络学习速度慢、泛化能力弱和易陷入局部极小值等问题,研究了自适应神经网络控制策略,提出了一种新结构的参数可调激活函数,其映射范围、斜率因子、水平位置和竖直位置等参数皆可自适应地调整,特别是激活函数的斜率因子可以通过模糊推理器进行整定,以使神经网络具有更强的非线性映射能力。仿真结果表明,除激活函数的结构和参数更新方法不同之外,在相同参数条件下,自适应 BP 神经网络算法比 FBP 神经网络算法具有更强的非线性表达能力、更快的收敛速度和更高的收敛精度,可以成功地驱动机械臂以参考速度跟踪参考位置轨迹,且具有更好的动态性能。

参考文献:

- [1] WANG L, LIU Z, CHEN C L, et al. Energy-efficient SVM learning control system for biped walking robots [J]. IEEE Transactions on Neural Networks & Learning Systems, 2013, 24(5): 831-837.
- [2] LI Z, LI J, KANG Y. Adaptive robust coordinated control of multiple mobile manipulators interacting with rigid environments [J]. Automatica, 2010, 46(12): 2028-2034.
- [3] CHEN C L P, WEN G X, LIU Y J, et al. Adaptive consensus control for a class of nonlinear multiagent

time-delay systems using neural networks [J]. IEEE Transactions on Neural Networks & Learning Systems, 2014, 25(6): 1217-1226.

- [4] CHEN C L P, LIU Y J, WEN G X. Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems [J]. IEEE Transactions on Cybernetics, 2014, 44(5): 583-593.
- [5] LIU Y J, TONG S. Adaptive NN tracking control of uncertain nonlinear discrete-time systems with non-affine dead-zone input [J]. IEEE Transactions on Cybernetics, 2014, 45(3): 497-505.
- [6] LIU Y J, CHEN C L P, WEN G X, et al. Adaptive neural output feedback tracking control for a class of uncertain discrete-time nonlinear systems [J]. IEEE Transactions on Neural Networks, 2011, 22(7): 1162-1167.
- [7] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning internal representations by error propagation [M] // Parallel Distributed Processing: Exploration of the Microstructure of Cognition. Cambridge, MA, USA: MIT Press, 1986: 318-362.
- [8] 林军, 倪宏, 孙鹏, 等. 一种采用神经网络 PID 控制的自适应资源分配方法 [J]. 西安交通大学学报, 2013, 47(4): 112-117.
LIN Jun, NI Hong, SUN Peng, et al. Adaptive resource allocation based on neural network PID control [J]. Journal of Xi'an Jiaotong University, 2013, 47(4): 112-117.
- [9] TSAI C, CHIH Y, WONG W H, et al. A hardware-efficient sigmoid function with adjustable precision for a neural network system [J]. IEEE Transactions on Circuits & Systems: II Express Briefs, 2015, 62(11): 1073-1077.
- [10] 柏子游, 张勇, 虞烈. 基于模糊数学和神经网络的数学形态学方法 [J]. 西安交通大学学报, 1999, 33(4): 104-105.
BAI Ziyu, ZHANG Yong, YU Lie. Mathematical morphological operations based on fuzzy set and neural network [J]. Journal of Xi'an Jiaotong University, 1999, 33(4): 104-105.
- [11] CARRASCO-ROBLES M, SERRANO L. A novel minimum-size activation function and its derivative [J]. IEEE Transactions on Circuits & Systems: II Express Briefs, 2009, 56(4): 280-284.
- [12] 梁霄, 张均东, 李巍, 等. 水下机器人 T-S 型模糊神经网络控制 [J]. 电机与控制学报, 2010, 14(7): 99-104.

(下转第 164 页)