文章编号：1000－8152(2002)04－05－0583

一种模糊神经网络的快速参数学习算法

陈　非，敬忠良，姚晓东
（上海交通大学 电子信息学院 航空航天信息与控制研究所，上海 200030）

摘要：提出了一种新的模糊神经网络的快速参数学习算法，采用一些特殊的处理，可以用递推最小二乘法(RLS)来调整所有的参数.以前的学习算法在调整模糊隶属度函数的中心和宽度的时候，用的是梯度下降法，具有容易陷入局部最小值点、收敛速度慢等缺点，而本算法则可以克服这些缺点，最后通过仿真验证了算法的有效性.

关键词：T-S 模糊推理系统；多层前向神经网络；改进 RLS 算法；模糊神经网络
中国分类号：TP18　　**文献标识码**：A

Fast parameter learning algorithm for fuzzy neural networks

CHEN Fei, JING Zhong-liang, YAO Xiao-dong
(School of Electronics and Information Technology, Institute of Aerospace Information and Control, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: A novel parameter learning algorithm for fuzzy neural networks (FNN) is proposed. The conventional methods usually use the gradient descent based backpropogation algorithm to adjust the center and width of the membership functions. To avoid the inborn problem of BP algorithm, such as local minima and slow convergence, a modified RLS method is employed here to adjust the parameters of FNN, which is faster than the conventional BP algorithm. The validity of this method has been demonstrated by simulation results.

Key words: T-S fuzzy inference system; multi-layer neural networks; modified RLS algorithm; fuzzy neural networks (FNN)

1　引言(Introduction)

模糊系统和神经网络都是对人的智能的一种模拟，前者采用自顶向下的角度，而后者则是自底向上的角度.它们均可从给定的系统的输入/输出数据中，建立系统的非线性模型，并且在数据处理的形式上，它们均采用并行处理的结构.但是，模糊系统和神经网络又有不同之处.神经网络可以从例子中学习，具有很强的自适应学习能力，但由于神经网络模型是无模型的预报器，因此要使学习结果满意，需要很多数据进行训练.另外，神经网络所获得的输入/输出关系无法用容易被人接受的方式表达出来.相反，模糊系统是建立在被人容易接受的'如果-则'表达方式上，而且它是建立在专家知识的基础之上，因此收敛快，但如何自动生成和调整隶属度函数和模糊规则，是一个棘手的问题.

B.Kosko 等[1]将模糊理论与神经网络融合(FNN)在一起，发挥了两者的优点，提高了系统的学习能力和表达能力，同时利用了专家知识和数据信息.因此，模糊神经网络近年来得到了广泛的研究.J. R. Jang[2]指出，在使用模糊神经网络时，希望能找到快速有效地调整隶属度函数的参数的方法.他在文中回顾了传统的 BP 算法调整参数的方法，同时针对输出节点的权值对输出是线性的，因此可以用 RLS 算法来进行训练，但由于高斯隶属度函数的中心和宽度对输出是非线性的，因此仍然用 BP 算法对它们进行调整.

Y.Chen 和 C.Teng[3]在讨论用 FNN 来进行模型参考自适应控制(MRAC)的时候，推导出了在误差反传调整隶属度函数的中心和宽度的公式.公式具有 MFN(multilayered feedforward network)的 BP 算法通用的形式，即

$$\Delta W_i^{(l)} = \alpha\delta_i^{(l)}y^{(l-1)}. \qquad (1)$$

式中，W 可以是 FNN 中的任何参数，但文中仍然采用 BP 算法来调整隶属度函数的参数.

收稿日期：2000－07－17；收修改稿日期：2001－04－29.

递推最小二乘 RLS[4]算法要求输出对参数是线性的,或者如果节点输出是节点输入的单调函数,而且节点输入对参数是线性的时候,也能用 RLS 算法,其本质就是通过逆映射将输出误差转换到输入端,然后利用最小二乘法对参数进行调整.

本文在推导误差反传公式的时候,发现公式具有 BP 算法的通用格式,从而得到隶属度神经元层输出的误差信号,此时通过选取三角隶属度函数,并通过特殊处理之后,使得对 FNN 所有参数均可以用 RLS 算法来进行调整,从而克服了 BP 算法收敛速度慢,易陷入局部极小等缺点.

2 模糊神经网络(Fuzzy neural network)

下面我们给出模糊神经网络的结构,它有 n 个输入,每个输入划分为 m 个语言变量(m 个隶属度函数),为便于说明,仅考虑单输出的情况,如图 1 所示.多输出情况可类似得到.

图中,网络的第一层代表输入节点,它将输入直接传递到第二层节点处.第二层节点的作用是求出各个输入的隶属度.网络第三层对应模糊推理,计算得到每条规则对应的适应度,在计算适应度的时候,采用乘积的方法.网络的最后一层得到输出.采用 Sugeno[5]规则推理.

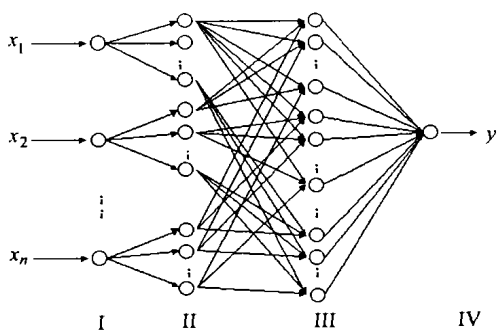


图 1 FNN 模型结构

Fig. 1 The structure of fuzzy neural network

我们将 FNN 每一层节点的输入和输出映射关系表达如下:

第Ⅰ层节点输出

$$y_i^{(1)} = x_i,\ i = 1,2,\cdots,n. \tag{2}$$

第Ⅱ层节点输出

$$y_{ij}^{(2)} = \exp\left(-\frac{(y_i^{(1)} - a_{ij})^2}{b_{ij}^2}\right),\quad i = 1,2,\cdots,n,\ j = 1,2,\cdots,m. \tag{3}$$

第Ⅲ层节点输出

$$y_{i_1 i_2 \cdots i_n}^{(3)} = \prod_{k=1}^{n} y_{ki_k}^{(2)},\ i_k \in \mathbb{N},\ i_k \in [1,m]. \tag{4}$$

第Ⅳ层节点输出

$$y^{(4)} = \frac{\sum y_{i_1 i_2 \cdots i_n}^{(3)} w_{i_1 i_2 \cdots i_n}}{\sum y_{i_1 i_2 \cdots i_n}^{(3)}}. \tag{5}$$

上述式中,a,b 为对应隶属度函数的中心和宽度,w 为 FNN 第三、四层节点之间的连接权值.$y_{i_1 i_2 \cdots i_n}^{(3)}$ 的下标按输入顺序排列,即该值由第一个输入的第 i_1 个隶属度函数值与第二个输入的第 i_2 个隶属度函数值一直到第 n 个输入的第 i_n 个隶属度函数值相乘运算的结果,该值对应模糊推理中某一语句的适应度.

3 修正 RLS 算法(Modified RLS algorithm)

RLS 算法要求输出对参数是线性的.谭永红[4]将它推广,如果节点输出是节点输入的单调函数,而且节点输入对参数是线性的时候,也能用 RLS 算法,其本质就是通过逆映射将输出信息转换到输入端,然后利用最小二乘法对参数进行调整,与标准 BP 算法相比,学习效率提高 10 倍以上.在文[6]中则提出了修正 RLS 算法,以解决文[4]中方法的数值稳定性问题,并给出了证明.现在对修正 RLS 作简要说明,详细推导读者可参看文[6]:

设多层神经网络的输入层为 $t = 1$,输出层为 $t = L$ 层.$r_{it}(k)$ 为 k 时刻第 t 层的第 i 个节点的期望输入,$d_{it}(k)$ 为 k 时刻第 t 层的第 i 个节点的期望输出.神经元的期望输出与其对应的期望输入有如下关系:

$$d_{it}(k) = f[r_{it}(k)]. \tag{6}$$

$x_{it}(k)$ 为 k 时刻第 t 层的第 i 个节点的实际输入,$y_{it}(k)$ 为 k 时刻第 t 层的第 i 个节点的实际输出.定义神经元的理想输出与实际输出之差为

$$\delta_{it}(k) = d_{it}(k) - y_{it}(k). \tag{7}$$

它又可由神经元理想输入和实际输入的差来近似:

$$\delta_{it}(k) \approx f'[x_{it}(k)] \cdot [r_{it}(k) - x_{it}(k)]. \tag{8}$$

多层神经网络各层权值的修正 RLS 调整算法在文献[6]中已经清楚给出,这里略去.

4 FNN 快速参数学习算法(Fast parameter learning algorithm for FNN)

本文新算法是在梯度推导过程中得到的,FNN 结构见图 1.设理想输出为 y_d,我们的目标函数取为

$$J = (y_d - y^{(4)})^2/2, \tag{9}$$

$$\frac{\partial J}{\partial a_{ij}} = -(y_d - y^{(4)})\frac{\partial y^{(4)}}{\partial a_{ij}} =$$

$$-(y_d-y^{(4)})\frac{\left[\left(\frac{\partial\sum y^{(3)}_{i_1\cdots i_{i-1}i_ii_{i+1}\cdots i_n}w_{i_1\cdots i_{i-1}i_ii_{i+1}\cdots i_n}}{\partial a_{ij}}\right)\left(\sum y^{(3)}_{i_1i_2\cdots i_n}\right)-\left(\sum y^{(3)}_{i_1i_2\cdots i_n}w_{i_1i_2\cdots i_n}\right)\left(\frac{\partial\Sigma y^{(3)}_{i_1\cdots i_{i-1}i_ii_{i+1}\cdots i_n}}{\partial a_{ij}}\right)\right]}{\left(\sum y^{(3)}_{i_1i_2\cdots i_n}\right)^2}.$$

仅有当 $i_i = j$ 时，$y_{i_1\cdots i_{i-1}i_ii_{i+1}\cdots i_n}$ 对 a_{ij} 的偏导才非零，因此上式等于

$$-(y_d - y^{(4)})\frac{\left(\frac{\partial\sum y^{(3)}_{i_1\cdots i_{i-1}ji_{i+1}\cdots i_n}w_{i_1\cdots i_{i-1}ji_{i+1}\cdots i_n}}{\partial a_{ij}}\right)\left(\sum y^{(3)}_{i_1i_2\cdots i_n}\right)-\left(\sum y^{(3)}_{i_1i_2\cdots i_n}w_{i_1i_2\cdots i_n}\right)\left(\frac{\partial\Sigma y^{(3)}_{i_1\cdots i_{i-1}ji_{i+1}\cdots i_n}}{\partial a_{ij}}\right)}{\left(\sum y^{(3)}_{i_1i_2\cdots i_n}\right)^2} =$$

$$-(y_d - y^{(4)})\frac{\left(\sum\left(\prod_{\substack{k=1\\k\neq i}}^{n}y^{(2)}_{ki_k}\right)w_{i_1\cdots i_{i-1}ji_{i+1}\cdots i_n}\right)-y^{(4)}\left(\sum\left(\prod_{\substack{k=1\\k\neq i}}^{n}y^{(2)}_{ki_k}\right)w_{i_1\cdots i_{i-1}ji_{i+1}\cdots i_n}\right)}{\left(\sum y^{(3)}_{i_1i_2\cdots i_n}\right)}\frac{\partial y^{(2)}_{ij}}{\partial a_{ij}}. \tag{10}$$

我们看到，上式的结果与 BP 算法反传的误差很相似，现在令第 i 个输入的第 j 个隶属度函数节点的理想输出与实际输出之差为

$$\delta_{ij} = -(y_d-y^{(4)})\left\{\left(\sum\left(\prod_{\substack{k=1\\k\neq i}}^{n}y^{(2)}_{ki_k}\right)w_{i_1\cdots i_{i-1}ji_{i+1}\cdots i_n}\right)-y^{(4)}\left(\sum\left(\prod_{\substack{k=1\\k\neq i}}^{n}y^{(2)}_{ki_k}\right)w_{i_1\cdots i_{i-1}ji_{i+1}\cdots i_n}\right)\right\}/\left(\sum y^{(3)}_{i_1i_2\cdots i_n}\right). \tag{11}$$

如果隶属度函数的参数与节点的输入为线性的话，那么我们就可以尝试用 RLS 算法了．三角型隶属函数(如图 2)的方程为：

$$\mu_A(x) = \begin{cases}0, & x < a,\\ (x-a)/(b-a), & a \leqslant x < b,\\ (x-c)/(b-c), & b \leqslant x < c,\\ 0, & x \geqslant c.\end{cases} \tag{12}$$

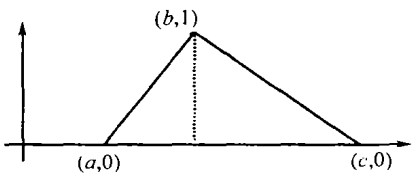


图 2　三角隶属度函数

Fig. 2　Triangle membership function

显然，直线由两个参数来决定，而且函数值与参数值是线性的，对三角隶属度函数来讲，对左、右边的线段来说都是线性的，因此解决了参数与节点输入的线性问题．现在又有一个问题，三角隶属度函数是非单调的，一个函数值可以对应两个自变量，所以不能直接利用逆映射及 RLS 算法．现在我们提出一个想法，即将输入值与三角隶属函数的顶点相比较，如果输入值小于顶点值，则用左边的线段来计算误差，如果输入值大于顶点值，则用右边的线段来计算误差．这样，我们解决了逆映射求节点输入端误差的难题．下面我们就可以用 RLS 算法来调整所有的 FNN 参数了．在用 RLS 算法的时候，还有一些技巧，因为直线 $y = kx + p$ 的参数 k 与 p 由两个点来决定，但还并不是这两个点的坐标，因此还需要进行换算．

对三角隶属函数左边的直线，有：

$$y = \frac{1}{b-a}x - \frac{a}{b-a} = kx + p. \tag{13}$$

同时，可由参数 k 与 p 反过来求 a 和 b 的值：

$$\begin{cases}a = -p/k,\\ b = (1-p)/k,\end{cases} \tag{14}$$

同理，对三角隶属函数右边的直线，有

$$y = -\frac{1}{c-b}x + \frac{c}{c-b} = kx + p, \tag{15}$$

$$\begin{cases}b = (1-p)/k,\\ c = -p/k.\end{cases} \tag{16}$$

在用 RLS 算法调整隶属度函数参数时，首先由式(11)计算出输出端偏差，然后通过式(8)将误差转换到输入端，式中，激励函数的导数实际就是直线的斜率，这里根据输入值在三角隶属函数顶点的左、右侧决定用哪边的直线方程．这样，我们就不断调整三角隶属函数的参数 a, b, c，使得目标函数(9)达到最小．

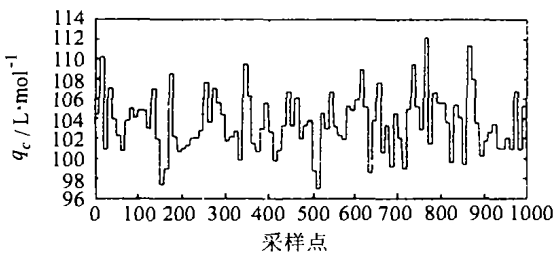
5　仿真结果(Simulation results)

我们的仿真例子选化工过程常见的 CSTR(continuous stirred tank reactor，连续搅拌反应器)模型[7]．CSTR 特性可由以下连续时间的非线性微分方程组来表示：

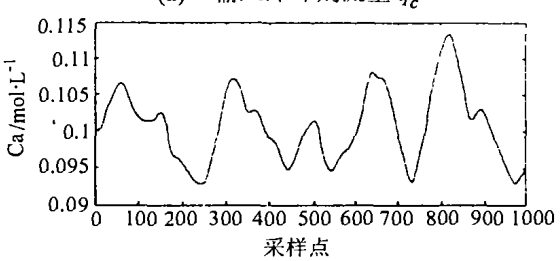
$$\begin{cases} \dot{Ca} = \dfrac{q}{v}(Ca_0 - Ca(t)) - k_0 Ca(t) e^{(-\frac{E}{RT(t)})}, \\ \dot{T} = \dfrac{q}{v}(T_0 - T(t)) - k_1 Ca(t)) e^{(-\frac{E}{RT(t)})} + \\ \qquad k_2 q_c(t)(1 - e^{(-\frac{k_3}{q_c(t)})})(T_{c0} - T(t)). \end{cases} \tag{17}$$

式中，$Ca(t)$ 是产品的平衡浓度，$T(t)$ 为反应温度；Ca_0 是进料浓度(1mol/L)，q 为物料流量，T_0，T_{c0} 分别为物料温度和冷却剂温度；同样，k_0，E/R，v，k_1，k_2，k_3 作为化学反应的系数在此时保持常数，CSTR的参数详见附录.

CSTR 的工作过程为：两种化学物质在 CSTR 中混合形成一种浓度为 $Ca(t)$ 的化合物 A，其混合温度为 $T(t)$. 此反应为放热反应，产生的热量会降低反应速度，因此必须通过引入冷却剂，其流量为 $q_c(t)$，带走热量，冷却温度，保证产品的浓度得以控制；产品的平衡浓度为 $Ca = 0.1$mol/L；此时的反应器温度及冷却剂流量分别为 $T = 438.54$K；$q_c = 103.41$L/min. 这里，我们所针对的是 SISO 系统，控制目标是使产品的浓度 Ca 在给定的范围内波动，操纵变量(即控制输入)为冷却剂流量 q_c，因此我们需要首先建立浓度 Ca 与流量 q_c 之间的模型. 选取冷却剂流量 q_c 为稳态值叠加白噪声输入，我们得到输出的浓度.



(a) 输入冷却剂流量 q_c



(b) 输出浓度 Ca

图 3 CSTR 的输入/输出数据

Fig. 3 The input/output data of CSTR

设系统结构未知，仅知其输入和输出，我们选取辨识结构为：

$$Ca(k+1) = f(Ca(k), Ca(k-1), T(k)). \tag{18}$$

因此，FNN 有三个输入，每个输入取两个隶属度函数，辨识前，我们将数据归一化. 将数据分为两组，前 500 个数据用来训练模糊神经网络，后 500 个数据用作检验，应用本文提出的 RLS 算法进行训练. 将 500 个数据训练完后，我们得到训练数据的误差为 Err = 7.5718e - 005. 用训练好的神经网络进行检验，对后 500 个数据，误差为 Err = 2.9766e - 004. 图 4 所示为对前 500 个数据的训练结果，实线代表 CSTR 实际输出值，虚线代表模糊神经网络输出值，两者几乎重合. 图 5 所示为对后 500 个数据的检验结果，曲线代表 CSTR 实际输出与模糊神经网络输出之差，其中纵坐标的权值为 10e - 3.

训练前我们的初始隶属度函数均匀分割论域，图 6 示出辨识训练前与训练后隶属函数的形状.

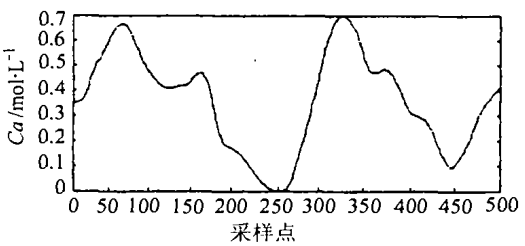


图 4 训练结果

Fig. 4 The result of training

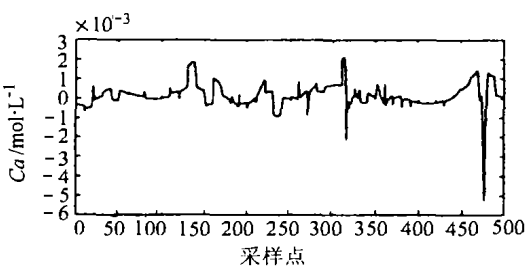
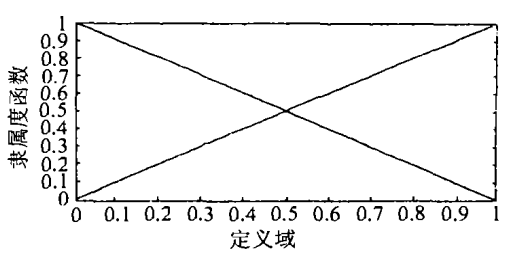
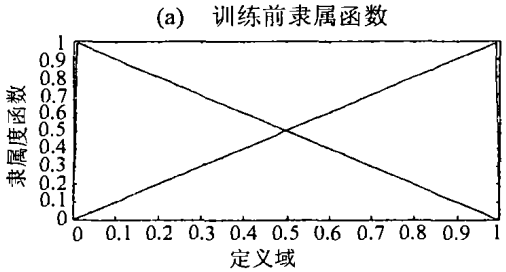


图 5 检验结果

Fig. 5 The result of testing



(a) 训练前隶属函数



(b) 训练后隶属函数

图 6 隶属函数的变化

Fig. 6 The variation of the membership functions

6 与具有动量项的 BP 算法的比较(Comparison with BP with Momentum)

为了加快 BP 算法的收敛速度,人们对它进行了改进,其中一种有效的方法就是引入一个动量项.通过将本文提出的算法和具有动量项的 BP 算法仿真结果进行比较,验证了本算法的优越性.

具有动量项的 BP 算法的权值修正公式为

$$w(n+1) = w(n) - \eta \nabla w(n) + \alpha \Delta w(n-1). \tag{19}$$

式中,n 是迭代次数,η 是学习速率,α 是动量项系数.η 根据每一次迭代是否使误差减小而变化.当迭代一次后,误差减小,则在下一次迭代中,η 乘以一个系数 $\phi > 1$(例如 $\phi = 1.05$);当迭代一次后,误差增大,则权值保持原值,η 乘以一个系数 $\beta < 1$(例如 $\beta = 0.8$),α 置为零,丢弃以前的迭代方向,然后重新下一次迭代.当迭代使误差减小,α 重置为初始值.

用具有动量项的 BP 算法和本文算法训练模糊神经网络的结果如图 7 和图 8.从图中可以看出,BP 算法的收敛速度很慢,一千步迭代后,误差平方和才达到 0.0263,而且误差达到 0.01 后,就很难进一步减小了.而用本文算法算法训练模糊神经网络的速度很快,25 步误差平方和就达到 0.01.

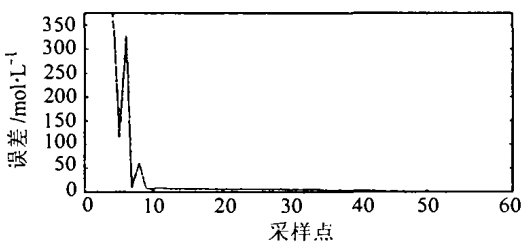


图 7　具有动量项的 BP 算法训练收敛速度

Fig. 7　Convergence of BP with momentum

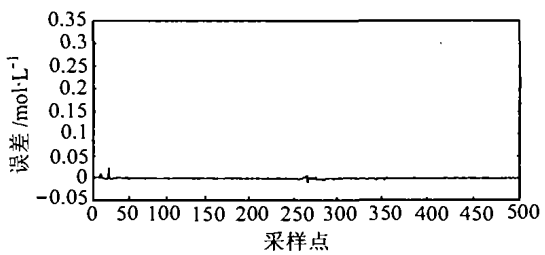


图 8　RLS 算法训练收敛速度

Fig. 8　Convergence of RLS

7 结论(Conclusion)

文中我们选取隶属函数是三角函数,这样,隶属函数参数对左、或右边的直线来说是线性的.但由于三角函数不是单调函数,因此不能直接利用 RLS 算法,我们采用一些特殊处理,使得由输出可以唯一确定一个输入值,这样我们就可以用线性最小二乘法了.它克服了 BP 算法的易限于局部极小值点及收敛速度慢等缺点,仿真证明了我们的算法的有效性.

附　录(Appendix)

CSTR 的参数:

对象参数	物理意义	名义值
q	过程流量	100 L/min
v	反应器体积	100 L
k_0	反应时间常数	$7.2 * 10^{10}$min
E/R	反应激活能	$1 * 10^4$ K
T_0	馈入温度	350 K
T_{c0}	冷却剂温度	350 K
ΔH	反应热	$-2 * 10^5$ cal/mol
C_p, C_{pc}	质量定压热容	1 cal/g/k
ρ, ρ_c	液体密度	$1 * 10^3$ g/l
H_a	热交换系数	$7 * 10^5$ cal/min/K

其中,$k_1 = -\Delta H k_0/\rho C_p$;$k_2 = \rho_c C_{pc}/\rho C_p v$;$k_3 = h_a/\rho_c C_{pc}$.

参考文献(References)

[1] Kong S G, Kosko B. Adaptive fuzzy systems for backing up a truck-and-trailer [J]. IEEE Trans. Neural Networks, 1992, 3(2): 211 - 233

[2] Jang J R. ANFIS: adaptive-network-based fuzzy inference system [J]. IEEE Trans. on SMC, 1993, 23(3): 665 - 683

[3] Chen Yie-Chien, Teng Ching-Cheng. A model reference control structure using a fuzzy neural network [J]. Fuzzy Sets and Systems, 1995, 73(3): 291 - 312

[4] Tan Y. RLS training algorithm for multi-layer feed-forward neural networks and its application to system identification [J]. Control Theory and Applications, 1994, 11(5): 594 - 599

[5] Sugeno M. An introductory survey of fuzzy control [J]. Information and Science, 1985, 36(1): 59 - 83

[6] Hu Y, Wu G. Modified RLS training algorithm for multi-layer feed-forward neural networks [J]. System Engineering and Electronics Technology, 2000, 22(1): 77 - 80

[7] Morningred J D, Paden B E, Seborg D E, et al. An adaptive nonlinear predictive controller [A]. Proc. Amer. Contr. Conf. [C]. San Diego, US, 1990, 2, 1614 - 1619

本文作者简介

陈　非　1976 年生.2000 年毕业于大连理工大学自动控制理论与应用专业,获硕士研究生学位.现在上海交通大学航空航天信息与控制研究所攻读博士学位.研究领域为信号处理,智能信息融合与控制.　Email: fchen@sjtu.edu.cn

敬忠良　1960 年生.教授.博士生导师.国家教育部"长江学者奖励计划特聘教授".现为上海交通大学航空航天信息与控制研究所所长.研究领域为随机控制,目标跟踪,智能信息融合与控制.

姚晓东　1970 年生.2000 年毕业于西北工业大学,获博士学位.现为上海交通大学航空航天信息与控制研究所博士后.研究方向为信号处理,智能信息融合与控制.