

文章编号: 1000- 582X(2006)01- 0150- 04

基于 Tan-Sigmoid函数参数调整的 BP神经网络改进算法*

罗 兵, 黄万杰, 杨 帅

(重庆大学 经济与工商管理学院, 重庆 400030)

摘 要: 为提高 BP神经网络的性能, 对网络的联接权值 W 和神经元的 \tan - sigmoid 转换函数的参数 T 、 θ 进行调整, 使信息分布存储于权值矩阵及转换函数中, 比传统的算法具有更强的非线性映射能力. 经严密的数学推导, 给出了最终的改进算法公式和 1 个预测需求量的算例, 结果表明, 改进后的算法能有效地减少隐层节点数, 且能加快收敛速度和提高收敛精度.

关键词: BP神经网络; \tan - sigmoid 转换函数; 分布存储

中图分类号: O231.2

文献标识码: A

Rumelhart等^[1]提出的传统 BP(Back Propagation)神经网络的收敛速度很慢, 精度不高. 现有的改进包括网络结构和学习算法两个方面^[2-9]. 网络结构的改进主要在互连方式、转换函数及隐层节点、添加惯性冲量、跳跃学习等方面; 学习算法的改进主要有: Levenberg-Marquardt算法、拟牛顿法、弹性 BP法、共轭梯度法、自适应学习速率法等.

传统 BP算法实质上是对 W (一般为联接权值矩阵, 以下均称为权值) 进行调整, 而假定各神经元的转换函数是相同的和不可变更的, 即信息仅分布存储于 W 之中. 生物的学习过程, 除了各神经元细胞间联接强度的调整外, 其本身的响应特性也在变动和调整. 过去由于计算机内存等硬件条件的限制, 为简化计算, 一般都假设生物体中所有神经细胞都具有相同且不变的响应特性. 而事实上, 只要计算机的硬件条件允许, 生物体中各个神经细胞应视为具有可调整的响应特性, 这样可大大地提高 BP神经网络的收敛速度和精度.

胡耀垓等^[2]在 Rumelhart等^[1]研究的基础上, 提出了对 W 和 \log - sigmoid 转换函数参数的 T 、 θ 同时进行调整的改进算法, 该算法克服了一些传统 BP神经网络的缺陷. 笔者在胡耀垓等^[2]研究的基础上, 对网络的权值 W 和神经元一般形式的 \tan - sigmoid 转换函数参数如缩放系数 T 、位移参数 θ 等进行了调整.

1 算法的数学推导

1.1 网络的构成

BP神经网络由许多并行运算的简单神经元组成, 这些神经元类似于生物神经系统的神经元. 虽然单个神经元的结构简单, 功能有限, 但大量神经元构成的网络系统所能实现的功能却非常强大.

BP神经网络 e 层拓扑结构如图 1 所示. 图中 $a(i)$ 为输入层节点, $i=1, 2, \dots, m$; $a(j)$ 为隐层节点, $j=1, 2, \dots, p$; $a(k)$ 为输出层节点, $k=1, 2, \dots, n$ (m, p, n 分别为输入层、隐层和输出层节点数); x_i 为 $a(i)$ 的输入 (也是输出); H_j, h_j 分别为 $a(j)$ 的输入、输出; Y_k, y_k 分别为 $a(k)$ 的输入、输出; y_k^* 为 $a(k)$ 的期望输出; w_{ij} 为 $a(i)$ 与 $a(j)$ 之间的权值 (以下称为隐层的权值); w_{jk} 为 $a(j)$ 与 $a(k)$ 之间的权值 (以下称为输出层的权值). 故有:

$$H_j = \sum_{i=1}^m x_i w_{ij} \quad (1)$$

$$h_j = f_1(H_j), \quad (2)$$

$$Y_k = \sum_{j=1}^p h_j w_{jk}, \quad (3)$$

$$y_k = f_2(Y_k). \quad (4)$$

从式 (1) \rightarrow (4) 完成了 $x_i \rightarrow y_k$ 的映射. 式中 f_1, f_2

* 收稿日期: 2005- 09- 10

作者简介: 罗兵 (1964-), 男, 重庆人, 重庆大学副教授, 博士, 主要从事物流与供应链管理和管理决策优化的研究.

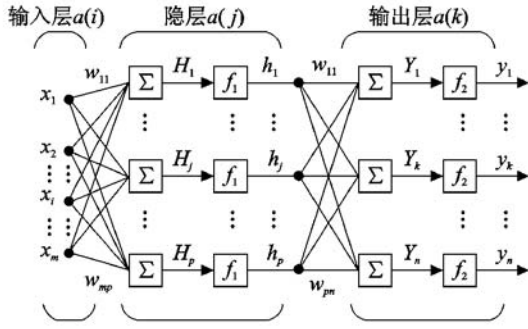


图 1 BP 神经网络 3 层拓扑结构

分别为隐层和输出层的转换函数, 一般为 sigmoid 函数, 包括 log-sigmoid 或 tan-sigmoid 转换函数, 能将 $(-\infty, +\infty)$ 的输入分别映射到区间 $(0, 1)$ 和 $(-1, +1)$ 中. 具体采用哪一种转换函数, 要根据输入/输出关系而定: 当输出不含有负值时, 采用 log-sigmoid 函数; 当输出含有负值时, 采用 tan-sigmoid 函数. 这些转换函数反映了单个神经元的响应特性 (即输入/输出关系), 人工神经网络的非线性映射能力正是来源于此.

tan-sigmoid 转换函数的一般形式为:

$$f(x, T, \theta) = \frac{e^{\frac{x-\theta}{T}} - e^{-\frac{x-\theta}{T}}}{e^{\frac{x-\theta}{T}} + e^{-\frac{x-\theta}{T}}} = 1 - \frac{2}{1 + e^{\frac{2(x-\theta)}{T}}}. \quad (5)$$

缩放系数 T 和位移参数 θ 决定了单个神经元的响应特性, 与 \mathbf{W} 结合形成功能强大的大规模并行信息处理网络的整体功能. 信息分布存储于各权值 \mathbf{W} 及各神经元转换函数中, 在神经节点相同的情况下, 比传统网络具有更强大的信息分布存储能力及非线性映射能力. 神经网络的特点就在于信息的分布式存储和并行协同处理.

对 $f(x, T, \theta)$ 关于 x, T, θ 分别求偏导有:

$$\frac{\partial f}{\partial x} = \frac{1}{T} (1 - f^2), \quad (6)$$

$$\frac{\partial f}{\partial T} = \frac{\theta - x}{T^2} (1 - f^2), \quad (7)$$

$$\frac{\partial f}{\partial \theta} = -\frac{1}{T} (1 - f^2). \quad (8)$$

根据式 (1) - (4), 可以得到:

$$\frac{\partial H_j}{\partial v_{ij}} = x_i \quad (9)$$

$$\frac{\partial H_j}{\partial T_j} = \frac{1}{T_j} (1 - h_j^2), \quad (10)$$

$$\frac{\partial Y_k}{\partial v_{jk}} = h_j \quad (11)$$

$$\frac{\partial Y_k}{\partial T_k} = \frac{1}{T_k} (1 - y_k^2). \quad (12)$$

对误差函数 $E = \frac{1}{2} \sum_{k=1}^n (y_k - y_k^*)^2$ 关于 y_k 求偏

导有:

$$\frac{\partial E}{\partial y_k} = y_k - y_k^*. \quad (13)$$

联立式 (12)、(13), 可得:

$$\frac{\partial E}{\partial Y_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial Y_k} = \frac{1}{T_k} (y_k - y_k^*) (1 - y_k^2). \quad (14)$$

给定输入/输出后, 误差函数 E 由 T, θ 和 \mathbf{W} 决定, 即 $E = G(\mathbf{W}, T, \theta)$ 为关于 T, θ 和 \mathbf{W} 的 $(n + m) \times p + 2(p + n)$ 元函数, 则 BP 算法对网络的训练被看成是在一个高维空间中寻找一个多元函数的极小点, 这是一个目标函数为 E 的优化问题.

1.2 权值 \mathbf{W} 的调整

1) 输出层的权值 w_{jk} 的调整

$$\begin{aligned} \frac{\partial E}{\partial v_{jk}} &= \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial v_{jk}} = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial Y_k} \cdot \frac{\partial Y_k}{\partial v_{jk}} = \\ &= \frac{1}{T_k} (y_k - y_k^*) (1 - y_k^2) h_j. \end{aligned} \quad (15)$$

2) 隐层的权值 w_{ij} 的调整

$$\frac{\partial E}{\partial v_{ij}} = \frac{\partial E}{\partial H_j} \cdot \frac{\partial H_j}{\partial v_{ij}} = \frac{\partial E}{\partial H_j} \cdot \frac{\partial h_j}{\partial H_j} \cdot \frac{\partial H_j}{\partial v_{ij}} =$$

$$\sum_{k=1}^n \left[\frac{\partial E}{\partial Y_k} \cdot \frac{\partial Y_k}{\partial h_j} \right] \frac{\partial h_j}{\partial H_j} \cdot \frac{\partial H_j}{\partial v_{ij}} =$$

$$\sum_{k=1}^n \left[\frac{\partial E}{\partial Y_k} \cdot \frac{\partial \sum_{j=1}^p h_j w_{jk}}{\partial h_j} \right] \frac{\partial h_j}{\partial H_j} \cdot \frac{\partial H_j}{\partial v_{ij}} =$$

$$\frac{1}{T_j} \sum_{k=1}^n \left[\frac{1}{T_k} (y_k - y_k^*) (1 - y_k^2) w_{jk} \right] \cdot (1 - h_j^2) x_i. \quad (16)$$

则 $\Delta v(N) = -L_w \frac{\partial E}{\partial v(N)}$, 为减少网络振荡, 需考虑上次修改, 引入动量项, 则有:

$$\Delta v(N) = -L_w \frac{\partial E}{\partial v(N)} + \alpha_w \Delta v(N-1), \quad (17)$$

其中, N 为训练循环次数, L_w 为 \mathbf{W} 的学习率, α_w 为其动量 (惯性) 系数 ($0 \leq \alpha_w \leq 1$).

1.3 缩放系数 T 的调整

1) 输出层中的神经元的缩放系数 T_k 的调整

$$\frac{\partial E}{\partial T_k} = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial T_k} = \frac{1}{T_k^2} (y_k - y_k^*) (\theta_k - y_k) (1 - y_k^2). \quad (18)$$

2) 隐层中的神经元的缩放系数 T_j 的调整

$$\begin{aligned} \frac{\partial E}{\partial T_j} &= \sum_{k=1}^n \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial T_j} = \sum_{k=1}^n \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial Y_k} \cdot \frac{\partial Y_k}{\partial T_j} = \\ &= \sum_{k=1}^n \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial Y_k} \cdot \frac{\partial \sum_{j=1}^p h_j w_{jk}}{\partial T_j} = \sum_{k=1}^n \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial Y_k} \cdot \frac{\partial h_j w_{jk}}{\partial T_j} = \\ &= \sum_{k=1}^n \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial Y_k} \cdot w_{jk} \cdot \frac{\partial h_j}{\partial T_j} = \\ &= \frac{1}{T_j^2} \sum_{k=1}^n \left[\frac{1}{T_k} (y_k - y_k^*) (1 - y_k^2) w_{jk} \right] (\theta_j - H_j) (1 - h_j^2). \end{aligned} \tag{19}$$

则 $\Delta T(N) = -L_T \frac{\partial E}{\partial T(N)}$, 引入动量项, 有:

$$\Delta T(N) = -L_T \frac{\partial E}{\partial T(N)} + \alpha_T \Delta T(N-1), \tag{20}$$

其中, L_T 为 T 的学习率, α_T 为其动量(惯性)系数 ($0 \leq \alpha_T \leq 1$).

1.4 位移参数 θ 的调整

1) 输出层中的神经元的位移参数 θ_k 的调整

$$\frac{\partial E}{\partial \theta_k} = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial \theta_k} = -\frac{1}{T_k} (y_k - y_k^*) (1 - y_k^2). \tag{21}$$

2) 隐层中的神经元的位移参数 θ_j 的调整

$$\begin{aligned} \frac{\partial E}{\partial \theta_j} &= \sum_{k=1}^n \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial \theta_j} = \sum_{k=1}^n \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial Y_k} \cdot \frac{\partial Y_k}{\partial \theta_j} = \\ &= \sum_{k=1}^n \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial Y_k} \cdot \frac{\partial \sum_{j=1}^p h_j w_{jk}}{\partial \theta_j} = \sum_{k=1}^n \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial Y_k} \cdot w_{jk} \frac{\partial h_j}{\partial \theta_j} = \\ &= -\frac{1}{T_j} \sum_{k=1}^n \left[\frac{1}{T_k} (y_k - y_k^*) (1 - y_k^2) w_{jk} \right] (1 - h_j^2). \end{aligned} \tag{22}$$

则 $\Delta \theta(N) = -L_\theta \frac{\partial E}{\partial \theta(N)}$, 引入动量项, 有:

$$\Delta \theta(N) = -L_\theta \frac{\partial E}{\partial \theta(N)} + \alpha_\theta \Delta \theta(N-1), \tag{23}$$

其中, L_θ 为 θ 的学习率, α_θ 为其动量(惯性)系数 ($0 \leq \alpha_\theta \leq 1$).

2 算例

笔者将改进后的算法用于某医院的库存控制, 预测 17GY 头皮式留置针的需求量. 网络输入是一个 1×9 的矩阵: $X = [x_1, x_2, \dots, x_9]$

$x_1 = T(n)$ 为使用周期(天); $x_2 = S(n)$ 为期初库存量(支); $x_3 = Q(n)$ 为阶段订购批量(支); $x_4 = SC(n)$ 为单位存货阶段储存成本(元/单位·阶段); $x_5 =$

$P1(n)$ 为手术室病人人数(人); $x_6 = P2(n)$ 为小儿科病人人数(人); $x_7 = P3(n)$ 为肝胆科病人人数(人); $x_8 = P4(n)$ 为普外科病人人数(人); $x_9 = P5(n)$ 为放疗科病人人数(人).

网络输出是一个 1×1 的矩阵: $Y = [y_k]$

$y_k = U$ 为 17GY 留置针阶段使用量(支).

采用 C 语言编制了改进算法的程序, 如图 2 所示.

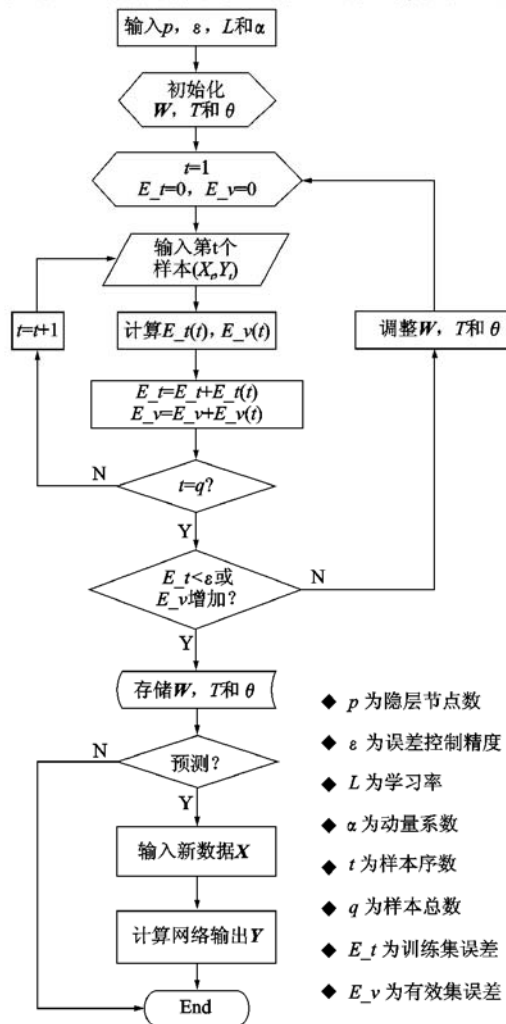


图 2 程序流程

其中:

- 1) 程序对 W, T 和 θ 取相同的学习率和动量系数;
- 2) 训练样本一次性输入, 对不同类别样本交替学习;

3) 计算误差 (MSE), $E = \frac{\sum_{t=1}^q \sum_{k=1}^n [y_k(t) - y_k^*(t)]^2}{q \times n}$,

式中 $y_k(t)$ 和 $y_k^*(t)$ 分别为第 t 个样本对应的输出层节点 $a(k)$ 的实际和期望输出值, $t = 1, 2, \dots, q$ (q 为训练样本总数).

选取不同隐层节点数和学习率进行网络训练, 采取方案及结果如表 1 所示.

表 1 网络训练方案及结果

隐层节点数	训练集 MSE	测试集 MSE	L_{min}	L_{max}	训练结果
2	8.50×10^{-6}	1.08×10^{-4}	0.4	0.65	每次迭代的时间短, 但所需迭代次数多, 故收敛速度慢; 训练集的误差比较大; 训练能全部成功.
3	6.50×10^{-6}	8.66×10^{-5}	0.25	0.575	
4	4.62×10^{-6}	7.84×10^{-5}	0.25	0.65	综合考虑训练集误差和测试集误差, 此范围的隐层节点数为最佳. 在所列 L 范围, 收敛速度快, 训练均成功, 但超出此范围后, 若 L 过小, 收敛速度慢, 跳出局部极值的能力弱, 若 L 过大, 出现振荡, 不易收敛, 甚至发散.
5	3.30×10^{-6}	7.63×10^{-5}	0.25	0.575	
6	2.27×10^{-6}	8.97×10^{-5}	0.125	0.425	
7	1.86×10^{-6}	9.86×10^{-5}	0.07	0.5	
8	1.60×10^{-6}	1.08×10^{-4}	0.08	0.45	
9	1.26×10^{-6}	1.15×10^{-4}	0.039	0.4	
10	1.26×10^{-6}	1.26×10^{-4}	0.054	0.45	
11	1.42×10^{-6}	1.37×10^{-4}	0.054	0.35	收敛速度快, 训练所需迭代次数少, 由于测试集误差变大, 成功率较低.
12	1.49×10^{-6}	1.41×10^{-4}	0.009	0.25	
13	1.10×10^{-6}	1.50×10^{-4}	0.03	0.25	
14	1.01×10^{-6}	1.70×10^{-4}	0.029	0.35	每次迭代的时间长, 但所需迭代次数少, 故收敛速度快; 训练集的误差相当小; 因测试集误差相当大, 故不能用此范围的隐层节点数.
15	8.99×10^{-7}	1.86×10^{-4}	0.008	0.35	
16	1.17×10^{-6}	2.07×10^{-4}	0.008	0.3	

表 2 给出改进前、后算法在预测 17GY 头皮式留置针需求量中训练网络的不同效果. 结果表明, 改进算法能有效地减少隐层节点数和加快收敛速度.

表 2 网络训练的效果比较

算法	最佳隐层节点数 / 个	能达到的最高精度 (均方误差 MSE)	平均一次迭代时间 / ms	平均迭代次数	总迭代时间 / s
改进前	9	10×10^{-5}	108	58	6.26
改进后	6	0.05×10^{-5}	152	26	3.95

图 3 显示了训练 (8-6-1) 网络的误差曲线. 虚线和实线分别表示算法改进前、后网络训练误差曲线. 图 3 表明, 改进后的算法能加快收敛速度并提高收敛精度.

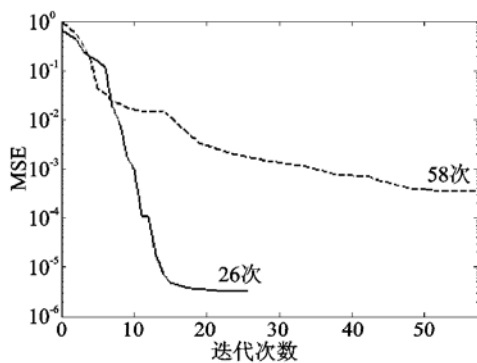


图 3 算法改进前、后的网络训练误差曲线

3 结 论

传统的 BP 算法通常只对权值进行调整, 而忽视了对转换函数的调整, 这种“简化”导致网络学习收敛速度慢, 甚至无法训练成功等一系列问题. 笔者提出的同时对权值 W 和 tan-sigmoid 转换函数参数进行调整的改进算法, 具有以下特点:

1) 非线性映射能力强. 使用更少的隐层节点数,

存储信息的能力强.

2) 健壮性. 网络训练成功率高, 能很大程度上跳出局部极小值.

3) 收敛速度快. 一旦进入收敛区, 能快速收敛至误差趋于 0

4) 收敛精度高. 能训练出误差 $E \leq 0.05 \times 10^{-5}$ 的网络.

参考文献:

- [1] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning Representations by Back-propagating Errors [J]. Nature, 1986, 323(9): 533-536
- [2] 胡耀斌, 李凯扬, 钟毓宁. 一种改进的神经网络 BP 算法 [J]. 武汉大学学报 (自然科学版), 1999, 45(1): 25-29
- [3] 王建成, 高大启, 王静, 等. 改进的遗传和 BP 杂交算法及神经网络经济预警系统设计 [J]. 系统工程理论与实践, 1998, 4(4): 136-141
- [4] 王晓萍, 黄海, 蒋化冰. BP 神经网络 Vogl 快速算法的改进 [J]. 浙江大学学报 (工学版), 2000, 34(2): 143-146
- [5] 李晓峰, 刘光中. 人工神经网络 BP 算法的改进及其应用 [J]. 四川大学学报 (工程科学版), 2000, 32(2): 105-109
- [6] 刘鹰, 赵琳. 神经网络 BP 算法的改进和仿真 [J]. 计算机仿真, 1999, 16(3): 12-14
- [7] HAGAN M T, MENHAJ M. Training Feed Forward Networks with the Marquardt Algorithm [J]. IEEE Transactions on Neural Networks, 1994, 5(6): 989-993
- [8] HIROSHI T, HISAKI H. The Functional Localization of Neural Networks Using Genetic Algorithms [J]. Neural Networks, 2003, 16(1): 55-67
- [9] NARENDRA K S, MUKHOPADHYAY S. Adaptive Control Using Neural Networks and Approximate Models [J]. IEEE Transactions on Neural Networks, 1997, (8): 475-485

(下转第 158 页)

Research of Investment and Cash Flow Constraint Under Ownership Concentration

Q U Long guang^{1,2}

(1. College of Economics & Business Administration, Chongqing University, Chongqing 400030, China)

2. Department of Economics & Business Administration, Yibin College, Sichuan 644007, China)

Abstract This article summarizes the theory achievements, arguments, the investment and ownership concentration model which studied in investment-cash flow and the stockholder's rights structure questions. The financing restraint is explained. The special social stratum shareholder is elaborated which influences relation between investment and internal cash flow by the financing restraint existing in the enterprise, the management opportunism hypothesis and the information asymmetrical theory separately. Finally the authors induce and study the real diagnosis model of the supposition above. The model is important for the empirical study of ownership structure influencing to corporate investment.

Key words investment, cash flow constraint, ownership concentration

(编辑 刘道芬)

(上接第 153 页)

Improved Algorithm of BP Neural Networks Based on Parameters Adjustment of Tan-Sigmoid Transfer Function

LUO Bing, HUANG Wan-jie, YANG Shuai

(College of Economics and Business Administration, Chongqing University, Chongqing 400030, China)

Abstract In order to improve the performance of BP neural networks, the network weight matrix W and the parameters T , θ of neuron's tan-sigmoid transfer function are adjusted. Information is stored in the weight matrix and the transfer functions dispersedly. The improved algorithm has stronger nonlinear mapping capability than traditional algorithm. The final formula of the improved algorithm is presented after rigorous mathematical deducing. An example of demand forecast is presented. The result indicates that the improved algorithm can reduce the nodes of hidden layer, accelerate the convergence and improve the convergence precision effectively.

Key words BP neural networks, tan-sigmoid transfer function, dispersed storage

(编辑 刘道芬)