

基于灵敏度分析法的ELM剪枝算法

李凡军^{1,2}, 韩红桂¹, 乔俊飞¹

(1. 北京工业大学 电子信息与控制工程学院, 北京 100124; 2. 济南大学 数学科学学院, 济南 250022)

摘要: 针对极端学习机(ELM)网络结构设计问题, 提出基于灵敏度分析法的ELM剪枝算法. 利用隐含层节点输出和相对应的输出层权值向量, 定义学习残差对于隐含层节点的灵敏度和网络规模适应度, 根据灵敏度大小判断隐含层节点的重要性, 利用网络规模适应度确定隐含层节点个数, 删除重要性较低的节点. 仿真结果表明, 所提出的算法能够较为准确地确定与学习样本相匹配的网络规模, 解决了ELM网络结构设计问题.

关键词: 前馈神经网络; 极端学习机; 灵敏度分析; 剪枝算法

中图分类号: TP273

文献标志码: A

Pruning algorithm for extreme learning machine based on sensitivity analysis

LI Fan-jun^{1,2}, HAN Hong-gui¹, QIAO Jun-fei¹

(1. College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing 100124, China; 2. School of Mathematical Science, Ji'nan University, Ji'nan 250022, China. Correspondent: QIAO Jun-fei, E-mail: junfeiq@bjut.edu.cn)

Abstract: In order to design the structure of extreme learning machine(ELM), a pruning algorithm is proposed by using the sensitivity analysis method. The residual error's sensitivities to the hidden nodes are defined by their outputs and weight vectors connecting to the output layer. The model scale adaptability is calculated and the hidden nodes are sorted by using the defined sensitivities. Then, the number of requisite hidden nodes is estimated by the model scale adaptability. The redundant nodes with smaller sensitivities are removed from the existent network. The simulation results show that the proposed approach can construct the compact structure for ELM effectively.

Key words: feedforward neural networks; extreme learning machine; sensitivity analysis; pruning algorithm

0 引言

针对单隐含层前馈神经网络(SLFNs), 文献[1-2]提出了ELM算法. 该算法随机选取输入层权值和阈值, 并保持不变, 采用最小二乘法求得输出层权值. 与传统的梯度下降算法相比, ELM算法具有更快的学习速度、较强的学习和泛化能力, 在理论研究和实际应用等诸多领域引起了人们的广泛关注^[3-6].

ELM算法惟一需要设定的参数是隐含层节点个数, 往往需要比传统方法更高维数的网络结构才能达到较好的学习效果^[6]. 而隐含层节点过多容易产生过拟合现象, 影响网络的泛化能力^[7-9]. 如何设计与学习样本相匹配的网络结构是ELM算法需要解决的问题.

剪枝算法是前馈神经网络结构设计的有效方

法^[6-9], 该类算法首先构造一个足够大的网络, 然后通过训练删除或合并某些节点或权值, 实现设计紧凑网络结构的目的. 文献[6]提出基于替代函数和贝叶斯框架的1范数ELM算法(N1-ELM), 在二次损失函数基础上引入1范数正则项, 使部分对网络影响较小的输出层权值逐渐趋近于零, 当隐含层节点的输出层权值小于某一给定阈值时, 将该节点删除. N1-ELM算法在保持较高测试精度的同时, 能够有效简化网络结构. 但N1-ELM算法主要适用于回归问题, 且计算量较大, 学习时间较长. 针对分类问题, Rong等^[7]提出ELM剪枝算法(P-ELM), 以概率的方法计算隐含层节点与样本类别的相关性, 剔除不相关和相关性较小的隐含层节点, 在保持较高分类准确率的前提下, 能够得到

收稿日期: 2013-03-22; 修回日期: 2013-06-22.

基金项目: 国家自然科学基金项目(61034008, 61203099, 61225016); 北京市自然科学基金项目(4122006); 教育部博士点新教师基金项目(20121103120020).

作者简介: 李凡军(1977-), 男, 博士生, 从事智能信息处理的研究; 乔俊飞(1968-), 男, 教授, 博士生导师, 从事智能控制与智能信息处理等研究.

规模较小的网络. 该算法主要适用于分类问题, 且学习时间较长. Miche 等^[8]提出的极端学习机最优剪枝算法(OP-ELM)可以用于分类和函数逼近问题, 其基本思想是: 首先利用 ELM 算法建立一个规模较大的 SLFN; 然后利用多响应稀疏回归算法(MRSR)对隐含层节点排序; 最后采用留一准则(LOO)对隐含层节点进行删除. OP-ELM 算法能够有效精简网络规模, 但计算过程较为复杂. 灵敏度分析法(SA)可以根据网络输出分析网络参数对网络输出的影响, 用于前馈神经网络结构修剪已取得较好的效果^[10-11]. 而传统的灵敏度分析法不能直接应用于 ELM 算法的网络结构修剪, 其困难在于寻找恰当的灵敏度计算公式和确定网络规模的方法.

针对以上问题, 本文给出了学习残差相对于隐含层节点的灵敏度和网络规模适应度的计算方法, 并提出基于 SA 的 ELM 剪枝算法(SA-ELM), 解决了 ELM 算法的网络结构设计问题. 在保持较高测试精度的同时, 构建更加紧凑的网络结构. SA-ELM 算法根据隐含层节点输出及其对应的输出层权值定义训练残差相对于隐含层节点的灵敏度, 据此对隐含层节点排序, 并计算网络规模适应度, 选择灵敏度较低的隐含层节点删除, 被删除节点的输入层权值和阈值采用叠加的方法平均分配给其他保留节点. 实验结果表明, SA-ELM 算法能够获得与样本相匹配的网络结构. 与 N1-ELM、P-ELM 和 ELM 算法相比, SA-ELM 算法构建的神经网络结构更加紧凑, 泛化能力更强.

1 基于 SA 的 ELM 剪枝算法

1.1 SLFNs 和 ELM 算法

给定样本集 $\{(x_i, t_i) | x_i \in R^n, t_i \in R^m, 1 \leq i \leq N\}$. 其中: $x_i = (x_{i1}, x_{i2}, \dots, x_{in})^T$, $t_i = (t_{i1}, t_{i2}, \dots, t_{im})^T$. 隐含层节点个数为 L , 激活函数为 $g(x)$, 则 SLFNs 的数学模型表示为

$$f_L(x_i) = \sum_{j=1}^L \beta_j g(w_j x_i + b_j) = o_i, \quad 1 \leq i \leq N. \quad (1)$$

其中: $w_j = (w_{j1}, w_{j2}, \dots, w_{jn})^T$ 为第 j 个隐含层节点输入层权值, b_j 为其阈值; $\beta_j = (\beta_{j1}, \beta_{j2}, \dots, \beta_{jm})^T$ 为第 j 个隐含层节点与输出节点连接的权值向量; $o_i = (o_{i1}, o_{i2}, \dots, o_{im})^T$ 为第 i 个样本对应的网络输出.

文献[1]已证明, 通过适当调节网络参数, SLFNs 可以以任意精度拟合训练样本, 所以式(1)可进一步改写为

$$f_L(x_i) = \sum_{j=1}^L \beta_j g(w_j x_i + b_j) = t_i,$$

$$1 \leq i \leq N. \quad (2)$$

为便于表述, 将上述 N 个方程写成矩阵方程的形式 $H\beta = T$. 其中: $H_{N \times L}$ 为隐含层输出矩阵, $\beta_{L \times m}$ 为网络输出层权值矩阵, T 为目标输出, 分别为

$$H = \begin{bmatrix} g_1(x_1) & \cdots & g_L(x_1) \\ \vdots & \ddots & \vdots \\ g_1(x_N) & \cdots & g_L(x_N) \end{bmatrix}_{N \times L}, \quad (3)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}, \quad (4)$$

$$g_j(x_i) = g(w_j x_i + b_j),$$

$$i = 1, 2, \dots, N, \quad j = 1, 2, \dots, L.$$

针对上述单隐含层神经网络, 给出 ELM 算法的具体步骤^[1-2]如下.

Step 1: 随机选取输入层权值向量 w_j 和阈值 b_j ($j = 1, 2, \dots, L$), 并保持不变.

Step 2: 计算隐含层输出矩阵 H .

Step 3: 计算 $\beta = H^+ T$, 其中 H^+ 为矩阵 H 的 Moore-Penrose 逆.

1.2 灵敏度及网络规模适应度定义

记

$$k_{ji} = g_j(x_i) = g(w_j x_i + b_j), \quad i = 1, 2, \dots, N,$$

则由式(2)可得

$$t_i = k_{1i}\beta_1 + k_{2i}\beta_2 + \cdots + k_{Li}\beta_L. \quad (5)$$

不失一般性, 假设删除节点 $j = 1$, 则网络输出变为

$$t'_i = k_{2i}\beta_2 + k_{3i}\beta_3 + \cdots + k_{Li}\beta_L. \quad (6)$$

由式(5)和(6)可得残差

$$\|t_i - t'_i\| = \|k_{1i}\beta_1\| = |k_{1i}| \|\beta_1\|. \quad (7)$$

由式(7)可见, 对于第 i 个样本, 去掉第 1 个隐含层节点引起的网络输出误差为节点输出的绝对值与相应输出层权值向量模的乘积. 因此, 定义学习残差相对于第 j 个隐含层节点的灵敏度为

$$S_j = \frac{1}{N} \sum_{i=1}^N |k_{ji}| \|\beta_j\|. \quad (8)$$

其中: N 为训练样本个数, k_{ji} 为第 j 个隐含层节点对于第 i 个样本的输出, β_j 为与第 j 个隐含层节点连接的输出层权值向量. S_j 越大表明第 j 个隐含层节点对于学习残差越重要, 所以由灵敏度可对隐含层节点进行排序

$$S'_1 \geq S'_2 \geq \cdots \geq S'_L. \quad (9)$$

因为隐含层节点删除越多, 学习残差越大, 删除的隐

含层节点越重要, 学习残差越大, 所以网络规模适应度(MSA)可由灵敏度定义为

$$M_k = \frac{\sum_{j=1}^k S'_j}{L}, \quad 1 \leq k \leq L. \quad (10)$$

M_k 越大, 网络规模 k 越大, 学习残差越小, 因此与学习样本相匹配的网络规模可由网络规模适应度定义为

$$M = \min\{k | M_k \geq \gamma, 1 \leq k \leq L\}. \quad (11)$$

其中: γ ($0 \leq \gamma \leq 1$) 为网络规模适应度阈值, M 为网络保留隐含层节点数量, $L - M$ 为网络中冗余节点个数. 关于阈值 γ 的选择, 问题依赖性比较强, 至今没有一个很好的方法. ELM算法的性能对隐含层节点个数不是非常敏感^[12], γ 取值不需要非常精确, 所以可以由 $\gamma = 0.5$ 开始, 以 ± 0.1 增减试凑的方式确定, 操作较为方便. 为了防止样本信息因节点被删除而消失, 被保留节点的输入权值需要进一步更新.

1.3 权值更新公式

根据灵敏度 S_j 将 L 个隐含层节点进行排序如下:

$$S_{j_1}^1 \geq S_{j_2}^2 \geq \dots \geq S_{j_M}^M \geq S_{j_{M+1}}^{M+1} \geq \dots \geq S_{j_L}^L. \quad (12)$$

其中: 下标 $1 \leq j_i \leq L$ 为排序前的隐节点序号, 上标 $1 \leq i \leq L$ 为排序后的隐节点位次序号. 由式(10)和(11)选择位次靠前的 M 个隐含层节点保留, 后 $L - M$ 个删除, 记 $R = \{j_1, j_2, \dots, j_M\}$ 为保留节点集, $D = \{j_{M+1}, j_{M+2}, \dots, j_L\} = \{d_1, d_2, \dots, d_{L-M}\}$ 为删除节点集. 为了保存被删除节点的样本信息, 同时消除过拟合信息, 采用权值横向平均传播的方法更新保留节点的输入层权值, 更新公式为

$$w_{j_l, i}^{\text{new}} = w_{j_l, i}^{\text{old}} + \frac{1}{L - M} \sum_{k=1}^{L-M} w_{d_k, i}. \quad (13)$$

其中: $1 \leq i \leq n$, $1 \leq l \leq M$, $w_{j_l, i}^{\text{old}}$ 和 $w_{j_l, i}^{\text{new}}$ 分别为更新前后的第 i 个输入节点与第 j_l 个保留节点的连接权值, $w_{d_k, i}$ 为第 i 个输入节点与第 d_k 个被删除节点的连接权值.

冗余节点删除后, 利用矛盾线性方程组的最小范数最小二乘解求得网络输出层权值为

$$\beta' = (H')^+ T. \quad (14)$$

其中: H' 为节点删除后神经网络的隐含层输出矩阵, β' 为新的网络输出层权值.

根据上述分析, 给出 SA-ELM 算法的详细步骤如下.

Step 1: 给定 N 个训练样本, 采用 ELM 算法构建

一个规模较大的 SLFN, 隐含层节点个数记为 $L < N$, 记隐含层输出矩阵为 H , 输出层权值矩阵为 β , 设置网络规模适应度阈值 γ , 转至 Step 2.

Step 2: 根据 H 和 β , 由式(8)计算学习残差相对于第 j 个隐含层节点的灵敏度 S_j ($1 \leq j \leq L$), 并对隐含层节点进行排序

$$S_{j_1}^1 \geq S_{j_2}^2 \geq \dots \geq S_{j_M}^M \geq S_{j_{M+1}}^{M+1} \geq \dots \geq S_{j_L}^L.$$

其中: 下标 $1 \leq j_i \leq L$ 为排序前的隐节点序号, 上标 $1 \leq i \leq L$ 为排序后的隐节点位次序号. 转至 Step 3.

Step 3: 根据学习残差灵敏度 S_j , 由式(10)计算网络规模适应度 M_k ($1 \leq k \leq L$), 转至 Step 4.

Step 4: 根据 M_k 和 γ , 由式(11)和(12)选择被删除节点, 转至 Step 5.

Step 5: 根据式(13)对保留隐含层节点输入权值进行更新.

Step 6: 对剪枝后的网络计算隐含层输出矩阵 H' , 并由式(14)计算网络输出层权值 β' .

由式(9)~(12)可见, 被删除节点的重要性相对较小, 因此在满足训练精度条件下, SA-ELM 算法能够得到结构更加紧凑的神经网络.

2 仿真结果

为了测试网络初始规模和网络规模适应度阈值对 SA-ELM 算法性能的影响, 第 2.1 节选取含噪声的模型数据进行仿真研究. 第 2.2 节选取 4 个实际问题, 分别与 N1-ELM、P-ELM 和 ELM 算法进行比较研究, 以验证 SA-ELM 算法的有效性. 所有实验程序运行环境为: 联想台式计算机 (CPU 3.4 Hz), 操作系统 Windows 7, Matlab 软件版本 7.1.

2.1 含噪声 Sinc 函数逼近

Sinc 函数^[2]为

$$y(x) = \begin{cases} \frac{\sin(x)}{x}, & x \neq 0; \\ 1, & x = 0. \end{cases} \quad (15)$$

其中 x 为 $(-10, 10)$ 上的均匀分布数列. 由式(15)产生 5000 个训练样本, 为了表明实验的可靠性, 训练样本叠加均匀分布在区间 $[-0.2, 0.2]$ 上的随机噪声, 同时采用相同的方法产生没有叠加噪声的 5000 个测试样本.

为了测试网络初始规模对 SA-ELM 算法性能的影响, 初始隐含层节点个数选择区间为 $[50, 500]$, 每间隔 50 取一次, 共计 10 种情况进行测试, 测试结果如图 1 所示. 初始隐节点个数增加到 100 后, 网络的学习性能基本不再变化, 学习与初始隐节点个数成正比. 建议 SA-ELM 算法网络初始规模不易过大, 一般要远小于学习样本数量. 为了测试网络规模适应度阈

值 γ 对网络性能的影响, γ 选取区间为 $[0, 1]$, 选取间隔为 0.1, 共计选取 11 种情况进行仿真研究. 图 2 为 γ 取值对网络性能的影响, 当 γ 取 $1 \sim 0.3$ 时, 隐节点数逐渐减少, 而网络性能基本没有发生变化, $\gamma = 0.3$ 以后性能急剧下降, 表明 SA-ELM 算法越早删除的隐节点对网络的重要性越小, 进一步表明本文定义的灵敏度能够准确地对隐节点进行排序, 而且可以通过调节 γ 的取值构建结构紧凑的神经网络.

为了研究 SA-ELM 算法的环境适应能力, 测试数

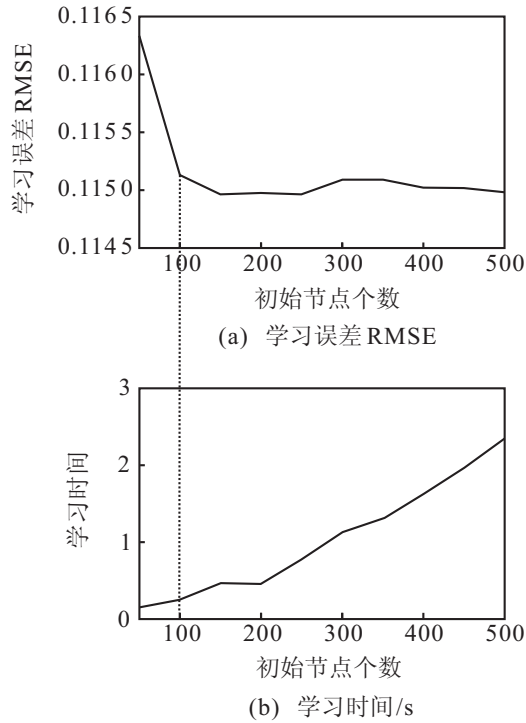


图 1 初始隐节点数对网络性能的影响

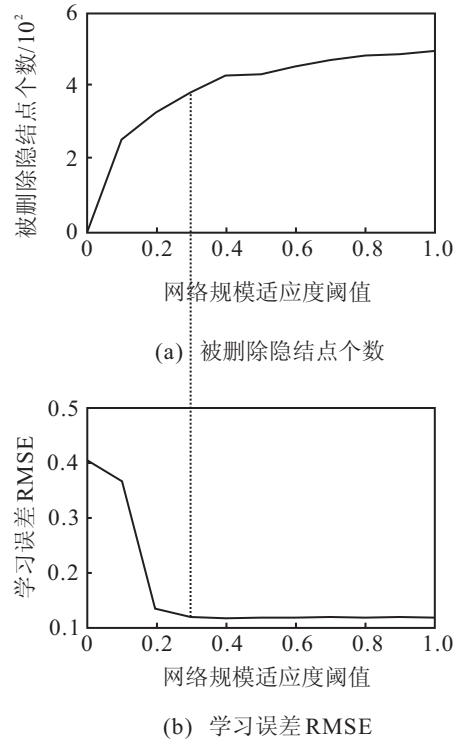


图 2 网络规模适应度阈值对隐节点个数的影响

据分别添加 11 种不同的噪声进行仿真研究, 噪声的标准差在区间 $[0, 0.1]$ 之间取值, 间隔为 0.01, 均值 0 不变, 仿真结果如表 1 所示. 由表示 1 可见, 在不同的噪声环境下, SA-ELM 算法只需要微调 γ 的取值, 网络即可自动找到与学习样本相匹配 (满足精度要求, 测试误差 $RMSE \leq 0.05$) 的隐节点数量. 同时可见, SA-ELM 算法可以通过调节网络规模适应度阈值, 大幅度删减网络冗余隐含层节点, 从而获得紧凑的网络结构, 并保持较高的学习精度.

表 1 噪声对 SA-ELM 算法的影响

噪声标准差	噪声均值	初始隐节点个数	学习误差	测试误差	最终隐节点个数	网络规模适应度阈值
0	0	500	0.026	0.035 6	22	0.24
0.01	0	500	0.024 5	0.037 3	16	0.17
0.02	0	500	0.026 9	0.039 9	20	0.25
0.03	0	500	0.026 8	0.043 7	20	0.2
0.04	0	500	0.024 7	0.040 5	24	0.24
0.05	0	500	0.027 7	0.043 2	18	0.21
0.06	0	500	0.027 2	0.041 4	20	0.21
0.07	0	500	0.028 3	0.041 1	17	0.19
0.08	0	500	0.027 2	0.041 2	19	0.21
0.09	0	500	0.028	0.043 6	23	0.25
0.10	0	500	0.028 6	0.044 9	26	0.25

表 2 数据描述

数据集	训练样本数	测试样本数	输入维数	输出维数/类别	问题类型
Abalone	2 000	2 177	7	1	回归
Census (house8L)	10 000	12 784	6	1	回归
Delta elevators	4 000	5 517	6	1	回归
Image segment	1 500	810	18	7	分类
Satellite image	4 400	2 000	36	6	分类

2.2 实际问题实验

为了进一步验证SA-ELM算法的有效性, 分别与N1-ELM、P-ELM和原始ELM算法进行比较研究. 实验数据全部采用UCI机器学习数据库中实际问题的真实数据, 以保证比较结果的可靠性和真实性, 同时, 便于比较研究回归问题^[5]和多分类问题测试算法^[6]的性能. 训练样本和测试样本的抽取方式与文献[5-6]相同, 表2给出了数据的简单描述.

4种方法隐含层节点激活函数全部为Sigmoid函数, 输入数据归一化至区间 $[-1, 1]$, 回归问题的输出数据归一化至区间 $[0, 1]$, 多分类问题根据类别构造对应的输出向量, 第 i 类对应输出向量的第 i 个位置取1, 其余位置取-1. 由于SA-ELM算法可以通过调节网络规模适应度阈值来限制隐含层节点数量, 本文全

部实验初始隐含层节点个数统一设置为400. γ 取值区间为 $(0, 1)$, 由0.5开始, 选取间隔为0.1, 选取结果见表3. 在相同条件下, 所有实验重复20次, 将训练时间、网络规模、测试误差的均值和方差进行比较. 回归问题的实验结果见表4, N1-ELM结果与文献[5]相同. 分类问题的结果见表5, P-ELM结果与文献[6]相同. 表4和表5中, 最优效果以粗体标出.

表3 SA-ELM算法参数设置

数据集	初始隐层节点数	网络规模适应度阈值
Abalone	400	0.3
Census-house(8L)	400	0.4
Delta elevators	400	0.2
Image segment	400	0.4
Satellite image	400	0.7

表4 回归实验结果比较

算法	Abalone				Census-house(8L)				Delta elevators			
	隐节点数	训练时间/s	测试误差		隐节点数	训练时间/s	测试误差		隐节点数	训练时间/s	测试误差	
			均值	标准差			均值	标准差			均值	标准差
SA-ELM	13	0.2161	0.0749	0.0006	40	1.0155	0.07	0.0013	15	1.5296	0.0537	0.0004
N1-ELM	33.45	1.1594	0.077	0.0014	50	4.2962	0.0705	0.002	20.25	2.5562	0.0534	0.0004
ELM	25	0.0187	0.0805	0.0037	120	0.9196	0.0701	0.0017	125	0.2812	0.0568	0.0028

表5 分类实验结果比较

算法	Image segmentation				Satellite image			
	隐节点数	训练时间/s	测试比率/%		隐节点数	训练时间/s	测试比率/%	
			均值	标准差			均值	标准差
SA-ELM	52.2	0.1975	95.124	0.00348	212	0.592	89.815	0.0046
P-ELM	198	16.22	94.83	0.622	397	49.533	89.955	0.563
ELM	200	1.4	95.01	0.78	500	14.92	89.04	1.5

由表4中3个回归实验的结果可见, 在测试误差基本相同的前提下, SA-ELM算法的隐节点个数比原始ELM算法分别减少了47%、66%和92%, 比N1-ELM算法分别减少了60%、20%和25%, 表明SA-ELM算法构建了结构更加紧凑的神经网络. 同时, 在学习时间上, SA-ELM算法比N1-ELM算法分别减少了81.4%、76.5%和40%, 学习速度更快. 由表5可见, 在保持相近测试比率的条件下, SA-ELM算法的网络规模分别为P-ELM算法的26%、42%; 在两个实验训练时间上, SA-ELM算法仅占P-ELM算法的1.2%, 表明SA-ELM算法能够快速准确地删除重要性较小的冗余节点, 确定与学习样本相匹配的隐节点个数. 由表4和表5的标准差可见, N1-ELM算法和P-ELM算法确定的网络规模对于学习样本的适应度是不确定的, 每次删除的隐节点对网络的性能影响变化较大. 对于学习样本的适应度, SA-ELM算法所确定的网络规模由定义的网络规模适应度确定, 每次删除的隐节

点的贡献率是一定的, 所以对网络性能的影响变化不大, 从而SA-ELM算法更加稳定.

通过实际问题数据实验表明, SA-ELM算法能够根据不同的样本数据, 在统一的网络初始结构前提下, 通过简单调节网络规模适应度阈值, 构建结构更加紧凑的神经网络. 在处理大规模实际问题数据时, 具有较快的学习速度和简单的网络结构, 拥有较好的学习和泛化性能, 且稳定性较好, 适用于实际问题建模.

3 结论

本文针对ELM算法的网络结构设计问题, 提出了基于灵敏度分析法的ELM神经网络结构修剪算法SA-ELM, 给出了学习残差相对于隐含层节点的灵敏度计算方法, 并通过灵敏度定义了新的网络规模适应度. SA-ELM算法首先利用隐含层节点输出及其对应的输出层权值向量计算节点对于学习残差的灵敏度, 根据灵敏度大小对隐含层节点排序, 同时计算网络规模适应度. 通过对比网络规模适应度与给定的适应度

阈值, 确定与训练样本相匹配的网络规模. 隐含层节点删除后, 通过横向传播, 将被删节点的权值平均叠加到其他节点的权值上, 以保留被删隐含层节点的样本信息, 从而实现在保持较高测试精度的条件下控制网络规模的目的. 通过比较 SA-ELM 算法与其他算法的实验结果, 得出如下结论:

1) SA-ELM 算法能够构建结构紧凑的神经网络, 并保持较高的测试精度.

2) 定义的灵敏度和网络规模适应度能够较为准确地确定与学习样本相匹配的网络规模.

综上所述, SA-ELM 算法能够很好地解决 ELM 算法的网络结构设计问题. 该算法能够在保持较高测试精度的同时构建更加紧凑的神经网络, 并且稳定性好, 泛化能力强, 适用于实际问题建模.

参考文献(References)

- [1] Bons Igel'nik, Pao Yoh-han. Stochastic choice of basis functions in adaptive function approximation and the functional-link net[J]. IEEE Trans on Neural Networks, 1995, 6(6): 1320-1329.
- [2] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: Theory and applications[J]. Neurocomputing, 2006, 70(1/2/3): 489-501.
- [3] Decherchi S. Circular-ELM for the reduced-reference assessment of perceived image quality[J]. Neurocomputing, 2013, 102(1): 78-89.
- [4] 陈华, 章兢, 张小刚, 等. 一种基于 Parzen 窗估计的鲁棒 ELM 烧结温度检测方法[J]. 自动化学报, 2012, 38(5): 841-849.
(Chen H, Zhang J, Zhang X G, et al. A robust-ELM approach based on parzen window's estimation for kiln sintering temperature detection[J]. Acta Automatic Sinica, 2012, 38(5): 841-849.)
- [5] Huang G B, Chen L, Siew C K. Universal approximation using incremental constructive feedforward networks with random hidden nodes[J]. IEEE Trans on Neural Networks, 2006, 17(4): 879-892.
- [6] 韩敏, 李德才. 基于替代函数及贝叶斯框架的 1 范数 ELM 算法[J]. 自动化学报, 2011, 37(11): 1344-1350.
(Han M, Li D C. An norm 1 regularization term ELM algorithm based on surrogate function and bayesian framework[J]. Acta Automatic Sinica, 2011, 37(11): 1344-1350.)
- [7] Rong H J, Ong Y S, Tan A H, et al. A fast pruned-extreme learning machine for classification problem[J]. Neurocomputing, 2008, 72(1): 4-8.
- [8] Miche Y, Sorjamaa A, Bas P, et al. OP-ELM: Optimally pruned extreme learning machine[J]. IEEE Trans on Neural Networks, 2010, 21(1): 158-162.
- [9] 张昭昭, 乔俊飞, 韩红桂. 一种基于神经网络复杂度的修剪算法[J]. 控制与决策, 2010, 25(6): 821-830.
(Zhang Z Z, Qiao J F, Han H G. A pruning algorithm based on neural complexity[J]. Control and Decision, 2010, 25(6): 821-830.)
- [10] 乔俊飞, 韩红桂. RBF 神经网络的结构动态优化设计[J]. 自动化学报, 2010, 36(6): 865-872.
(Qiao J F, Han H G. Optimal structure design for RBFNN structure[J]. Acta Automatic Sinica, 2010, 36(6): 865-872.)
- [11] Han H G, Qiao J F. A structure optimization algorithm for feedforward neural network construction[J]. Neurocomputing, 2013, 99(1): 347-357.
- [12] Huang G B, Wang D H, Lan Y. Extreme learning machines: A survey[J]. Int J of Machine Learning and Cybernetics, 2011, 2(2): 107-122.

(责任编辑: 郑晓蕾)