

网络结构增长的极端学习机算法

李凡军^{1,2}, 乔俊飞^{1†}, 韩红桂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, 山东 济南 250022)

摘要: 针对极端学习机(extreme learning machine, ELM)结构设计问题, 基于隐含层激活函数及其导函数提出一种前向神经网络结构增长算法. 首先以Sigmoid函数为例给出了一类基函数的派生特性: 导函数可以由原函数表示. 其次, 利用这种派生特性提出了ELM结构设计方法, 该方法自动生成双隐含层前向神经网络, 其第1隐含层的结点随机逐一生成. 第2隐含层的输出由第1隐含层新添结点的激活函数及其导函数确定, 输出层权值由最小二乘法分析获得. 最后给出了所提算法收敛性及稳定性的理论证明. 对非线性系统辨识及双螺旋分类问题的仿真结果证明了所提算法的有效性.

关键词: 前向神经网络; 极端学习机; 导数; 结构设计

中图分类号: TP273 **文献标识码:** A

Incremental constructive extreme learning machine

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

(1. College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing 100124, China;

2. School of Mathematical Science, University of Jinan, Jinan Shandong 250022, China)

Abstract: Focusing on the problem of architectural design of extreme learning machine (ELM), we propose a novel constructive algorithm by the activation function and its derivatives. Firstly, taking the Sigmoid function as an example, we give in detail the derived characteristics for a class of base functions: derivative functions can be expressed by their primitive functions. By making use of these derived characteristics, we propose a method to design the structure of ELM, which automatically generate feedforward neural networks with double hidden layers. The new units in the first hidden layer are generated randomly one by one; then, the outputs of the second hidden layer (derivation) are calculated by the activation function of the new node in the first layer and its derivatives. The weights of the output layer are calculated analytically by the least squares method. Finally, the analysis of convergence and stability are presented. The effectiveness of the proposed method is demonstrated by simulation results on nonlinear system identification and two-spiral classification problem.

Key words: feedforward neural networks; extreme learning machine; derivative; architectural design

1 引言(Introduction)

针对单隐含层前向神经网络(single hidden layer feedforward neural networks, SLFNs)学习速度慢的问题, Huang等人提出了极端学习机(extreme learning machine, ELM)^[1]算法, 该算法首先随机选取输入层权值及阈值, 然后通过隐含层输出矩阵的广义逆矩阵解析地求出输出层权值. ELM算法无需调整任何参数, 具有较快的学习速度和较强的泛化能力, 是前向神经网络研究领域的重要课题^[2-3], 在模式识别、图像处理、非线性系统的建模和优化等方面得到了广泛的应用^[4-7].

隐含层神经元个数是ELM神经网络(采用ELM算

法的SLFN)唯一需要确定的参数. 隐结点过少网络不能很好地学习训练样本, 过多则容易产生过拟合现象, 影响网络的泛化能力. 因此其应用的关键问题是设计与学习样本相匹配的网络结构. 前向神经网络的学习能力与其结构复杂度(自由参数的个数)¹成正比^[8], ELM神经网络需要较高维数的网络结构才能很好地学习训练样本. 而过于复杂的网络结构将导致过拟合现象, 影响网络的泛化能力. 所以ELM网络结构设计仍是一个普遍关注而尚未解决的问题.

近年来, 增量式算法已成为ELM神经网络结构设计的一个主流方向^[9-11]. 2006年, Huang等人提出了一种增长型的ELM(I-ELM)算法^[9], 该算法通过随机

收稿日期: 2013-09-23; 录用日期: 2014-01-20.

†通信作者. E-mail: junfeiq@bjut.edu.cn.

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

¹神经网络结构复杂度一般指的是自由参数或者可调参数的数量^[8], 因此本文定义ELM神经网络的结构复杂度为随机生成的隐结点数量或者是随机生成的输入层权值向量的维数.

增加隐结点的方式实现了神经网络结构的自适应增长, 具有很强的学习能力. 但Huang等人研究发现 I-ELM算法比较容易产生不良的隐结点, 增大了神经网络的结构复杂度, 并对I-ELM算法进行改进, 提出了EI-ELM(enhanced incremental ELM, EI-ELM)算法^[10]. 该算法每次随机产生多个隐结点(结点池), 从中选取效果最好的添加到网络中去, 能够构建结构更加紧凑的神经网络, 有效地降低ELM神经网络的结构复杂度. 但神经网络规模较大时, I-ELM及EI-ELM算法逐一添加隐结点的方式增加了算法的循环次数, 影响神经网络的泛化能力. Feng等人提出的误差最小化ELM(error minimized ELM, EM-ELM)算法^[11], 解决了单一增加隐结点的模式, 减少了ELM神经网络的学习时间. 但EM-ELM算法容易产生不良隐结点, 降低神经网络的泛化能力. 文献[9-11]提出的增量式算法, 在保持相同测试精度的前提下, 需要的隐结点个数要少于通过交叉检验方法所确定的隐结点个数.

针对上述问题, 本文基于神经网络隐含层激活函数及其导函数, 提出了一种网络结构自适应增长的ELM(incremental constructive ELM, IC-ELM)算法, 并从理论上证明了该算法的收敛性和稳定性. IC-ELM算法构造了一个双隐含层的前向神经网络(如图1所示), 第1隐含层的结点随机产生, 而第2隐含层(称为“衍生层”)结点的激活函数由第1隐含层结点的激活函数及其导函数确定. 该算法实现了网络结构的自适应增长, 降低了ELM神经网络的结构复杂度, 通过衍生层激活函数的多样化提高了神经网络的泛化能力. 对非线性系统辨识及双螺旋分类问题的仿真结果验证了IC-ELM算法的有效性.

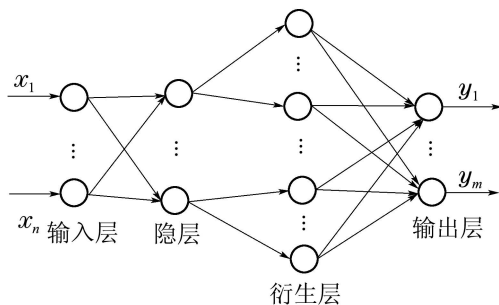


图 1 IC-ELM网络模型
Fig. 1 Structure of IC-ELM

2 网络结构增长的ELM算法(IC-ELM algorithm)

IC-ELM算法第1隐含层结点的激活函数为Sigmoid函数, 衍生层的激活函数由第1隐含层的激活函数及其导函数组成, 下面, 首先简单介绍Sigmoid函数及其导函数的特性.

2.1 Sigmoid函数及其导函数特性(Sigmoid function and its derivatives)

文献[12]研究发现, Sigmoid函数与其各阶导函数

之间具有一种派生关系: 高阶导函数可以由低阶导函数及其原函数简单表示. 以单极限Sigmoid函数为例, 其1, 2, 3阶导函数的派生关系如下:

$$f_0(u) = \frac{1}{1 + e^{-\lambda u}}, \quad (1)$$

$$f_1(u) = \frac{d}{du} f_0(u) = f_0(u)(1 - f_0(u)), \quad (2)$$

$$f_2(u) = \frac{d}{du} f_1(u) = f_1(u)(1 - 2f_0(u)), \quad (3)$$

$$f_3(u) = \frac{d}{du} f_2(u) = f_1(u)(1 - 6f_1(u)), \quad (4)$$

其中: $f_0(u)$ 为Sigmoid函数, $f_1(u)$, $f_2(u)$, $f_3(u)$ 分别为其1, 2, 3阶导函数. 图2(a)给出了Sigmoid函数及其导函数图像, 图2(b)给出的是Sigmoid函数及其归一化后的导函数图像. 由式(2)–(4)可以看出, 各阶导函数都是连续可导的有界函数, 并且图2表明它们归一化后的形状类似于常用的激活函数如: 正弦函数、高斯函数等. 所以, $f_1(u)$, $f_2(u)$, $f_3(u)$ 或归一化后的导函数可作为神经网络隐含层结点的激活函数. IC-ELM算法利用Sigmoid函数的派生特性计算其导函数, 将其作为衍生层的结点输出, 在基本实现正弦、高斯等函数映射作用的前提下, 可以减少ELM算法中随机权值的数量, 进而降低ELM神经网络的结构复杂度. 下面详细介绍IC-ELM算法的具体步骤.

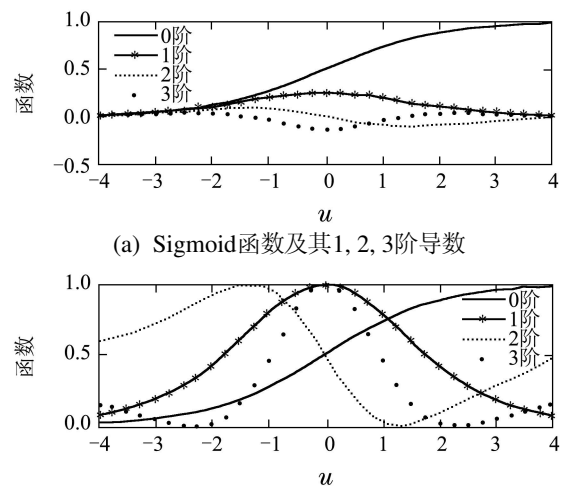


图 2 Sigmoid函数及其导函数

图 2 Sigmoid and its derivatives
Fig. 2 Sigmoid and its derivatives

2.2 IC-ELM算法步骤(Steps of IC-ELM)

将式(1)–(4)推广可知, 单极限Sigmoid函数的任意阶导函数均可由原函数的幂线性表示, 即

$$(f_0, f_1, \dots, f_{k-2}, f_{k-1}) = (f_0, f_0^2, \dots, f_0^{k-1}, f_0^k)A, \quad (5)$$

其中 A 为可逆的上三角矩阵. 由 $0 < f_0 < 1$ 可知 $f_0^k \rightarrow 0 (k \rightarrow \infty)$. 即随着 k 的增大, f_k 可以由 f_0, f_1, \dots, f_{k-1} 近似线性表示, 成为冗余结点而影响网络的

泛化性能. 因此本文将选择 f_0, f_1, f_2, f_3 作为衍生层结点的激活函数.

给定任意 N 个不同的训练样本 $\{(x_i, t_i) | (x_i, t_i) \in \mathbb{R}^n \times \mathbb{R}^m, 1 \leq i \leq N\}$, 记 $X = (x_1, x_2, \dots, x_N)$, $T = (t_1, t_2, \dots, t_N)^T$, 则对于含有 L 个隐含层结点, 激活函数 $g(x)$ 的 SLFN, ELM 算法的数学模型为

$$f_L(x_i) = \sum_{j=1}^L \beta_j g(w_j x_i + b_j) = t_i, 1 \leq i \leq N, \quad (6)$$

其中: $w_j = (w_{j1}, w_{j2}, \dots, w_{jn})^T$ 为隐含层第 j 个节点的输入权值向量, b_j 为其阈值, $\beta_j = (\beta_{j1}, \dots, \beta_{jm})^T$ 为连接隐含层第 j 个节点与输出层的权值向量. 其中 w_j 与 b_j 取较小的随机数, 且在学习过程中保持不变. 将上述 N 个方程写成矩阵形式 $H\beta = T$, 其中:

$$H = \begin{bmatrix} g_1(x_1) & \cdots & g_L(x_1) \\ \vdots & & \vdots \\ g_1(x_N) & \cdots & g_L(x_N) \end{bmatrix}_{N \times L}, \quad (7)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}. \quad (8)$$

ELM 算法目的在于寻找最小范数的 β 使得误差 $\|H\beta - T\|$ 最小化. 并采用伪逆算法得到输出权值 $\beta = H^+T$. 而 ELM 算法的网络规模难以确定, 网络的结构复杂度(随机权值的个数)较高. 为了自动确定 ELM 算法的网络规模, 降低网络的结构复杂度, IC-ELM 算法采用随机隐结点的激活函数及其导函数构造了第 2 隐含层(衍生层), 扩大了隐含层输出矩阵的维数. IC-ELM 算法的数学模型为

$$f_L(x_i) = \sum_{j=1}^L \sum_{k=0}^3 \beta_j^k f_{jk}(w_j x_i + b_j) = t_i, \quad 1 \leq i \leq N, \quad (9)$$

其中: w_j 与 b_j 与 ELM 算法中的相应参数定义相同, β_j^k 为第 j 个隐结点对应的第 k 个衍生结点与输出层的连接权值向量, f_{jk} 为第 j 个隐结点激活函数的 k 阶导函数. IC-ELM 算法目的在于寻找最小范数的 β 以及最小的 L 使得误差 $\|H\beta - T\|$ 最小化, 其中

$$H = \begin{bmatrix} f_{10}(x_1) & \cdots & f_{13}(x_1) & \cdots & f_{L3}(x_1) \\ \vdots & & \vdots & & \vdots \\ f_{10}(x_N) & \cdots & f_{13}(x_N) & \cdots & f_{L3}(x_N) \end{bmatrix}_{N \times (4L)}. \quad (10)$$

显然在具有相同随机权值的条件下, IC-ELM 算法衍生层输出矩阵的维数是 ELM 算法隐含层输出矩阵维数的 4 倍. 即在输出矩阵维数相同的条件下, IC-ELM 算法降低了网络结构复杂度, 同时丰富了隐结点激活函数的类型. 下面详细介绍 IC-ELM 算法的具体步骤.

给出含有任意 N 个互不相同训练样本的集合 $\{(x_i, t_i) | (x_i, t_i) \in \mathbb{R}^n \times \mathbb{R}^m, 1 \leq i \leq N\}$, 记 $X =$

(x_1, x_2, \dots, x_N) , $T = (t_1, t_2, \dots, t_N)^T$, 隐含层结点个数上限为 L , 学习目标误差设定为 E_0 , IC-ELM 算法具体步骤如下:

步骤 1 随机初始化网络输入层权值 W , 输出层权值 $\beta = W^T$, 建立一个没有隐含层的两层前向神经网络, 并计算网络输出误差, 即 $E = \|X^T \beta - T\|$, 令隐含层结点个数 $k = 0$, 转入步骤 2;

步骤 2 当 E 大于 E_0 并且 $k < L$ 时, 转入步骤 3, 否则停止计算;

步骤 3 随机产生一个新的隐含层结点, 新结点的输入权值记为 W^{new} , 阈值记为 b^{new} , 计算新结点的输入 $V^{\text{new}} = W^{\text{new}} X + b^{\text{new}}$, 由式 (1) 计算其输出 $U^{\text{new}} = f_0^{\text{new}}(V^{\text{new}})$, 根据式 (2)–(4) 计算出 U^{new} 对 V^{new} 的 1, 2, 3 阶导数, 即 $f_i(V^{\text{new}})$, $i = 1, 2, 3$, 转入步骤 4;

步骤 4 当 $k = 0$ 时, 创建一个衍生层, 构造一个 4 层前向神经网络(如图 1), 衍生层的结点输出分别为步骤 3 计算出的 $f_i(V^{\text{new}})$ ($i = 0, 1, 2, 3$), 衍生层的输出矩阵为 $H(1) = [f_0^T(V^{\text{new}}) \quad f_1^T(V^{\text{new}}) \quad f_2^T(V^{\text{new}}) \quad f_3^T(V^{\text{new}})]$, 转入步骤 5; 当 $k \neq 0$ 时, 衍生层新增 4 个结点, 输出分别为步骤 3 计算出的 $f_i(V^{\text{new}})$ ($i = 0, 1, 2, 3$), 衍生层的输出矩阵为 $H(k+1) = [H(k) \quad f_0^T(V^{\text{new}}) \quad f_1^T(V^{\text{new}}) \quad f_2^T(V^{\text{new}}) \quad f_3^T(V^{\text{new}})]$, 转入步骤 5;

步骤 5 利用 ELM 算法计算输出层的权值矩阵 $\beta(k+1) = H(k+1)^+T$, 其中 $H(k+1)^+$ 为 $H(k+1)$ 的广义逆矩阵, 转入步骤 6;

步骤 6 计算网络输出误差 $E(k+1) = \|H(k+1)\beta(k+1) - T\|$, $k = k+1$, 转入步骤 2.

由 IC-ELM 算法步骤可以看出, 隐含层每新增一个随机结点, 衍生层相应增加 4 个不同激活函数的输出结点, 因此减少了随机结点的数量, 降低了 ELM 神经网络的结构复杂度, 丰富了隐节点激活函数类型.

3 IC-ELM 算法的收敛及稳定性分析 (Convergence and stability of IC-ELM)

记衍生层第 k 次增加结点的输出分别为 $f_{k,0}, f_{k,1}, f_{k,2}, f_{k,3}$, 则衍生层输出矩阵 $H(k)$ 表示为

$$H(k) = H(f_{1,0}^T, \dots, f_{1,3}^T, \dots, f_{k,0}^T, \dots, f_{k,3}^T), \quad (11)$$

$$H(k+1) = H(H(k), f_{k+1,0}^T, \dots, f_{k+1,3}^T). \quad (12)$$

令

$$\Delta H(k+1) = [f_{k+1,0}^T \quad \cdots \quad f_{k+1,3}^T], \quad (13)$$

则

$$H(k+1) = H(H(k), \Delta H(k)). \quad (14)$$

定义衍生层输出矩阵 H 的误差函数 $E(H) = \min \|H\beta - T\|$, 则由文献 [11] 中结论可得

$$E(H(k+1)) = \min \|H(k+1)\beta(k+1) - T\| \leq$$

$$\min \|H(k)\beta(k) - T\| = E(H(k)). \quad (15)$$

即

$$E(H(1)) \geq E(H(2)) \geq \dots \geq E(H(k)). \quad (16)$$

由式(1)–(4)可以看出函数 $f_{ij}(1 \leq i \leq k, 1 \leq j \leq 3)$ 在整个实数域上连续有界且任意阶可导, 那么由文献[1]中结论可得, 对任意 N 个不同的样本 $(x_i, t_i) \in \mathbb{R}^n \times \mathbb{R}^m$ 及随机选取的输入层权值及阈值, 以概率1保证存在正整数 h , 当 $[N/4] < h \leq [N/4] + 1$ 时²,

$$E(H(h)) = 0. \quad (17)$$

由式(16)–(17)可得, 对任意小的 $\varepsilon > 0$ 以概率1保证存在整数 $k > 0$ 使得 $E(H(k)) < \varepsilon$, 即IC-ELM算法以概率1保证是收敛的. 由式(16)看出, 学习误差随着隐含层结点个数的增加而单调下降, 因此随着网络结构的不断增长, IC-ELM算法是稳定的.

4 仿真及结果分析(Simulations and results)

本部分程序运行环境为: 台式PC(CPU 3.4Hz), 操作系统Windows 7, 软件MATLAB 7.0.

表 1 实验4.1结果比较

Table 1 The comparison result in experiment 4.1

算法	学习时间/s		训练误差		测试误差		隐节点数	
	均值	均方差	均值	均方差	均值	均方差	均值	均方差
IC-ELM* ¹	1.17e-02	1.72e-03	6.73e-03	1.96e-03	5.80e-03	1.83e-03	6.2	0.421
ELM* ¹	1.25e-02	1.11e-03	6.79e-03	3.23e-03	6.08e-03	2.52e-03	25	0
EM-ELM	8.81e-03	2.99e-03	6.57e-03	2.22e-03	5.98e-3	1.97e-03	25.4	2.27
IC-ELM* ²	3.43e-02	2.68e-03	1.87e-06	1.58e-06	1.78e-06	1.58e-06	12.1	0.316
ELM* ²	4.13e-03	1.49e-03	5.45e-06	3.60e-06	3.20e-06	2.09e-06	48	0

由表1中(IC-ELM*¹, ELM*¹)学习误差及隐节点数量可看出, 与EM-ELM及ELM算法相比, IC-ELM算法在保持网络学习能力的前提下利用衍生层降低ELM神经网络的结构复杂度. 标准差说明, 由Sigmoid函数及其导函数构造的衍生层减少了输入层权值的个数, 降低了网络对随机权值的依赖性, IC-ELM算法的稳定性明显增强. 为了进一步验证IC-ELM算法的性能, IC-ELM及EM-ELM算法设定训练RMSE为 10^{-5} , ELM算法的网络隐节点数设置为48个, 其他条件不变. 在相同条件下每组实验进行10遍, EM-ELM均不能达到学习目标, IC-ELM及ELM的实验结果列于表1中(IC-ELM*², ELM*²). 由表1(IC-ELM*², ELM*²)数据可以看出, 在隐含层(衍生层)输出维数相同的情况下, IC-ELM算法由于采用Sigmoid函数及其导函数为激活函数, 丰富了激活函数类型, 与ELM算法相比, 明显提高了网络的学习及泛化能力. 实验中进一步设定训练误差RMSE设定为 10^{-6} , IC-ELM测试误差可以达到 10^{-7} 级别, 而ELM算法则无法达到这样的测试精度要求.

图3给出测试误差RMSE为 $3.65e - 007$ 时的系统

² $[N/4]$ = 不大于 $N/4$ 的最大整数

4.1 非线性动态系统辨识(Nonlinear system identification)

非线性动态系统由下列方程表示^[13]:

$$y(t+1) = \frac{y(t)y(t-1)[y(t)+2.5]}{1+y^2(t)+y^2(t-1)} + u(t), \quad (18)$$

其中: $u(t) = \sin(2\pi t/25)$, $t \in [1, 400]$, $y(0) = 0$, $y(1) = 0$, 选取100个点作为训练样本, 其中 $t \in [1, 100]$, 另外100个点作为测试样本, $t \in [301, 400]$, 训练均方根误差(root-mean-square error, RMSE)设定为0.01, 激活函数选取 $\lambda = 1$ 的Sigmoid函数; IC-ELM算法隐含层结点个数上限取100; 采用试错法确定ELM算法隐含层结点数为25; EM-ELM算法初始结点数设定为1, 每次添加4个隐结点, 隐含层结点个数上限取100. 在同样的条件下分别作10遍实验, 取CPU运行时间、训练误差、测试误差、隐结点个数(衍生层结点数)各自的平均值及标准差进行比较, 表1给出了比较结果(IC-ELM*¹, ELM*¹), 其中*¹表示训练均方根为0.01的实验; *²表示训练均方根为 10^{-5} 的实验.

辨识效果, 表明IC-ELM可以很好的逼近给定的非线性动态系统, 有很强的函数逼近能力和良好的泛化性能. 图4给出IC-ELM*²的学习误差曲线, 说明该算法具有良好的学习能力和稳定性. 图5则说明IC-ELM算法随着循环次数的增加, 衍生层结点也在不断增加, 实现网络结构的自适应增长. 非线性系统辨识结果表明IC-ELM算法实现了网络结构的自增长, 降低了网络的结构复杂度, 提高了ELM网络的函数逼近能力.

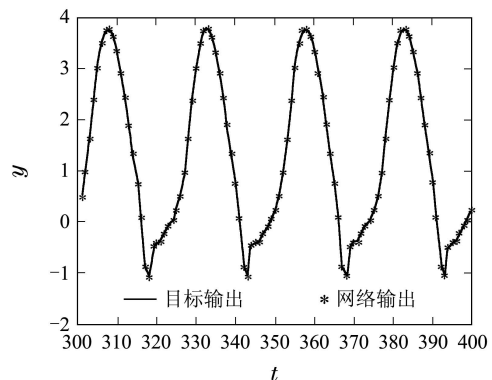


图 3 实验4.1中IC-ELM泛化效果

Fig. 3 Generalization of IC-ELM in experiment 4.1

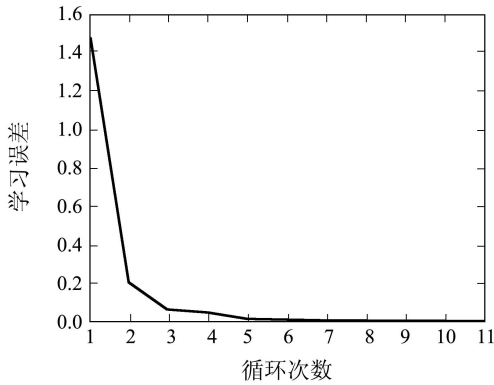


图4 实验4.1中IC-ELM学习误差曲线

Fig. 4 Training MSE of IC-ELM in experiment 4.1

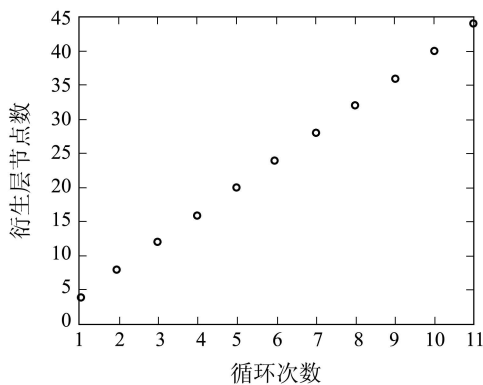


图5 IC-ELM隐结点增长图示

Fig. 5 Growing hidden units of IC-ELM

4.2 双螺旋分类问题(Two-spiral classification)

双螺旋是一个2分类问题, 要求把分布在 $x-y$ 平面上的分别属于2个不同螺旋线的点 $p(x, y)$ 加以正确分类. 由于两类螺旋线相互缠绕, 呈现高度非线性关系, 加大了模式分类难度, 用常规的BP网络很难解决^[14], 在神经网络研究领域里称为标杆(benchmark)问题. 因此, 双螺旋分类问题被认为是检验人工神经网络算法分类性能的基准问题. 本实验的训练样本集由下列方程产生:

$$\begin{cases} \theta = i \times \pi/16, \\ \rho = 6.5 \times (104 - i)/104, \\ x = \rho \times \sin \theta, \\ y = \rho \cos \theta, \end{cases} \quad (19)$$

其中: $i = 0, 1, \dots, 96$, (x, y) 及 $(-x, -y)$ 构成两类共计194个样本点(图6所示). 并且在 $[-7, 7] \times [-7, 7]$ 区域内以间隔0.1选取了19881个点作为测试样本.

在实验过程中IC-ELM算法隐含层结点激活函数采用 $\lambda = 1$ 的Sigmoid函数, 训练RMSE设定为0.1, 训练10次, 平均使用30个隐结点就可以达到学习要求, 并能100%正确区分两类螺旋, 而且对测试样本的分类效果也非常好, 如图8所示. 通过尝试, ELM算法隐含层结点设置为130个, 也能达到学习要求, 并能对训练样本正确分类, 但对测试样本的分类效果比较差,

如图7所示. 双螺旋实验说明, IC-ELM算法可以很好地解决两分类问题, 提高了ELM算法的泛化能力.

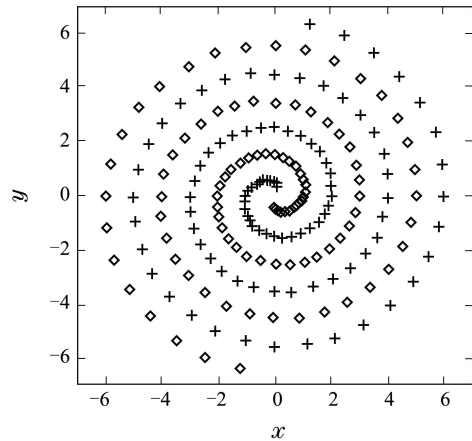


图6 实验4.2训练样本图示

Fig. 6 Training samples in experiment 4.2

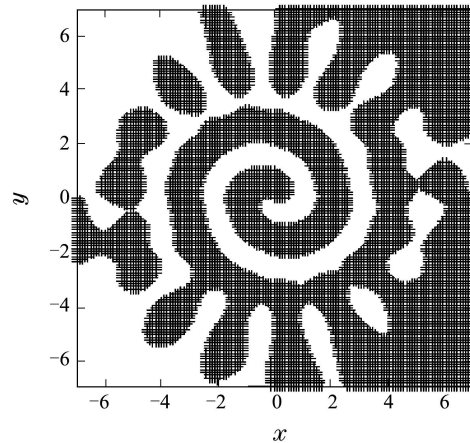


图7 ELM在实验4.2中的泛化效果

Fig. 7 Generalization of ELM in experiment 4.2

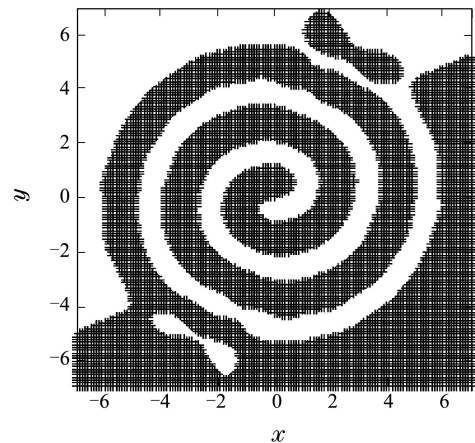


图8 IC-ELM在实验4.2中的泛化效果

Fig. 8 Generalization of IC-ELM in experiment 4.2

在相同条件下, IC-ELM及ELM对于双螺旋问题独立重复10次, 选取训练时间、训练分类率、测试分类率及随机结点的数量进行统计, 详细结果列于表2.

由表1和表2统计的误差和测试比率可以看出, IC-ELM通过衍生层输出丰富了隐含层结点的激活函数类型, 使学习误差和测试误差明显降低, 提高了ELM算法的学习和泛化性能. 同时表中学习时间证实, IC-ELM算法由于利用中间结果计算衍生层的输出,

无需进行复杂的运算, 提高了ELM算法学习速度. 而表1中数据显示, EM-ELM算法拥有更快的学习速度, IC-ELM算法则具有更好的学习和泛化能力. 表1和表2中的方差和最大最小值说明, IC-ELM比其他两个算法更加稳定.

表2 实验4.2结果比较

Table 2 The comparison result in experiment 4.2

算法	学习时间/s			学习精度	测试精度			隐节点数		
	均值	最小值	最大值	均值	均值	最小值	最大值	均值	最小值	最大值
IC-ELM	0.6014	0.4368	0.8892	1	0.9294	0.8849	0.9779	31.85	30	34
ELM	0.6708	0.4992	0.8736	1	0.9085	0.8753	0.9429	126.2	120	136

5 结论(Conclusions)

本文针对ELM网络结构设计问题, 提出了增量式IC-ELM算法, 并从理论上分析了该算法的收敛性及稳定性. IC-ELM算法利用Sigmoid函数及其导函数, 在单隐含层前向神经网络的基础上构造了一个双隐含层增长型前向神经网络. 该算法通过隐含层结点及衍生层的自动增长实现了前向神经网络结构自适应增长. 利用Sigmoid函数及其导函数值作为衍生层的输出, 丰富了衍生层结点激活函数的类型, 减少了隐含层随机结点的数量, 增强了网络的稳定性, 降低了ELM神经网络的结构复杂性. 实验结果证实IC-ELM算法最终获得的ELM神经网络结构紧凑, 泛化能力强.

参考文献(References):

- [1] HUANG G B, ZHU Q Y, SIEW C K. Extreme learning machine: theory and applications [J]. *Neurocomputing*, 2006, 70(1): 489 – 501.
- [2] HUANG G B, CHEN L. Convex incremental extreme learning machine [J]. *Neurocomputing*, 2007, 70(16): 3056 – 3062.
- [3] LAN Y, SOH Y C, HUANG G B. Two-stage extreme learning machine for regression [J]. *Neurocomputing*, 2010, 73(16): 3028 – 3038.
- [4] SUN Z L, NG K M, SOSZYNSKA-BUDNY J, et al. Application of the LP-ELM model on transportation system lifetime optimization [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2011, 12(4): 1484 – 1494.
- [5] 韩敏, 王新迎. 多元混沌时间序列的加权极端学习机预测 [J]. *控制理论与应用*, 2013, 30(11): 1467 – 1472.
(HAN Min, WANG Xinying. Multivariate chaotic time series prediction based on weighted extreme learning machine [J]. *Control Theory & Applications*, 2013, 30(11): 1467 – 1472.)
- [6] YEU C W T, LIM M H, HUANG G B, et al. A new machine learning paradigm for terrain reconstruction [J]. *IEEE Geoscience and Remote Sensing Letters*, 2006, 3(3): 382 – 386.
- [7] 肖冬, 王继春, 潘孝礼, 等. 基于改进PCA-ELM方法的穿孔机盘转速测量 [J]. *控制理论与应用*, 2010, 27(1): 19 – 24.

(XIAO Dong, WANG Jichun, PAN Xiaoli, et al. Modeling and control of guide-disk speed of rotary piercer [J]. *Control Theory & Applications*, 2010, 27(1): 19 – 24.)

- [8] TEOH E J, TAN K C, XIANG C. Estimating the number of hidden neurons in a feedforward network using the singular value decomposition [J]. *IEEE Transactions on Neural Networks*, 2006, 17(6): 1623 – 1629.
- [9] HUANG G B, CHEN L, SIEW C K. Universal approximation using incremental constructive feedforward networks with random hidden nodes [J]. *IEEE Transactions on Neural Networks*, 2006, 17(4): 879 – 892.
- [10] HUANG G B, CHEN L. Enhanced random search based incremental extreme learning machine [J]. *Neurocomputing*, 2008, 71(16): 3460 – 3468.
- [11] FENG G, HUANG G B, LIN Q, et al. Error minimized extreme learning machine with growth of hidden nodes and incremental learning [J]. *IEEE Transactions on Neural Networks*, 2009, 20(8): 1352 – 1357.
- [12] LI F J, LI Y. Cascade-correlation algorithm with trainable activation functions [J]. *Computer and Information Science*, 2011, 4(6): 28 – 37.
- [13] HAN H G, CHEN Q L, QIAO J F. Research on an online self-organizing radial basis function neural network [J]. *Neural Computing & Applications*, 2010, 19(5): 667 – 676.
- [14] 张昭昭, 乔俊飞, 杨刚. 一种功能分区的BP神经网络结构设计方法 [J]. *控制与决策*, 2011, 26(11): 1659 – 1664.
(ZHANG Zhaozhao, QIAO Junfei, YANG Gang. Structure model of function-dividing design for BP neural network [J]. *Control and Decision*, 2011, 26(11): 1659 – 1664.)

作者简介:

李凡军 (1977–), 男, 博士研究生, 主要研究方向为智能信息处理及神经网络结构优化设计, E-mail: ss_lifj@ujn.edu.cn;

乔俊飞 (1968–), 男, 教授, 博士生导师, 主要研究方向为智能控制与智能信息处理等, E-mail: junfei@bjut.edu.cn;

韩红桂 (1983–), 男, 博士, 主要研究方向为智能信息处理、智能控制理论及应用等, E-mail: rechardhan@bjut.edu.cn.