

## 改进式混合增量极限学习机算法

王超<sup>1</sup>, 王建辉<sup>1</sup>, 顾树生<sup>1</sup>, 张宇献<sup>2</sup>, 武玮<sup>1</sup>

(1. 东北大学 信息科学与工程学院, 沈阳 110004; 2. 沈阳工业大学 电气工程学院, 沈阳 110870)

**摘要:** 针对增量型极限学习机(I-ELM)中存在大量降低学习效率及准确性的冗余节点的问题, 提出一种基于Delta检验(DT)和混沌优化算法(COA)的改进式增量型核极限学习算法. 利用COA的全局搜索能力对I-ELM中的隐含层节点参数进行寻优, 结合DT算法检验模型输出误差, 确定有效的隐含层节点数量, 从而降低网络复杂程度, 提高算法的学习效率; 加入核函数可增强网络的在线预测能力. 仿真结果表明, 所提出的DCI-ELMK算法具有较好的预测精度和泛化能力, 网络结构更为紧凑.

**关键词:** 极限学习机; 增量学习算法; Delta检验; 混沌优化算法; 增量型极限学习机

**中图分类号:** TP273

**文献标志码:** A

## Improved hybrid incremental extreme learning machine algorithm

WANG Chao<sup>1</sup>, WANG Jian-hui<sup>1</sup>, GU Shu-sheng<sup>1</sup>, ZHANG Yu-xian<sup>2</sup>, WU Wei<sup>1</sup>

(1. College of Information Science and Engineering, Northeastern University, Shenyang 110004, China; 2. School of Electrical Engineering, Shenyang University of Technology, Shenyang 110870, China. Correspondent: WANG Chao, E-mail: supper\_king1018@163.com)

**Abstract:** Focusing on the problem that redundant nodes in incremental extreme learning machine(I-ELM) can lead to ineffective iteration increases and reduce the learning efficiency, an improved I-ELM algorithm based on Delta test(DT) and chaotic optimization algorithm(COA) is proposed. The COA is used to optimize the hidden layer neuron parameters of I-ELM by global searching ability, and is combined with the DT algorithm which tests the output error of model to determine the effective hidden layer neurons number. The learning efficiency of the algorithm is improved by reducing the network complexity, and the DCI-ELM with kernel can enhance the online prediction ability of network. The simulations show that the DCI-ELMK algorithm with more compact network structure has higher prediction accuracy and better ability of generalization compared with other algorithms.

**Keywords:** extreme learning machine(ELM); incremental learning algorithm; Delta test; chaotic optimization algorithm; incremental ELM

## 0 引言

传统的单隐含层前向神经网络(SLFNs)普遍存在训练效率低、学习能力不强等问题. Huang等<sup>[1]</sup>提出了极端学习机(ELM)算法, 该算法无需调整任何参数, 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, 具有较高的学习速率和较强的泛化能力. 极限学习机作为系统辨识与模式识别的有力工具, 具有广泛的应用前景<sup>[2-3]</sup>. 由于固定型极限学习机为了获得较强的学习能力, 通常采用维数较高的网络结构, 由此便产生了最优隐含层节点个数及有效控制模型规模等问题. 对此, Huang等<sup>[4-5]</sup>提出了增量型极限学

习机(I-ELM), 采用增量型算法自适应地选择网络的隐含层节点数量, 在每次迭代过程中产生 $k(k \geq 1)$ 个隐含层节点, 并更新输出权值. 然而, 增量型极限学习机同样存在一些需要解决的问题. 如: I-ELM中存在输出权值较小的隐含层节点, 导致增量型极限学习机中许多隐含层节点对最终网络输出起到的作用很小, 大量冗余的节点只能增加网络结构的复杂性, 降低学习效率; 同时, 由于收敛速率较低, I-ELM通常需要更多的隐含层节点数, 有时甚至超过学习样本的数量.

近年来, 极限学习机的研究主要集中在增量型极限学习机的优化与改进. Huang等<sup>[6]</sup>对I-ELM算法进

收稿日期: 2014-08-25; 修回日期: 2014-11-17.

基金项目: 国家自然科学基金项目(61102124); 辽宁省科学技术计划项目(JH2/I01).

作者简介: 王超(1985-), 男, 博士生, 从事智能控制和机器学习的研究; 王建辉(1957-), 女, 教授, 博士生导师, 从事复杂控制系统的建模与控制、网络环境下先进控制技术及其应用等研究.

行改进,提出了EI-ELM(enhanced incremental ELM)算法.通过优选方法选取有效节点用于构建网络,在一定程度上降低了ELM的网络结构复杂程度.但是,当神经网络规模较大时,隐含层节点的递增方式使算法的循环次数大幅度增加,影响神经网络的泛化能力.Huang等<sup>[7]</sup>提出了基于Barron凸优化算法的极限学习机(CI-ELM),通过重新计算隐含层节点增加后的现有节点输出权值,获得较快的收敛速率.Wang等<sup>[8]</sup>提出了改进凸优化增量型极限学习机(ECI-ELM),通过改进的随机搜索方法确定隐含层参数,能够有效地逼近目标函数.但CI-ELM和ECI-ELM算法均容易产生冗余隐含层节点,增加了ELM的迭代次数,降低了神经网络的学习能力.

本文提出一种基于Delta检验(DT)算法和混沌优化算法(COA)的核极限学习机算法(DT-COA-IELMK).先利用COA的全局搜索能力对I-ELM的隐含层节点参数( $a, b$ )进行参数辨识,再根据DT计算输出权值,可有效地减少I-ELM网络中的隐含层节点数量,缩短I-ELM算法的学习时间,降低网络结构的复杂程度,提高算法的学习能力.同时,采用未知核映射代替已知隐含层映射,可增强算法的在线预测能力.通过UCI数据集检验,并对比上述算法,所得结果表明DT-COA-IELMK算法具有更快的收敛速率和更好的泛化能力.

## 1 核增量型极限学习机(I-ELMK)

### 1.1 极限学习机(ELM)

假设前馈神经网络模型有 $K$ 个隐含层节点,激活函数为 $g(\cdot)$ .对于 $N$ 个不同的学习样本( $X, Y$ ), $X = \{x_i[n]\} \in \mathbf{R}^{N \times L}$ , $Y = \{y_i[n]\} \in \mathbf{R}^{N \times L}$ ,ELM的数学表达式为

$$\sum_{i=1}^L w_i g(W_{in(i)} \cdot x_j + b_i) = o_j, \quad j = 1, 2, \dots, N. \quad (1)$$

其中: $W_{in(i)} \in \mathbf{R}^p$ 是连接输入节点与 $i$ -th隐层节点的权值向量, $W_{in(i)} \cdot x_j$ 表示 $W_{in(i)}$ 与 $x_j$ 的内积, $b_i \in \mathbf{R}$ 是 $j$ -th隐层偏置值, $g(\cdot)$ 是激活函数, $w_i \in \mathbf{R}^p$ 是连接 $i$ -th与输出节点的权值向量, $o_j \in \mathbf{R}$ 是ELM的输出, $L$ 是隐层节点的数量. $g(\cdot)$ 可以是任意无穷可微函数,一般取Sigmoid函数<sup>[9-10]</sup>.式(1)可以改写成矩阵向量形式

$$Hw = o. \quad (2)$$

其中

$$H = \begin{bmatrix} g(W_{in(1)} \cdot x_1 + b_1) & \cdots & g(W_{in(L)} \cdot x_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(W_{in(1)} \cdot x_N + b_1) & \cdots & g(W_{in(L)} \cdot x_N + b_L) \end{bmatrix}_{N \times L},$$

$$o = [o_1, o_2, \dots, o_N]^T, \quad w = [w_1, w_2, \dots, w_L]^T. \quad (3)$$

矩阵 $H$ 为ELM的隐含层输出矩阵,其 $i$ -th行 $h_i$ 是相对于输入 $x_i$ 的隐含层输出向量,输入权值 $W_{in(i)}$ 和偏置值 $b_i$ 随机初始化且在训练过程中保持不变,输出权值 $w$ 是唯一需要训练确定的参数.如果含有 $L$ 个隐层节点的ELM模型能够学习数量为 $N$ 的训练样本,并且无残差存在,则意味着存在 $w_i$ 使得

$$\sum_{i=1}^L w_i g(W_{in(i)} \cdot x_j + b_i) = t_j, \quad j = 1, 2, \dots, N, \quad (4)$$

其中 $t_j$ 为目标值.式(4)可以改写成矩阵向量的形式

$$Hw = t, \quad (5)$$

其中 $t = [t_1, t_2, \dots, t_N]^T$ 为目标向量.输出权值 $w$ 是唯一需要训练确定的参数,可采用如下算法确定:

$$w = H^\dagger t, \quad (6)$$

其中 $H^\dagger$ 是 $H$ 的广义逆运算.

### 1.2 核增量型极限学习机

基于核方法的智能算法在处理非线性不可分问题时具有很强的分类能力和良好的拟合能力,Guo等<sup>[11]</sup>将核方法引入ELM,利用固定训练样本集建立离线训练学习模型.ELM的核矩阵表示如下:

$$K_{\text{ELM}} = HH^T, \quad (7)$$

$$K_{\text{ELM}} = h(x_i) \cdot h(x_j) = K(x_i, x_j). \quad (8)$$

因此,ELM的输出函数可以表示为

$$f(x) = h(x)H^T \left( HH^T + \frac{1}{C} \right)^{-1} t = \begin{bmatrix} K(x, x_1) \\ K(x, x_2) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T \left( K_{\text{ELM}} + \frac{1}{C} \right)^{-1} t. \quad (9)$$

其中: $H(x)$ 为数据在ELM特征空间中的映射, $h(x)$ 为ELM中隐含层的映射, $C$ 为正则化参数, $t$ 为目标向量.

I-ELMK算法的实现步骤如下.

Step 1: 设 $A = [K_{\text{ELM}} + 1/C]$ ,对于时刻 $t$ ,有

$$A_t = \begin{bmatrix} \frac{1}{C} + K(x_1, x_1) & \cdots & K(x_1, x_N) \\ \vdots & \ddots & \vdots \\ K(x_N, x_1) & \cdots & \frac{1}{C} + K(x_N, x_N) \end{bmatrix}. \quad (10)$$

Step 2: 对于 $t+1$ 时刻,有

$$A_{t+1} = \begin{bmatrix} \frac{1}{C} + K(x_1, x_1) & \cdots & K(x_1, x_{N+k}) \\ \vdots & \ddots & \vdots \\ K(x_{N+k}, x_1) & \cdots & \frac{1}{C} + K(x_{N+k}, x_{N+k}) \end{bmatrix}. \quad (11)$$

简化矩阵  $A_{t+1}$ , 设

$$U_t = \begin{bmatrix} K(x_1, x_{N+1}) & \cdots & K(x_1, x_{N+k}) \\ \vdots & \ddots & \vdots \\ K(x_N, x_{N+1}) & \cdots & K(x_N, x_{N+k}) \end{bmatrix}, \quad (12)$$

$$D_t = \begin{bmatrix} \frac{1}{C} + K(x_{N+1}, x_{N+1}) & \cdots & K(x_{N+1}, x_{N+k}) \\ \vdots & \ddots & \vdots \\ K(x_{N+k}, x_{N+1}) & \cdots & \frac{1}{C} + K(x_{N+k}, x_{N+k}) \end{bmatrix}, \quad (13)$$

则得  $A_{t+1} = \begin{bmatrix} A_t & U_t \\ U_t^T & D_t \end{bmatrix}$ . 利用新数据对参数  $A_{t+1}$  求逆运算, 得

$$A_{t+1}^{-1} = \begin{bmatrix} A_t^{-1} + A_t^{-1}U_t C_t^{-1}U - t^T A_t^{-1} & -A_t^{-1}U_t C_t^{-1} \\ -C_t^{-1}U_t^T A_t^{-1} & C_t^{-1} \end{bmatrix}, \quad (14)$$

其中  $C_t = D_t - U_t^T A_t^{-1}U_t$ .

**Step 3:** 在线更新测试数据. 给定数据量为  $M$ , 对于测试数据  $X_{\text{test}} = [x_{\text{test}1}, x_{\text{test}2}, \dots, x_{\text{test}M}]$ , 利用  $A_{t+1}^{-1}$  估计输出值  $\hat{Y}_{\text{test}}$ , 即

$$\hat{Y}_{\text{test}} = \begin{bmatrix} K(x_{\text{test}1}, x_1) & \cdots & K(x_{\text{test}1}, x_M) \\ \vdots & \ddots & \vdots \\ K(x_{\text{test}N}, x_1) & \cdots & K(x_{\text{test}N}, x_M) \end{bmatrix} \begin{bmatrix} A_M^{-1}y_1 \\ \vdots \\ A_M^{-1}y_M \end{bmatrix}, \quad (15)$$

其中输出值  $\hat{Y}_{\text{test}} = [y_1, y_2, \dots, y_M]^T$ .

## 2 混沌优化算法 (COA)

基于混沌序列的优化算法, 其基本思想是用类似载波的方式将混沌状态引入优化变量中, 把混沌运动的遍历范围映射到优化变量的取值范围 (定义域), 然后利用混沌变量直接进行搜索<sup>[12]</sup>, 有

$$\begin{aligned} X_{n+1} &= f(c, X_n) = \sin(c/X_n), \\ -1 &\leq X_n \leq 1, X_0 \neq 0, n = 0, 1, \dots, k; \\ \min f(X_i), & i = 1, 2, \dots, n; \\ \text{s.t. } & a_i \leq X_i \leq b_i. \end{aligned} \quad (16)$$

其中:  $n$  为变量个数,  $[a_i, b_i]$  为各个变量  $X_i$  的定义域,  $f(X_i)$  为优化目标函数. 按照下式进行二次载波:

$$X_i^k = X_i^{\text{best}} + \alpha(k) \cdot y. \quad (17)$$

利用载波后的混沌变量进行迭代, 计算相应的性能指标  $f(X_i^k)$ , 当满足终止条件时终止搜索, 输出最优解  $X^{\text{best}}$  和最优目标值  $f^*$ <sup>[13]</sup>.

## 3 Delta 检验 (DT) 算法

DT 算法是基于  $K$  近邻原则的无参数技术, 是一种对于未知模型数据集的噪声方差估计快速可扩展算法, 能够在得到最小均方误差的同时抑制过拟合现象. Jones<sup>[14]</sup>介绍了 Delta 检验算法的原理及其应用, Stefánsson 等<sup>[15]</sup>采用 DT 方法评价 NNE (nonparametric noise estimator) 并取得了较为理想的效果.

假定标准 DT 分析中, 输入向量与输出向量的形式为  $(x_i, y_i | 1 \leq i \leq N)$ . 其中:  $N$  为样本数量, 输入向量  $x_i \in R^d$  被限定在封闭的集合  $C \subset R^d$  中. 输入向量与输出向量之间的关系为  $y_i = f(x_i) + r_i$ . 其中:  $f$  为未知函数,  $r$  为噪声. DT 是一种评价输入与输出向量间非线性相关性的有效方法, 被选择的输入变量代表着输入变量与输出变量最确定的变量关系. DT 利用未知函数  $f$  的连续性假设对噪声  $r$  的方差进行估计, 如果  $x$  和  $x'$  两点在输入空间接近, 则由未知函数  $f$  的连续性可知,  $f(x)$  和  $f(x')$  在输出空间同样接近. 如果对应的输出值  $y$  和  $y'$  在输出空间中无法接近, 则表明噪声  $r$  对  $f(x)$  和  $f(x)'$  产生了影响. 利用  $x_{f(i)}$  表示数据集  $x_1, x_2, \dots, x_N$  中  $x_i$  的最近邻点, 因此, DT 中的  $\delta$  可表示为<sup>[16]</sup>

$$\delta = \frac{1}{2N} \sum_{i=1}^N |y_{f(i)} - y_i|^2, \quad (18)$$

其中  $y_{f(i)}$  为  $x_{f(i)}$  的输出. Jones<sup>[14]</sup>对 Delta 检验的收敛性进行了验证.

## 4 混合优化极限学习机 (DCI-ELMK)

构造基于 Delta 检验的混沌优化算法的极限学习机 (DCI-ELMK), 设 ELM 中隐含层节点参数  $Y_L^*$  为需要优化的变量, 目标函数为残差  $E$ . DCI-ELMK 的实现步骤如下.

**Step 1** 计算最佳隐含层节点数  $L_{\text{best}}$ .

**Step 1.1:** 给定  $N$  个训练样本  $(x_i, t_i)$ ,  $x_i = [x_{i1}, \dots, x_{in}]^T$ ,  $t_i \in R (i = 1, 2, \dots, N)$ , 给定期望的网络输出误差  $\eta$ , 输出预测误差  $\varphi$ ;

**Step 1.2:** 设神经网络隐含层节点数  $L = 0$ , 网络误差  $e_0^* = T$ , 迭代次数  $k = 0$ ;

**Step 1.3:** 设隐含层节点  $L = L + \lambda$ ;

**Step 1.4:** 根据 Delta 检验计算预测误差

$$E_R = \frac{1}{2N} \sum_{i=1}^N |f_i(x) - f|;$$

**Step 1.5:** 利用 DT-COA 优化获得最优隐含层节点参数值  $Y_L^*$ ;

**Step 1.6:** 计算输出权值

$$\beta_L^* = \frac{E_L \cdot H_L^T(Y_L^*)}{H_L(Y_L^*) \cdot H_L^T(Y_L^*)};$$

Step 1.7: 计算输出误差

$$E_L = E_{L-1} - \beta_L^* \|H_L(Y_L^*), x\|;$$

Step 1.8: 若  $E_L < \eta$  且  $E_R < \varphi$ , 则终止训练, 否则转 Step 1.3.

Step 2 在线计算预测输出值  $\hat{Y}_{test}$ .

Step 2.1: 设  $A = [K_{ELM} + 1/C]$ ,  $t$  时刻为  $A_t$ ,  $t + 1$  时刻为  $A_{t+1}$ ;

Step 2.2: 对参数  $A_{t+1}$  求逆运算得  $A_{t+1}^{-1}$ ;

Step 2.3: 在线更新测试数据, 用  $A_{t+1}^{-1}$  估计输出值  $\hat{Y}_{test}$ , 有

$$\hat{Y}_{test} = \begin{bmatrix} K(x_{test1}, x_1) & \cdots & K(x_{test1}, x_M) \\ \vdots & \ddots & \vdots \\ K(x_{testN}, x_1) & \cdots & K(x_{testN}, x_M) \end{bmatrix} \begin{bmatrix} A_M^{-1} y_1 \\ \vdots \\ A_M^{-1} y_M \end{bmatrix};$$

Step 2.4: 计算输出值  $\hat{Y}_{test} = [y_1, y_2, \dots, y_M]^T$ .

当 Step 1.3 中  $\lambda = 1$  时, 表示 DCI-ELMK 中隐含层节点数量为单一增加; 当  $\lambda$  为大于 1 的整数时, 表示算法中隐含层节点数量为组群增加; 当  $\lambda$  等于神经网络最大隐含层节点设定数量时, DCI-ELMK 可以一次性计算所有隐含层节点参数和输出权值.

### 5 仿真及结果分析

为验证本文所提出 DCI-ELMK 算法的有效性, 利用 UCI 真实数据集 (<http://archive.ics.uci.edu/ml/>) 对 CI-ELM、EI-ELM、ECI-ELM 和 DCI-ELMK 进行回归测试. 10 组典型真实数据集分别是: Abalone、Energy efficiency、Combined cycle power plant (CCPP)、Boston housing、Concrete compressive strength (CCS) 等. 本文所有仿真均在同一系统环境下运行所得, 系统运行环境为 PC (Duo CPU 2.99 Hz, 6.00 GB RAM), 操作系统为 Windows 7, 仿真软件为 Matlab R2010a. 实验数据集分训练数据集和测试数据集, 具体描述如表 1 所示.

测试实验中, 将输入数据在  $[-1, 1]$  区间进行归一化, 相应的输出数据归一化至  $[0, 1]$  区间. ELM 网络

表 1 数据集描述

数据集	训练样本数	测试样本数	特征数	问题类型
Friedman	18 080	12 600	11	回归
CCPP	7 000	4 300	4	回归
Servo	100	67	4	回归
California housing	4 430	2 020	8	回归
CCS	1 030	900	9	回归
Abalone	4 100	3 000	8	分类
Energy efficiency	768	500	8	分类
Boston housing	466	300	13	分类
Bank	3 000	1 300	8	分类
Delta ailerons	2 800	1 700	5	分类

的激活函数均为 Sigmoid 函数, 每组数据进行 30 次实验, 将 30 次实验结果的平均值作为最终结果. 图 1 所示为利用 Combined cycle power plant 数据集进行预测实验, 在非线形动态系统辨识的情况下, 相比于其他 3 种算法, DCI-ELMK 算法可以获得更快的学习速率和更少的隐层节点数. 通过实验可知, 其他 4 组 UCI 数据集的测试效果也显示了同样的泛化能力.

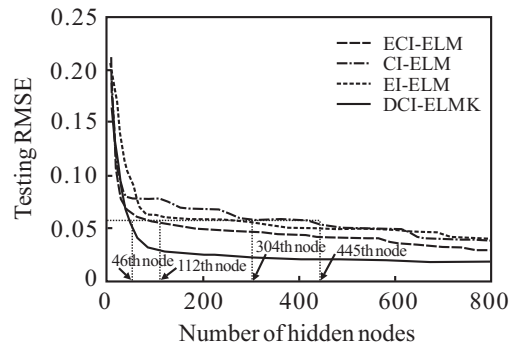


图 1 CCPP 数据集算法误差曲线对比

表 2 给出了不同 I-ELM 算法基于回归问题的标准化比较结果. 以 Concrete compressive strength 数据集为例, 设定最大隐含层节点数为 100, 误差终止条件为  $RMSE = 0.045$ . DCI-ELMK 算法的训练误差为 0.043 1, 测试误差为 0.038 2, 回归正确率明显高于其他 3 种算法.

表 2 回归问题的训练和测试误差对比

数据集	#Nodes	CI-ELM		EI-ELM		ECI-ELM		DCI-ELMK	
		训练误差	测试误差	训练误差	测试误差	训练误差	测试误差	训练误差	测试误差
Friedman (0.1)	50	0.160 1	0.167 3	0.146 5	0.158 9	0.142 4	0.135 3	<b>0.095 6</b>	<b>0.109 1</b>
Combined cycle power plant (0.052)	100	0.057 6	0.061 7	0.055 1	0.052 9	0.053 5	0.060 4	<b>0.051 3</b>	<b>0.050 8</b>
Servo (0.155)	50	0.167 7	0.178 5	0.161 2	0.160 1	0.154 7	0.154 2	<b>0.153 1</b>	<b>0.151 4</b>
California housing (0.15)	100	0.162 3	0.159 9	0.148 7	0.153 3	0.152 1	0.150 9	<b>0.146 7</b>	<b>0.140 3</b>
Concrete compressive strength (0.045)	100	0.067 1	0.072 7	0.052 9	0.051 3	0.044 7	0.042 4	<b>0.043 1</b>	<b>0.038 2</b>

表 3 给出了针对回归问题的算法性能比较. 可以看出, 基于 DCI-ELM 算法的实验结果对比其他 3 种现有算法, 其神经网络节点数明显减少, 以 Combined cycle power plant 数据集为例, 当算法终止条件设定为  $RMSE = 0.052$  时, DCI-ELMK 算法所需节点数平均值为 43.35, 明显低于其他 3 种现有算法: 111.92 (ECI-ELM), 305.18 (EI-ELM), 390.76 (CI-ELM), 可见 DCI-ELMK 表现出更紧凑的网络结构. 由于采用 DT-COA 对网络结构进行简化处理, 在保证预测精度的前提下, DCI-ELMK 算法并没有过多消耗训练时间, 在

训练时间上仅有部分情况略长于 CI-ELM 算法, 并且全部实验数据集训练的时间均低于 EI-ELM 和 ECI-ELM.

表 4 和表 5 为基于分类问题的算法分类准确性、网络复杂度和训练时间的实验对比结果. 通过 UCI 真实数据集验证, 本文所提出的 DCI-ELMK 算法有效地减少了冗余隐含层节点数, 降低了网络的结构复杂度和训练时间, 提高了算法的学习效率, 各项指标均优于其他 3 种改进 I-ELM 算法.

表 3 回归问题的网络复杂度和训练时间对比

数据集	CI-ELM		EI-ELM		ECI-ELM		DCI-ELMK	
	Nodes	Time/s	Nodes	Time/s	Nodes	Time/s	Nodes	Time/s
Friedman (0.1)	140.26	<b>2.218 3</b>	87.82	4.216 2	62.37	4.002 1	<b>19.75</b>	3.113 4
Combined cycle power plant (0.052)	390.76	<b>2.010 4</b>	305.18	2.562 2	111.92	3.017 8	<b>45.35</b>	2.074 3
Servo (0.155)	70.01	0.038 1	58.56	0.423 3	14.22	0.455 1	<b>11.74</b>	<b>0.037 3</b>
California housing (0.15)	220.73	0.878 2	19.24	1.051 5	112.98	1.420 2	<b>17.42</b>	<b>0.721 6</b>
Concrete compressive strength (0.045)	70.22	0.532 9	65.02	0.743 9	51.23	0.824 1	<b>49.33</b>	<b>0.412 5</b>

表 4 分类问题的训练和测试误差对比

数据集	#Nodes	CI-ELM		EI-ELM		ECI-ELM		DCI-ELMK		%
		均值	标准差	均值	标准差	均值	标准差	均值	标准差	
Abalone (0.075)	200	83.18	0.002 2	84.02	0.001 8	93.76	<b>0.001 4</b>	<b>94.23</b>	0.001 5	
Energy efficiency (0.045)	100	90.18	0.001 4	92.42	0.001 6	94.12	0.001 1	<b>96.52</b>	<b>0.000 8</b>	
Boston housing (0.1)	30	68.02	0.011 3	71.78	0.010 1	84.82	0.007 2	<b>93.01</b>	<b>0.004 1</b>	
Bank (0.065)	150	90.21	0.021 7	92.03	0.012 2	91.82	0.012 4	<b>95.89</b>	<b>0.007 6</b>	
Delta ailerons (0.04)	200	89.23	0.037 2	86.89	0.022 1	92.23	0.011 7	<b>92.71</b>	<b>0.010 8</b>	

表 5 分类问题的网络复杂度和训练时间对比

数据集	CI-ELM		EI-ELM		ECI-ELM		DCI-ELMK	
	Nodes	Time/s	Nodes	Time/s	Nodes	Time/s	Nodes	Time/s
Abalone (0.075)	143.05	<b>0.285 2</b>	113.94	0.921 6	23.64	1.121 5	<b>19.15</b>	0.622 1
Energy efficiency (0.045)	72.21	<b>0.150 2</b>	80.43	0.302 3	67.03	0.282 2	<b>50.21</b>	0.202 3
Boston housing (0.1)	38.41	0.097 3	21.11	0.073 4	39.17	0.116 8	<b>22.06</b>	<b>0.094 2</b>
Bank (0.065)	199.43	0.674 5	170.27	0.789 1	<b>10.21</b>	0.892 2	10.97	<b>0.627 6</b>
Delta ailerons (0.04)	332.69	1.123 9	290.21	2.174 2	39.66	1.365 3	<b>36.03</b>	<b>0.691 2</b>

图 2 和图 3 分别表示利用 Combined cycle power plant 和 Concrete compressive strength 数据集验证 DCI

-ELMK 算法的预测效果. 考虑到实际工况下数据通常以一组或多组的在线时序型输入, 将现有数据集划

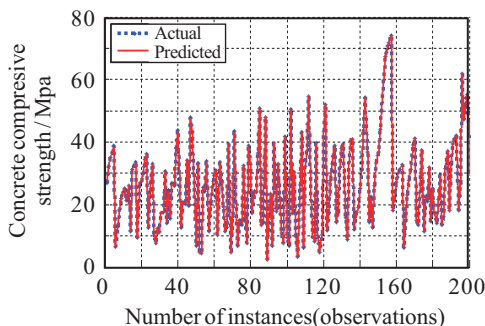


图 2 CCS 数据集增量数据拟合曲线

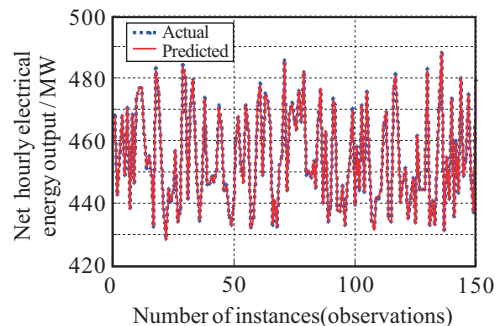


图 3 CCGP 数据集增量数据拟合曲线

分为固定数据和增量数据, 设定增量规模为5组. 通过仿真验证, 本文提出的DCI-ELMK算法能够准确预测在线时序数据, 具有较强的泛化能力.

## 6 结 论

本文针对增量型ELM中冗余节点可导致无效迭代次数增加、降低学习效率的问题, 提出了基于DT和COA的核极限学习机DCI-ELMK算法. 利用DT-COA优化隐含层节点参数, 减少了隐含层无效节点的数量, 增强了网络的稳定性, 降低了ELM网络的结构复杂程度, 提高了算法的学习效率. 对于在线时序型数据, 利用核函数减弱了算法对新数据的敏感性, 使算法具有更好的在线预测能力. 选取10组UCI典型数据集作为测试数据进行实验对比, 结果表明, 本文提出的DCI-ELMK算法网络结构更为紧凑, 具有较好的预测精度和泛化能力.

## 参考文献(References)

- [1] Huang G B, Zhu Q, Siew C K. Extreme learning machine: Theory and applications[J]. *Neurocomputing*, 2006, 70 (1/2/3): 489-501.
- [2] Zong W, Huang G B. Face recognition based on extreme learning machine[J]. *Neurocomputing*, 2011, 74(16): 2541-2551.
- [3] 颜七笙, 王士同, 张延飞, 等. 基于经验模式分解和极限学习机的铀资源价格预测方法[J]. *控制与决策*, 2014, 29(7): 1187-1192.  
(Yan Q S, Wang S T, Zhang Y F, et al. Uranium resource price prediction based on empirical mode decomposition and extreme learning machine[J]. *Control and Decision*, 2014, 29(7): 1187-1192.)
- [4] Huang G B, Li M B, Chen L, et al. Incremental extreme learning machine with fully complex hidden nodes[J]. *Neurocomputing*, 2008, 71: 576-583.
- [5] 田慧欣, 王安娜. 基于增量学习思想的改进AdaBoost建模方法[J]. *控制与决策*, 2012, 27(9): 1433-1436.  
(Tian H X, Wang A N. Improved adaboost modeling method based on incremental learning[J]. *Control and Decision*, 2012, 27(9): 1433-1436.)
- [6] Huang G B, Chen L. Enhanced random search based incremental extreme learning machine[J]. *Neurocomputing*, 2008, 71: 3460-3468.
- [7] Huang G B, Chen L. Convex incremental extreme learning machine[J]. *Neurocomputing*, 2007, 70: 3056-3062.
- [8] Wang W, Zhang R. Improved convex incremental extreme learning machine based on enhanced random search[J]. *Unifying Electrical Engineering and Electronics Engineering*, 2014, 238: 2033-2040.
- [9] Zhang R, Lan Y, Huang G B, et al. Universal approximation of extreme learning machine with adaptive growth of hidden nodes[J]. *IEEE Trans on Neural Networks and Learning Systems*, 2012, 23: 365-371.
- [10] 李凡军, 乔俊飞, 韩红桂. 网络结构增长的极端学习机算法[J]. *控制理论与应用*, 2014, 31(5): 638-643.  
(Li F J, Qiao J F, Han H G. Incremental constructive extreme learning machine[J]. *Control Theory & Applications*, 2014, 31(5): 638-643.)
- [11] Guo L, Hao J H, Liu M. An incremental extreme learning machine for online sequential learning problems[J]. *Neurocomputing*, 2014, 128: 50-58.
- [12] Yang Y M, Wang Y N, Yuan X F, et al. Hybrid chaos optimization algorithm with artificial emotion[J]. *Applied Mathematics and Computation*, 2012, 218(11): 6585-6591.
- [13] Yang Y M, Wang Y N, Yuan X F. Parallel chaos search based incremental extreme learning machine[J]. *Neural Processing Letters*, 2013, 37: 277-301.
- [14] Jones A J. New tools in non-linear modeling and prediction[J]. *Computational Management Science*, 2004, 1(2): 109-149.
- [15] Stefánsson A, Koncar N, Jones A J. A note on the gamma test[J]. *Neural Comput*, 1997, 5(3): 131-133.
- [16] Yu Q, Heeswijk M V, Miche Y, et al. Ensemble delta test-extreme learning machine(DT-ELM) for regression[J]. *Neurocomputing*, 2014, 129: 153-158.

(责任编辑: 李君玲)