第 33 卷第 3 期
2016 年 3 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 33 No. 3
Mar. 2016

DOI: 10.7641/CTA.2016.50640

# 基于多层学习克隆选择的改进式增量型超限学习机算法

王　超[1†], 王建辉[1], 顾树生[1], 王　枭[1], 张宇献[2]

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

**摘要**: 针对增量型超限学习机(incremental extreme learning machine, I–ELM)中大量冗余节点可导致算法学习效率降低, 网络结构复杂化等问题, 提出基于多层学习(multi-learning)优化克隆选择算法(clone selection algorithm, CSA)的改进式I–ELM. 利用Baldwinian learning操作改变抗体信息的搜索范围, 结合Lamarckian learning操作提高CSA的搜索能力. 改进后的算法能够有效控制I–ELM的隐含层节点数, 使网络结构更加紧凑, 提高算法精度. 仿真结果表明, 所提出的基于多层学习克隆选择的增量型核超限学习机(multi-learning clonal selection I–ELMK, MLCSI–ELMK)算法能够有效简化网络结构, 并保持较好的泛化能力, 较强的学习能力和在线预测能力.

**关键词**: 克隆选择算法; 鲍德温学习; 拉马克学习; 神经网络; 增量型超限学习机; 软计算

**中图分类号**: TP273　　**文献标识码**: A

# Improved incremental extreme learning machine based on multi-learning clonal selection algorithm

WANG Chao[1†], WANG Jian-hui[1], GU Shu-sheng[1], WANG Xiao[1], ZHANG Yu-xian[2]

(1. College of Automation Science and Technology, Northeastern University, Shenyang Liaoning 110004, China;
2. School of Electrical Engineering, Shenyang University of Technology, Shenyang Liaoning 110870, China)

**Abstract:** The great number of redundant nodes in an incremental extreme learning machine (I–ELM) may lower the learning efficiency of the algorithm, and complicate the network structure. To deal with this problem, we propose the improved I–ELM with kernel (I–ELMK) on the basis of multi-learning clonal selection algorithm (MLCSA). The MLCSA uses Baldwinian learning and Lamarckian learning, to exploit the search space by employing the information of antibodies, and reinforce the exploitation capacity of individual information. The proposed algorithm can limit the number of hidden layer neurons effectively to obtain more compact network architecture. The simulations show that MLCSI–ELMK has higher prediction accuracies online and off-line, while providing a better capacity of generalization compared with other algorithms.

**Key words:** clonal selection algorithm; Baldwinian learning; Lamarckian learning; neural networks; incremental extreme learning machine; soft computing

## 1　引言(Introduction)

超限学习机(extreme learning machine, ELM)算法, 是新近发展起来的一种高效的神经网络学习方法[1]. 该算法克服了传统神经网络的缺陷, 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, ELM作为前向神经网络研究领域的重要课题, 在模式识别[2–3]、故障检测与诊断[4–5]、图像识别与处理[6–8]和软测量计算[9–10]等方面得到了广泛的应用. 针对ELM网络中隐含层节点数量对其模型预测效果具有较大影响这一问题, Huang等提出了增量型超限学习机(incremental ELM, I–ELM)[11], 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$k(k \geqslant 1)$个隐含层节点, 并更新输出权值. 然而, I–ELM 算法容易产生冗余的隐含层结点, 不但增大了神经网络的结构复杂度, 而且降低学习效率, 大量增加的隐含层节点数, 有时甚至会超过学习样本的数量.

近年来, 很多学者和研究人员针对如何控制I–ELM隐含层节点数的问题做出了一系列的优化与改进, 一些经典算法为I–ELM的发展做出巨大的贡献. Huang等人对I–ELM算法进行改进, 提出增强增量超限学习机 (enhanced I–ELM, EI–ELM)[12]算法和基于Barron凸优化算法(convex optimization learning)的超

收稿日期: 2015−07−24; 录用日期: 2015−11−05.

†通信作者. E-mail: supper_king1018@163.com; Tel.: +86 13804072128.

本文责任编委: 苏剑波.

国家自然科学基金项目(61102124), 辽宁省科学技术计划项目(JH2/101)资助.

Supported by National Natural Science Foundation of China (61102124) and Liaoning Key Industry Programme(JH2/101).

限学习机(convex I–ELM, CI–ELM)[13], 分别降低了网络结构复杂程度和提高了收敛速度. Yang等人[14]通过parallel chaos search的方法搜索最优的输入权值和阈值, 从而有效地约束I–ELM中隐含层节点的数量, 但是算法的准确性取决于搜索步数$N$, 因此, 过小的搜索步数会导致输出精度下降, 无限制地提高$N$值则会大幅度增加混纯优化超限学习机(parallel chaos ELM, PC–ELM)的训练时间. 近一段时间, 增量型超限学习机的算法又有了一些新的突破. Shang等人[15]提出的置信度加权超限学习机(confidence–weighted ELM, CW–ELM)算法, 以留一法增量超限学习机(leave-one-out I–ELM, LOO–IELM)[16]为基础并且对其进行Tikhonov正则化, 采用内点法(interior point algorithm)增强对复杂噪声的适应能力. Yu等[17]提出的ErrCor算法和Ding等[18]提出的改进型增量正则化超限学习机(improved incremental regularized ELM, II–RELM)算法分别在限制隐含层节点数量方面和预测控制中展现了较强的能力.

本文所提出的基于多层免疫克隆算法优化的超限学习机算法(multi-learning clonal selection I–ELM with kernel, MLCSI–ELMK), 总结了先进的多层学习方法[19–20], 利用Multi-learning思想优化人工免疫算法, 加强克隆选择的目的性和快速性, 采用基于Baldwin effect的Baldwinian learning指导基因变化, 进而通过种群中更新的抗体信息改变搜索空间的范围, 结合Lamarckian learning构成multi-learning策略, 增强克隆选择算法(clonal selection algorithm, CSA)的搜索能力, 对决定I–ELM学习能力和网络规模的隐含层节点参数进行整定, 定义最小输出误差值计算为亲和度的判断标准, 保存亲和度最高的抗体作为节点参数信息进行迭代, 保证有序地累积当前最优节点参数, 在控制输出误差精度的同时大幅度降低隐含层节点数量, 降低了网络复杂程度, 提高算法的学习效率. 同时, 采用核函数映射代替隐含层映射, 增强了算法的在线预测能力. 通过UCI真实数据集和实际工业生产数据集检验, 本文提出的MLCSI–ELMK算法显示出更紧凑的网络结构和泛化能力.

## 2　基于多层学习的克隆选择优化算法(Multi-learning clonal selection algorithm)

### 2.1　克隆选择算法(Clonal selection algorithm)

Burnet克隆选择学说是人工免疫系统(artificial immune system, AIS)中最重要的一个理论, 引入生物免疫系统机制处理, 根据亲和度计算产生一个新的抗体种群, 扩大搜索范围, 并通过更新低亲和度的抗体对种群进行重组, 保持抗体种群的多样性, 从而构建出CSA[21], 并且在组合优化, 网络检测和故障诊断等许多领域得到了广泛应用[22–23]. 在人工免疫系统中, 克隆选择是由亲和度诱导的抗体随机映射. 根据克隆选择学说, 定义克隆选择算子对抗体种群$A(t)$依次进行操作: 克隆操作$T^{\mathrm{C}}$, 基因变异操作$T^{\mathrm{M}}$和克隆选择操作$T^{\mathrm{S}}$. 抗体群的状态转移情况可以表示成如下的随机过程[24]:

$$\begin{aligned}&A(t)\xrightarrow{\textbf{clone}(T^{\mathrm{C}})}X(t)\xrightarrow{\textbf{mutation}(T^{\mathrm{M}})}Y(t)\\&\xrightarrow{\textbf{recombination}(Y(t)\cup A(t))}D(t)\\&\xrightarrow{\textbf{selection}(T^{\mathrm{S}})}A(t+1).\end{aligned}$$

在人工免疫系统中, 抗原对应于优化问题的目标函数和各种约束条件, 抗体对应于问题的候选解, 抗原和抗体之间的亲和度一般指候选解对问题的适应性度量, 即解与目标函数的匹配程度. 定义初始抗体群$A(N)$.

1) 克隆操作$T^{\mathrm{C}}$: 定义克隆操作为$X(t)=T^{\mathrm{C}}(A(t))$=$\{T^{\mathrm{C}}(A_1(t)),T^{\mathrm{C}}(A_2(t)),\cdots,T^{\mathrm{C}}(A_n(t))\}$, 其中: $X_i(t)=T^{\mathrm{C}}(A_i(t))=\{X_{i1}(t),X_{i2}(t),\cdots,X_{iq_i}(t)\}$, $X_{ij}(t)=A_i(t)$, $i=1,2,\cdots,n$, $j=1,2,\cdots,q_i$, $q_i$为种群的克隆规模.

2) 基因变异操作$T^{\mathrm{M}}$: 根据概率$p_{\mathrm{m}}$对克隆后的抗体群$X(t)$进行变异操作:

$$\begin{aligned}&Y(t)=T^{\mathrm{M}}(X(t))=\{T^{\mathrm{M}}(X_1(t)),T^{\mathrm{M}}(X_1(t)),\\&\qquad\cdots,T^{\mathrm{M}}(X_1(t))\},\\&Y_i(t)=\{Y_{i1}(t),Y_{i2}(t),\cdots,Y_{iq_i}(t)\},\\&Y_{ij}(t)=T^{\mathrm{M}}(X_{ij}(t)),\\&i=1,2,\cdots,n,\ j=1,2,\cdots,q_i.\end{aligned}$$

为了保留抗体原始种群的信息, 变异算子并不作用到抗体种群$A(t)$.

3) 克隆选择操作$T^{\mathrm{S}}$: 克隆选择操作$T^{\mathrm{S}}$能够从种群$Y(t)$和$A(t)$重组后的种群中$D(t)$选择优秀的个体, 从而形成新的种群, 这个过程可以由式(1)表示:

$$\begin{aligned}&A(t+1)=T^{\mathrm{S}}(Y(t)\cup A(t))=\\&\begin{cases}Y_i^*(t),\ \text{if } F(Y_i^*(t))>F(A_i(t)),\\A_i(t),\ \ \text{otherwise},\end{cases}\end{aligned}\tag{1}$$

其中: $Y_i^*(t)$为$Y_i(t)$中亲和度最高的抗体,

$$\begin{aligned}&A(t+1)=T^{\mathrm{S}}(Y(t)\cup A(t))=\\&\{T^{\mathrm{S}}(Y_1(t)\cup A_1(t)),\cdots,T^{\mathrm{S}}(Y_n(t)\cup A_n(t))\}=\\&\{A_1(t+1),A_2(t+1),\cdots,A_n(t+1)\}.\end{aligned}\tag{2}$$

### 2.2　基于多层学习的克隆选择优化算法(MLCSA)

本文提出利用Baldwinian learning[25–26]和Lamarckian learning[27]组成多层学习策略, 扩大CSA中抗体信息的搜索空间, 加强高亲和度抗体种群的繁衍能力, 通过交叉进化搜索更新克隆操作后的种群, 为克隆选择提供最优适应度值的抗体参数.基于多层学

习的克隆选择优化算法中抗体状态可表示为

$$A(t)\xrightarrow{\textbf{clone}(H^{\mathrm{C}})}X(t)\xrightarrow{\textbf{B-learning}(H_{\mathrm{L}}^{\mathrm{B}})}Y(t)\xrightarrow{\textbf{L-learning}(H_L^L)}Z(t)\xrightarrow{\textbf{recombination}(Z(t)\cup A(t))}D(t)\xrightarrow{\textbf{selection}(H^{\mathrm{S}})}A(t+1),$$

其中克隆优化算法主要操作如下:

1) 克隆操作$H^{\mathrm{C}}$.

定义抗体种群$A(t)=\{A_1(t), A_2(t),\cdots, A_n(t)\}$, 则克隆操作为

$$X(t)=\{H^{\mathrm{C}}(A_1(t)),\ H^{\mathrm{C}}(A_2(t)),\cdots, H^{\mathrm{C}}(A_n(t))\},$$

其中:

$$X_i(t)=H^{\mathrm{C}}(A_1(t))=\{X_{i1}(t), X_{i2}(t),\cdots, X_{iq_i}(t)\},$$
$$X_{ij}(t)=A_i(t),\ i=1,2,\cdots,n,\ j=1,2,\cdots,q_i,\ q_i$$

为种群的克隆规模.

2) Baldwinian learning $H_{\mathrm{L}}^{\mathrm{B}}$.

Baldwinian learning基于Baldwinian效应, 通过改变搜索空间的形状产生进化过程中的最优解. 定义

$$X(t)=\{X_1(t), X_2(t),\cdots, X_n(t)\},$$
$$X_i(t)=\{x_{i1}(t), x_{i2}(t),\cdots, x_{iq_i}(t)\},$$

Baldwinian learning操作如下所示:

$$Y_{i1}(t)=H_L^{\mathrm{S}}(x_{ij}(t))=\begin{cases}x_{ij}(t)+s\cdot(x_l(t)-x_m(t)), & \text{if rand}\leqslant p_i,\\ x_{ij}(t), & \text{else},\end{cases}\tag{3}$$

其中: $l,m=1,2,\cdots,n,\ l\neq m\neq i,\ x_l(t)$和$x_m(t)$分别为$X_l(t), X_m(t)$中随机选取的抗体; $s$为Baldwinian learning的学习强度, $p_i$为Baldwinian learning的概率, rand为[0, 1]均匀分布中的随机数. 经过Baldwinian learning操作的种群变为$Y(t)=\{Y_1(t), Y_2(t),\cdots, Y_n(t)\}$, 其中:

$$Y(t)=\{y_{i1}(t), y_{i2}(t),\cdots, y_{iq_i}(t)\},$$
$$y_{ij}(k)=H_{\mathrm{L}}^{\mathrm{B}}(x_{ij}(t)),$$
$$j=1,2,\cdots,q_i,\ i=1,2,\cdots,n.$$

3) Lamarckian learning $H_{\mathrm{L}}^{\mathrm{L}}$.

随机产生$q_i$个方向向量, 设定局部搜索的初始方向, 每个$\overrightarrow{a_i}$为初始点, 利用Lamarckian learning可扩展到搜索$\overrightarrow{a_i}$的近邻全局最优解. Lamarckian learning操作如下:

$$H_L^L(B(t))=Z(t)=\{H_L^L(\overrightarrow{B}_1^1(t),\overrightarrow{d}_1^1)+\cdots+H_L^L(\overrightarrow{B}_1^{q_1}(t),\overrightarrow{d}_1^{q_1})\}+\{H_L^L(\overrightarrow{B}_2^1(t),\overrightarrow{d}_2^1)+\cdots+H_L^L(\overrightarrow{B}_2^{q_1}(t),\overrightarrow{d}_2^{q_1})\}+\cdots+\{H_L^L(\overrightarrow{B}_n^1(t),\overrightarrow{d}_n^1)+\cdots+H_L^L(\overrightarrow{B}_n^{q_1}(t),\overrightarrow{d}_n^{q_1})\},\tag{4}$$

其中$d_i^j,\ i=1,2,\cdots,n,\ j=1,2,\cdots,q_i$.

4) 克隆选择操作$H^{\mathrm{S}}$.

定义$z_i^*(t)\in Z_i(t), i=1,2,\cdots,n$, $z_i^*(t)$为$Z_i(t)$中亲和度最高的抗体, 并且

$$a_i(t+1)=H^{\mathrm{C}}(Z_i(t)\cup a_i(t))=\begin{cases}z_i^*(t), & \text{if } F(z_i^*(t))>F(a_i(t)),\\ a_i(t), & \text{else.}\end{cases}\tag{5}$$

重组经过多层学习操作形成的种群$Z(t)$和$A(t)$, 对重组后的种群进行克隆选择操作$H^{\mathrm{S}}$产生新的抗体种群$A(t+1)$, 具体操作如下:

$$A(t+1)=H^{\mathrm{C}}(Z(t)\cup A(t))=\{H^{\mathrm{C}}\ (Z_1(t)\cup A_1(t)),\cdots, H^{\mathrm{C}}(Z_n(t)\cup A_n(t))\}=\{A_1(t+1), A_2(t+1),\cdots, A_n(t+1)\},\tag{6}$$

其中$A_i(t+1)=H^{\mathrm{C}}(Z_i(t)\cup A_i(t)), i=1,2,\cdots,n$.

**Step 1** 初始化: 随机产生$N$个抗体, 组成初始抗体种群$A(t)$, 并计算每个抗体的亲和度, 设$t=0$;

**Step 2** 克隆操作$H^{\mathrm{C}}$: 根据亲和度, 对抗体种群$A(t)$中的抗体进行克隆增殖操作, 得到扩增后的种群$X(t)$, 克隆规模是抗体亲和度的单调递增函数;

**Step 3** 多层学习操作:

**Step 3.1** Baldwinian learning $H_{\mathrm{L}}^{\mathrm{B}}$: 对种群$X(t)$中的抗体进行B-learning操作, 生成抗体种群$Y(t)$;

**Step 3.2** Lamarckian learning $H_{\mathrm{L}}^{\mathrm{L}}$: 对种群$X(t)$中的抗体进行L-learning操作, 生成抗体种群$Z(t)$;

**Step 4** 重组操作: 计算$Z(t)$中抗体的亲和度, 对$Z(t)$和$A(t)$重组, 产生种群$D(t)$;

**Step 5** 克隆选择$H^{\mathrm{S}}$: 对$D(t)$进行克隆选择操作, 得到新的抗体种群$A(t+1)$;

**Step 6** 终止: 如果终止条件被满足, 停止并且输出$A(t+1)$中亲和力最高的抗体, 否则, $t=t+1$, 转至Step 2.

### 2.3 克隆选择优化算法的实验结果与分析(Experiment and analysis of MLCSA)

为了测试本节提出的基于多层学习的克隆选择优化算法(multi–learning CSA, MLCSA)对函数优化时的求解性能, 测试实验采用了12个不同类型的测试函数对算法进行测试, 其中包括单峰测试函数, 多峰测试函数, 旋转多峰测试函数, 并与现有5种算法进行对比, 分别为: 克隆选择算法(CSA)、鲍德温克隆选择算法(Baldwinian clonlal selection algorithm, BCSA)、拉马克克隆选择算法(Lamarckian clonal selection algorithm, LCSA)、微分进化算法(differential evolution algorithm, DEA). 测试函数中: $f_1-f_5$取维数$D=2$进行测试, $f_6-f_{12}$分别取维数$D=10$和$D=30$进行测试, 抗体种群规模$NP=30$, 设定克隆规模$nc=5$, 算法终止条件为迭代次数为300, Baldwinian学习强度为0.5. 测试函数的具体形式见表1.

表 1　用于算法验证的12个测试函数

Table 1　The 12 test functions used in this study

| 函 数 名 | $D$ | $S$ |
|---|---|---|
| $f_1(x)=(4-2.1x_1^2+x_1^{4/3})\cdot x_1^2+x_1x_2+(-4+4x_1^2)\cdot x_2^2$ | 2 | $x_1\in[-3,3],\ x_2\in[-2,2]$ |
| $f_2(x)=0.5+[(\sin\sqrt{x_1^2+x_2^2})^2-0.5]/[1.0+0.001(x_1^2+x_2^2)]^2$ | 2 | $x_1,x_2\in[-100,100]$ |
| $f_3(x)=(x_1^2+x_2^2)^{0.25}[\sin^2(50(x_1^2+x_2^2)^{0.1}+1.0)]$ | 2 | $x_1,x_2\in[-100,100]$ |
| $f_4(x)=(1+(x_1+x_2+1)^2\cdot(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2))\cdot$ $(30+(2x_1-3x_2)^2\cdot(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2))$ | 2 | $x_1,x_2\in[-2,2]$ |
| $f_5(x)=a(x_2-\frac{5.1}{4\cdot\pi^2}\cdot x_1^2+\frac{5}{\pi}\cdot x_1-6)^2+10\cdot(1-\frac{1}{8\cdot\pi})\cdot\cos x_1+10$ | 2 | $x_1\in[-5,10],\ x_2\in[0,15]$ |
| $f_6(x)=\{\sum_{i=1}^{D}i\cdot\cos[(i+1)x+i]\}\times\{\sum_{i=1}^{D}i\cdot\cos[(i+1)y+i]\}$ | 10/30 | [–10, 10] |
| $f_7(x)=-20*\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2})-\exp(-\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i))+22.71282$ | 10/30 | [–5, 5] |
| $f_8(x)=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})$ | 10/30 | [–600, 600] |
| $f_9(x)=\sum_{i=1}^{D-1}[100\cdot(x_{i+1}-x_i^2)^2+(1-x_i)^2]$ | 10/30 | [–2.048, 2.048] |
| $f_{10}(x)=\sum_{i=1}^{D-1}[x_i^2-10\cdot\cos(2\pi x_i)+10]$ | 10/30 | [–5.12, 5.12] |
| $f_{11}(x)=\sum_{i=1}^{D}x_i^2$ | 10/30 | [–1, 1] |
| $f_{12}(x)=418.9829\times D-\sum_{i=1}^{D-1}\sin(\lvert x_i\rvert^{1/2})$ | 10/30 | [–500, 500] |

表 2　$D=2,10,30$, CSA, BCSA, LCSA, DEA和MLCSA的测试对比

Table 2　Mean and standard deviation values obtained by CSA, BCSA, LCSA, DEA and MLCSA, when $D=2,10,30$

| 函数 | $D$ | CSA | BCSA | LCSA | DEA | MLCSA |
|---|---|---|---|---|---|---|
| | | Mean±Std | Mean±Std | Mean±Std | Mean±Std | Mean±Std |
| $f_1$ | 2 | 8.2291E–001±2.1247E–006 | 3.2378E–001±1.4892E–009 | 1.0316E+000±1.2883E–009 | 2.6123E+000±1.2883E–009 | **2.0234E–001±7.6545E–011** |
| $f_2$ | 2 | 9.9939E–001±1.7823E–002 | 4.2801E–001±2.2531E–016 | **0±0** | 7.1313E–001±3.7782E–004 | 3.9723E–001±1.2004E–021 |
| $f_3$ | 2 | 1.1215E+001±4.2127E–005 | 3.3297E+000±3.2129E–004 | 8.7245E+000±1.2562E–005 | 2.1563E+000±2.4533E–002 | **2.1123E+000±1.1351E–007** |
| $f_4$ | 2 | 1.0124E+004±2.9548E+001 | 2.7433E+000±3.4621E+000 | 3.4528E+000±7.2849E–001 | 5.1384E+003±2.6213E+000 | **0±0** |
| $f_5$ | 2 | 1.7376E+002±2.5521E–003 | 5.0178E–001±3.1514E–004 | 4.6562E+001±5.0782E–004 | 1.7376E+002±2.4721E–003 | **1.5266E–002±1.7219E–006** |
| $f_6$ | 10 | 1.0011E+001±8.3621E–004 | 2.0832E+001±7.6523E–003 | 2.3030E+000±6.0134E–004 | 5.6548E+001±2.6411E–003 | **1.7676E+000±5.2372E–006** |
| | 30 | 1.1250E+001±1.0294E–003 | 1.5513E+001±4.6599E–005 | 7.2203E+000±5.6654E–005 | 3.0981E+001±3.1024E–004 | **6.9426E+000±1.9834E–005** |
| $f_7$ | 10 | 2.8739E–003±8.2820E–003 | 4.4409E–015±**0** | 7.0449E+000±7.9257E+000 | 1.9864E–012±1.2303E–012 | **3.0628E–015**±2.8824E–016 |
| | 30 | 1.0234E–002±3.2108E–002 | 1.5614E–011±3.9345E–013 | 2.0479E–010±6.9213E–012 | 1.0717E–005±5.9408E–006 | **8.0001E–012±3.2883E–015** |
| $f_8$ | 10 | 2.6429E–002±2.1023E–002 | 4.4409E–015±**0** | 2.5438E–013±1.4537E–012 | 1.9864E–012±1.2303E–012 | **5.2123E–014**±7.4503E–022 |
| | 30 | 1.7234E–003±1.0002E–004 | 8.6168E–003±1.1259E–002 | 3.2127E–006±9.2130E–009 | 1.8079E–003±4.7543E–003 | **9.7018E–009±1.0105E–011** |
| $f_9$ | 10 | 7.9837E+000±2.8720E+000 | 6.8559E–013±1.7711E–012 | 1.1245E+001±2.2796E+001 | 3.4243E–006±2.7598E–006 | **3.7319E–008±3.2613E–019** |
| | 30 | 3.3894E+001±2.9022E+000 | 1.1896E–011±4.2596E–011 | 2.6377E–007±5.2001E–009 | 1.2323E+001±1.1123E+000 | **0±0** |
| $f_{10}$ | 10 | 3.2883E+001±4.9892E+000 | 4.9146E–015±2.6900E–014 | 8.1951E+001±6.8196E+000 | 1.7691E+001±5.4715E+000 | **4.3253E–016±7.3311E–017** |
| | 30 | 7.7992E+001±3.8839E+000 | **0±0** | 5.2112E–001±8.2854E–004 | 1.0458E+002±3.0790E+001 | 4.3253E–002±7.3311E–005 |
| $f_{11}$ | 10 | 9.7662E–009±5.6508E–012 | 6.1735E–030±9.1544E–030 | **0±0** | 1.9407E–023±2.0307E–023 | **0±0** |
| | 30 | 2.0023E–005±6.9922E–009 | 2.9665E–025±8.4182E–025 | 2.0023E–005±8.6284E–009 | 2.3391E–009±5.2742E–009 | **5.8004E–044±7.7219E–050** |
| $f_{12}$ | 10 | 3.8211E+002±1.0128E+002 | 7.6982E+000±3.1017E–002 | **7.9329E–001±2.0322E–003** | 4.1282E+000±2.0063E–002 | 9.1360E–001±2.7632E–003 |
| | 30 | 1.1998E+003±2.3190E+002 | 4.5350E+002±7.9492E–001 | 6.1130E+002±2.2401E–001 | 1.7160E+003±3.1302E+002 | **2.0183E+002±9.8877E–002** |

表2为MLCSA算法与现有4种优化算法的性能比较, 其中$f_1-f_5$为测试函数$f_1-f_5$在维度$D=2$时独立运行30次测试的平均值结果, 从结果可以看出, MLCSA算法的展现出了更好的函数求解性能.

当维度$D=10$时, 通过对测试函数$f_6-f_{12}$的实验结果可以看出, MLCSA算法能准确地找到测试函数$f_6$, $f_7$, $f_8$, $f_9$, $f_{11}$, $f_{12}$全局最优解和$f_{10}$的近似全局最优解. 当维度$D=30$时, MLCSA算法均能准确地找到测试函数$f_6$, $f_7$, $f_8$, $f_9$, $f_{10}$, $f_{11}$的全局最优解, 实验结果表明, 除$f_6(D=2)$, $f_{10}(D=30)$, $f_{12}(D=10)$算法LCSA和BCSA表现出较好的性能以外, 本文所提出的MLCSA算法在低维和高维不同测试下均能展现出更好的搜索全局最优解能力, 并且通过多层学习方法减少对抗体信息的依赖程度, 提高算法解决复杂优化问题的性能.

## 3 核增量型超限学习机(Incremental extreme learning machine with kernel)

### 3.1 超限学习机(Extreme learning machine)

假设前馈神经网络模型$K$个隐含层节点, 激活函数$g(\cdot)$. 对于$N$个不同的学习样本$(X,Y)$, $X=\{x_i[n]\}\in\mathbb{R}^{N\times L}$, $Y=\{y_i[n]\}\in\mathbb{R}^{N\times L}$, ELM的数学表达式如下[1]:

$$\sum_{i=1}^{L} w_i g(W_{in(i)}\cdot x_j+b_i)=o_j,\ j=1,2,\cdots,N, \quad (7)$$

其中$g(\cdot)$为激活函数, 可以是任意无穷可微函数, 一般取Sigmoid函数[28]. 式(7)可改写成矩阵向量形式:

$$Hw=o. \quad (8)$$

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, 那么意味着存在$w_i$使得

$$\sum_{i=1}^{L} w_i g(W_{in(i)}\cdot x_j+b_i)=t_j,\ j=1,2,\cdots,N, \quad (9)$$

其中$t_j$为目标值. 式(4)可以改写成矩阵向量的形式:

$$Hw=t, \quad (10)$$

其中$t=[t_1,t_2,\cdots,t_N]^{\mathrm{T}}$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:

$$w=H^{\dagger}t, \quad (11)$$

其中$H^{\dagger}$是$H$的广义逆运算.

### 3.2 核增量型超限学习机(I–ELMK)

基于核方法的智能算法在处理非线性不可分问题中具很强的分类能力和良好的拟合能力, Frenay和Lu等[29–30]将核方法引入ELM, 利用固定训练样本集来建立离线训练学习模型. ELM的核矩阵表示如下:

$$K_{\mathrm{ELM}}=HH^{\mathrm{T}}, \quad (12)$$

$$K_{\mathrm{ELM}}=h(x_i)\cdot h(x_j)=K(x_i,x_j). \quad (13)$$

因此, ELM的输出函数可以表示为

$$f(x)=h(x)H^{\mathrm{T}}(HH^{\mathrm{T}}+\frac{1}{C})^{-1}t=\begin{bmatrix}K(x,x_1)\\K(x,x_2)\\\vdots\\K(x,x_N)\end{bmatrix}^{\mathrm{T}}(K_{\mathrm{ELM}}+\frac{1}{C})^{-1}t, \quad (14)$$

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

设$A=[K_{\mathrm{ELM}}+\frac{1}{C}]$, 对于时刻$t$有

$$A_t=\begin{bmatrix}\frac{1}{C}+K(x_1,x_1) & \cdots & K(x_1,x_N)\\\vdots & \vdots & \vdots\\K(x_N,x_1) & \cdots & \frac{1}{C}+K(x_N,x_N)\end{bmatrix}. \quad (15)$$

对于$t+1$时刻数据:

$$A_{t+1}=\begin{bmatrix}\frac{1}{C}+K(x_1,x_1) & \cdots & K(x_1,x_{N+k})\\\vdots & \vdots & \vdots\\K(x_{N+k},x_1) & \cdots & \frac{1}{C}+K(x_{N+k},x_{N+k})\end{bmatrix}. \quad (16)$$

简化矩阵$A_{t+1}$, 设

$$U_t=\begin{bmatrix}K(x_1,x_{N+1}) & \cdots & K(x_1,x_{N+k})\\\vdots & \vdots & \vdots\\K(x_N,x_{N+1}) & \cdots & K(x_N,x_{N+k})\end{bmatrix}, \quad (17)$$

$$D_t=\begin{bmatrix}\frac{1}{C}+K(x_{N+1},x_{N+1}) & \cdots & K(x_{N+1},x_{N+k})\\\vdots & \vdots & \vdots\\K(x_{N+k},x_{N+1}) & \cdots & \frac{1}{C}+K(x_{N+k},x_{N+k})\end{bmatrix}, \quad (18)$$

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

$$A_{t+1}^{-1}=\begin{bmatrix}A_t^{-1}+A_t^{-1}U_tC_t^{-1}U-t^{\mathrm{T}}A_t^{-1} & -A_t^{-1}U_tC_t^{-1}\\-C_t^{-1}U_t^{\mathrm{T}}A_t^{-1} & C_t^{-1}\end{bmatrix}, \quad (19)$$

其中$C_t=D_t-U_t^{\mathrm{T}}A_t^{-1}U_t$.

在线更新测试数据, 设给定数据量为$M$, 对于测试数据$X_{\mathrm{test}}=[x_{\mathrm{test}1},x_{\mathrm{test}2},\cdots,x_{\mathrm{test}M}]$, 利用$A_{t+1}^{-1}$估计输出值$\hat{Y}_{\mathrm{test}}$, $\hat{Y}_{\mathrm{test}}$可表示为

$$\hat{Y}_{\mathrm{test}}=$$

$$\begin{bmatrix} K(x_{\text{test}1}, x_1) & \cdots & K(x_{\text{test}1}, x_M) \\ \vdots & \vdots & \vdots \\ K(x_{\text{test}N}, x_1) & \cdots & K(x_{\text{test}N}, x_M) \end{bmatrix} \begin{bmatrix} A_M^{-1} y_1 \\ \vdots \\ A_M^{-1} y_M \end{bmatrix}, \tag{20}$$

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

## 4 基于多层学习克隆选择的增量型超限学习机(MLCSI–ELMK)

### 4.1 MLCSI–ELMK的学习步骤(Learning steps of MLCSI–ELMK)

构造基于多层学习算法的超限学习机, 设ELM隐含层节点参数$w_i$为需要优化的变量. 目标函数为给定期望的网络输出误差$\eta$. 设定$\Omega_i(w_i)$为抗体种群内需要进行克隆优化选择的待优化抗体, 种群规模为$N$, 初始隐含层节点数为$L$, 随机生成一定规模的抗体种群, 对种群内的抗体进行二进制编码操作, 并对各抗体进行编码操作, 根据多层克隆学习算法计算出最优隐含层参数$\Omega_i^*(w_i, b)$, 增加隐含层节点数$L = L + \lambda$, 再次初始生成种群规模为M的抗体种群, 交叉选择次优参数, 如果网络输出误差到达设定值, 多层克隆学习优化过程终止. 网络权值编码规则如图1所示.

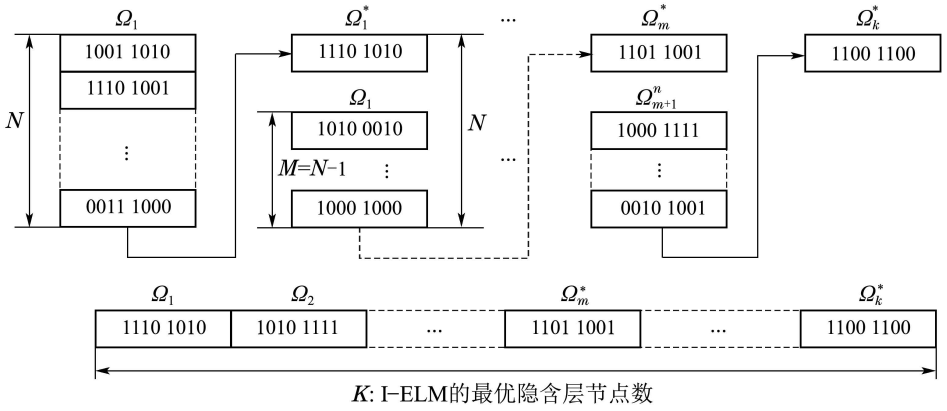



图 1 基于MLCSA的网络权值编码规则

Fig. 1 The encoding principle of network weights based on MLCSA

MLCSI–ELMK算法的具体实现步骤如下:

**第1阶段** 计算最佳隐含层节点数$L_{\text{best}}$.

**Step 1** 初始化. 随机产生$M$个抗体, 组成初始抗体种群$A(t)$, 并计算每个抗体的亲和度, 给定$N$个训练样本$(x_i, t_i)$, $x_i = [x_{i1}, x_{i2}, \cdots, x_{in}]$, $t_i \in \mathbb{R}(i = 1, 2, \cdots, N)$, $\Omega_N(w_i, b) = A$给定期望的网络输出误差$\eta$;

**Step 2** 设神经网络隐含层节点数$L = 0$, 网络初始误差$E_L$, $L = L + \lambda$;

**Step 3** 克隆操作. 据亲和度, 对抗体种群$A(t)$中的抗体进行克隆增殖操作, 得到扩增后的种群;

**Step 4** 多层学习操作. 对种群$X(t)$中的抗体进行Baldwinian learning(B–L)操作, 生成抗体种群$Y(t)$, 对种群$Y(t)$中的每个克隆抗体进行Lamarckian learning(L–L)操作形成$Z(t)$;

**Step 5** 评价. 计算$Z(t)$中抗体的亲和度;

**Step 6** 克隆选择. 对$Z(t)$和$A(t)$进行克隆选择操作, 得到新的抗体种群$A(t+1)$;

**Step 7** 终止. 如果终止条件被满足, 停止并且输出$A(t+1)$中亲和力最高的抗体, 最优隐含层否则, $t = t + 1$转至Step 2.

**Step 8** 计算输出权值$w_L^* = \dfrac{E_L \cdot H_L^{\text{T}}(\Omega_L^*)}{H_L(\Omega_L^*) \cdot H_L^{\text{T}}(\Omega_L^*)}$, 计算输出误差$E_L = E_{L-1} - w_L^* \| H_L(\Omega_L^*), x \|$;

**Step 9** 若$E_L < \eta$则终止训练, 否则跳到Step 3.

**第2阶段** 在线计算预测输出值$\hat{Y}_{\text{test}}$.

**Step 10** 设$G = [K_{\text{ELM}} + \dfrac{1}{C}]$, 对于$t$时刻, $G_{t+1}$;

**Step 11** 对参数$G_{t+1}$求逆运算得$G_{t+1}^{-1}$;

**Step 12** 在线更新测试数据$\hat{Y}_{\text{test}}$;

**Step 13** 计算输出值$\hat{Y}_{\text{test}} = [y_1, y_2, \cdots, y_M]^{\text{T}}$.

为了提高神经网络的运算速度, 当Step 2中$\lambda$可以取大于1的整数, 即表示算法中隐含层节点数量为组群增加, 如果$\lambda$等于神经网络最大隐含层节点设定数量时, 基于多层学习克隆选择算法可以保存隐含层节点参数$\Omega_i(w_i, b)$的次最优解作为全局最优解, 并输出隐含层节点参数和输出权值.

### 4.2 MLCSI–ELMK的收敛性证明(Proof of convergence for MLCSI–ELMK)

**引理** 给定任意有界连续或分段连续激励函数, 对于任意连续目标函数$f$, 存在任意输出矩阵$H$和前馈单隐层神经网络使得

$$\lim_{L\to+\infty}\|e_L\|=\lim_{L\to+\infty}\|H_L\beta_L-f\|=0,$$

且网络误差$\|e_L\|$随隐含层节点数$L$单调下降, 若$\beta_L=\dfrac{f\cdot H_L^{\mathrm{T}}}{H_L\cdot H_L^{\mathrm{T}}}$.

**定理** 给定的离散样本$(x_i,t_i)$, $x\in\mathbb{R}^n$, $t\in\mathbb{R}^m$, 给定任意$\varepsilon>0$, 必定存在具有$q$个神经元的前馈单隐层神经网络, 使得$\|H_q\beta_q-f\|<\varepsilon$, 若神经网络的参数由MLCSI–ELMK算法训练获得.

**证** 设基于多层学习的克隆选择算法的最大迭代次数为$K$, 当前隐含层节点数为$L$. 初始设定$k=0$, $k<K$, 相应的网络输出余差$\|E_L(H_L^{k=0})\|=\|H_L(\Omega_{k=0}^*)\beta_L-f\|$.

若MLCSA迭代$K$次, 即$k=K$, 可得隐含层节点数为$L$时的网络输出余差: $\|H_L(\Omega_K^*)\beta_L-f\|\leqslant\|H_L(\Omega_0^*)\beta_L-f\|$. 此时, 若增加隐含层节点$L=L+1$, 根据引理1可得, 相应的网络输出误差满足$\|H_{L+1}(\Omega_K^*)\beta_{L+1}-f\|\leqslant\|H_{L+1}(\Omega_0^*)\beta_{L+1}-f\|\leqslant\|H_L(\Omega_0^*)\beta_L-f\|$. 因此, 当$L=N$且$k=K$时, 可得

$$\|E_N(H_N^K)\|\leqslant\|E_N(H_N^0)\|\leqslant\cdots\leqslant$$
$$\|E_2(H_2^0)\|\leqslant\|E_1(H_1^K)\|\leqslant\|E_1(H_1^0)\|,$$

即

$$\|H_N(\Omega_K^*)\beta_N-f\|\leqslant$$
$$\|H_N(\Omega_0^*)\beta_N-f\|\leqslant\cdots\leqslant$$
$$\|H_2(\Omega_0^*)\beta_2-f\|\leqslant\|H_1(\Omega_K^*)\beta_1-f\|\leqslant$$
$$\|H_1(\Omega_0^*)\beta_1-f\|,$$

且$\lim\limits_{N\to+\infty}\|H_N(\Omega_K^*)\beta_N-f\|=0$. 因此, 若给定任意$\varepsilon>0$, 存在隐含层节点数为$q<N$的神经网络, 使得$\|H_q(\Omega_K^*)\beta_q-f\|<\varepsilon$成立.

由定理的证明可以看出, 隐含层节点数为$q$的MLCSI–ELMK网络能够通过增量型算法准确地学习$N$个任意不同的样本, 并且可以将算法应用于回归和分类问题.

## 5 仿真及结果分析(Experiment of simulation and the analysis of results)

为验证本文所提出MLCSI–ELMK算法的有效性, 利用UCI数据集(http://archive.ics.uci.edu/ml/)对CI–ELM, EI–ELM, PC–ELM, LOO–IELM, ErrCor, CW–ELM, II–RELM和MLCSI–ELMK进行测试验证, 8组典型真实数据集分别是: Machine CPU, Combined Cycle Power Plant (CCPP), Friedman, Parkinsons, Waveform II, Energy efficiency, Letter recognition, Satellite image.

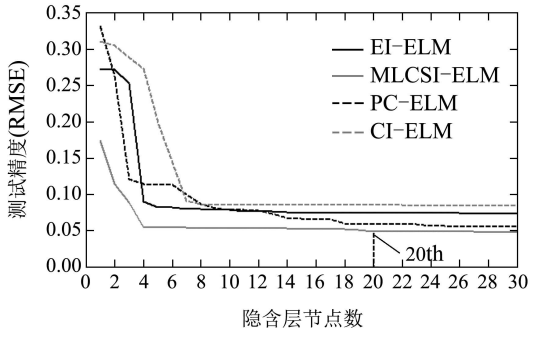
本文所有仿真均在同一系统环境下运行所得, 系统运行环境PC (Quad CPU 2.7 Hz, 16.00 GB RAM), 操作系统Windows 7, 仿真软件MATLAB R2012a, 抗体种群规模$NP=30$, 设定克隆规模$nc=5$, 算法终止条件为迭代次数为300, Baldwinian学习强度为0.5. 实验数据集分训练数据集和测试数据集, 具体描述如表3所示.

表 3 数据集描述

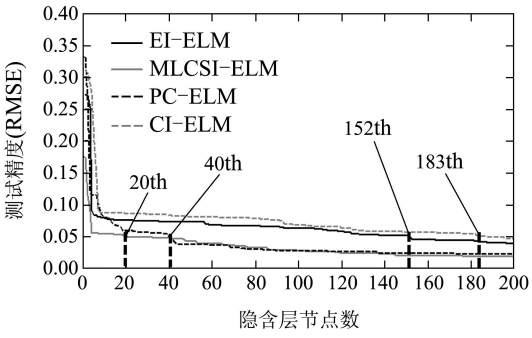
Table 3 Specification of 10 benchmark problem

| 数据集 | 训练样本数 | 测试样本数 | 特征数 | 问题类型 |
|---|---|---|---|---|
| Machine CPU | 100 | 85 | 6 | Regression |
| CCPP | 7, 000 | 4300 | 4 | Regression |
| Friedman | 18, 080 | 12, 600 | 11 | Regression |
| Parkinsons | 2, 800 | 2, 350 | 18 | Regression |
| Waveform II | 2, 000 | 2, 000 | 40 | Classification |
| Energy efficiency | 768 | 500 | 8 | Classification |
| Letter recognition | 10, 000 | 10, 000 | 16 | Classification |
| Satellite image | 9, 300 | 8, 000 | 36 | Classification |

数据测试中, 对输入数据在[–1, 1]区间进行归一化, 相应的输出数据在[0, 1]区间, ELM网络的激活函数均采用Sigmoid函数, 每组对测试数据进行40次实验, 输出结果为40次实验结果的平均值. 图2(a)表示, 利用Waveform II数据集进行分类验证, 当MLCSI–ELMK算法的隐含层节点数(Node)为20时, 其他4种经典的ELM优化算法的RMSE均明显高于0.005. 在非线性动态系统辨识的情况下, 相比于其他4种算法, MKCSI–ELM算法可以获得更快的学习速率和更少的隐层节点数. 图2(b)所示, 设定算法终止条件为RMSE$=0.05$或达到最大迭代次数$M=500$, 5种ELM算法均能够展现快速收敛的能力, 但是对比4种算法的最大隐含层节点数可以发现, 当MLCSI–ELMK算法测试精度达到0.05时, 所需隐含层节点数为30, 明显小于其他几种算法: CI–ELM(183th), EI–ELM(152th), PC–ELM(40th). 通过实验可知, 其余UCI数据集的测试效果也显示了同样的泛化能力.




(a) 隐含层节点数, Node = 30




(b) 隐含层节点数, Node = 200

图 2 基于Waveform II数据集的测试精度对比

Fig. 2 Testing accuracy of different algorithms based on Waveform II

表4给出了本文所提出的MLCSI–ELMK算法和7种增量型ELM算法基于回归问题标准化比较结果. 以Machine CPU数据集为例, 在训练和测试误差对比中, 设定最大隐含层节点数为50, 误差终止条件为RMSE = 0.05, MLCSI–ELMK算法的训练误差为0.0481, 测试误差为0.0494, 回归正确率明显高于其他7种算法. 在回归问题的算法性能比较中, 可以看出基于MLCSI–ELMK算法的实验结果对比其他算法, 当达到初始设定的网络输出误差时, 神经网络所需隐含层节点数值最小. 以Parkinsons数据集为例, 当算法终止条件设定为RMSE = 0.12时, MLCSI–ELMK算法所需节点数平均值为44.86, 算法训练时间为2.5403 s, 所需隐含层节点数明显低于其他7种算法: 50.19(ErrCor, 2.5563 s), 77.12(II–RELM, 2.6001 s), 79.09(PC–ELM, 2.6665 s), 112.38(CW–ELM, 4.1605 s), 140.21(LOO–IELM, 4.4634 s), 150.45(EI–ELM, 4.4861 s), 192.66(CI–ELM, 4.7707 s), 表现出来更紧凑的网络结构.

表5为基于分类问题的准确性, 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MLCSI–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程度, 提高了算法的学习效率, 在保证运算精度的同时确保了运算的快速性, 表现出了优于其他7种改进I–ELM的算法性能.

表 4 回归问题的训练和测试误差对比

Table 4 The error comparison of training and testing of the regression cases

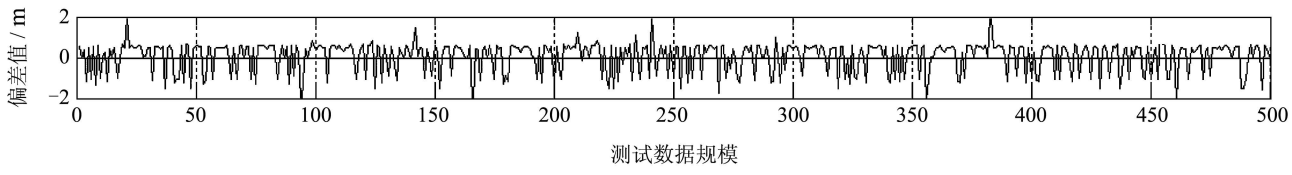
| 数据集 | 算法 | 均方根误差(训练& 测试) | | | 隐含层节点&平均时间 | |
|---|---|---|---|---|---|---|
| | | 节点数(固定) | 训练 | 测试 | #节点数 | 时间/s |
| Machine CPU(0.05) | CI–ELM$^{(2007)}$ | 50 | 0.0882 | 0.1004 | 65.37 | 0.2274 |
| | EI–ELM$^{(2008)}$ | 50 | 0.0674 | 0.0663 | 52.93 | 0.2731 |
| | PC–ELM$^{(2012)}$ | 50 | 0.0573 | 0.0680 | 17.02 | 0.1897 |
| | LOO–IELM$^{(2014)}$ | 50 | 0.0627 | 0.0704 | 80.21 | 0.2343 |
| | ErrCor$^{(2014)}$ | 50 | 0.0492 | 0.0533 | 19.72 | **0.1254** |
| | CW–ELM$^{(2015)}$ | 50 | 0.0582 | 0.0575 | 66.76 | 0.2501 |
| | II–RELM$^{(2015)}$ | 50 | 0.0556 | 0.0598 | 32.64 | 0.2861 |
| | MLCSI–ELMK | 50 | **0.0481** | **0.0494** | **14.89** | 0.1945 |
| Combined Cycle Power Plant(0.052) | CI–ELM | 100 | 0.0573 | 0.0602 | 388.92 | 2.0081 |
| | EI–ELM | 100 | 0.0563 | 0.0566 | 309.18 | 2.5711 |
| | PC–ELM | 100 | 0.0527 | 0.0531 | 47.73 | **1.9061** |
| | LOO–IELM | 100 | 0.0531 | 0.0529 | 190.37 | 2.7867 |
| | ErrCor | 100 | 0.0507 | 0.0511 | 38.08 | 2.0868 |
| | CW–ELM | 100 | 0.0524 | 0.0544 | 166.53 | 2.3064 |
| | II–RELM | 100 | 0.0522 | 0.0529 | 50.32 | 2.7747 |
| | MLCSI–ELMK | 100 | **0.0502** | **0.0513** | **29.15** | 2.0103 |
| Friedman (0.1) | CI–ELM | 50 | 0.1643 | 0.1656 | 72.57 | 0.0431 |
| | EI–ELM | 50 | 0.1501 | 0.1572 | 59.03 | 0.4087 |
| | PC–ELM | 50 | 0.1092 | 0.1101 | 13.98 | 0.0822 |
| | LOO–IELM | 50 | 0.1563 | 0.1589 | 60.61 | 0.4272 |
| | ErrCor | 50 | 0.0972 | 0.1002 | 12.78 | 0.0416 |
| | CW–ELM | 50 | 0.1514 | 0.1511 | 58.04 | 0.3906 |
| | II–RELM | 50 | 0.1083 | 0.1087 | 14.66 | 0.0991 |
| | MLCSI–ELMK | 50 | **0.0892** | **0.0883** | **10.18** | **0.0383** |
| Parkinsons(0.12) | CI–ELM | 200 | 0.1123 | 0.1080 | 192.66 | 4.7707 |
| | EI–ELM | 200 | 0.0889 | 0.0921 | 150.45 | 4.4861 |
| | PC–ELM | 200 | 0.0511 | 0.0493 | 79.09 | 2.6665 |
| | LOO-IELM | 200 | 0.0621 | 0.0633 | 140.21 | 4.4634 |
| | ErrCor | 200 | 0.0382 | 0.0407 | 50.19 | 2.5563 |
| | CW–ELM | 200 | 0.0596 | 0.0589 | 112.38 | 4.1605 |
| | II–RELM | 200 | 0.0468 | 0.0472 | 77.12 | 2.6001 |
| | MLCSI–ELMK | 200 | **0.0321** | **0.0327** | **44.86** | **2.5403** |

表 5 分类问题的训练和测试误差对比

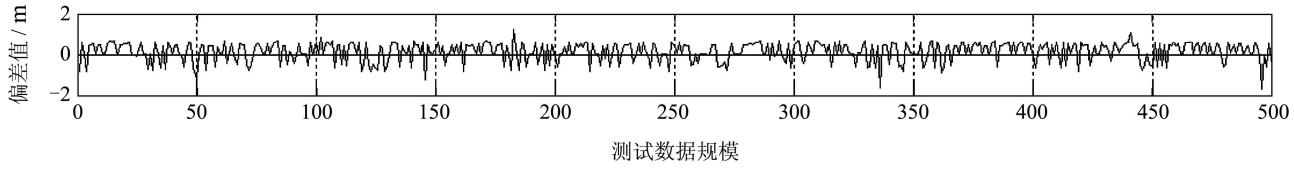
Table 5 The error comparison of training and testing of the classification cases

| 数据集 | 算法 | 测试精度 | | | 隐含层节点& 平均时间 | |
|---|---|---|---|---|---|---|
| | | 节点数(固定) | 平均值 | 标准差 | #节点数 | 时间/s |
| Waveform II (0.05) | CI–ELM$^{(2007)}$ | 50 | 0.8229 | 0.0447 | 182.55 | 2.9803 |
| | EI–ELM$^{(2008)}$ | 50 | 0.7981 | 0.0451 | 155.28 | 2.3321 |
| | PC–ELM$^{(2012)}$ | 50 | 0.8603 | 0.0301 | 39.03 | 3.3371 |
| | LOO–IELM$^{(2014)}$ | 50 | 0.8412 | 0.0307 | 133.07 | 2.8816 |
| | ErrCor$^{(2014)}$ | 50 | 0.8935 | 0.0232 | 27.12 | **2.1225** |
| | CW–ELM$^{(2015)}$ | 50 | 0.8489 | 0.0285 | 118.54 | 2.7006 |
| | II–RELM$^{(2015)}$ | 50 | 0.0871 | 0.0311 | 46.82 | 2.8151 |
| | MLCSI–ELMK | 50 | **0.9098** | **0.0212** | **20.54** | 3.0024 |
| Energy efficiency (0.045) | CI–ELM | 100 | 0.9030 | 0.0014 | 75.05 | **0.3514** |
| | EI–ELM | 100 | 0.9239 | 0.0013 | 83.51 | 0.6049 |
| | PC–ELM | 100 | 0.9652 | 0.0009 | 52.67 | 0.4231 |
| | LOO–IELM | 50 | 0.9454 | 0.0011 | 62.19 | 0.4911 |
| | ErrCor | 50 | 0.9693 | **0.0008** | 28.44 | 0.4750 |
| | CW–ELM | 50 | 0.9582 | 0.0010 | 59.83 | 0.4855 |
| | II–RELM | 50 | 0.9507 | 0.0009 | 52.51 | 0.4512 |
| | MLCSI–ELMK | 100 | **0.9697** | **0.0008** | **25.65** | 0.4683 |
| Letter recognition (0.04) | CI–ELM | 200 | 0.7503 | 0.0118 | 175.31 | 0.9957 |
| | EI–ELM | 200 | 0.7773 | 0.0101 | 150.22 | 0.7306 |
| | PC–ELM | 200 | 0.9046 | 0.0034 | 41.21 | 1.3106 |
| | LOO–IELM | 50 | 0.9201 | 0.0056 | 130.09 | 1.2236 |
| | ErrCor | 50 | 0.9489 | 0.0021 | 37.52 | 0.7808 |
| | CW–ELM | 50 | 0.9312 | 0.0049 | 110.41 | 1.2774 |
| | II–RELM | 50 | 0.9088 | 0.0031 | 51.15 | 1.2006 |
| | MLCSI–ELMK | 200 | **0.9505** | **0.0019** | **33.52** | **0.7218** |
| Landsat satellite image (0.062) | CI–ELM | 200 | 0.8440 | 0.0044 | 55.65 | **1.4787** |
| | EI–ELM | 200 | 0.8556 | 0.0039 | 56.93 | 1.7342 |
| | PC–ELM | 200 | 0.9103 | 0.0019 | 25.81 | 2.4543 |
| | LOO–IELM | 50 | 0.9219 | 0.0017 | 60.12 | 2.8655 |
| | ErrCor | 50 | 0.9287 | 0.0015 | 19.30 | 1.6508 |
| | CW–ELM | 50 | 0.9304 | 0.0016 | 54.41 | 2.9223 |
| | II–RELM | 50 | 0.9261 | 0.0016 | 28.86 | 2.3409 |
| | MLCSI–ELMK | 200 | **0.9318** | **0.0013** | **16.99** | 1.7002 |

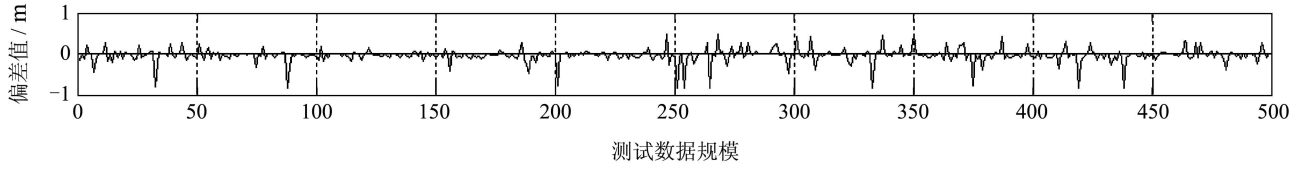
将本文所提出的算法应用到实际工程中, 并与其他几种算法进行对比. 在连续退火过程中, 带钢经过在预热段(preheating section, PHS)、辐射管加热段 (radiation-tube heating section, RHS)、缓冷段 (slow cooling section, SCS)、快冷段 (rapid cooling section, RCS)等温度分区时, 由于自身的金属特性会发生不同程度的伸长或缩短的现象. 利用现有数据针对带钢延伸(或缩短)的长度进行软测量计算可以减少设备的投入, 也适用于无法安装测量传感器的特殊环境的需求. 模型输入数据为: 带钢运行速度、炉内分区温度、各张力辊的张力值等, 输出值为带钢长度变化值. 图3为8种算法的预测偏离值对比图, 从中可以看出基于MLCSI–ELMK算法的带钢长度预测偏差值的波动最小, 偏差范围在$[-1, 1]$之间, 完全满足工业现场的实际要求, 进一步验证了所提出算法较强的泛化能力.



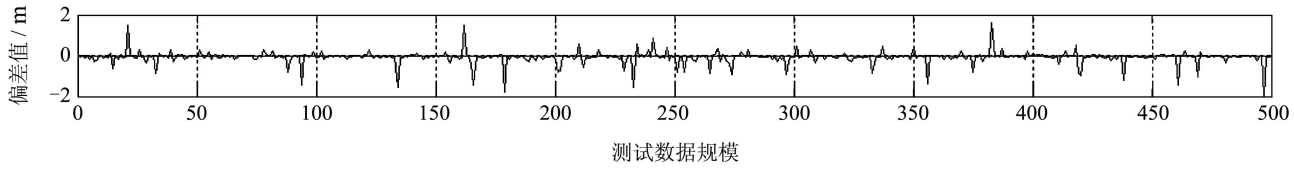

(a) CI–ELM



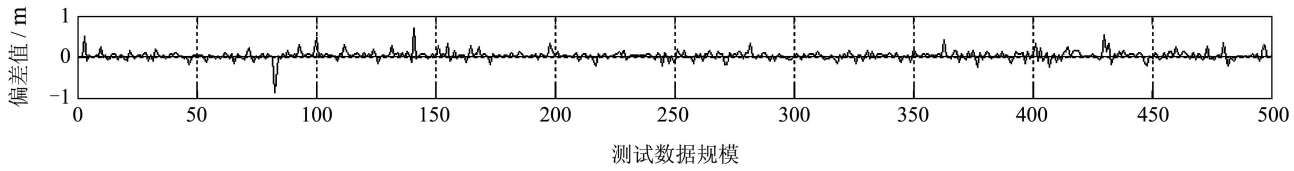

(b) EI–ELM



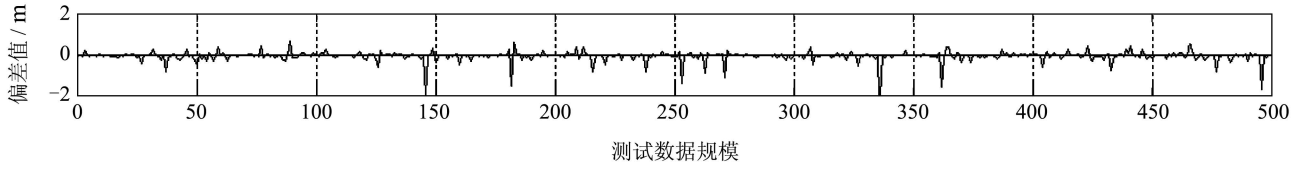

(c) PC–ELM



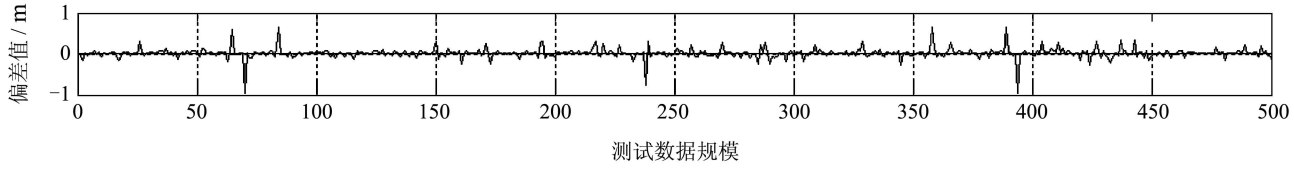

(d) LOO–IELM



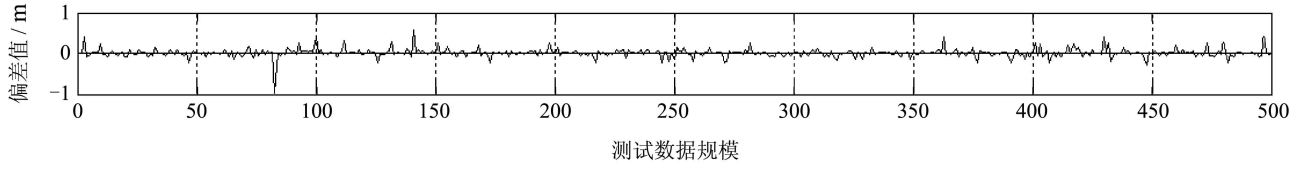

(e) ErrCor




(f) CW–ELM




(g) II–RELM




(h) MLCSI–ELMK

图 3 基于带钢延伸量数据的算法预测对比

Fig. 3 The comparisons of deviation prediction based on strip-elongation data

## 6 结论(Conclusions)

ELM作为系统辨识与模式识别的有力工具, 具有较好的泛化性能和广泛的应用前景, 但是由于ELM算法存在最优隐含层节点难以确定的问题, 考虑到I–ELM算法能够通过累加隐藏层节点的方法避免隐含层节点的盲目选择和过拟合现象, 因此本文的研究主要集中在I–ELM算法的隐含层节点控制问题上. 文本利用多层学习算法代替传统CSA中的基因变异操作过程, 降低了抗体信息的依赖程度, 增强了高亲和度抗体的繁衍能力. 基于MLCSA的I–ELM算法能够对网络权值进行有针对性的搜索, 并保持最优的权值信息作为隐含层节点增加对网络输出误差影响的校验条件, 快速和有效地控制节点数的增加, 降低了网络的复杂程度, 在UCI数据和实际工业生产数据测试中表现出了更好的回归和分类能力. 对于在线数据更新问题, 采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. 虽然算法解决了网络结构复杂化的问题, 但是由于MLCSA算法需要以牺牲少量运算时间为代价换取更高的精确度, 导致本文提出的MLCSI–ELMK无法始终保持最短的运算时间, 因此, 进一步提高模型运行速度是今后研究的重点.

## 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] LIU Xueyi, SONG Chunyue, LI Ping. Model complexity control of extreme learning machine using Vapnik-Chervonenkis generalization bounds [J]. *Control Theory & Applications*, 2014, 31(5): 644 – 653.
(刘学艺, 宋春跃, 李平. 基于Vapnik-Chervonenkis泛化界的极限学习机模型复杂性控制 [J]. 控制理论与应用, 2014, 31(5): 644 – 653.)

[3] ALEXANDRE E, CUADRA L, SALCEDO-SANZ S, et al. Hybridizing extreme learning machines and genetic algorithms to select acoustic features in vehicle classification [J]. *Neurocomputing*, 2015, 152(25): 58 – 68.

[4] KONG W W, ZHANG C, LIU F, et al. Irradiation dose detection of irradiated milk powder using visible and near-infrared spectroscopy and chemometrics [J]. *Journal of Dairy Science*, 2015, 96(8): 4921 – 4927.

[5] SUN L, CHEN B D, TOH K A, et al. Sequential extreme learning machine incorporating survival error potential [J]. *Neurocomputing*, 2015, 155(1): 194 – 204.

[6] BAZI Y, ALAJLAN N, MELGANI F, et al. Differential evolution extreme learning machine for the classification of hyperspectral images [J]. *IEEE Geoscience and Remote Sensing Letters*, 2014, 11(6): 1066 – 1070.

[7] BENCHERIF M A, BAZI Y, GUESSOUM A, et al. Fusion of extreme learning machine and graph-based optimization methods for active classification of remote sensing images [J]. *IEEE Geoscience and Remote Sensing Letters*, 2015, 12(3): 527 – 531.

[8] TANG J X, DENG C W, HUANG G B, et al. Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine [J]. *IEEE Transactions on Geoscience and Remote Sensing*, 2015, 53(3): 1174 – 1185.

[9] CHANG Y Q, WANG S, TIAN H X, et al. Multiple regression machine system based on ensemble extreme learning machine for soft sensor [J]. *Sensor Letters*, 2013, 11(4): 710 – 714.

[10] HE Yanlin, WANG Xiao, ZHU Qunxiong. Modeling of acetic acid content in purified terephthalic acid solvent column using principal component analysis based improved extreme learning machine [J]. *Control Theory & Applications*, 2015, 32(1): 80 – 85.
(贺彦林, 王晓, 朱群雄. 基于主成分分析—改进的极限学习机方法的精对苯二甲酸醋酸含量软测量 [J]. 控制理论与应用, 2015, 32(1): 80 – 85.)

[11] HUANG G B, LI M B, CHEN L, et al. Incremental extreme learning machine with fully complex hidden nodes [J]. *Neurocomputing*, 2008, 71(4/5/6): 576 – 583.

[12] HUANG G B, CHEN L. Enhanced random search based incremental extreme learning machine [J]. *Neurocomputing*, 2008, 71(16/17/18): 3460 – 3468.

[13] HUANG G B, CHEN L. Convex incremental extreme learning machine [J]. *Neurocomputing*, 2007, 70(16/17/18): 3056 – 3062.

[14] YANG Y M, WANG Y N, YUAN X F. Parallel chaos search based incremental extreme learning machine [J]. *Neural Processing Letters*, 2013, 37(3): 277 – 301.

[15] SHANG Z G, HE J Q. Confidence-weighted extreme learning machine for regression problems [J]. *Neurocomputing*, 2015, 148(19): 544 – 550.

[16] YU Q, MICHE Y, SÉVERIN E, et al. Bankruptcy prediction using extreme learning machine and financial expertise [J]. *Neurocomputing*, 2014, 128(27): 296 – 302.

[17] YU H, REINER P D, XIE T T. An incremental design of radial basis function networks [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2014, 25(10): 1793 – 1803.

[18] DING J L, WANG F, SUN H, et al. Improved incremental regularized extreme learning machine algorithm and its application in two-motor decoupling control [J]. *Neurocomputing*, 2015, 149(3): 215 – 223.

[19] ZHANG C M, CHEN J, XIN B. Distributed memetic differential evolution with the synergy of Lamarckian and Baldwinian learning [J]. *Applied Soft Computing*, 2013, 13(5): 2947 – 2959.

[20] ZHANG R, SONG S, WU C. A hybrid differential evolution algorithm for job shop scheduling problems with expected total tardiness criterion [J]. *Neurocomputing*, 2013, 13(3): 1448 – 1458.

[21] LEANDRO N, FERNANDO J. Learning and optimization using the clonal selection principle [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(3): 239 – 251.

[22] SHI Xuhua, QIAN Feng. Artificial immune network multi-agent optimization strategy for dynamic environment [J]. *Control Theory & Applications*, 2011, 28(7): 921 – 930.
(史旭华, 钱锋. 动态环境的人工免疫网络多Agent优化策略 [J]. 控制理论与应用, 2011, 28(7): 921 – 930.)

[23] DAI H, YANG Y, LI H, et al. Bi-direction quantum crossover-based clonal selection algorithm and its applications [J]. *Expert Systems with Applications*, 2014, 41(16): 7248 – 7258.

[24] PENG Y, LU B L. Hybrid learning clonal selection algorithm [J]. *Information Sciences*, 2015, 296(1): 128 – 146.

[25] GONG M G, JIAO L C, ZHANG L N. Baldwinian learning in clonal selection algorithm for optimization [J]. *Information Sciences*, 2010, 180(8): 1218 – 1236.

[26] YUAN Q, QIAN F, DU W. A hybrid genetic algorithm with the Baldwin effect [J]. *Information Sciences*, 2010, 180(5): 640 – 652

[27] ZHANG C, CHEN J, XIN B, Distributed memetic differential evolution with the synergy of Lamarckian and Baldwinian learning [J]. *Applied Soft Computing*, 2013, 13(5): 2947 – 2959.

[28] ZHANG R, LAN Y, HUANG G B, et al. Universal approximation of extreme learning machine with adaptive growth of hidden nodes [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2012, 23(2): 365 – 371.

[29] FRENAY B, VERLEYSEN M. Parameter-insensitive kernel in extreme learning for non-linear support vector regression [J]. *Neurocomputing*, 2011, 74(16): 2526 – 2531.

[30] GUO L, HAO J H, LIU M. An incremental extreme learning machine for online sequential learning problems [J]. *Neurocomputing*, 2014, 128(27): 50 – 58.

## 作者简介:

**王　超**　(1985−), 男, 博士研究生, 主要研究方向为智能控制和机器学习, E-mail: supper_king1018@163.com;

**王建辉**　(1957−), 女, 教授, 博士生导师, 主要研究方向为复杂控制系统的建模与控制, E-mail: wangjianhui@mail.neu.edu.cn;

**顾树生**　(1939−), 男, 教授, 博士生导师, 主要研究方向为先进控制技术等, E-mail: gushusheng@mail.neu.edu.cn;

**王　枭**　(1990−), 男, 博士研究生, 主要研究方向为智能控制和微电网, E-mail: wangxiao.owl@gmail.com;

**张宇献**　(1979−), 男, 教授, 硕士生导师, 主要研究发现为智能优化控制, E-mail: yuxian524524@163.com.