

文章编号:1007-5321(2018)02-0009-06

DOI:10.13190/j.jbupt.2017-110

# 一种可应用于嵌入式系统的低复杂度超限学习机训练方法

张克终<sup>1,2</sup>, 徐力<sup>3</sup>, 尉志青<sup>1</sup>, 黄赛<sup>1</sup>, 冯志勇<sup>1</sup>

(1. 北京邮电大学 信息与通信工程学院, 北京 100876; 2. 北京工业大学 北京未来网络科技高精尖创新中心, 北京 100124;  
3. 中国科学院 计算技术研究所, 北京 100190)

**摘要:**为解决超限学习机复杂度较高的问题,提出了一种新型的超限学习机更新策略,称为序列超限学习机。避免了复杂度较高的逆矩阵运算,而且能够应用于嵌入式系统中。序列超限学习机比各种广泛应用的机器学习分类器具有更低的计算复杂度。基于实际数据集的仿真结果表明,序列超限学习机的分类精度比传统超限学习机和其他广泛应用的分类器更高,而且具有更短的训练时间。

**关键词:**超限学习机;神经网络;分类算法;低复杂度;嵌入式系统

中图分类号:TP183

文献标志码:A

## Low Complexity Training Strategy for Extreme Learning Machine Used in Embedded System

ZHANG Ke-zhong<sup>1,2</sup>, XU Li<sup>3</sup>, WEI Zhi-qing<sup>1</sup>, HUANG Sai<sup>1</sup>, FENG Zhi-yong<sup>1</sup>

(1. School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China;  
2. Beijing Advanced Innovation Center for Future Internet Technology, Beijing University of Technology, Beijing 100124, China;  
3. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

**Abstract:** Extreme learning machine (ELM) achieves faster training speed and higher classification accuracy, compared with other widely used classifiers, such as back propagation (BP), support vector machine (SVM), spectral clustering (SC), and so forth. However, ELM suffers from some drawbacks: 1) ELM utilizes the calculation of inverse matrix for training, which cannot be adopted in the embedded system; 2) the training time of ELM increases dramatically for large-scale applications. To solve these drawbacks of ELM, a new training strategy called Sequential ELM (SELM) was proposed, which avoids the calculation of inverse matrix. Therefore, SELM can be adopted in the embedded system. It is proven that SELM achieves lower complexity than other widely used algorithms. Furthermore, simulations based on practical datasets indicate that the classification accuracy of SELM is higher than traditional ELM and other widely used classifiers with shorter training time.

**Key words:** extreme learning machine; neural network; classification; low complexity; embedded system

收稿日期:2017-06-18

基金项目:国家自然科学基金项目(61631003, 61525101);深圳市海外高层次人才创新创业专项资金“孔雀团队”项目(KQTD2015033114415450)

作者简介:张克终(1989—),男,博士生,E-mail:k.z.zhang\_cn@ieee.org;冯志勇(1971—),女,教授,博士生导师。

# 1 超限学习机研究背景

机器学习算法的突出优点是能够逼近复杂的非线性函数,而且具有并行性好、快速适应性,近年来在诸多领域得到了广泛的应用<sup>[1]</sup>,而且能够处理许多传统方法无法处理的分类问题. 反向传播算法<sup>[1]</sup>、支持向量机<sup>[2]</sup>和谱聚类<sup>[3]</sup>是最常见的机器学习分类算法. 这些算法虽然得到了广泛的应用,但也存在一些共同的缺点:1) 容易得到局部最优解;2) 算法复杂度较高. 这些不足限制了传统机器学习分类算法的应用. 所以需要提出新的分类算法以弥补传统机器学习分类算法的不足.

针对上述分类算法的缺点,黄广斌等提出了超限学习机(ELM, extreme learning machine)<sup>[1]</sup>. 传统单层神经网络的隐藏层权重是由梯度下降等方法计算得到,而超限学习机的隐藏层权重是随机产生的,这大大缩短了训练时间<sup>[1]</sup>. 然而传统超限学习机也具有如下缺点:1) 训练阶段需要计算逆矩阵,无法应用于嵌入式系统中;2) 随着训练数据量的增加,传统超限学习机的训练时间将会快速上升,无法处理大数据量的应用. 另外,超限学习机存在许多改良算法. Wang 等<sup>[4]</sup>提出了有效超限学习机,该算法选取特定的隐藏层参数,使隐藏层矩阵满秩,仿真证明其分类精度比传统超限学习机更高. Liu 等<sup>[5]</sup>提出了半监督超限学习机,在训练神经网络之前,半监督超限学习机对训练数据进行聚类,从而提高了分类精度. 然而上述超限学习机的改良算法在传统超限学习机优化方程的基础上,仅仅更改了目标函数或者增加了约束条件,却仍然需要计算逆矩阵,和传统超限学习机具有相同的缺点.

针对传统分类算法和基于超限学习机的各种算法的缺点,提出了一种低复杂度的超限学习机训练策略,称为序列超限学习机(SELM, sequential ELM). 序列超限学习机采用了不同的训练策略序列,亦即对于超限学习机的优化方程采用了不同的求解方法,避免了逆矩阵的计算,使得超限学习机可以应用于嵌入式系统中,这是许多传统机器学习分类器无法做到的. 可证明,和传统的基于机器学习的分类器相比,序列超限学习机具有更低的算法复杂度. 基于实际数据集的实验结果表明,序列超限学习机具有较高的分类精度和较短的训练时间.

## 1.1 超限学习机介绍

设  $g$  表示激活函数,  $\mathbf{w}_l$  表示第  $l$  个隐藏节点和

输入节点之间的权重,  $b_l$  表示第  $l$  个隐藏节点和输入节点之间的偏置. 其中  $\mathbf{w}_l, b_l (l=1, 2, \dots, L)$  由任意连续分布的函数产生<sup>[6-7]</sup>. 其隐藏层输出矩阵为  $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_L] \in \mathbf{R}^{N \times L}$ . 设  $h_{nl}$  是  $\mathbf{H}$  第  $(n, l)$  个元素,那么  $h_{nl} = g(\mathbf{w}_l^T \mathbf{x}_n + b_l)$ , 其中  $\mathbf{x}_n \in \mathbf{R}^{N_0}, n=1, 2, \dots, N$ , 表示输入数据. 训练超限学习机可以分为:1) 构建具有固定数目的隐藏层节点,其中隐藏层的参数随机产生;2) 固定隐藏层的节点数和参数,训练超限学习机的输出权重.

## 1.2 算法步骤

超限学习机的训练过程可表示为如下优化问题:

$$\underset{\beta}{\text{minimize}} \quad C' \|\mathbf{T} - \mathbf{H}\beta\|_F^2 + \|\beta\|_F^2 \quad (1)$$

其中  $\|\cdot\|_F$  表示 Frobenius 范数, 设输出层的权重矩阵为  $\beta = [\beta_1, \beta_2, \dots, \beta_L]^T \in \mathbf{R}^{L \times M}$ , 训练目标矩阵表示为  $\mathbf{T} = [t_1, t_2, \dots, t_N]^T \in \mathbf{R}^{N \times M}$ . 那么根据式(1)可得,超限学习机输出层的输出权重矩阵可以表示为

$$\beta = \left( \frac{1}{C'} \mathbf{I} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{T} \quad (2)$$

其中  $\mathbf{I} \in \mathbf{R}^{L \times L}$  表示单位矩阵. 在训练阶段,固定隐藏层节点数、隐藏层参数  $\mathbf{w}_l, b_l (l=1, 2, \dots, L)$  和输出权重  $\beta$ . 验证阶段隐藏层输出矩阵  $\mathbf{H}' = \{h'_{nl}\}$  可以表示  $h'_{nl} = g(\mathbf{w}_l^T \mathbf{x}'_n + b_l)$ , 其中  $\mathbf{x}'_n (n=1, 2, \dots, N)$  表示验证阶段的输入数据.

## 1.3 超限学习机算法复杂度

隐藏层输出矩阵  $\mathbf{H}$  的计算复杂度为

$$T_H = O(N_0 NL) \quad (3)$$

其中:  $N_0$  和  $N$  分别表示训练数据和训练目标的数;  $L$  表示神经网络隐藏层节点的数. 然后需要  $S_1 = NL^2 + NL - \frac{L^2}{2} + \frac{3}{2}L$  个浮点操作数去计算  $\frac{1}{C'} \mathbf{I} +$

$\mathbf{H}^T \mathbf{H}$ , 而  $\frac{1}{C'} \mathbf{I} + \mathbf{H}^T \mathbf{H}$  的逆矩阵需要  $S_2 = L^3 + L^2 + L$  个浮点操作数. 计算式(2)中的矩阵乘法需要的浮点操作数为  $S_3 = 2NL^2 - NL + 2NML - ML$ , 其中  $M$  表示类的数. 机器学习算法在实际应用中,训练数据数  $N$  往往较大,满足  $N \gg L, M, N_0$ <sup>[8]</sup>. 综上可得,设  $O$  为朗道符号,传统超限学习机的计算复杂度为

$$T_{\text{ELM}} = T_H + O(S_1 + S_2 + S_3) = O(NL^2 + NML + N_0 NL) \quad (4)$$

# 2 序列超限学习机

## 2.1 算法步骤

序列超限学习机逐行更新输出权重矩阵. 在训

训练阶段,定义误差矩阵  $\mathbf{E}_l = \mathbf{T} - \sum_{k \neq l} \mathbf{h}_k \boldsymbol{\beta}_k^T$ , 那么超限学习机优化方程可以表示为

$$\underset{\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_L]^T}{\text{minimize}} \quad C \|\mathbf{E}_l - \mathbf{h} \boldsymbol{\beta}_l^T\|_F^2 + \sum_{l=1}^L \|\boldsymbol{\beta}_l\|_2^2 \quad (5)$$

其中  $\|\cdot\|_2$  表示  $l_2$  范数. 式(5)代入 KKT 条件, 有

$$\nabla_{\boldsymbol{\beta}_l} \left( C \|\mathbf{E}_l - \mathbf{h} \boldsymbol{\beta}_l^T\|_F^2 + \sum_{l=1}^L \|\boldsymbol{\beta}_l\|_2^2 \right) = 0 \quad (6)$$

通过计算式(6),  $\boldsymbol{\beta}_l (l=1, 2, \dots, L)$  可以表示为

$$\boldsymbol{\beta}_l = (C^{-1} + \mathbf{h}_l^T \mathbf{h}_l)^{-1} \mathbf{E}_l^T \mathbf{h}_l \quad (7)$$

综合上述分析可知, 训练阶段的流程如下:

1) 初始化阶段. 随机产生隐藏层参数  $\mathbf{w}_l, b_l$ ,

设定门限值  $\varepsilon$ ;

2) 计算隐藏层权重矩阵  $\mathbf{H}$ ;

3) 对于  $l=1, 2, \dots, L$ , 有

a) 计算误差矩阵  $\mathbf{E}_l = \mathbf{T} - \sum_{k \neq l} \mathbf{h}_k \boldsymbol{\beta}_k^T$ ;

b) 通过式(7)更新输出权重的第  $l$  列  $\boldsymbol{\beta}_l$ ;

c) 如果  $C \|\mathbf{T} - \mathbf{H}\boldsymbol{\beta}\|_F^2 + \|\boldsymbol{\beta}\|_F^2 \geq \varepsilon$ , 返回 3),

如果  $C \|\mathbf{T} - \mathbf{H}\boldsymbol{\beta}\|_F^2 + \|\boldsymbol{\beta}\|_F^2 \leq \varepsilon$ , 转到 4);

4) 返回输出层权重  $\boldsymbol{\beta}$ .

验证阶段的流程如下:

1) 固定参数  $\mathbf{w}_l, b_l (l=1, 2, \dots, L)$  及输出权重  $\boldsymbol{\beta}$ ;

2) 计算隐藏层权重  $\mathbf{H}'$ ;

3) 计算输出矩阵  $\mathbf{O}' = \mathbf{H}'\boldsymbol{\beta}$ .

序列超限学习机仅需要简单地四则运算, 适用于嵌入式系统中. 由下文的分析以及实验结果可知, 序列超限学习机算法复杂度低, 分类准确性高. 这是由于传统超限学习机训练时, 所有输出层权重同时更新, 这导致某个隐藏层节点的扰动会扩散到整个输出层权重中, 而序列超限学习机的输出层权重是逐行更新的, 某个隐藏层节点的扰动只会影响与之连接的输出层权重, 不会扩散到整个输出层权重中.

## 2.2 序列超限学习机算法复杂度

序列超限学习机误差矩阵  $\mathbf{E}_l = \mathbf{T} - \sum_{k \neq l} \mathbf{h}_k \boldsymbol{\beta}_k^T$

所需浮点操作数为  $S_{E_l} = NM(L+1)$ , 其第一行  $\boldsymbol{\beta}_1$  更新的浮点操作数为  $S_{\boldsymbol{\beta}_1} = 3NM + NML - M + 2N + 2$ , 误差矩阵  $\mathbf{E}_l (l=2, 3, \dots, L)$  可以表示为  $\mathbf{E}_l = \mathbf{E}_1 - \mathbf{h}_1 \boldsymbol{\beta}_1 + \mathbf{h} \boldsymbol{\beta}_l$ , 计算复杂度是  $S_{E_l} = 4NM$ . 矩阵  $\boldsymbol{\beta}$  第  $l$  行 ( $l=2, 3, \dots, L$ ) 的浮点操作数是  $S_{\boldsymbol{\beta}_l} = 6NM - M +$

$2N + 2$ , 结合式(3)和式(4), 序列超限学习机的计算复杂度为

$$T_{\text{SELM}} = O(S_H + S_{\boldsymbol{\beta}_1} + S_{E_1} + (L-1)(S_{\boldsymbol{\beta}_l} + S_{E_l})) = O(NML + N_0NL) \quad (8)$$

下面, 给出并且证明了定理 1.

**定理 1** 如果隐藏层节点数  $L$  和类的数  $M$  满足不等式  $L \geq \frac{1}{3}(5M+4)$ , 那么序列超限学习机的计算复杂度小于传统超限学习机的计算复杂度.

将不等式  $L \geq \frac{1}{3}(5M+4)$  代入式(4)和式(8),

可知当隐藏层节点数  $L$  和类的数  $M$  满足  $L \geq \frac{1}{3}(5M+4)$  时, 有  $T_{\text{SELM}} \leq T_{\text{ELM}}$ , 定理 1 得证.  $\square$

由定理 1 可知, 如果有足够的隐藏层节点, 那么序列超限学习机比传统超限学习机的复杂度更低.

## 3 常用分类器算法复杂度对比

### 3.1 支持向量机的算法复杂度

二分类支持向量机复杂度为  $T_{\text{bin}} = O(N^3 + N_0N^2)$ , 其中  $N_0$  是输入数据的维度<sup>[9]</sup>. 多分类问题有 2 种策略: 一对多策略 (OVA, one versus all) 和一对一策略 (OVO, one versus one)<sup>[2]</sup>. OVA 解  $M(M-1)/2$  个二分类问题. 训练数据数目  $N$  通常远大于  $L, M, N_0$ <sup>[8]</sup>, OVA 复杂度为

$$T_{\text{OVA}} = O(MN^3 + N_0MN^2) = O(N^3)$$

第  $m (m=1, 2, \dots, M)$  类有  $N_m$  个数据. 则 OVO 复杂度为

$$T_{\text{OVO}} = \sum_{1 \leq i < j \leq N} O((N_i + N_j)^3) \\ (a+b)^3 \leq 4(a^3 + b^3) \text{ 对任取 } a, b \geq 0 \text{ 都成立, 所以} \\ T_{\text{OVO}} \leq \sum_{1 \leq i < j \leq M} O(N_i^3 + N_j^3) \leq O(MN^3 + N_0MN^2) \quad (9)$$

故 OVO 复杂度

$$T_{\text{OVO}} = O(MN^3 + N_0MN^2) = O(N^3)$$

### 3.2 谱聚类的算法复杂度

由于谱聚类算法 (SC, spectral clustering) 计算邻接矩阵的复杂度为  $T_{\text{adj}} = O(N_0N^2)$ <sup>[3]</sup>. 邻接矩阵奇异值分解的复杂度为  $T_{\text{SVD}} = O(N^3 + N_0L^2 + N_0^2L)$ <sup>[3]</sup>. 故谱聚类的算法复杂度为  $T_{\text{SC}} = T_{\text{adj}} + T_{\text{SVD}} = O(N^3)$ .

### 3.3 有效超限学习机的算法复杂度

下面对有效超限学习机 (EELM, effective ELM) 的

算法复杂度进行分析. 设  $[\mathbf{x}_i^T, \mathbf{t}_i^T]^T \in \mathbf{R}^d$ , 可以证明<sup>[4]</sup>, 有效超限学习机选择隐藏层权重和偏置的复杂度为  $T_{\text{extra}} = O(Nd)$ .  $[\mathbf{x}_i^T, \mathbf{t}_i^T]^T$  维度通常不会远大于  $L$ , 即  $d = O(N)$ , 所以有  $T_{\text{extra}} = O(N^2)$ . 有效超限学习机的算法复杂度为

$$T_{\text{EELM}} = T_H + T_{\text{ELM}} + T_{\text{extra}} = O(N^2 + NL^2 + NML + N_0NL)$$

(10)

3.4 半监督超限学习机的算法复杂度

下面分析计算半监督超限学习机 (USELM, unsupervised ELM)<sup>[5]</sup> 的复杂度. 设  $\mathbf{W} = \{w_{ij}\}$  表示相似度矩阵, 第  $(i, j)$  个元素为

$$w_{ij} = \exp(-2\sigma^{-2} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2)$$

其中  $\sigma$  表示输入数据  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  的标准差. 训练输出矩阵需要计算  $(\mathbf{I} + \lambda \mathbf{H}^T \mathbf{L} \mathbf{H}) \boldsymbol{\nu} = \gamma \mathbf{H}^T \mathbf{H} \boldsymbol{\nu}$  的特征值, 其中  $\mathbf{L} = \mathbf{D} - \mathbf{W}$  表示拉普拉斯矩阵<sup>[5]</sup>.  $\mathbf{D}$  是对角矩阵, 其第  $i$  行、第  $i$  列元素为  $D_{ii} = \sum_{j=1}^N w_{ij}$ .  $\lambda$  表示权衡参数.  $\gamma$  和  $\boldsymbol{\nu}$  分别为特征值和对应的特征向量. 计算  $w_{ij}$  的复杂度为  $T_{w_{ij}} = 3N_0 + 1$ , 其中  $N_0$  表示输入数据  $\mathbf{x}_n$  ( $n = 1, 2, \dots, N$ ) 的维度. 由于矩阵  $\mathbf{W}$  是对称矩阵, 计算矩阵  $\mathbf{W}$  的复杂度为

$$T_W = \frac{1}{2}N(N+1)T_{w_{ij}} = \frac{1}{2}N(N+1)(3N_0+1)$$

(11)

计算  $\gamma$  和  $\boldsymbol{\nu}$  的复杂度为  $T_{\text{eig}} = O(N^3)$ <sup>[10]</sup>. 半监督超限学习机的复杂度为

$$T_{\text{USELM}} = T_{\text{ELM}} + T_W + T_{\text{eig}} = O(N^3)$$

综上可得, 各种算法的复杂度如表 1 所示.

表 1 各种算法的复杂度

算法	复杂度
序列超限学习机	$O(NML + N_0NL)$
传统超限学习机	$O(NL^2 + NML + N_0NL)$
支持向量机(OVO)	$O(N^3)$
支持向量机(OVA)	$O(N^3)$
谱聚类	$O(N^3)$
有效超限学习机	$O(N^2 + NL^2 + NML + N_0NL)$
半监督超限学习机	$O(N^3)$

通过表 1 可以发现, 支持向量机<sup>[2]</sup>、谱聚类<sup>[3]</sup>、有效超限学习机<sup>[4]</sup>、半监督超限学习机<sup>[5]</sup> 的算法复杂度比序列超限学习机的算法复杂度高. 另外, 定理 1 已经证明, 在满足一定条件的情况下, 系列超限学习机的算法复杂度比传统超限学习机更低. 而且

定理 1 需要的条件在实际应用中很容易满足. 在实际应用中, 序列超限学习机的算法复杂度不仅比支持向量机、谱聚类等传统分类算法低, 也比传统超限学习机及其改进算法, 如有效超限学习机、半监督超限学习机等复杂度低.

4 仿真实验和分析

下面将序列超限学习机的效果与其他算法进行对比. 支持向量机和谱聚类效果比反向传播算法更好<sup>[2-3]</sup>, 所以选择支持向量机和谱聚类算法作为对比. 超限学习机的改进算法中, 选择半监督超限学习机和有效超限学习机算法. 选择基于真实数据, 将广泛应用的 UCI 数据集<sup>[11]</sup> 在 Matlab 2016b 上运行, 中央处理器主频为 2.40 GHz. 每组仿真进行了 1 000 次, 并且求取平均值. 选择 sigmoid 函数作为激活函数, 各数据集的特性如表 2 所示.

表 2 数据集特性

数据集	数据数	数据维度	类的数
Glass	214	10	7
Segment	2 310	19	7
Diabetes	442	10	2
Wine	178	13	3
Human	10 299	561	6
Wine quality	4 898	12	6
Bank	45 211	17	12
Diabetes	442	10	2
Iris	150	4	3

4.1 分类精度比较

基于表 2 中的数据集, 对各种分类器的效果进行了对比, 分类效果如表 3 所示. 可以发现, 基于超限学习机的各种算法, 即序列超限学习机、传统超限

表 3 各种分类器的分类准确度对比 %

数据集	支持 向量机	谱聚类	传统 超限 学习机	半监督 超限 学习机	有效 超限 学习机	序列 超限 学习机
Glass	68.8	61.4	66.7	60.6	72.7	86.5
Segment	67.5	13.0	66.7	69.9	68.3	73.1
Diabetes	76.6	43.5	77.5	43.7	76.8	85.9
Wine	98.4	83.3	98.6	98.7	96.6	99.4
Human	45.7	91.9	53.9	98.4	84.6	99.7
Wine Quality	99.8	13.4	99.2	99.2	99.1	99.9
Bank	89.2	49.3	98	85.1	98.1	99.9
Iris	93.3	95.7	97.6	97.4	98.3	99.7



学习机、有效超限学习机和半监督超限学习机算法的分类效果比支持向量机和谱聚类的分类效果好。而基于超限学习机的各种算法中,序列超限学习机的算法精度比其他几种基于超限学习机的算法的分类精度更高,这意味着序列超限学习机可有效地应用于分类问题中。

4.2 训练时间比较

不同分类器的训练时间如表 4 所示,序列超限学习机的训练时间比其他各种分类器的训练时间少。这验证了对各种算法复杂度的分析结果,序列超限学习机可以处理大数据量分类问题。

表 4 各种分类器的训练时间对比

数据集	支持 向量机	谱聚类	传统 超限 学习机	半监督 超限 学习机	有效 超限 学习机	序列 超限 学习机
Glass	2.088	0.022	0.008	0.018	0.013	0.006
Segment	0.139	0.074	0.011	0.026	0.022	0.008
Diabetes	0.065	0.007	0.001	0.001	0.008	0.001
Wine	0.178	0.17	0.025	0.107	0.07	0.024
Human	20.66	306.84	2.088	4.306	2.506	1.455
Wine Quality	0.912	3.466	0.042	0.145	0.062	0.038
Bank	684.45	275.67	0.25	10.245	0.318	0.101
Iris	0.407	0.013	0.003	0.011	0.007	0.002

4.3 分类效果和隐藏层节点数的关系

通过表 3 和表 4 可以发现,基于超限学习机的各种分类器(序列超限学习机、传统超限学习机、半监督超限学习机以及有效超限学习机)在分类精度和训练时间方面,比支持向量计算法、谱聚类算法、超限学习机及其改进算法都具有更好的效果。

下面分析基于超限学习机的不同分类器的性能差异。首先分析基于超限学习机各种算法的收敛性,采用 Human 数据集,分析基于超限学习机的各种分类器的分类效果和隐藏层节点数之间的关系。图 1 所示为不同分类器的分类效果和隐藏层节点数之间的关系,用以衡量不同分类器的收敛性。可以发现,序列超限学习机算法的收敛速度和半监督超限学习机相近,而且序列超限学习机的收敛速度比半监督超限学习机的收敛速度更快。实际上,针对表 3 和表 4 中其他数据集进行的测试得到了和 Human 数据集类似的结果。由于隐藏节点数大于 200 的情况下,序列超限学习机具有较好的分类效果,所以隐藏节点数选择为  $L \geq 200$ 。

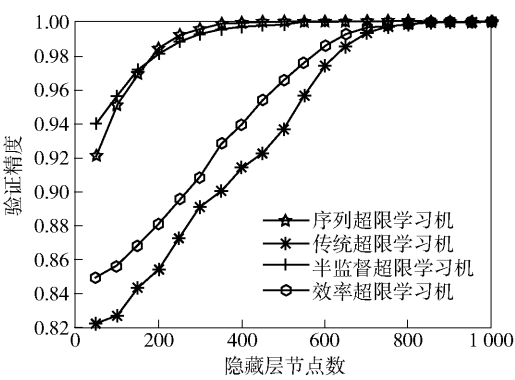


图 1 分类精度和隐藏层节点数的关系

传统超限学习机的收敛速度比序列超限学习机慢,这是因为传统超限学习机需要将整个隐藏层输出矩阵  $\beta$  视为整体进行计算,这使得某个节点上的误差和扰动会扩散到所有神经网络的隐藏层节点。

4.4 训练时间和隐藏层节点数的关系

不同算法的复杂度对比如图 2 所示。相同隐藏层节点数的情况下,序列超限学习机的训练时间比其他算法更少。这验证了上文中对于各种算法复杂度的分析结果。根据实验结果以及定理 1 的结论,为了使得序列超限学习机比其他基于超限学习机的分类器具有更少的训练时间,选择  $L \geq 200$ 。

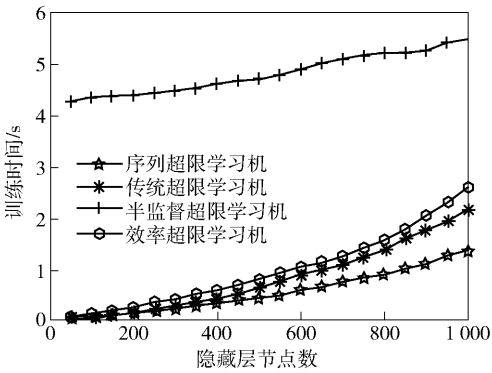


图 2 训练时间和隐藏层节点数的关系

通过图 1 和图 2 的分析,在提高分类精度的情况下,为了尽量降低训练时间,选择  $L = 200$ 。通过图 1 和图 2 的结果可见,为了达到相同的分类精度,序列超限学习机所需要的节点数比其他基于超限学习机的分类器更少。由于序列超限学习机避免了逆矩阵的计算,所以可以应用于嵌入式系统中。

5 结束语

提出了一种新型的超限学习机训练方法,称为序列超限学习机。证明了序列超限学习机的复杂度

比传统超限学习机以及各种常用分类器的复杂度更低,而且基于真实数据(UCI 数据集)的实验证明了上述结论. 序列超限学习机的输出矩阵是逐行更新的,避免了传统超限学习机的逆矩阵运算,所以能够应用于嵌入式系统中,且具有更低的复杂度. 与广泛应用的数据集相比序列超限学习机的分类效果更好,收敛速度更快,训练时间更短. 所以序列超限学习机能够得到更加广泛的研究和应用.

### 参考文献:

- [1] Tang Jiexiong, Deng Chenwei, Huang Guangbin. Extreme learning machine for multilayer perceptron[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(4): 809-821.
- [2] Chakrabarty A, Buzzard G T, Žak S H. Output-tracking quantized explicit nonlinear model predictive control using multiclass support vector machines[J]. IEEE Transactions on Industrial Electronics, 2017, 64(5): 4130-4138.
- [3] Yang Yang, Shen Fumin, Huang Zi, et al. Discrete non-negative spectral clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(9): 1834-1845.
- [4] Huang Gao, Huang Guangbin, Song Shiji, et al. Trends in extreme learning machines: a review[J]. Neural Networks, 2015, 61(Supplement C): 32-48.
- [5] Liu Tianchi, Yang Yan, Huang Guangbin, et al. Driver distraction detection using semi-supervised machine learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(4): 1108-1120.
- [6] Jin Bo, Jing Zhongliang, Zhao Haitao. Incremental and decremental extreme learning machine based on generalized inverse[J]. IEEE Access, 2017(5): 20852-20865.
- [7] Zhang Lei, Zhang David. Evolutionary cost-sensitive extreme learning machine[J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(12): 3045-3060.
- [8] 刘渭滨, 邹智元, 邢薇薇. 模式分类中的特征融合方法[J]. 北京邮电大学学报, 2017, 40(4): 1-8.  
Liu Weibin, Zou Zhiyuan, Xing Weiwei. Feature fusion methods in pattern classification[J]. Journal of Beijing University of Posts and Telecommunications, 2017, 40(4): 1-8.
- [9] Huang Kaizhu, Jiang Haochuan, Zhang Xuyao. Field support vector machines[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2017, 1(6): 454-463.
- [10] Kuang Kelvin, Lee Chi, Chen Chiaoen. An eigen-based approach for enhancing matrix inversion approximation in massive MIMO systems[J]. IEEE Transactions on Vehicular Technology, 2017, 66(6): 5480-5484.
- [11] Zhang Xianchao, Zong Linlin, Liu Xinyue, et al. Constrained clustering with nonnegative matrix factorization[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(7): 1514-1526.