

拉普拉斯多层极速学习机*

丁世飞^{1,2}, 张楠^{1,2}, 史忠植²



¹(中国矿业大学 计算机科学与技术学院, 江苏 徐州 221116)

²(中国科学院 计算技术研究所 智能信息处理重点实验室, 北京 100190)

通讯作者: 丁世飞, E-mail: dingsf@cumt.edu.cn

摘要: 极速学习机不仅仅是有效的分类器,还能应用到半监督学习中.但是,半监督极速学习机和拉普拉斯光滑孪生支持向量机一样,是一种浅层学习算法.深度学习实现了复杂函数的逼近并缓解了以前多层神经网络算法的局部最小性问题,目前在机器学习领域中引起了广泛的关注.多层极速学习机(ML-ELM)是根据深度学习和极速学习机的思想提出的算法,通过堆叠极速学习机-自动编码器算法(ELM-AE)构建多层神经网络模型,不仅实现了复杂函数的逼近,并且训练过程中无需迭代,学习效率高.把流形正则化框架引入 ML-ELM 中,提出拉普拉斯多层极速学习机算法(Lap-ML-ELM).然而,ELM-AE 不能很好地解决过拟合问题.针对这一问题,把权值不确定引入 ELM-AE 中,提出权值不确定极速学习机-自动编码器算法(WU-ELM-AE),可学习到更为鲁棒的特征.最后,在前面两种算法的基础上提出权值不确定拉普拉斯多层极速学习机算法(WUL-ML-ELM),它堆叠 WU-ELM-AE 构建深度模型,并用流形正则化框架求取输出权值.该算法在分类精度上有明显提高并且不需花费太多的时间.实验结果表明,Lap-ML-ELM 与 WUL-ML-ELM 都是有效的半监督学习算法.

关键词: 极速学习机;半监督学习;多层极速学习机;流形正则化;权值不确定

中图法分类号: TP181

中文引用格式: 丁世飞,张楠,史忠植.拉普拉斯多层极速学习机.软件学报,2017,28(10):2599-2610. <http://www.jos.org.cn/1000-9825/5128.htm>

英文引用格式: Ding SF, Zhang N, Shi ZZ. Laplacian multi layer extreme learning machine. Ruan Jian Xue Bao/Journal of Software, 2017,28(10):2599-2610 (in Chinese). <http://www.jos.org.cn/1000-9825/5128.htm>

Laplacian Multi Layer Extreme Learning Machine

DING Shi-Fei^{1,2}, ZHANG Nan^{1,2}, SHI Zhong-Zhi²

¹(School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

²(Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Extreme learning machine (ELM) not only is an effective classifier in supervised learning, but also can be applied on semi-supervised learning. However, semi-supervised ELM (SS-ELM) is merely a surface learning algorithm similar to Laplacian smooth twin support vector machine (Lap-STSV). Deep learning has the advantage of approximating the complicated function and alleviating the optimization difficulty associated with deep models. Multi layer extreme learning machine (ML-ELM) has been developed according to the idea of deep learning and extreme learning machine, which stacks extreme learning machines, based auto encoder (ELM-AE) to create a multi-layer neural network. ML-ELM not only approximates the complicated function but also avoids the need to iterate during

* 基金项目: 国家自然科学基金(61672522, 61379101); 国家重点基础研究发展计划(973)(2013CB329502)

Foundation item: National Natural Science Foundation of China (61672522, 61379101); National Key Basic Research Program of China (973) (2013CB329502)

收稿时间: 2015-06-03; 修改时间: 2015-11-23; 采用时间: 2016-05-25; jos 在线出版时间: 2016-10-11

CNKI 网络优先出版: 2016-10-12 16:26:37, <http://www.cnki.net/kcms/detail/11.2560.TP.20161012.1626.006.html>

training process, exhibiting the merits of high learning efficiency. In this article, manifold regularization is introduced into the model of ML-ELM and a Laplacian ML-ELM (Lap-ML-ELM) is put forward. Furthermore, in order to solve the over fitting problem with ELM-AE, weight uncertainty is brought into ELM-AE to form a weight uncertainty ELM-AE (WU-ELM-AE) which can learn more robust features. Finally, a weight uncertainty Laplacian ML-ELM (WUL-ML-ELM) is proposed based on the above two algorithms, which stacks WU-ELM-AE to create a deep network and uses the manifold regularization framework to obtain the output weights. Lap-ML-ELM and WUL-ML-ELM are more efficient than SS-ELM in classification and do not need to spend too much time. Experimental results show that Lap-ML-ELM and WUL-ML-ELM are efficient semi-supervised learning algorithms.

Key words: extreme learning machine; semi-supervised learning; multi layer extreme learning machine; manifold regularization; weight uncertainty

极速学习机(extreme learning machine,简称 ELM)是 Huang 等人提出的高效、简洁的单隐层前馈神经网络的学习算法^[1,2].ELM 只需要设定网络的隐层节点数,并且不需要调整网络的输入权值以及隐层的偏置,只需要学习一次就可得到唯一的最优解,所以 ELM 具有学习速度快、泛化性能好的优点.许多学者研究 ELM 在分类和回归上的应用^[3-5].邓万宇等人结合结构风险最小化理论提出了正则化极速学习机(regularized ELM,简称 RELM)^[6].Zong 等人针对非均衡数据提出了加权极速学习机(weighted ELM,简称 WELM)^[7].Huang 等人结合流形正则化提出了半监督极速学习机(semi-supervised ELM,简称 SS-ELM)^[8],给极速学习机注入了新的活力.还有不少学者从理论上证明了极速学习机的可行性^[9,10].

半监督学习是机器学习的热点之一,目前的半监督学习算法大致可划分为 3 类:基于聚类假设的算法^[11]、基于协同训练的算法^[12]、基于图正则化框架的算法^[13].基于图模型的半监督学习算法是半监督学习的一个热点,流形正则化(manifold regularization,简称 MR)是常用的图正则化框架.拉普拉斯光滑孪生支持向量机^[14]和 SS-ELM 都是基于 MR 的半监督学习算法,但它们都是浅层学习算法.深度学习是一种多隐层多层感知器的人工神经网络学习算法,实现复杂函数的逼近并缓解了以前多层神经网络算法的局部最小性问题^[15,16].2006 年,Hinton 等人首次提出深度学习的概念,提出多层自动编码器深层结构^[17].LeCun 等人提出卷积神经网络(CNNs)^[18].后来,深度置信网(DBNs)^[19]和堆叠自动编码器(SAE)^[20]相继被提出.2013 年,Huang 等人结合 ELM 和深度学习的概念提出多层极速学习机算法(multi layer-ELM,简称 ML-ELM)^[21].ML-ELM 通过堆叠极速学习机-自动编码器算法(ELM-AE)构建多层神经网络模型,它不仅具有深度学习的优点,还有较快的学习速度.

我们把流形正则化框架引入 ML-ELM,提出拉普拉斯多层极速学习机算法(Laplacian ML-ELM,简称 Lap-ML-ELM).Lap-ML-ELM 把标记样本和未标记样本一起训练,利用极速学习机-自动编码器算法(ELM-AE)逐层训练,得到隐层上的权值,最后,用流形正则化框架求取输出权值.然而,ELM-AE 不能很好地解决神经网络中常见的过拟合问题.权值不确定(weight uncertainty)是一种解决过拟合问题常用的方法^[22],其把权值矩阵看作随机变量.例如,每层权值矩阵中的随机变量可用具有相同均值与方差的高斯分布表示,这样,一个网络可以看作是许多子网络的集合.我们可以把权值不确定应用到 ELM-AE 中去,把 ELM-AE 的输入权值看作符合高斯分布的随机变量,这样可以得到一组与每个输入权值对应的输出权值,我们把这种算法称作权值不确定极速学习机-自动编码器算法(weight uncertainty ELM-AE,简称 WU-ELM-AE).我们把 WU-ELM-AE 引入 Lap-ML-ELM 算法模型,提出权值不确定拉普拉斯多层极速学习机算法(weight uncertainty Laplacian ML-ELM,简称 WUL-ML-ELM).它利用 WU-ELM-AE 的输出权值的均值逐层确定 Lap-ML-ELM 的输入权值和隐层间的权值,输出权值还是通过用流形正则化框架求取.

本文研究并提出了权值不确定拉普拉斯多层极速学习机算法.第 1 节简述传统的极速学习机和正则化极速学习机算法.第 2 节介绍多层极速学习机原理和模型结构.第 3 节详述我们提出的拉普拉斯多层极速学习机算法、权值不确定极速学习机-自动编码器算法与权值不确定拉普拉斯多层极速学习机算法.第 4 节首先用人工数据集与 UCI 数据集测试 Lap-ML-ELM 算法与 WUL-ML-ELM 算法的性能,然后把这两种算法应用到图像识别上.实验结果表明:与 SS-ELM 相比,Lap-ML-ELM 与 WUL-ML-ELM 在分类精度上有明显提高,并且无需花费太多的时间.

1 极速学习机

极速学习机是 Huang 等人提出的一种高效、简洁的单隐层前馈神经网络的学习算法^[1,2].极速学习机模型由输入层、单隐层和输出层这 3 层组成.对于 N 个训练样本 $(x_i, y_i), i=1, \dots, N, x_i \in R^j, y_i \in R^m$, 如果 ELM 模型结构具有 j 个输入层神经元、 n 个隐层神经元、 m 个输出层神经元和隐层上的激活函数是 $g(x)$, 那么隐层的输出可以表示为公式(1), 隐层的输出和输出神经元的输出数值关系可以表示为公式(2).

$$h=g(ax+b) \quad (1)$$

$$h(x_i)\beta = y_i^T, \text{其中}, i=1, 2, \dots, N \quad (2)$$

公式(2)又可以写成

$$H\beta=Y \quad (3)$$

其中,

$$H = \begin{bmatrix} g(a_1, b_1, x_1) & g(a_2, b_2, x_1) & \cdots & g(a_n, b_n, x_1) \\ g(a_1, b_1, x_2) & g(a_2, b_2, x_2) & \cdots & g(a_n, b_n, x_2) \\ \vdots & \vdots & \ddots & \vdots \\ g(a_1, b_1, x_N) & g(a_2, b_2, x_N) & \cdots & g(a_n, b_n, x_N) \end{bmatrix}, \beta = \begin{bmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_n^T \end{bmatrix}_{n \times m}, Y = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times m}.$$

用极速学习机这种方法求取隐层和输出层的输出矩阵,可分为 3 步.

- (1) 随机选择 0-1 之间的数值输入输出层和隐层之间的输入权值和隐层上的偏置;
- (2) 计算网络隐层的输出矩阵 H ;
- (3) 计算网络隐层和输出层之间的输出权值 β ;

$$\beta = H^+ Y \quad (4)$$

其中, H^+ 表示隐层输出矩阵 H 的广义逆矩阵.

ELM 训练速度快,泛化性能良好,但是没有较好的鲁棒性.邓万宇等人结合经验风险和结构风险提出了正则化极速学习机(RELM),具有更好的鲁棒性^[7].RELM 通过最小化最小二乘估计的正则化代价函数求取输出权值,从而得到如下公式:

$$\min L_{\text{RELM}} = \frac{1}{2} \|\beta\|^2 + \frac{C}{2} \|Y - H\beta\|^2 \quad (5)$$

其中, C 是调节经验风险和结构风险的参数.

如果对 L_{RELM} 求导等于 0, 我们得到:

$$\beta + CH^T(Y - H\beta) = 0 \quad (6)$$

输出权值 β 可以通过如下公式得到:

$$\beta = \begin{cases} \left(\frac{I}{C} + H^T H \right)^{-1} H^T Y, N \geq n \\ H^T \left(\frac{I}{C} + H H^T \right)^{-1} Y, N < n \end{cases} \quad (7)$$

其中, I 是单位矩阵.

2 多层极速学习机

2.1 极速学习机-自动编码器算法(ELM-AE)

自动编码器是深度学习中常用的人工神经网络模型.自动编码器是一种无监督的神经网络,其输入和输出是相同的,就是一种尽可能复现输入信号的神经网络^[17].极速学习机-自动编码器算法(ELM-AE)是 Huang 等人提出的一种新的神经网络方法,可以像自动编码器一样复现输入信号^[21].

ELM-AE 模型由输入层、单隐层和输出层这 3 层组成,假设其模型结构具有 j 个输入层神经元、 n 个隐层

神经元、 j 个输出层神经元和在隐层上的激活函数是 $g(x)$. ELM-AE 可以根据代表输入信号的隐层输出分为 3 种不同的表示方法:压缩表示、等维表示和稀疏表示.

ELM-AE 与传统的 ELM 模型最主要的区别是:ELM 是有监督算法,输出的是标签,ELM-AE 是无监督学习算法,其输出与输入一致.那么,对于 N 个训练样本 $x_i, i=1, \dots, N, x_i \in R^l$, ELM-AE 的隐层输出可以表示为公式(8),隐层输出和输出神经元输出数值关系可以表示为公式(9).

$$h = g(ax+b), \text{其中}, a^T a = I, b^T b = 1 \quad (8)$$

$$h(x_i)\beta = x_i^T, \text{其中}, i = 1, \dots, N \quad (9)$$

ELM-AE 求隐层和输出层的输出矩阵也可分为 3 步,除了隐层参数在随机生成后还要正交化外,输出权值 β 的计算方法与传统 ELM 也略有不同.

对于压缩表示与稀疏表示的 ELM-AE,输出权值 β 可由公式(10)计算得到.

$$\beta = \begin{cases} \left(\frac{I}{C} + H^T H \right)^{-1} H^T X, N \geq n \\ H^T \left(\frac{I}{C} + H H^T \right)^{-1} X, N < n \end{cases} \quad (10)$$

对于等维表示的 ELM-AE,输出权值 β 可由公式(11)计算得到.

$$\beta = H^{-1} X \quad (11)$$

2.2 多层极速学习机算法(ML-ELM)

2006 年, Hinton 等人提出了在非监督数据上建立多层神经网络的一种有效方法:首先,使用无监督学习训练各层参数;然后,有监督学习对网络进行微调^[17]. 2013 年, Huang 等人提出多层极速学习机(ML-ELM)^[21]. 与其他深度学习模型一样, ML-ELM 刚开始利用无监督学习方法 ELM-AE 对各层参数训练,但有一点不同的是, ML-ELM 不需要对网络进行微调.这样,与其他深度学习算法相比, ML-ELM 就不需要花费很长的时间对网络模型进行训练.

ML-ELM 使用 ELM-AE 逐层训练, ML-ELM 利用 ELM-AE 训练时,第 i 个隐层的输出和第 $(i-1)$ 个隐层上的输出的数值关系可用公式(12)表示.

$$H_i = g(H_{i-1}\beta_i^T) \quad (12)$$

其中, H_i 表示第 i 个隐层的输出,如果 $i=1$, 此时, H_{i-1} 表示的是输入层的输出,也就是整个网络的输入; β_i 表示 ELM-AE 对第 $(i-1)$ 个隐层和第 i 个隐层训练时的权值矩阵,此时, ELM-AE 的输入为 H_{i-1} , 隐层的神经元数目与 ML-ELM 上第 i 个隐层上的神经元数目一致. ML-ELM 上的输出权值 β 通过最小化最小二乘估计的正则化代价函数求得.

3 拉普拉斯多层极速学习机

3.1 一般的拉普拉斯多层极速学习机算法

基于图算法的正则化框架是当前半监督学习研究的热点,它利用了流形假设.它把标记和未标记的样本当作数据图的顶点,顶点的边当作权重.如果两个样本 x_i 和 x_j 接近,那么它们的标签 y_i 和 y_j 也应该彼此接近.为了实现这个假设,流形正则化框架使以下代价函数最小化:

$$\begin{aligned} L_m &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} (y_i - y_j)^2 = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} (y_i^2 - 2y_i y_j + y_j^2) \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n -2w_{ij} y_i y_j + \frac{1}{2} \sum_{i=1}^n y_i^2 \left(\sum_{j=1}^n w_{ij} \right) = \hat{Y}^T (D - W) \hat{Y} = \hat{Y}^T L \hat{Y} \end{aligned} \quad (13)$$

其中, $\hat{Y} = [\bar{y}_i \bar{y}_j]^T$, D 是对角矩阵, L 是拉普拉斯矩阵.

我们把流形正则化框架引入多层极速学习机模型,提出了拉普拉斯多层极速学习机 Lap-ML-ELM. Lap-ML-ELM 模型结构与 ML-ELM 相同,但是 Lap-ML-ELM 把标记样本和未标记样本一起训练. Lap-ML-ELM 和 ML-ELM 最大的不同之处是最后的输出权值 β 的求法不同:ML-ELM 是直接通过最小化最小二乘估计的广义正则化代价函数求得, Lap-ML-ELM 是利用流形正则化框架求得.

在半监督学习中,我们拥有少量的标记样本 $\{X_l, Y_l\} = \{x_i, y_i\}_{i=1}^l$ 和大量的未标记样本 $X_u = \{x_i\}_{i=1}^u$. Lap-ML-ELM 把标记样本和未标记样本使用 ELM-AE 对逐层进行训练. 假设 Lap-ML-ELM 有 k 个隐层,则可通过公式(12)把第 k 隐层输出 H_k 求得. 我们通过使以下代价函数最小化求出输出权值 β .

$$\min L_{\text{Lap-ML-ELM}} = \frac{1}{2} \|\beta\|^2 + \frac{C}{2} W \|Y - H_k \beta\|^2 + \frac{\lambda}{2} \text{Tr}(\beta^T H_k^T L H_k \beta) \quad (14)$$

其中, C 是外部的正则化参数, λ 是内在的正则化参数, $Y \in R^{(l+u) \times m}$ 是由标记样本的标签 Y_l 和未标记样本的标签 $Y_u=0$ 组成, W 是 $(l+u) \times (l+u)$ 的对角矩阵, 前 l 个对角元素是 $W_{ii}=1/N_i$, 其余为 0.

如果对 $L_{\text{Lap-ML-ELM}}$ 求导等于 0, 则我们得到:

$$\beta - CH^T W(Y - H_k \beta) + \lambda H_k^T L H_k \beta = 0 \quad (15)$$

输出权值 β 可以通过如下公式得到:

$$\beta = \begin{cases} (I + CH_k^T W H_k + \lambda H_k^T L H_k)^{-1} H_k^T C W Y, & l+u \geq n_k \\ H_k^T (I + C W H_k H_k^T + \lambda L H_k H_k^T)^{-1} C W Y, & l+u < n_k \end{cases} \quad (16)$$

其中, n_k 是 Lap-ML-ELM 中第 k 个隐层的节点数.

3.2 权值不确定拉普拉斯多层极速学习机算法

在 Lap-ML-ELM 学习过程中,我们需要用 ELM-AE 逐层确定输入权值和隐层间的权值,而且这些权值都可看作是实值,训练这样一个多层模型很可能出现过拟合问题. 神经网络中常见的解决过拟合问题的方法有 Dropout、权值不确定等. 权值不确定方法把神经网络中的每个权值看作一个可能值的概率分布,而不是以前的单一的固定值,这样,学习到的特征更为鲁棒. 其中,权值的波动也可以理解为训练数据的变化. 因此,我们把权值不确定应用到 ELM-AE 中去,提出权值不确定极速学习机-自动编码器 WU-ELM-AE 算法.

我们把 WU-ELM-AE 的每个输入权值看作一个可能值的概率分布. 假设这种分布是高斯分布,这样每个输入权值都是符合高斯分布的随机变量. 由于 ELM-AE 是一次学习完成, WU-ELM-AE 需要计算多次输出权值,每次的输入权值都是符合固定均值与方差的高斯分布的随机变量,我们用多次输出权值的平均值堆叠多层神经网络.

WU-ELM-AE 模型也是由输入层、单隐层和输出层这 3 层组成,如图 1 所示,其模型结构也具有 j 个输入层神经元、 n 个隐层神经元、 j 个输出层神经元和隐层上的激活函数是 $g(x)$, 并且其计算输出权值的次数为 k 次(在本文中 k 的值为 10). 与 ELM-AE 一样, WU-ELM-AE 首先随机确定输入权值的均值 μ (按照 ELM-AE 确定输入权值的方法来确定)与隐层的偏置 b . 不同的是, WU-ELM-AE 还要知道输入权值的标准差 $\sigma = \log(1 + \exp(\rho))$ (其中, ρ 在本文的取值范围为 $[0, 0.001]$). 在每次计算输出权值时,都要用均值 μ 和标准差 σ 得到一个输入权值 $w^{(c)}$: $w^{(c)} = \mu + \sigma \varepsilon^{(c)}$ (其中, $\varepsilon^{(c)} \sim N(0, 1)$, \odot 表示对位相乘). 这样,我们就可以通过公式(8)计算每次隐层的输出,然后利用公式(10)、公式(11)计算出对应的输出权值 $\beta^{(c)}$. WU-ELM-AE 算法的具体流程见算法 1.

算法 1. 权值不确定极速学习机-自动编码器(WU-ELM-AE)算法.

输入:

- 样本 $X = \{x_i \in R^j\}_{i=1}^N$;
- 隐层节点数目 n ;
- 计算输出权值的次数为 k ;
- 隐层的激活函数 $g(x)$;

输出: k 次输出权值的均值 β .

- Step 1. 随机产生输入权值的均值 μ 和隐层上的偏置 b ;
- Step 2. **for** $c=1, \dots, k$ **do**
- Step 3. 随机生成 $\varepsilon^{(c)} \sim N(0, I)$, 然后确定 ρ 的值 (ρ 在本文的取值范围是 $[0, 0.001]$), 最后确定第 c 次的输入权值 $w^{(c)}: w^{(c)} = \mu + \log(1 + \exp(\rho)) \circ \varepsilon^{(c)}$;
- Step 4. 通过式(8)计算出隐层的输出 $H^{(c)}$;
- Step 5. 最后计算出第 c 次的输出权值:

$$\beta^{(c)} = \begin{cases} H^{(c)-1} X, & j = n \\ \left(\frac{I}{C} + H^{(c)T} H^{(c)} \right)^{-1} H^{(c)T} X, & j \neq n \text{ 且 } N \geq n; \\ H^{(c)T} \left(\frac{I}{C} + H^{(c)} H^{(c)T} \right)^{-1} X, & j \neq n \text{ 且 } N < n \end{cases}$$

Step 6. **end for**

Step 7. k 次输出权值的均值 $\beta = \frac{1}{k} \sum_{c=1}^k \beta^{(c)}$.

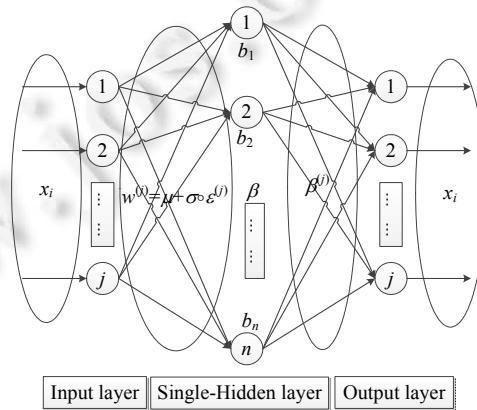


Fig.1 Model structure of WU-ELM-AE

图 1 WU-ELM-AE 的模型结构

与 ELM-AE 相比, WU-ELM-AE 可以学习到更为鲁棒的特征, 我们可以堆叠 WU-ELM-AE 构建多层神经网络, 利用 WU-ELM-AE 的输出权值的均值逐层确定多层神经网络的输入权值和隐层间的权值, 最后的输出权值是通过流形正则化框架求取, 我们把这种算法叫作权值不确定拉普拉斯多层极速学习机算法 WUL-ML-ELM.

当有少量的标记样本 $\{X_l, Y_l\} = \{x_l, y_l\}_{l=1}^l$ 和大量的未标记样本 $X_u = \{x_i\}_{i=1}^u$ 时, WUL-ML-ELM 有 k 个隐层以及每层的节点数为 $\{n_i\}_{i=1}^k$, 如图 2 所示, 我们可以利用 WU-ELM-AE 学习样本 $\{X_l, X_u\}$ 的特征, 得到输入权值和隐层间的权值 $\{w_i\}_{i=1}^k$. 假设利用 WU-ELM-AE 学习第 $(i-1)$ 个隐层 ($i-1=0$ 时, 该层为输入层) 和第 i 个隐层训练时的权值矩阵为 w_i , 此时, WU-ELM-AE 的输入为

$$H_{i-1} = \begin{cases} \{X_l, X_u\}, & i-1=0 \\ g(H_{i-2} w_{i-1}), & i-1>0 \end{cases} \quad (17)$$

若第 i 个隐层节点数为 n_i , 且 WU-ELM-AE 学习到的输出权值的均值为 β_i , 那么 $w_i = \beta_i^T$. 我们求取输出权值 β 也是通过定义公式(14)那样的代价函数得到的. 也就是说, 在确定输入权值和隐层间的权值 $\{w_i\}_{i=1}^k$ 后, 我们可以得到最后一个隐层的输出 H_k , 最后可以通过公式(16)求出输出权值 β .

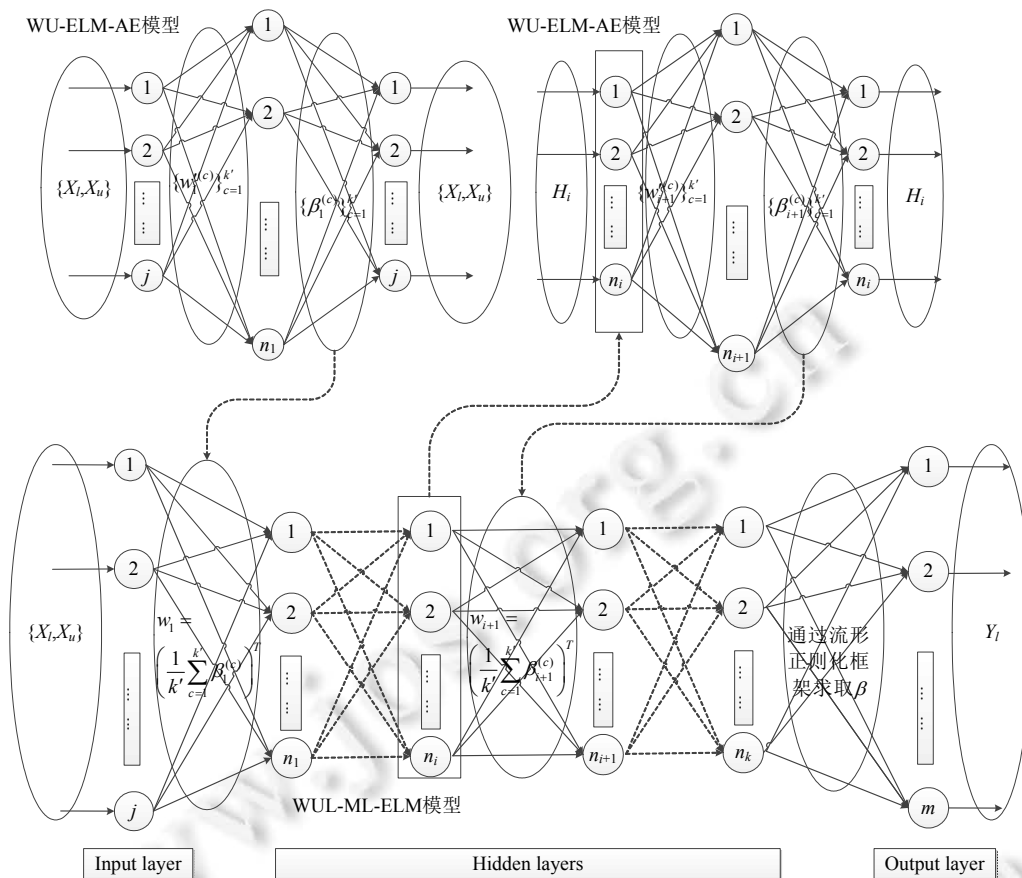


Fig.2 Model structure of WUL-ML-ELM

图 2 WUL-ML-ELM 的模型结构

WUL-ML-ELM 算法的具体流程见算法 2.

算法 2. 权值不确定拉普拉斯多层极速学习机(WUL-ML-ELM)算法.

输入:

- 标记样本 $\{X_l, Y_l\} = \{x_i \in R^j, y_i \in R^m\}_{i=1}^l$;
- 未标记样本 $X_u = \{x_i \in R^j\}_{i=1}^u$;
- 隐层的层数 k 以及每层的节点数 $\{n_i\}_{i=1}^k$;
- 隐层的激活函数均为 $g(x)$;

输出:WUL-ML-ELM 的映射函数 $f:R^j \rightarrow R^m$.

Step 1. 用样本 $\{X_l, X_u\}$ 计算拉普拉斯矩阵 L ;

Step 2. **for** $i=1, \dots, k$ **do**

Step 3. 利用 WU-ELM-AE 学习第 $(i-1)$ 个隐层和第 i 个隐层训练时的权值矩阵 w_i ;

Step 4. 计算第 i 个隐层的输出 H_i ;

$$H_i = \begin{cases} g(\{X_l, X_u\}w_i), & i=1 \\ g(H_{i-1}w_i), & i>1 \end{cases}$$

Step 5. **end for**

Step 6. 利用公式(16)计算输出权值 β :

$$\beta = \begin{cases} (I + CH_k^TWH_k + \lambda H_k^T LH_k)^{-1} H_k^T CWY, & l+u \geq n_k \\ H_k^T (I + CWH_kH_k^T + \lambda LH_kH_k^T)^{-1} CWY, & l+u < n_k \end{cases}$$

Step 7. 返回 WUL-ML-ELM 的映射函数 $f: f(x) = g(\dots g(xw_1) \dots)w_k \beta$.

4 实验与分析

为了测试所提出算法的性能,我们用人工数据集、UCI 数据集与图像数据集进行测试,并与 RELM 和 SS-ELM 的测试结果进行比较.以上几种方法都是在 Intel(R) Xeon(R) CPU E4500 0 @3.6GHZ 处理器、18G 内存、Windows7 64 位操作系统和 MATLAB2012B 的环境中运行的.我们把每个数据集分为 4 个部分:少量的标记样本数据集 l 、大量的未标记样本数据集 u 、验证数据集 v 和测试数据集 t .实验所用数据集有人工数据集(G50C)、UCI 数据集^[23](DBWorld e-mails,CNAE-9 和 Amazon Commerce reviews set)与图像数据集(UMIST^[24], COIL20^[25],Caltech 101 Silhouettes^[26]和 MNIST^[27]),其详细信息见表 1.

Table 1 Details of of benchmark data sets

表 1 基准数据集的详细信息

数据集	样本数目				样本维数	类别数目
	$ l $	$ u $	$ v $	$ t $		
G50C	50	313	50	137	50	2
DBWorld e-mails	5	38	4	17	4702	2
CNAE-9	18	882	18	108	856	9
Amazon commerce reviews set	100	925	100	375	10 000	50
UMIST	20	392	20	143	10 304	20
COIL20	40	1 000	40	360	16 384	20
Caltech 101 silhouettes	202	6 055	202	2 307	784	101
MNIST	100	10 000	100	59 800	784	10

RELM 和 SS-ELM 在所有数据集上的隐层节点数均为 2 000,Lap-ML-ELM 和 WUL-ML-ELM 在所有数据集上都是三隐层的模型结构,但在具体数据集上略有不同.Lap-ML-ELM 和 WUL-ML-ELM 在 G50C 和 DBWorld e-mails 这两个维数或样本少的数据集上的隐层结构为 40-40-200,在 CNAE-9,Caltech 101 Silhouettes 和 MNIST 这些维数适中的数据集中的隐层结构为 500-500-2000,在 Amazon Commerce reviews set,UMIST 和 COIL20 这些维数较大的数据集上的隐层结构为 1000-1000-2000.RELM,SS-ELM,Lap-ML-ELM 和 WUL-ML-ELM 的隐层上激活函数都选择 Sigmoid 函数.实验用的几种算法都有正则化参数选择,并且这些参数都从 $[10^{-4} 10^{-3} \dots 10^3 10^4]$ 选择.表 2 给出了算法在不同数据集上选取的参数,并且 Lap-ML-ELM 和 WUL-ML-ELM 在同一数据集上的选择是一致的.

Table 2 Regularization parameters' specifications of contrast algorithms in different data sets

表 2 算法在不同数据集上的正则化参数信息

数据集	RELM	SS-ELM		Lap-ML-ELM/ WUL-ML-ELM			
	C	C	λ	C_1	C_3	C	λ
G50C	10^{-3}	10^{-2}	10^{-1}	10^{-2}	10^{-1}	10^3	10^{-2}
DBWorld e-mails	10^0	10^{-4}	10^{-4}	10^0	10^{-1}	10^{-1}	10^1
CNAE-9	10^{-3}	10^{-4}	10^{-3}	10^{-2}	10^2	10^{-4}	10^3
Amazon commerce reviews set	10^{-3}	10^{-2}	10^{-4}	10^{-1}	10^3	10^1	10^{-1}
UMIST	10^{-2}	10^2	10^4	10^1	10^3	10^2	10^4
COIL20	10^2	10^4	10^{-1}	10^{-1}	10^0	10^2	10^4
Caltech 101 silhouettes	10^{-3}	10^{-2}	10^{-4}	10^{-3}	10^1	10^2	10^{-2}
MNIST	10^1	10^{-1}	10^{-2}	10^{-3}	10^0	10^4	10^3

在表中, C 与 λ 分别是计算输出权值时的外部与内在正则化参数, C_1 与 C_3 分别是 Lap-ML-ELM/WUL-ML-ELM 上输入层和第 1 层隐层与第 2 层隐层和最后一层之间的连接权值.由于是有随机初始化权值的步骤,算法的每次结果会有小幅波动,因此,将 4 种算法在每个数据集上都运行 10 次,并计算其错误率的平均值,统计结果

见表 3.

Table 3 Error rates of algorithms on different datasets

表 3 算法在测试数据集上的性能比较

算法 数据集	#.	RELM (%)	SS-ELM (%)	Lap-ML-ELM (%)	WUL-ML-ELM (%)
G50C	<i>u</i>	9.30±0.80	7.76±0.58	7.76±1.57	7.03±1.06
	<i>v</i>	9.60±1.58	10.00±2.49	7.60±2.07	6.80±1.69
	<i>t</i>	6.35±1.29	5.62±1.09	5.04±0.94	4.89±1.24
DBWorld e-mails	<i>u</i>	39.21±0.83	17.89±4.77	23.12±12.08	20.97±9.05
	<i>v</i>	40.00±12.91	37.50±24.30	32.75±14.52	30.00±15.89
	<i>t</i>	51.76±2.48	17.65±6.20	12.65±14.47	10.24±11.48
CNAE-9	<i>u</i>	39.68±1.12	38.91±0.77	30.81±0.86	30.53±0.89
	<i>v</i>	43.33±4.38	38.89±3.70	39.44±1.76	39.44±1.76
	<i>t</i>	35.39±2.19	34.44±1.11	27.67±1.48	26.11±1.70
Amazon commerce reviews set	<i>u</i>	72.04±0.55	72.94±1.05	71.32±1.70	71.16±2.47
	<i>v</i>	72.10±2.28	70.00±2.67	67.20±3.36	69.00±3.13
	<i>t</i>	68.56±1.95	68.56±2.11	64.72±2.61	64.08±2.22
UMIST	<i>u</i>	53.90±2.47	54.34±1.66	51.96±0.99	47.93±1.53
	<i>v</i>	45.50±6.85	43.00±5.87	39.50±4.38	37.00±4.22
	<i>t</i>	49.86±1.75	49.16±1.58	49.16±2.01	44.34±2.19
COIL20	<i>u</i>	28.60±0.93	27.18±1.00	23.41±1.18	22.84±1.99
	<i>v</i>	33.25±4.72	32.50±4.08	24.50±1.58	23.75±2.12
	<i>t</i>	33.86±1.02	32.28±0.74	26.64±1.19	25.94±1.42
Caltech 101 silhouettes	<i>u</i>	67.30±0.64	61.90±0.67	63.64±0.50	63.15±0.51
	<i>v</i>	36.78±0.84	35.10±1.03	33.37±1.28	33.51±0.96
	<i>t</i>	61.10±0.88	60.87±1.34	56.29±1.48	55.48±1.14
MNIST	<i>u</i>	27.73±0.38	25.40±0.41	20.20±0.24	20.02±0.23
	<i>v</i>	28.20±0.92	26.40±1.17	21.00±1.05	20.10±1.29
	<i>t</i>	29.48±0.45	27.42±0.35	21.37±0.24	21.25±1.24

从表 2 中可以看出,WUL-ML-ELM 几乎在所有数据集上都能得到令人满意的实验结果.RELM 和 SS-ELM 的唯一区别就是:由于目标函数的不同,SS-ELM 的目标函数中增加了流行正则化项,从而导致输出权值的计算方法有所不同.

从表 2 中可以看出:除了 Amazon Commerce reviews set 与 UMIST 的未标记样本数据集外,SS-ELM 在其他数据集上的错误率都比 RELM 要低.这表明,流行正则化项可以有效地描述样本间的结构信息,提高了算法的准确率.与 SS-ELM 相比,Lap-ML-ELM 利用 ELM-AE 堆叠了多层网络结构,并且除了输出权值的计算方法与 SS-ELM 一致之外,其他的权值都是利用 ELM-AE 算法学习到的,这样,隐层的输出可以更好地描述样本,而不是 SS-ELM 那样的输入权值是随机确定的.

从表 2 中也可以看出:除了 DBWorld e-mails 和 Caltech 101 Silhouettes 的未标记样本数据集与 CNAE-9 的验证数据集外,Lap-ML-ELM 在其他数据集上的错误率都比 SS-ELM 要低.这表明,利用 ELM-AE 算法堆叠的多层网络结构可以学习到较好的样本的代表特征,提高了算法的准确率.然而,ELM-AE 不能很好地解决神经网络中常见的过拟合问题,WU-ELM-AE 就是针对这一问题提出的.WUL-ML-ELM 与 Lap-ML-ELM 唯一的不同之处就是:Lap-ML-ELM 利用 ELM-AE 堆叠了多层网络结构,而 WUL-ML-ELM 使用了 WU-ELM-AE.

同样,从表 2 中也可以看出:除了 Amazon Commerce reviews set 的验证数据集外,WUL-ML-ELM 在其他数据集上的错误率基本上都比 Lap-ML-ELM 要低.这表明,WUL-ML-ELM 可以学习到更为鲁棒的特征,提高了算法的准确率.

我们可以说 Lap-ML-ELM 与 WUL-ML-ELM 都是有效的半监督学习算法.为了进一步比较算法的运行效率,表 4 给出了每种算法在各个数据集上的训练时间.

Table 4 Training time of algorithms on different datasets

表 4 算法在不同数据集上的训练时间

数据集 \ 算法	RELM (s)	SS-ELM (s)	Lap-ML-ELM (s)	WUL-ML-ELM (s)
G50C	0.023 4	0.341 6	0.243 4	0.489 8
DBWorld e-mails	0.480 5	0.592 8	0.104 5	0.819 0
CNAE-9	0.101 4	2.405 5	5.388 3	25.846 2
Amazon commerce reviews set	1.174 7	12.427 0	37.331 0	139.096 7
UMIST	0.819 0	3.316 6	27.175 4	85.089 2
COIL20	0.224 6	56.147 9	74.250 2	229.165 5
Caltech 101 silhouettes	1.386 8	14.611 1	20.517 3	86.870 7
MNIST	0.117 0	100.004 4	128.685 2	367.008 0

从表 4 中可以看到,WUL-ML-ELM 的训练时间最长.这是因为该算法不仅要计算拉普拉斯矩阵,还要使用 WU-ELM-AE 算法学习除了输出权值以外网络中所有的连接权值,并且 WU-ELM-AE 需要计算多次它的输出权值才能学习到更为鲁棒的特征.

从表 4 中我们还可以得到以下结论.

- 1) RELM 与 SS-ELM 在同一数据集上的网络结构是一致的,但是 SS-ELM 训练时间比 RELM 要长,这表明,流行正则化项虽然可以提高算法的准确率,但它是以计算效率为代价的,并且拉普拉斯矩阵的计算复杂度主要与样本数目有关,样本越多,训练时间越长;
- 2) 除了 UMIST 数据集外,Lap-ML-ELM 的训练时间与 SS-ELM 基本是一个数量级的.这表明,Lap-ML-ELM 虽然利用 ELM-AE 堆叠了多层网络结构,但这在大多数情况下并没有影响计算效率;
- 3) WUL-ML-ELM 的训练时间是 Lap-ML-ELM 的 2~5 倍,这表明,WUL-ML-ELM 虽然可以学习到更为鲁棒的特征,但同时也消耗了一定的训练时间.

总体来说,WUL-ML-ELM 虽然比 Lap-ML-ELM 要消耗更多的训练时间,但其也提高了算法的准确率,与提高的精度相比,其所增加的那部分时间代价也是可以接受的.

5 结束语

半监督极速学习机是一种浅层学习算法,在此基础上,我们提出了拉普拉斯多层极速学习机算法(Lap-ML-ELM),它利用 ELM-AE 训练多层结构,然后利用流形正则化框架计算输出权值.Lap-ML-ELM 在训练过程中无需迭代,学习效率高,不仅提升了算法的精度,而且在大多数情况下没有影响计算效率,进而,我们针对堆叠极速学习机-自动编码器算法(ELM-AE)中的过拟合问题,提出了权值不确定极速学习机-自动编码器算法(WU-ELM-AE),它需要计算多次输出权值,每次的输入权值都是符合固定均值与方差的高斯分布的随机变量,我们用多次输出权值的平均值堆叠出权值不确定拉普拉斯多层极速学习机算法(WUL-ML-ELM).WUL-ML-ELM 虽然需要消耗一定的时间代价,但却有效地提高了算法的准确率.

实验结果表明,Lap-ML-ELM 与 WUL-ML-ELM 都是有效的半监督学习算法.但是,Lap-ML-ELM 与 WUL-ML-ELM 还有一些不足之处.

- 当未标记样本太多时,计算拉普拉斯矩阵花费的时间太长;
- 模型结构不好选择.

针对以上问题的研究,就是今后的工作.

References:

- [1] Huang GB, Zhu QY, Siew CK. Extreme learning machine: A new learning scheme of feedforward neural networks. In: Proc. of the Int'l Joint Conf. on Neural Networks (IJCNN 2004). Budapest: IEEE Press, 2004. 25-29. [doi: 10.1109/IJCNN.2004.1380068]
- [2] Huang GB, Zhu QY, Siew CK. Extreme learning machine: Theory and applications. Neurocomputing, 2006,70(1):489-501. [doi: 10.1016/j.neucom.2005.12.126]

- [3] Cambria E, Huang GB, Kasun LLC, Zhou HM, Vong CM, Lin J, Yin JP, Cai ZP, Liu Q, Li K, Leung VCM, Feng L, Ong YS, Lim A, Akusok A, Lendasse A, Corona F, Nian R, Miche Y, Gastaldo P, Zunino R, Decherchi S, Yang XF, Mao KZ, Oh BS, Jeon J, Toh KA, Beng A, Teoh J, Kim J, Yu HC, Chen YQ, Liu JF. Extreme learning machines [trends & controversies]. *IEEE Intelligent Systems*, 2013,28(6):30–59. [doi: 10.1109/MIS.2013.140]
- [4] Huang GB. An insight into extreme learning machines: Random neurons, random features and kernels. *Cognitive Computation*, 2014,6(3):376–390. [doi: 10.1007/s12559-014-9255-2]
- [5] Ding SF, Xu XZ, Nie R. Extreme learning machine and its applications. *Neural Computing and Applications*, 2014,25(3):549–556. [doi: 10.1007/s00521-013-1522-8]
- [6] Deng WY, Zheng QH, Chen L, Xu XB. Research on extreme learning of neural networks. *Chinese Journal of Computers*, 2010, 33(2):279–287 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016 .2010.00279]
- [7] Zong WW, Huang GB, Chen YQ. Weighted extreme learning machine for imbalance learning. *Neurocomputing*, 2013,101:229–242. [doi: 10.1016/j.neucom.2012.08.010]
- [8] Huang G, Song SJ, Gupta JND, Wu C. Semi-Supervised and unsupervised extreme learning machines. *IEEE Trans. on Cybernetics*, 2014,44(12):2405–2417. [doi: 10.1109/TCYB.2014.2307349]
- [9] Chen H, Peng JT, Zhou YC, Li LQ, Pan ZB. Extreme learning machine for ranking: Generalization analysis and applications. *Neural Networks*, 2014,53:119–126. [doi: 10.1016/j.neunet.2014.01.015]
- [10] Lin SB, Liu X, Fang J, Xu ZB. Is extreme learning machine feasible? A theoretical assessment (Part II). *IEEE Trans. on Neural Networks and Learning Systems*, 2015,26(1):21–34. [doi: 10.1109/TNNLS.2014.2336665]
- [11] Tison C, Nicolas JM, Tupin F, Maître H. A new statistical model for Markovian classification of urban areas in high-resolution SAR images. *IEEE Trans. on Geoscience and Remote Sensing*, 2004,42(10):2046–2057. [doi: 10.1109/TGRS.2004.834630]
- [12] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: *Proc. of the 11th Annual Conf. on Computational Learning Theory*. ACM Press, 1998. 92–100. [doi: 10.1145/279943.279962]
- [13] Belkin M, Niyogi P, Sindhvani V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 2006,7:2399–2434.
- [14] Chen WJ, Shao YH, Hong N. Laplacian smooth twin support vector machine for semi-supervised classification. *Int'l Journal of Machine Learning and Cybernetics*, 2014,5(3):459–468. [doi: 10.1007/s13042-013-0183-3]
- [15] Bengio Y, Lecun Y. Scaling learning algorithms toward AI. *Large-Scale Kernel Machines*, 2007,34(5):1–41.
- [16] Bengio Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2009,2(1):1–127. [doi: 10.1561/2200000006]
- [17] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science*, 2006,313(5786):504–507. [doi: 10.1126/science.1127647]
- [18] LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1989,1(4):541–551. [doi: 10.1162/neco.1989.1.4.541]
- [19] Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. *Neural Computation*, 2006,18(7):1527–1554. [doi: 10.1162/neco.2006.18.7.1527]
- [20] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 2010,11:3371–3408.
- [21] Kasun LLC, Zhou HM, Huang GB, Vong CM. Representational learning with extreme learning machine for big data. *IEEE Intelligent System*, 2013,28(6):31–34.
- [22] Blundell C, Cornebise J, Kavukcuoglu K. Weight uncertainty in neural networks. In: *Proc. of the 32nd Int'l Conf. on Machine Learning*. Lille, 2015. 1613–1622. <http://proceedings.mlr.press/v37/blundell15.html>
- [23] Filannino M, Liu Z, Cuturi M. UCI machine learning repository. 2007. <http://archive.ics.uci.edu/ml/>
- [24] Graham DB, Allinson NM. Characterizing virtual eigensignatures for general purpose face recognition. In: Wechsler H, Phillips PJ, Bruce V, Fogelman-Soulie F, Huang TS, eds. *Face Recognition: From Theory to Applications*. NATO ASI Series F, Computer and Systems Sciences, 1998,163:446–456. [doi: 10.1007/978-3-642-72201-1_25]

- [25] Nene SA, Nayar SK, Murase H. Columbia object image library (COIL-20). Technical Report, CU-CS-005-96, Columbia University, 1996.
- [26] Marlin BM, Swersky K, Chen B, de Freitas N. Inductive principles for restricted Boltzmann machine learning. In: Proc. of the 13th Int'l Conf. on Artificial Intelligence and Statistics. 2010,9:509-516. <https://www.cs.ox.ac.uk/publications/publication7466-abstract.html>
- [27] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based learning applied to document recognition. Proc. of the IEEE, 1998, 86(11):2278-2324. [doi: 10.1109/5.726791]

附中文参考文献:

- [6] 邓万宇,郑庆华,陈琳,许学斌.神经网络极速学习方法研究.计算机学报,2010,33(2):279-287. [doi: 10.3724/SP.J.1016.2010.00279]



丁世飞(1963—),男,山东青岛人,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为人工智能,机器学习,数据挖掘.



史忠植(1941—),男,研究员,博士生导师,CCF 会士,主要研究领域为智能科学,机器学习,数据挖掘.



张楠(1991—),男,博士生,主要研究领域为人工智能,机器学习,数据挖掘.