Discriminative Globality-Locality Preserving Extreme Learning Machine for Image Classification

Yonghe Chu[a], Hongfei Lin[a*], Liang Yang[a], Yufeng Diao[a], Dongyu Zhang[a], Shaowu Zhang[a], Xiaochao Fan[a], Chen Shen[a], Bo Xu[a], Deqin Yan[b]

[a] Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China

[b] School of Computer and Information Technology, Liaoning Normal University, Dalian 116081, China

Abstract: Extreme learning machines (ELM) have been widely used in classification due to their simple theory and good generalization ability. However, there remains a major challenge: it is difficult for ELM algorithms to maintain the manifold structure and the discriminant information contained in the data. To address this issue, we propose a discriminant globality-locality preserving extreme learning machine (DGLELM) in this paper. In contrast to ELM, DGLELM not only considers the global discriminative structure of the dataset but also makes the best use of the local discriminative geometry information. DGLELM optimizes the projection direction of the ELM output weights by maximizing the inter-class dispersion and minimizing the intra-class dispersion for global and local data. Experiments on several widely used image databases validate the performance of DGLELM. The experimental results show that our approach achieves significant improvements over state-of-the-art ELM algorithms.

Keywords: extreme learning machine; manifold structure; discriminating information; local discriminative geometry

1 Introduction

Huang et al. [1-2] proposed an extreme learning machine (ELM) based on the structure of single-hidden layer feedforward networks (SLFNs). ELMs randomly generate the input weights and offset values of the hidden layer nodes. Only the output weights of all parameters need to be analysed and determined, thus transforming the traditional neural network solving process into a linear model. The literature [3-4] showed that the ELM randomly generates the input weight and offset value of the hidden layer node through analysis, and the output weight maintains the universal approximation ability of SLFNs. Compared with the traditional neural network method based on the gradient idea, it has higher efficiency and can obtain an optimal solution close to the global solution. The literature [5-6] noted that ELM has better generalization ability than support vector machine (SVM) [7] and its improved algorithm. ELM has been successfully applied to many applications for its speed, efficiency and easy implementation [8-14].

Researchers have studied ELM in different ways and proposed various improvements. Huang et al. [15] further studied the general approximation ability of ELM. Wang [16] et al. proposed a local generalization error model for the problem of ELM generalization ability, and the researchers also compared ELM with other classification algorithms. In the applications, researchers have proposed various improvements based on the ELM algorithm for the specific needs of specific problems. Zong et al. [17] studied the defects of ELM in imbalanced and missing data and proposed a weighted extreme learning machine for imbalanced learning. Huang [18] et al., Yang [19] and others applied ELM to semi-supervised learning using manifold learning techniques. Zhang [20] et al. proposed the evolutionary cost-sensitive extreme learning machine (ECELM) by analysing the influence of data on the ELM model from the perspective of the cost sensitivity coefficient. Researchers used a priori information on the data to apply ELM models to improve the generalization performance of ELM. Zhang [21] et al., Silvestre [22] and others used the prior information of the data to improve ELM. With the continuous improvement of its theory and the expansion of its application range, ELM has been applied to biomedical information [23], computer vision [24], data mining [25], and control and robotics [26].

The above mentioned improvements have enhanced the generalization performance and application range of ELM. However, one major drawback of the above approaches is that it is difficult for them to maintain the manifold structure of the data and the discriminative information contained in the data. In recent years, to effectively reveal the local geometry contained in the sample, the literature [27-30] proposed several representative manifold learning methods: isometric mapping (Isomap), locally linear embedding (LLE), Laplacian eigenmaps (LE), and locality preserving projections (LPP). Based on the idea of manifold learning and linear

[*] Corresponding author E-mail: hflin@dlut.edu.cn.

discriminant analysis (LDA) [31], researchers have proposed various improvements for ELM to maintain the geometry and discriminative information of the data. Iosifidis et al. introduced an intra-class divergence matrix and a global divergence matrix into the ELM model using the principle of linear discriminant analysis to explore the geometry of data and proposed the minimum class variance extreme learning machine (MCVELM) [32] and the minimum variance extreme learning machine (MVELM) [33]. On this basis, Iosifidis et al. introduced the Laplacian feature mapping framework into the ELM model with subspace learning (SL) criteria and proposed a graph-embedded extreme learning machine algorithm (GEELM) [34] to optimize the network output weights of ELM. Liu et al. proposed the robust discriminative extreme learning machine (RDELM) [35] to account for the lack of discriminant information between data sample classes in the MCVELM algorithm. The RDELM algorithm not only accounts for the intra-class discrimination information of the data samples but also considers the inter-class discrimination information of the data samples. Based on the consistency property of data, which enforces similar samples to share similar properties, Peng et al. proposed a discriminative graph regularized extreme learning machine (GELM)[36]. GELM combines the discriminant information of the data samples to construct a Laplacian eigenmaps (LE) [29] structure, which is introduced into the ELM algorithm as a regular term. In addition, Peng et al. proposed a discriminative manifold extreme learning machine (DMELM) based on local intra-class discriminative information and local inter-class discriminant information and data geometric structure information [37]. However, the above algorithm does not comprehensively consider the global geometry and local combination structure of the data.

In this paper, to better use the global geometry and local geometry of the data and the discriminative information contained in the data, we introduce the basic principles of LDA and discriminant LPP(DLPP) [38] into ELM and propose a discriminative globality-locality preserving extreme learning machine (DGLELM). First, we use the basic principles of LDA to construct the global class divergence and inter-class divergence, reflecting the global geometry and global discriminant information of the data. Second, the idea of discriminant LPP is used to construct local intra-class divergence and local inter-class divergence, which reflects the local manifold structure and local discriminant information of the data. Finally, the global intra-class divergence and global inter-class divergence, as well as the local intra-class divergence and local inter-class divergence, are introduced into the ELM model to optimize the projection direction of the ELM output weight. It is worth mentioning that the literature [39-40] found that the intra-class divergence matrix, inter-class divergence and total divergence matrix in linear discriminant analysis (LDA) have the ability to maintain the global identification information and the global geometry of the training samples. The experiments on the image datasets show that the proposed algorithm with a discriminative globality-locality preserving extreme learning machine improves the performance of the ELM algorithm.

Our approach has two major contributions:

(1) Considering the global geometry and local geometry of the data, we introduced the basic principles of LDA and discriminant LPP into the ELM model.

(2) Introducing the discriminative information into the ELM model, we construct a global discriminative structure of the dataset and a local discriminative geometry of the dataset.

The rest of the paper is organized as follows. Section 2 introduces the basic principles of the ELM algorithm. We provide the framework of the DGLELM algorithm in section 3. Section 4 analyses the experimental results. Our summaries are drawn in section 5.

2 Extreme Learning Machine

Based on the single hidden layer feedforward neural network, Huang et al. [1] proposed the ELM algorithm, which avoids the gradient-based approach of traditional neural networks. Therefore, the solution speed has been greatly improved, and no excessive human intervention is required during the solution process. For N different samples $K=\{(\boldsymbol{x}_i,\boldsymbol{t}_i)\mid \boldsymbol{x}_i\in R^d,\boldsymbol{t}_i\in R^m,i=1,2,\ldots,N\}$, where $\boldsymbol{x}_i=(x_{i1},x_{i2},\ldots,x_{id})^T$ and $\boldsymbol{t}_i=(t_{i1},t_{i2},\ldots,t_{im})^T$ .The ELM model with L hidden layer node activation function $g(x)$ is as follows:

$$\sum_{j=1}^{L} \boldsymbol{\beta}_j g_j(\boldsymbol{x}_i) = \sum_{j=1}^{L} \boldsymbol{\beta}_j g(\boldsymbol{a}_j \cdot \boldsymbol{x}_i + b_j) = \boldsymbol{o}_i \tag{1}$$

where $\boldsymbol{a}_j = (a_{j1}, a_{j2}, \ldots, a_{jd})$ is the input weight vector connecting the j th hidden layer node with the input nodes, $\boldsymbol{\beta_j} = (\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm})^T$ is the output weight vector connecting the j th hidden layer node and the output node. b_j is the offset value of the j th hidden layer node. $\boldsymbol{a}_j \cdot \boldsymbol{x_i}$ represents the inner product of $\boldsymbol{a_j}$ and $\boldsymbol{x_i}$. $\boldsymbol{o}_i = (o_{i1}, oi_2, \ldots, o_{im})^T$ is the network output corresponding to sample $\boldsymbol{x_i}$. To integrate all data samples, equation (1) can be rewritten as follows:

$$\boldsymbol{\beta^T H} = \boldsymbol{T} \tag{2}$$

where $h(\boldsymbol{x}_i) = (g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_i + b_1), g(\boldsymbol{a}_2 \cdot \boldsymbol{x}_i + b_2), \ldots, g(\boldsymbol{a}_L \cdot \boldsymbol{x}_i + b_L))^T$ is the output vector of the hidden layer with respect to $\boldsymbol{x}_i$, $\boldsymbol{H}$ is the network hidden layer node output matrix, $\boldsymbol{\beta}$ is the output weight matrix, and $\boldsymbol{T}$ is the expected output matrix:

$$\boldsymbol{H} = \begin{bmatrix} g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_1 + b_1) & \cdots & g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_N + b_1) \\ \vdots & \vdots & \vdots \\ g(\boldsymbol{a}_L \cdot \boldsymbol{x}_1 + b_L) & \cdots & g(\boldsymbol{a}_L \cdot \boldsymbol{x}_N + b_L) \end{bmatrix}_{L \times N} = [h(\boldsymbol{x}_1), h(\boldsymbol{x}_2), \ldots, h(\boldsymbol{x}_N)] \tag{3}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1{}^T \\ \vdots \\ \boldsymbol{\beta}_L{}^T \end{bmatrix}_{L \times m}, T = [\boldsymbol{t}_1, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_1]_{m \times N} \tag{4}$$

We solve equation (2) by the least squares solution:

$$\tilde{\boldsymbol{\beta}}_{ELM} = \arg\min_{\boldsymbol{\beta}} \left\| \boldsymbol{\beta^T H} - \boldsymbol{T} \right\|^2 = (\boldsymbol{H}^T)^+ \boldsymbol{T}^T \tag{5}$$

where $(\boldsymbol{H}^T)^+$ is the generalized inverse matrix of the matrix.

To improve the stability and generalization ability of ELM, Huang proposed an ELM with equality optimization constraints[41](Equality optimization constraints based ELM). The ELM objective function of the equality optimization constraint can be written as:

$$J_{RELM} = \min_{\boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|^2 + \frac{1}{2} C \left\| \boldsymbol{\beta^T H} - \boldsymbol{T} \right\|^2 \tag{7}$$

where C is a penalty coefficient on the training errors. We have $\partial J_{RELM} / \partial \boldsymbol{\beta} = 0$, and $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta}_{RELM} = \left(\frac{\boldsymbol{I}}{C} + \boldsymbol{HH}^T \right)^{-1} \boldsymbol{HT}^T \tag{8}$$

In summary, the ELM algorithm solving process can be summarized as follows:

1) Initialize the training sample set;

2) Randomly specify the network input weight $\boldsymbol{a}_j$ and the offset value b_j, $j = 1, 2, \cdots, L$;

3) Calculate the hidden layer node output matrix $\boldsymbol{H}$ by the activation function;

4) Calculate the output weight matrix $\boldsymbol{\beta}$ according to equation (5) or equation (8).

3 Discriminative Globality-Locality Preserving Extreme Learning Machine

To solve the problem that ELM cannot maintain data geometry and discriminate information well, we propose a discriminative globality-locality preserving extreme learning machine (DGLELM) in this section. First, we introduce the basic principles of the LDA algorithm and discriminant LPP algorithm. Second, a manifold regularization framework is defined based on the LDA algorithm and the discriminant LPP algorithm. Finally, the optimization problem formulation of the DGLELM is given by using the manifold regularization framework.

3.1 Notations

Given dataset $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in R^{D \times N}$, D is the data dimension, $N = N_1 + N_2 + \cdots + N_C$ is the number of samples, and C is the total number of categories for the dataset. The dataset $\boldsymbol{X}$ is expressed as $\boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_N]\in R^{d\times N}$ in the low dimension. The projection transformation matrix is defined as $\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_d]\in R^{D\times d}$ to reduce the data from the original space R^D to the low-dimensional subspace R^d. The symbol $\|\cdot\|_2$ denotes the l_2 -norm. $Tr(\cdot)$ denotes the trace operator, and $N_k(\cdot)$ denotes the k nearest neighbours operator.

3.2 Linear Discriminant Analysis

The main idea of LDA is to enhance the global class discrimination after projection, which maximizes the rank of scattering matrices $\boldsymbol{S}_B$ between classes by minimizing the rank of scattering matrices $\boldsymbol{S}_W$ within the class to find a subspace to distinguish between the different categories. According to the derivation of LDA in the literature [31], $\boldsymbol{S}_W$ and $\boldsymbol{S}_B$ are defined as follows:

$$\boldsymbol{S}_W=\sum_{i=1}^{C}\sum_{j=1}^{N_i}\left(\boldsymbol{x}_j^{(i)}-\boldsymbol{m}^{(i)}\right)\left(\boldsymbol{x}_j^{(i)}-\boldsymbol{m}^{(i)}\right)^T \tag{10}$$

$$\boldsymbol{S}_B=\sum_{i=1}^{C}N_i\left(\boldsymbol{m}^{(i)}-\boldsymbol{m}\right)\left(\boldsymbol{m}^{(i)}-\boldsymbol{m}\right)^T \tag{11}$$

In equations (10) and (11), N_i is the number of samples in the i th class, $\boldsymbol{x}_j^{(i)}$ is the j th sample in the i th class. $\boldsymbol{m}^{(i)}$ is the mean vector of the i th class; $\boldsymbol{m}$ represents the mean vector of all samples, C is the total number of categories in the dataset. LDA has the following optimization guidelines:

$$L_{LDA}=\max_{\boldsymbol{U}}\frac{Tr\left(\boldsymbol{U}^T\boldsymbol{S}_B\boldsymbol{U}\right)}{Tr\left(\boldsymbol{U}^T\boldsymbol{S}_W\boldsymbol{U}\right)} \tag{12}$$

Equation (12) finds the projection transformation matrix by the Lagrange multiplier method; then, the corresponding low-dimensional expression of $\boldsymbol{X}$ is obtained by $\boldsymbol{Y}=\boldsymbol{U}^T\boldsymbol{X}$. To solve the singularity problem of the intra-class scattering matrix, we use the maximum marginal criterion [42] to rewrite equation (12) as follows:

$$L_{MMC}=\max_{\boldsymbol{U}}Tr\left(\boldsymbol{U}^T(\boldsymbol{S}_B-\boldsymbol{S}_W)\boldsymbol{U}\right) \tag{13}$$

3.3 Discrimination Locality Preserving Projections

It is prone to overlap in the sample points after LPP algorithm projection; DLPP[38] introduces discriminant information into the LPP algorithm by maximizing the inter-class divergence and minimizing the intra-class divergence. The feature subspace obtained after projection is more advantageous for classification purposes. The DLPP optimization equation is as follows:

$$L_{DLPP}=\min\frac{\sum_{c=1}^{C}\sum_{i,j=1}^{n_c}(\boldsymbol{y}_i^c-\boldsymbol{y}_j^c)^2W_{ij}^c}{\sum_{i,j}^{C}(\boldsymbol{m}_i-\boldsymbol{m}_j)^2B_{ij}} \tag{14}$$

C is the total number of categories in the dataset, n_c is the total number of samples in the c th class, $\boldsymbol{y}_i^c$ is the i th sample in the c th class, m_i and m_j are separately the mean weight vectors for the i th class and the j th class, $\boldsymbol{m}_i=\frac{1}{n_i}\sum_{k=1}^{n_i}\boldsymbol{y}_k^i$, $\boldsymbol{m}_j=\frac{1}{n_j}\sum_{k=1}^{n_j}\boldsymbol{y}_k^j$. n_i is the total number of samples in the i th class, n_j is the total number of samples in the j th class. Both W_{ij}^c and B_{ij} are weight matrices. We can rewrite (14) into the following form by $\boldsymbol{Y}=\boldsymbol{U}^T\boldsymbol{X}$:

$$
\begin{aligned}
&\frac{1}{2}\sum_{c=1}^{C}\sum_{i,j=1}^{n_c}(\boldsymbol{y}_i^c-\boldsymbol{y}_j^c)^2 W_{ij}^c=\frac{1}{2}\sum_{c=1}^{C}\sum_{i,j=1}^{n_c}(\boldsymbol{U}^T\boldsymbol{x}_i^c-\boldsymbol{U}^T\boldsymbol{x}_j^c)^2 W_{ij}^c\\
&=\sum_{c=1}^{C}\left(\sum_{i=1}^{n_c}\boldsymbol{U}^T\boldsymbol{x}_i^c D_{ii}^c(\boldsymbol{x}_i^c)^T\boldsymbol{U}-\sum_{i,j=1}^{n_c}\boldsymbol{U}^T\boldsymbol{x}_i^c W_{ij}^c(\boldsymbol{x}_j^c)^T\boldsymbol{U}\right)\\
&=\sum_{c=1}^{C}\boldsymbol{U}^T\boldsymbol{X}_c(\boldsymbol{D}_c-\boldsymbol{W}_c)\boldsymbol{X}_c^T\boldsymbol{U}=\boldsymbol{U}^T\boldsymbol{X}(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{X}^T\boldsymbol{U}\\
&=\boldsymbol{U}^T\boldsymbol{X}\boldsymbol{L}_1\boldsymbol{X}^T\boldsymbol{U}
\end{aligned}
\tag{15}
$$

where $\boldsymbol{W}_c$ is the weight matrix between any two samples in the c th class, and its elements are $W_{ij}^c=\exp(-\left\|x_i^c-x_j^c\right\|^2/t)$; where t is a parameter that can be determined empirically; $\boldsymbol{D}_c$ is a diagonal matrix, and its elements are the column sum of $\boldsymbol{W}_c$.

$$D_{ii}^c=\sum\nolimits_j W_{ij}^c \ . \ \boldsymbol{X}=[\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots,\boldsymbol{X}_C], \qquad \boldsymbol{L}_1=\boldsymbol{D}-\boldsymbol{W} \ ;$$

$$\boldsymbol{D}=\begin{bmatrix} D_1 & 0 & 0\\ 0 & \ddots & 0\\ 0 & 0 & \mathrm{D_C}\end{bmatrix} \qquad \boldsymbol{W}=\begin{bmatrix} W_1 & 0 & 0\\ 0 & \ddots & 0\\ 0 & 0 & \mathrm{W_C}\end{bmatrix}$$

$$
\begin{aligned}
&\frac{1}{2}\sum_{i,j=1}^{C}(\boldsymbol{m}_i-\boldsymbol{m}_j)^2 B_{ij}=\frac{1}{2}\sum_{i,j=1}^{C}\left(\frac{1}{n_i}\sum_{k=1}^{n_i}\boldsymbol{y}_k^i\text{ - }\frac{1}{n_j}\sum_{k=1}^{n_j}\boldsymbol{y}_k^j\right)^2 B_{ij}\\
&=\frac{1}{2}\sum_{i,j=1}^{C}\left(\frac{1}{n_i}\sum_{k=1}^{n_i}\boldsymbol{U}^T\boldsymbol{x}_k^i\text{ - }\frac{1}{n_j}\sum_{k=1}^{n_j}\boldsymbol{U}^T\boldsymbol{x}_k^j\right)^2 B_{ij}\\
&=\frac{1}{2}\sum_{i,j=1}^{C}\left(\boldsymbol{U}^T\left(\frac{1}{n_i}\sum_{k=1}^{n_i}\boldsymbol{x}_k^i\right)\text{ - }\boldsymbol{U}^T\left(\frac{1}{n_j}\sum_{k=1}^{n_j}\boldsymbol{x}_k^j\right)\right)^2 B_{ij}\\
&=\frac{1}{2}\sum_{i,j=1}^{C}\left(\boldsymbol{U}^T\boldsymbol{f}_i\text{ - }\boldsymbol{U}^T\boldsymbol{f}_j\right)^2 B_{ij}\\
&=\sum_{i=1}^{C}\boldsymbol{U}^T\boldsymbol{f}_i E_{ii}\boldsymbol{f}_i^T\boldsymbol{U}-\sum_{i,j}^{C}\boldsymbol{U}^T\boldsymbol{f}_j B_{ii}\boldsymbol{f}_j^T\boldsymbol{U}\\
&=\boldsymbol{U}^T\boldsymbol{F}(\boldsymbol{E}-\boldsymbol{B})\boldsymbol{F}^T\boldsymbol{U}=\boldsymbol{U}^T\boldsymbol{F}\boldsymbol{L}_2\boldsymbol{F}^T\boldsymbol{U}
\end{aligned}
\tag{16}
$$

where $\boldsymbol{F}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\ldots,\boldsymbol{f}_c]$, $\boldsymbol{f}_i$ is the mean face in the i th class, i.e., $\boldsymbol{f}_i=\frac{1}{n_i}\sum_{k=1}^{n_i}\boldsymbol{x}_k^i$, $\boldsymbol{L}_2=\boldsymbol{E}-\boldsymbol{B}$, where $\boldsymbol{B}$ is the between-class weight matrix, and its components are defined as: $B_{ij}=\exp(-\left\|f_i-f_j\right\|^2/t)$, where t is a parameter that can be determined empirically; $\boldsymbol{E}$ is a diagonal matrix, and its elements are the column sum of $\boldsymbol{B}$, $E_{ii}=\sum\nolimits_j B_{ij}$.

Substituting equations (15) and (16) into equation (14), we obtain the objective function of DLPP as:

$$L_{DLPP}=\min\frac{\boldsymbol{U}^T\boldsymbol{X}\boldsymbol{L}_1\boldsymbol{X}^T\boldsymbol{U}}{\boldsymbol{U}^T\boldsymbol{F}\boldsymbol{L}_2\boldsymbol{F}^T\boldsymbol{U}} \tag{17}$$

Equation (17) obtains the projection transformation matrix $\boldsymbol{U}$ by the Lagrange multiplier method and then obtains the corresponding low-dimensional expression of $\boldsymbol{X}$ through $\boldsymbol{Y}=\boldsymbol{U}^T\boldsymbol{X}$. To solve the singularity problem of the inter-class scattering matrix, we use the maximum marginal criterion [42] to rewrite formula (12) into the following form:

$$L_{DLPP} = \min(\boldsymbol{U}^T \boldsymbol{X}\boldsymbol{L}_1\boldsymbol{X}^T\boldsymbol{U} - \boldsymbol{U}^T\boldsymbol{F}\boldsymbol{L}_2\boldsymbol{F}^T\boldsymbol{U}) \tag{18}$$

3.4 DGLELM

Based on manifold learning [43], it is assumed that data samples that are close together have the same marginal distribution P_x ; if two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are close to each other, then the conditional probabilities $P(\boldsymbol{y} \mid \boldsymbol{x}_1)$ and $P(\boldsymbol{y} \mid \boldsymbol{x}_2)$ should be similar as well. The above assumption is widely used in the smoothness assumption in machine learning [43]. The manifold regularization framework can be obtained based on the LE algorithm [29]. Due to the difficulty the LE algorithm demonstrates on new test data [30], the LPP algorithm is proposed. However, the LPP algorithm can lead to overlapping sample points after projection, which will make the misclassification phenomenon easier in the classification process . The DLPP algorithm can make full use of sample class information to construct local intra-class divergence and local inter-class divergence so that similar samples are more aggregated and different types of sample points are far from each other. Thus, the projected sample points are more favourable for classification. Therefore, this paper constructs a manifold regularization framework based on the DLPP algorithm and LDA and introduces it into the ELM model. We introduce equations (13) and (18) into the ELM model and replace $\boldsymbol{U}^T$ with $\boldsymbol{\beta}^T$. The DGLELM objective function is as follows:

$$\begin{aligned} J_{DGLELM} &= \min_{\beta} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C_1\left\|\boldsymbol{\beta}^T H - T\right\|^2 + \frac{1}{2}C_2\left(L_{DLPP} - L_{MMC}\right) \\ &= \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C_1\left\|\boldsymbol{\beta}^T \boldsymbol{H} - \boldsymbol{T}\right\|^2 + C_2 Tr\left((\boldsymbol{\beta}^T\boldsymbol{H}\boldsymbol{L}_1\boldsymbol{H}^T\boldsymbol{\beta} - \boldsymbol{\beta}^T\boldsymbol{F}\boldsymbol{L}_2\boldsymbol{F}^T\boldsymbol{\beta})_{DLPP} - \left(\boldsymbol{\beta}^T(\boldsymbol{S}_B - \boldsymbol{S}_W)\boldsymbol{\beta}\right)_{MMC}\right) \\ &= \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C_1\left\|\boldsymbol{\beta}^T \boldsymbol{H} - \boldsymbol{T}\right\|^2 + C_2 Tr\left(\boldsymbol{\beta}^T\left(\boldsymbol{H}\boldsymbol{L}_1\boldsymbol{H}^T - \boldsymbol{F}\boldsymbol{L}_2\boldsymbol{F}^T - \boldsymbol{S}_B + \boldsymbol{S}_W\right)\boldsymbol{\beta}\right) \end{aligned} \tag{19}$$

where $Tr\left((\boldsymbol{\beta}^T\boldsymbol{H}\boldsymbol{L}_1\boldsymbol{H}^T\boldsymbol{\beta} - \boldsymbol{\beta}^T\boldsymbol{F}\boldsymbol{L}_2\boldsymbol{F}^T\boldsymbol{\beta})_{DLPP} - \left(\boldsymbol{\beta}^T(\boldsymbol{S}_B - \boldsymbol{S}_W)\boldsymbol{\beta}\right)_{MMC}\right)$ is the manifold regularization term, $\|\boldsymbol{\beta}\|^2$ is the l_2 -norm regularization term, C_1 is a penalty constant on the training errors, and C_2 is a penalty constant on the manifold regularization term.

We make $\partial J_{DGLELM} / \partial \boldsymbol{\beta} = 0$, and $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta} = \left[\frac{\boldsymbol{I}}{C_1} + \boldsymbol{H}\boldsymbol{H}^T + \frac{C_2}{C_1}\left(\boldsymbol{H}\boldsymbol{L}_1\boldsymbol{H}^T - \boldsymbol{F}\boldsymbol{L}_2\boldsymbol{F}^T - \boldsymbol{S}_B + \boldsymbol{S}_W\right)\right]^{-1}\boldsymbol{H}\boldsymbol{T}^T \tag{20}$$

```
DGLELM Algorithm.
Input: Initialize the training sample set $K = \{(\boldsymbol{x}_i, \boldsymbol{t}_i) \mid \boldsymbol{x}_i \in R^d, \boldsymbol{t}_i \in R^m, i = 1,2,\dots,N\}$, activation function $g(x)$, the number of hidden layer nodes is $L$ , and the regularization parameters are $C_1$ and $C_2$ ;
Step 1: Randomly specify the network input weights $\boldsymbol{a}_j$ and offset value $b_j$ , $j = 1,2,\cdots,L$ ;
Step 2: Calculate the hidden layer node output matrix $\boldsymbol{H}$ ;
Step 3: Calculate $L_{MMC}$ and $L_{DLPP}$ by equations (13) and (18) to construct a manifold regularization framework and introduce it to equation (19);
Step 4: Calculate the output weight matrix from equation (20) $\boldsymbol{\beta}$ .
```

4 Experiments

To verify the validity of the proposed algorithm DGLELM, we use the image dataset for experiments. A detailed description of the image dataset is given in Table 1. We compare the results of the DGLELM experiments with the experimental results of ELM, RELM, GEELM [34], RDELM [35], GELM [36] and so on.

4.1. Databases

The Yale B[1] dataset contains 2,414 images of 38 people, each containing 55 images. The image shows the state of each person's happiness, sadness, normality, etc. in different situations. In our

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

experiments, the images were cropped and scaled to an image size of 32×32.

The ORL[1] dataset contains 400 images of 40 people, each containing 10 images. We selected images of different expressions under different lighting conditions. In our experiments, the images were cropped and scaled to an image size of 32×32.

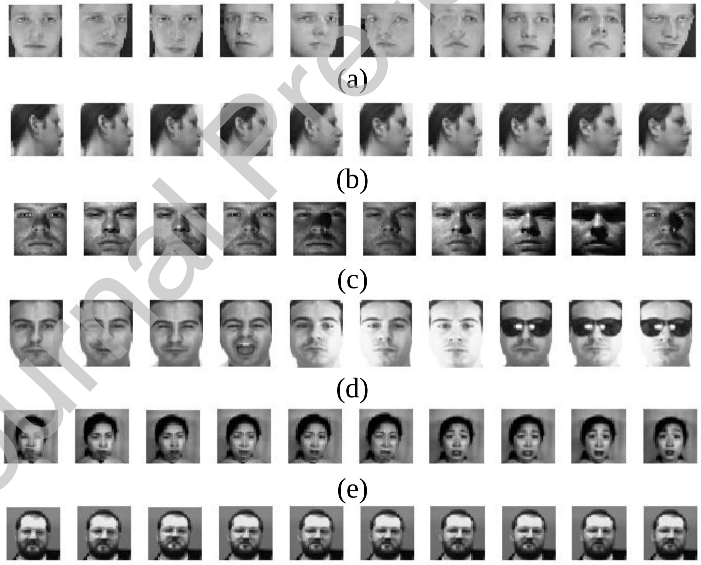
The AR face database[44] dataset contains 4,000 images of 126 people (70 men and 56 women). We use a subset that contains 2600 grey images of 100 human subjects and have selected images of different expressions under different lighting conditions. In our experiments, the images were cropped and scaled to an image size of 50×40.

The JAFFE (Japan Female Facial Expression) expression database[45] was jointly established by the Japanese ATR Human Information Processing Research Laboratory and the Department of Psychology of Kyushu University, Japan. The expression library contains 213 facial expressions of 10 Japanese women, each with 7 expressions (6 basic expressions plus neutral expressions: happy, sad, surprised, angry, disgusted, fearful, and neutral); each expression has 2-4 frames. To overcome the effects of noise such as illumination and face, the image is first rotated, trimmed, scaled, etc., and the image size is 32×32.

The UMIST[2] dataset contains 575 images of 20 people. The images show the state of each person in different situations, such as happiness, sadness, and normality. In our experiments, the images were cropped and scaled to an image size of 32×32.

The Face Recognition Data, University of Essex, UK (Faces94)[3] contains 3,059 images of 153 people (female (20), male (113), male staff (20)). The image shows the state of each person's happiness, sadness, normality, etc. in different situations. In our experiments, we first change the image from colour to a greyscale image and then cut and scale the image. The final image size is 32×32.

Some samples of different image datasets will be given in Figure 1. Table 1 provides specific details of the different image datasets.



(a)

(b)

(c)

(d)

(e)

(f)

Image samples used in the experiments. (a) ORL face database. (b) UMIST face database. (c) Yale B face database. (d) AR face database. (e) JAFFE face database. (e) Faces94 face database.

Table 1 Dataset description

Datasets	Dim	Samples	Classes
ORL	1024	400	10
UMIST	1024	575	20
Yale B	1024	2414	38
AR	2000	2600	100
JAFFE	1024	213	10
Faces94	1024	3059	153

[2] http://images.ee.umist.ac.uk/danny/database.html

[3] https://cswww.essex.ac.uk/mv/allfaces/faces94.html

4.2 Experimental Settings

For all the ELM algorithms, we select the sigmoid function as the activation function, and the number of hidden layer nodes is selected from $L = \{500,1000,1500\}$. For different ELM algorithms, we use the threefold cross-validation and grid search methods to identify the optimal parameters. For RELM, $C \in \{2^{-10},2^{-9},\ldots,2^{9},2^{10}\}$. For GEELM, RDELM, GELM, and DGLELM, the penalty parameter C_1 and the regularization parameter C_2 are taken, and the values are $C_1 \in \{2^{-10},2^{-9},\ldots,2^{9},2^{10}\}$, $C_2 \in \{2^{-10},2^{-9},\ldots,2^{9},2^{10}\}$. For the ORL dataset, we randomly select $l_{ORL} = \{2,3,4,5\}$ images per subject for training and the rest for testing. Similarly, for the remaining datasets, we set $l_{UMIST} = \{2,4,6,8\}$, $l_{YaleB} = \{15,25,35,45\}$, $l_{AR} = \{11,14,17,20\}$, $l_{JAFFE} = \{2,3,4,5\}$, and $l_{\mathrm{Faces94}} = \{2,3,4,5\}$. All experiments were randomly selected from the training set and the test set to run 10 times, and then the average of the 10 runs was calculated as the final recognition result. The recognition results of different ELM algorithms on the image dataset are shown in Tables 2, 3, 4, 5, 6 and 7. All experiments were conducted with MATLAB 2015b, on a computer with an Intel(R) Core(TM) 3.40 GHz CPU and 8 GB RAM.

Table 2 classification accuracy (mean± std%) of different methods with different numbers of training samples on the ORL database

Algorithms	ORL			
	2 Train	3 Train	4 Train	5 Train
ELM	78.88±2.64	84.82±2.71	89.87±2.27	92.10±2.59
RELM	79.19±1.79	85.93±1.94	89.92±2.34	92.55±1.72
GEELM	81.13±2.57	87.46±2.27	91.79±2.07	93.15±1.84
GELM	83.13±2.99	87.93±2.20	90.79±2.39	93.25±1.69
RDELM	80.16±1.47	86.54±2.47	90.71±1.74	93.45±1.21
DGLELM	**85.50±2.44**	**91.57±1.23**	**94.83±1.32**	**97.10±0.94**

Table 3 classification accuracy (mean± std%) of different methods with different numbers of training samples on the UMIST database

Algorithms	UMIST			
	2 Train	4 Train	6 Train	8 Train
ELM	81.07±3.50	91.25±1.63	95.30±1.62	96.31±2.05
RELM	80.18±2.84	91.71±2.43	95.55±2.06	96.69±1.62
GEELM	82.29±1.90	92.42±1.85	95.45±1.62	97.13±1.75
GELM	81.82±2.79	92.17±1.77	95.55±2.24	96.75±1.86
RDELM	82.07±3.13	92.29±1.35	95.45±1.98	96.94±1.75
DGLELM	**82.89±1.94**	**92.63±2.30**	**95.90±1.60**	**98.25±2.24**

Table 4 classification accuracy (mean± std%) of different methods with different numbers of training samples on the Yale B database

Algorithms	Yale B			
	15 Train	25 Train	35 Train	45 Train
ELM	86.49±0.79	92.35±0.87	94.49±0.73	96.84±1.10
RELM	89.41±1.14	95.49±0.86	97.59±0.50	97.87±0.61
GEELM	90.61±1.27	95.99±0.86	97.78±0.39	98.42±0.68
GELM	90.44±1.06	95.83±0.85	97.83±0.48	98.24±0.53
RDELM	90.62±1.25	95.98±0.78	97.68±0.34	98.21±0.75
DGLELM	**92.11±1.14**	**97.04±0.75**	**98.36±0.36**	**98.68±0.48**

Table 5 classification accuracy (mean± std%) of different methods with different numbers of training samples on the AR database

Algorithms	AR			
	11 Train	14 Train	17 Train	20 Train

ELM	92.87±0.89	95.41±0.50	97.03±0.48	97.73±0.82
RELM	94.61±0.63	95.87±0.37	97.47±0.49	97.62±0.67
GEELM	96.33±0.39	97.16±0.58	97.67±0.67	98.08±0.80
GELM	95.83±0.57	97.00±0.44	97.79±0.65	98.05±0.68
RDELM	95.72±0.67	96.66±0.30	97.70±0.56	98.28±0.92
DGLELM	**97.22±0.45**	**98.19±0.36**	**98.82±0.37**	**98.97±0.37**

Table 6 classification accuracy (mean± std%) of different methods with different numbers of training samples on the JAFFE database

Algorithms	JAFFE			
	2 Train	3 Train	4 Train	5 Train
ELM	91.28±3.78	94.41±1.74	97.44±1.23	98.07±1.46
RELM	90.72±5.52	94.94±3.40	96.88±1.21	97.40±1.65
GEELM	91.94±3.96	95.12±2.94	96.25±1.14	99.20±0.98
GELM	91.33±3.76	94.94±3.16	97.25±1.22	97.83±1.73
RDELM	91.28±4.90	95.53±1.57	96.94±1.65	98.00±1.09
DGLELM	**96.00±2.66**	**98.12±1.88**	**99.38±0.66**	**99.67±0.47**

Table 7 classification accuracy (mean± std%) of different methods with different numbers of training samples on the Faces94 database

Algorithms	Faces94			
	2 Train	3 Train	4 Train	5 Train
ELM	93.96±1.07	95.91±0.66	96.72±0.66	97.20±0.43
RELM	96.23±0.54	97.59±0.39	97.86±0.23	98.06±0.37
GEELM	96.53±0.38	97.62±0.37	97.73±0.22	97.96±0.30
GELM	96.79±0.46	97.64±0.37	97.85±0.30	98.10±0.41
RDELM	96.44±0.44	97.61±0.37	97.86±0.30	98.03±0.36
DGLELM	**96.98±0.38**	**98.07±0.39**	**98.37±0.20**	**98.48±0.31**

4.3 Experimental Analysis

Table 2-7 shows the recognition results of different ELM algorithms on the face image dataset, where the best results are shown in boldface. From Table 2-7, we found some interesting points as detailed below.

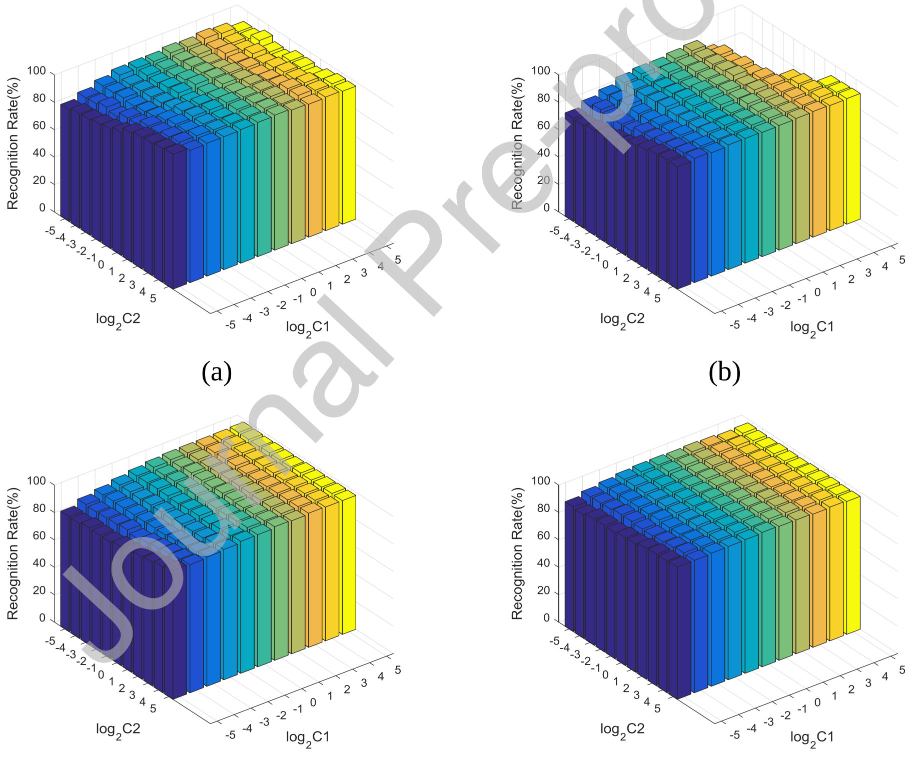
1) In most cases, it can be seen from the recognition results of Table 2-7 on all six datasets that the recognition result of the DGLELM algorithm proposed in this paper is better than other ELM algorithms. The above results verify that the DGLELM algorithm can maintain the global geometry and local geometry of the data, as well as the discriminant information contained in the data.
2) From Table 2-7, we can also see that as the number of data samples increases, the recognition rate of different ELM algorithms increases. This is because the RDELM algorithm introduces global intra-class divergence and global inter-class divergence into the ELM model and maintains the global geometry of the data and the discriminant information of the data.
3) The GEELM algorithm constructs a graph embedding framework to introduce it into the ELM model, thereby maintaining several geometric structures and discriminating information of the data samples. GELM and DGLELM are based on the idea of manifold learning: if data samples $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are drawn from the same marginal distribution P_x, if two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are close to each other, then the conditional probabilities $P(\boldsymbol{y}\mid\boldsymbol{x}_1)$ and $P(\boldsymbol{y}\mid\boldsymbol{x}_2)$ should be similar as well. GEELM, GELM, and DGLELM all maintain local structural information for the data. Table 2-7 shows that the recognition structure of the GEELM and GELM algorithms is better than that of the ELM and RELM algorithms, which verifies that the GEELM and GELM algorithms can effectively maintain the local manifold structure of the data.
4) As shown in Table 2-7, the identification structures of RDELM, GEELM, and GELM are very close. The recognition structure of the DGLELM algorithm is significantly better than other ELM algorithms, and the recognition structure of the RELM algorithm is superior to the ELM

algorithm. The necessity of minimizing the output weight in the original ELM can improve the recognition result.

Overall, we consider the global discriminative structure of the dataset and the local discriminative geometry of the dataset. By maximizing the global dispersion and minimizing the global intra-class dispersion while minimizing local intra-class dispersion and maximizing local inter-class divergence, we construct a manifold regularization framework and introduce it into the ELM model to optimize the output weight of the ELM. Thus, DGLELM can achieve state-of-the-art performance.

4.4 Parameter Sensitiveness Analysis for DGLELM

In this section, we examine the parameter sensitivity of DGLELM. There are two regularization parameters to be tuned in our proposed framework. The number of hidden layer nodes in the experiment is selected from $L = \{500,1000,1500\}$. Penalize the parameter C_1 and the regularization parameter C_2, which are $C_1 \in \{2^{-5}, 2^{-4}, \ldots, 2^{4}, 2^{5}\}$, $C_2 \in \{2^{-5}, 2^{-4}, \ldots, 2^{4}, 2^{5}\}$. We use image datasets to experimentally observe the effect of different parameter values on the final recognition rate where Tr(#) is the number of training samples per class. Figure 2 shows the recognition results of DGLELM on different face image datasets under different parameter values. In Figure 2, we can see that in the datasets ORL, Yale B, AR, Faces94, DGLELM is not sensitive to the values of the regularization parameters. On the datasets JAFFE and UMIST, DGLELM is not sensitive to the values of the regularization parameters when C_1 is not very large.

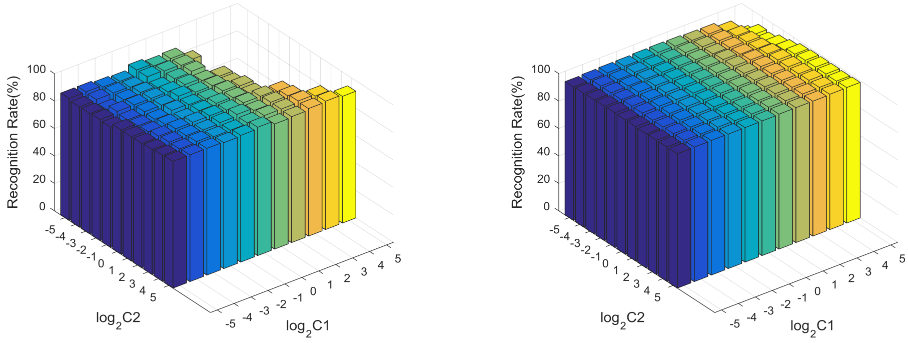


(a)

(b)

(c)

(d)



(e) (f)

Figure 2. On the image dataset, the influence of different values of penalty parameters C_1 and regularization parameter C_2 on the DGLELM algorithm (%). (a) ORL (#Tr 5). (b) UMIST (#Tr 4). (c) Yale B (#Tr 35). (d) AR (#Tr 14). (e) JAFFE(#Tr 2). (f) Faces94(#Tr 4).

4.5 Efficiency Comparison

In this subsection, we analyse the computational complexity of different ELM algorithms. All experiments are conducted using a MATLAB 2015b computer with an Intel(R) Core(TM) 3.40 GHz CPU and 8 GB RAM. We perform experiments on six image datasets. For the ORL dataset, we randomly select $l_{ORL} = 2$ images per subject for training and the rest for testing. Similarly, for the remaining datasets, we set $l_{UMIST} = 4$, $l_{YaleB} = 15$, $l_{AR} = 11$, $l_{JAFFE} = 2$, and $l_{\text{Faces94}} = 2$. The training time and test time of different ELM algorithms on the image dataset are given in Table 8. As seen in Table 8, ELM shows the superiority in CPU running time, while RELM, GEELM, GELM, RDELM and DGLELM have poor CPU running time performance, which is due to the introduction of regular terms. In Table 8, we find that DGLELM requires more running time than other evolutionary ELM methods, which is attributed to the combination of the global discriminative structure of the dataset and the local discriminative geometry of the dataset used in M-ELM. Integrating Table 2-8, we analyse the performance of the proposed algorithm DGLELM from the classification recognition rate and time efficiency. It can be seen in Table 2 and Table 3 that DGLELM has good classification ability and is superior to other ELM algorithms, and it can be seen from Table 4 that GELM does not have a considerable advantage on time overhead, and it is not the least time-consuming algorithm on some datasets, but the classification speed is fast and has good classification performance based on ELM. DGLELM can be used as an effective classifier in pattern recognition.

Table 8 Runtime comparisons of different methods (measured in seconds)

Alg.	ORL 2Train		UMIST 4Train		Yale B 15Train		AR 11Train		JAFFE 2Train		Faces94 2Train	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
ELM	0.1094	0.0625	0.0781	0.0625	0.5938	0.2301	1.7344	0.4063	0.0781	0.0781	0.2656	0.3750
RELM	1.5000	0.0625	1.3906	0.0625	1.5938	0.2188	1.8125	0.3594	7.9531	0.0156	1.5156	0.3906
GEELM	1.2344	0.1563	1.2506	0.1094	1.9531	0.5313	2.4688	0.6250	8.3281	0.0625	1.3750	1.2656
GELM	1.4219	0.0625	1.3906	0.0781	4.6250	0.2500	14.3438	0.3594	7.7500	0.0625	2.2656	0.3906
RDEM	2.7344	0.0625	2.2031	0.0625	2.9375	0.2500	5.8438	0.3281	8.2031	0.0625	7.4063	0.4063
DGLELM	4.6094	0.1719	2.6719	0.0625	5.2500	0.6250	8.8125	0.5938	8.7500	0.0625	12.0156	1.2969

5. Conclusions

In this paper, considering that the existing ELM algorithms cannot maintain the global geometry and local geometry and the discriminant information of the data, we exploit the idea of manifold learning to construct the global discriminative structure and the local discriminative geometry and introduce it into the ELM model to optimize the output weight of the ELM. We validate the GELLM algorithm by experiments using an image dataset. The experimental results verify the effectiveness of GLELM.

Conflicts of Interest

We declare that we have no conflicts of interest.

Acknowledgement

This work is partially supported by a grant from the Natural Science Foundation of China (No. 61632011, 61572102,61702080, 61602079) and the Fundamental Research Funds for the Central Universities (NO. DUT18ZD102, DUT17RC(3)016).

References

[1] G.-B. Huang , Q.-Y. Zhu , C.-K. Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. in: Proceedings of IEEE International Joint Conference on Neural Networks. 2004, 2:985–990 .

[2] G.-B. Huang. Q.-Y. Zhu, C-K. Siew, Extreme learning machine: theory and applications. Neurocomputing. 2006, 70: 489–501.

[3] G.B. Huang, Y. Q. Chen, H. A. Babri. Classification ability of single hidden layer feedforward neural networks. IEEE Trans. Neural Netw. 2000, 11(3): 799–801.

[4] R. Zhang, Y. Lan, G.-B. Huang, Z.-B. Xu. Universal approximation of extreme learning machine with adaptive growth of hidden nodes. IEEE Trans. Neural Netw. Learn. Syst, 2012, 23(2): 365–371.

[5] M. Fernández-Delgado, E. Cernadas, S. Barro, J. Ribeiro. Direct kernel perceptron (DKP): Ultra-fast kernel ELM-based classification with noniterative closed-form weight calculation. Neural Networks. 2014, 50: 60–71.

[6] G. Huang, S.-J. Song. Semi-supervised and unsupervised extreme learning machines. IEEE Transactions on Cybernetics. 2014, 44(12): 2405-2417.

[7] V. N. Vapnik. Statistical Learning Theory. Wiley. NY, 1998.

[8] Z.Y. Zhou , J. Chen , Y.C. Song. RFSEN-ELM: Selective ensemble of extreme learning machines using rotation forest for image classification. Neural Netw World. 2017, 27 (5): 499–517 .

[9] Y. Peng , W.Z. Kong , B. Yang. Orthogonal extreme learning machine for image classification. Neurocomputing. 2017, 266: 458–464.

[10] Z.Y. Zhou , R. Xu , D.C. Wu. Illumination correction of dyed fabrics approach using Bagging-based ensemble particle swarm optimization–extreme learning machine. Optical Engineering. 2016, 55(9) :093102 .

[12] Z.T. Liu, M. Wu, W.H. Cao. Speech emotion recognition based on feature selection and extreme learning machine decision tree. Neurocomputing. 2018, 273:271–280 .

[13]F. Du, J.S. Zhang , N.N. Ji, G. Shi, C.X. Zhang. An effective hierarchical extreme learning machine based multimodal fusion framework. Neurocomputing. 2018, 322: 141–150.

[14]Y.H. Chen, M. Kloft, Y. Yang, C.H Li, L Li. Mixed kernel based extreme learning machine for electric load forecasting. Neurocomputing. 2018, 312: 90-106.

[15] G. B. Huang, L. Chen, C. K. Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Transactions on Neural Networks. 2006,17(4): 879–892.

[16] X. Z. Wang, Q. Y. Shao, Q. Miao, J. H. Zhai. Architecture selection for networks trained with extreme learning machine using localized generalization error model. Neurocomputing. 2013, 102, 3–9.

[17] W. W. Zong, G.B. Huang, Y. Chen. Weighted extreme learning machine for imbalance learning. Neurocomputing. 2013, 101: 229–242.

[18] G. Huang, S. J. Song. Semi-supervised and unsupervised extreme learning machines. IEEE Transactions on Cybernetics. 2014, 44(12): 2405-2417.

[19] Y. Gu, Y. Q. Chen, J. F. Liu, X. L. Jiang. Semi-supervised deep extreme learning machine for Wi-Fi based localization. Neurocomputing. 2015,166:282–293.

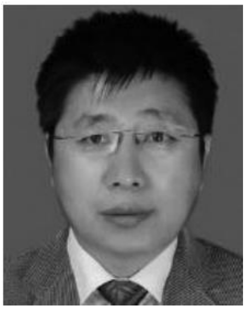
[20] L. Zhang, D.Zhang. Evolutionary Cost-sensitive Extreme Learning Machine.IEEE Transactions on Neural Networks and Learning Systems. 2016, PP(99):1-16.

[21] W. B Zhang, H. B. Ji, G.S. Liao, Y.Q. Zhang. A novel extreme learning machine using privileged information. Neurocomputing. 2015, 168:823–828.

[22] L.J. Silvestre, A.P. Lemos, J.P. Braga. Dataset structure as prior information for parameter-free regularization of extreme learning machines. Neurocomputing. 2015, 169: 288–

294.
[23] C. Savojardo, P. Fariselli, and R. Casadio. BETAWARE: a machine-learning tool to detect and predict transmembrane beta barrel proteins in Prokaryotes. Bioinformatics, 2013.
[24] H.P. Liu, F.X Li, X.Y Xu, F.C Sun. Multi-modal local receptive field extreme learning machine for object recognition. Neurocomputing. 2018, 277: 4-11.
[25] Y.W. Ming, E. Zhu, M. Wang. DMP-ELMs: Data and model parallel extreme learning machines for large-scale learning tasks. Neurocomputing. 2018 320: 85-97.
[26] S. Decherchi, P. Gastaldo, R.S. Dahiya, M. Valle, R. Zunino. Tactile data classification of contact materials using computational intelligence. IEEE Trans Robot. 2011, 27(3):635–639.
[27] J.B. Tenenbaum, V. D. Silva, J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science. 2000,290:2319- 2323.
[28]S.T. Roweis, L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science. 2000,290:2323- 2326.
[29]M. Belkin, P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation. 2003,15(6):1373- 1369.
[30] X.F. He, P. Niyogi. Locality preserving projections. Advances in Neural Information Processing Systems. 2004, 153–160.
[31] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1997, 19 (7): 711–720.
[32] A. Iosifidis, A. Tefas. Minimum class variance extreme learning machine for human action recognition. IEEE Transactions on Circuits and Systems for Video Technology. 2013, 23(11): 1968-1979.
[33] A. Iosifidis, A. Tefas, Pitas. Ioannis. Minimum Variance Extreme Learning Machine for human action recognition. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2014.
[34] A. Iosifidis, A. Tefas, Pitas. Ioannis. Graph Embedded Extreme Learning Machine. IEEE Transactions on Cybernetics. 2016, 46(1): 311-324.
[35] S.L. Liu, L. Feng, Y. Liu. Robust discriminative extreme learning machine for relevance feedback in image retrieval. Multidim Syst Sign Process. 2017, 28:1071–1089.
[36] Y. Peng, B.L. Lu. Discriminative graph regularized extreme learning machine and its application to face recognition. Neurocomputing. 2015, 149: 340–353.
[37] Y. Peng, B.L. Lu. Discriminative manifold extreme learning machine and applications to image and EEG signal classification. Neurocomputing. 2016, 174: 265–277.
[38] W.W. Yu, X.L Teng , C.Q Liu. Face recognition using discriminant locality preserving projections, Image and Vision Computing. 2006, 24: 239–248.
[39] X. F. He, S. C. Yan, Y. X. Hu. Face recognition using Laplacian faces.IEEE Trans on Pattern Analysis and Machine Intelligence. 2005, 27(3):328- 340.
[40]H. X. Wang, S. B. Chen, Z. L. Hu, W. M. Zeng. Locality-preserved maximum information projection. IEEE Trans on Neural Networks. 2008, 19(4):571- 585.
[41] G.B. Huang. An insight to extreme learning machines: Random neurons, random features and kernels. Cognitive Computation. 2014, 6(3):376–390.
[42] X. Li, T. Jiang, K. Zhang. Efficient and robust feature extraction by maximum margin criterion, IEEE Trans. Neural Netw. 2006,17 (1) :157–165.
[43]M. Belkin, P. Niyogi, V, Sindhwani. Manifold regularization:a geometric framework for learning from examples. Journal of Machine Learning Research. 2006, (7): 2399- 2434.
[44] A.M. Martinez, A.C. Kak. PCA versus LDA. IEEE Trans. Pattern Anal. Mach. Intell. 2001, 23 (2):228–233.
[45] M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba. Coding facial expressions with Gabor wavelets. in Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. Nara, Japan, 1998, 200–205.

Yonghe Chu received the M.S. degree in College of Computer and Information Technology, Liaoning Normal University, China, in 2017. Currently, he is working toward the Ph.D. degree in the School of Computer Science and Technology, Dalian University of Technology, China. His research interests include pattern recognition, computer vision and machine learning.



Hongfei Lin received the M.S. degree from the Dalian University of Technology, Dalian, China, in 1992 and the Ph.D. degree from Northeastern University, Shenyang, China, in 2000. He is currently a Professor with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. He has authored over 100 scientific papers in various journals, conferences, and books. His research projects are funded by Natural Science. His current research interests include text mining for biomedical literature, biomedical hypothesis generation, information extraction from huge biomedical resources, learning to rank, sentimental analysis, and opinion mining.

Liang Yang received the M.S. degree from the Dalian University of Technology, Dalian, China, and the Ph.D. degree from the Dalian University of Technology, Dalian, China, He is currently a instructor with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. His current research interests include sentimental analysis, and opinion mining.

Yufeng Diao is working toward the Ph.D. degree in the School of Computer Science and Technology, Dalian University of Technology, China. Her research interests include sentimental analysis, and opinion mining.

Dongyu Zhang is working toward the Ph.D. degree in the School of Computer Science and Technology, Dalian University of Technology, China. Her research interests include sentimental analysis, and opinion mining.

Shaowu Zhang is a Professor with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. His current research interests include sentimental analysis, and opinion mining.

Xiaochao Fan is working toward the Ph.D. degree in the School of Computer Science and Technology, Dalian University of Technology, China. His research interests include sentimental analysis, and opinion mining.

Chen Shen is working toward the Ph.D. degree in the School of Computer Science and Technology, Dalian University of Technology, China. His research interests include sentimental analysis, and opinion mining.

Bo Xu received the M.S. degree from the Dalian University of Technology, Dalian, China, and the Ph.D. degree from the Dalian University of Technology, Dalian, China. His current research interests include sentimental analysis, and opinion mining.

Deqin Yan is professor at College of Computer and Information Technology, Liaoning Normal University. He received Ph.D. at Nankai University in 1999. His research interest is pattern recognition.