Cogn Comput
DOI 10.1007/s12559-017-9474-4

Neuron Pruning-Based Discriminative Extreme Learning Machine for Pattern Classification

Tan Guo[1] · **Lei Zhang**[1] · **Xiaoheng Tan**[1]

Received: 30 September 2016 / Accepted: 25 April 2017

Abstract Extreme learning machine (ELM), as a newly developed learning paradigm for the generalized single hidden layer feedforward neural networks, has been widely studied due to its unique characteristics, i.e., fast training, good generalization, and universal approximation/classification ability. A novel framework of discriminative extreme learning machine (DELM) is developed for pattern classification. In DELM, the margins between different classes are enlarged as much as possible through a technique called ε-dragging. DELM is further extended to pruning DELM (P-DELM) using $L_{2,1}$-norm regularization. The performance of DELM is compared with several state-of-the-art methods on public face databases. The simulation results show the effectiveness of DELM for face recognition when there are posture, facial expression, and illumination variations. P-DELM can distinguish the importance of different hidden neurons and remove the worthless ones. The model can achieve promising performance with fewer hidden neurons and less prediction time on several benchmark datasets. In DELM model, the margins between different classes are enlarged by learning a nonnegative label relaxation matrix. The experiments validate the effectiveness of DELM. Furthermore, DELM is extended to P-DELM based on $L_{2,1}$-norm regularization. The developed P-DELM can naturally distinguish the importance of different hidden neurons, which will lead to a more compact network by neuron pruning. Experimental validations on some benchmark datasets show the advantages of the proposed P-DELM method.

Keywords Extreme learning machine · Label relaxation · Pattern classification · Pruning method

Introduction

Cognitive computation has been emerging as a discipline involving neurobiology, cognitive psychology, and artificial intelligence [1, 2]. A cognitive system is, broadly speaking, something that seeks to mimic or better understand the way that humans process complex situations. Extensive efforts have been made for the study of cognitive-inspired techniques/systems in the past few decades [3–5]. As a type of cognitive-inspired computation technique, feedforward neural networks (FNNs) have been widely investigated and applied since the introduction of the well-known back-propagation (BP) algorithm [6]. However, these gradient descent-based methods may face with the problems of local minima, learning rate, stopping criteria, and learning epochs [7].

Recently, extreme learning machine (ELM) has been proposed for training single hidden layer feedforward neural networks (SLFNs). Unlike the other traditional learning algorithms, e.g., BP-based neural networks (NNs) or support vector machine (SVM), the parameters of hidden layers in ELM are randomly generated without tuning. The hidden nodes in ELM can be established independent of the training data [8, 9]. Huang et al. [10, 11] have theoretically proved that the SLFNs with randomly generated hidden neurons and the output weights calculated by regularized least square maintain its universal approximation capability. The concrete biological evidences

✉ Xiaoheng Tan
txh@cqu.edu.cn

[1] College of Communication Engineering, Chongqing University, Chongqing 400044, China

Published online: 11 May 2017

for ELM have also been reported [12, 13]. With the learning theory, ELM tends to achieve faster and better generalization performance than those of NNs and SVM. ELMs have been extensively studied and demonstrated to have excellent learning accuracy and speed in a variety of applications, such as semisupervised and unsupervised learning [14], multilayer perceptron [15], dimensionality reduction [16], visual tracking [17], tactile object recognition [18], and transfer learning [19, 20].

In the architectural design of ELM network, a key problem is to determine the suitable number of hidden neurons. Too few or too many hidden neurons employed in an ELM network would lead to underfitting or overfitting [21]. The suitable number of hidden neurons is usually determined with human intervention in a trial-and-error way. There are mainly two heuristic techniques for the problem, i.e., constructive methods (or growing methods) and destructive methods (or pruning methods). Huang et al. [10] presented an incremental ELM (I-ELM), where the hidden nodes are added incrementally and the output weights are determined analytically. Lan et al. [22] proposed a constructive hidden node selection method for ELM (CS-ELM) by selecting the optimal number of hidden nodes when the unbiased risk estimation-based criterion C_p reaches the minimum value. Obviously, both I-ELM and CS-ELM are constructive methods. There are also some pruning methods. Miche et al. [23] proposed an optimally pruned ELM (OP-ELM), which first ranks the hidden neurons using the multi-response sparse regression algorithm (MRSR). OP-ELM then selects the hidden neurons through leave-one-out (LOO) validation. Recently, a pruning ensemble model of ELM with $L_{1/2}$ regularizer (PE-ELMR) has been proposed in [24]. PE-ELMR incorporates $L_{1/2}$ regularizer into the preliminary ELM, and the neurons in hidden layer are pruned with the ensemble model.

From the viewpoint of geometry, it is expected that the distances between data points in different classes are as large as possible after they are transformed [25]. However, the traditional ELM assumes that the hidden layer output can be exactly transformed into strict label matrix and does not consider such a geometrical criterion. These observations motivate us to introduce the geometrical criterion into classical ELM to fully exploit the discriminant information in data.

To this end, one feasible way is to enlarge the distances between regression labels of different classes. Figure 1 shows the idea of our method. For a two-class classification problem, the regression labels in original ELM are coded as [+1, −1] and [−1, +1]. They are denoted as red points in the figure. The maximal distance between them is fixed with little freedom. After dragged to the blue ones with proper dragging direction and value, the distance between them can be enlarged (d2 > d1). With this strategy, ELM is expected to have better generalization ability by fully exploiting the discriminative information in data. In detail, a technique called ε-dragging is employed to learn a nonnegative label relaxation matrix, which promotes the regression labels of different classes moving along with opposite directions. A slack label matrix is embedded into the ELM framework so that the distances between different classes can be enlarged. Besides, the proposed discriminative extreme learning machine (DELM) method is further extended for the architectural design of ELM network in a destructive manner. We introduce a structured norm regularization, namely $L_{2,1}$-norm, into DELM model to learn a row-sparse output weight matrix. The method, termed as pruning DELM (P-DELM), can distinguish the importance of different hidden neurons in information transmission. As a result, the worthless neurons can be adaptively removed for a more compact network. It is worth noting that there are some major differences between our work and the work in [17], though both of them adopt $L_{2,1}$-norm regularization for the purpose of neuron pruning. Firstly, we focus on pattern classification with single-task ELM in a supervised way. However, the model in [17] targets at visual tracking with multitask ELM in a semisupervised manner. Secondly, we first design a novel DELM model with label relaxation, and then $L_{2,1}$-norm regularization is introduced into the developed DELM. The obtained P-DELM network tends to be more compact and has better generalization ability. Thirdly, the model in [17] first ranks the neurons in hidden layer and then selects a fixed number of neurons. Differently, we develop an adaptive neuron

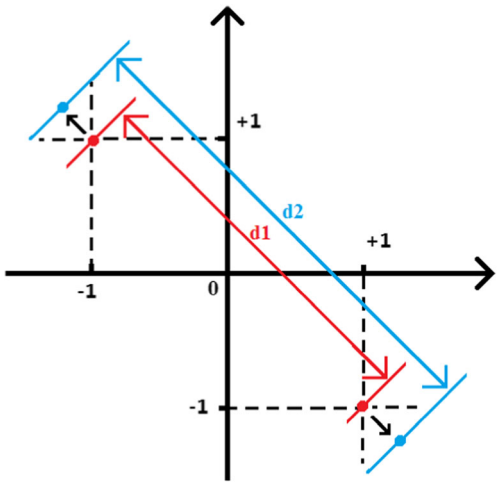

Fig. 1 A simplified illustration for proposed DELM. The regression labels in classic ELM are denoted as *red points*. After dragged to the *blue ones*, the distance between the samples' regression labels is enlarged. With this strategy, DELM is expected to fully exploit the discriminative information in data

selection method by the obtained row-sparse output weight matrix. Moreover, the working principle of $L_{2,1}$-norm regularization is analyzed in detail in this paper. The proposed P-DELM might be introduced into the model in [17] for visual tracking.

Several characteristics of the proposed DELM and P-DELM are as follows:

1. DELM inherits the merits of ELM, including the feature mapping with randomly generated input weights and bias, and good generalization.
2. Hadamard product of matrices is introduced into DELM to perform ε-dragging. A slack variable matrix is constructed, and thus, the margins between different classes can be enlarged. The resultant optimization problem can be solved iteratively by performing variable decoupling. Both theoretical analysis and experimental evaluations show the effectiveness of DELM.
3. The DELM is extended to P-DELM based on $L_{2,1}$-norm regularization. Worthless hidden neurons can be removed with the obtained row-sparse output weight matrix for a more compact network.

The remainder of this paper is outlined as follows. "Extreme Learning Machine and Discriminative Extreme Learning Machine" section reviews related works on ELM and presents our discriminative ELM (D-ELM) model. Our pruning DELM (P-DELM) model is introduced in "Neuron Pruning-Inspired Discriminative Extreme Learning Machine" section. "Experiments" section reports the experimental results. Conclusions are drawn in "Conclusions" section.

Extreme Learning Machine and Discriminative Extreme Learning Machine

Extreme Learning Machine

ELMs are a type of FNNs characterized by a random initialization of their hidden layer weights and a fast training algorithm for the output weights. The optimization function of ELM is

$$\begin{aligned} &\min_{\boldsymbol{\beta}\in\Re^{L\times c}} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\cdot\frac{1}{2}\sum_{i=1}^{N}\|\boldsymbol{\xi}_i\|^2 \\ &s.t.\mathbf{h}(\boldsymbol{x}_i)\boldsymbol{\beta}=\boldsymbol{t}_i-\boldsymbol{\xi}_i,\quad i=1,2\ldots N \Leftrightarrow \mathbf{H}\boldsymbol{\beta}=\mathbf{T}-\boldsymbol{\xi} \end{aligned} \tag{1}$$

where $\boldsymbol{\beta}\in\Re^{L\times c}$ denotes the output weights between hidden layer and output layer and $\boldsymbol{\xi}=[\boldsymbol{\xi}_1,\boldsymbol{\xi}_2\ldots\boldsymbol{\xi}_N]^{\mathrm{T}}\in\Re^{N\times c}$ are the prediction error matrices with respect to the training data. C is a penalty constant on the training errors, and $\mathbf{H}\in\Re^{N\times L}$ is the hidden layer output matrix, computed as

$$\mathbf{H}=\begin{bmatrix}\mathbf{h}(\boldsymbol{x}_1)\\ \vdots\\ \mathbf{h}(\boldsymbol{x}_N)\end{bmatrix}=\begin{bmatrix}h_1(\boldsymbol{x}_1) & \ldots & h_L(\boldsymbol{x}_1)\\ \vdots & \vdots & \vdots\\ h_1(\boldsymbol{x}_N) & \cdots & h_L(\boldsymbol{x}_N)\end{bmatrix}$$

where $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N]\in\Re^{d\times N}$ is the training dataset falling into c categories with label matrix $\mathbf{T}=[\boldsymbol{t}_1,\boldsymbol{t}_2\ldots\boldsymbol{t}_N]^{\mathrm{T}}\in\Re^{N\times c}$ and $\boldsymbol{t}_i=[-1\ldots+1\ldots-1]\in\Re^{c}$. Note that only the jth entry of $\boldsymbol{t}_i$ is +1 which indicates that sample $\boldsymbol{x}_i$ comes from the jth class. By substituting the constraints of (1) into its objective function, we get the following equivalent unconstrained optimization problem

$$\min_{\boldsymbol{\beta}\in\Re^{L\times c}}\frac{1}{2}\|\boldsymbol{\beta}\|^2+C\cdot\frac{1}{2}\|\mathbf{H}\boldsymbol{\beta}-\mathbf{T}\|^2 \tag{2}$$

The previous problem is widely known as the ridge regression or regularized least square. By setting the gradient of the objective function with respect to $\boldsymbol{\beta}$ to be $\mathbf{0}$, we have

$$\boldsymbol{\beta}^*+C\cdot\mathbf{H}^{\mathrm{T}}(\mathbf{H}\boldsymbol{\beta}-\mathbf{T})=\mathbf{0} \tag{3}$$

The closed-form solution $\boldsymbol{\beta}$ can be solved under the following two circumstances. If the number of training samples N is larger than L ($N>L$), the gradient equation is overdetermined, and the closed-form solution can be calculated as

$$\boldsymbol{\beta}^*=\mathbf{H}^{\dagger}\mathbf{T}=\left(\mathbf{H}^{\mathrm{T}}\mathbf{H}+\frac{\mathbf{I}_{L\times L}}{C}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{T} \tag{4}$$

where $\mathbf{I}_{L\times L}$ denotes the identity matrix with size of L and $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of $\mathbf{H}$. If the number N of training patterns is smaller than L ($L>N$), an underdetermined least square problem would be handled. One can restrict $\boldsymbol{\beta}$ to be a linear combination of the rows in $\mathbf{H}$ as $\boldsymbol{\beta}=\mathbf{H}^{\mathrm{T}}\boldsymbol{\alpha}$ ($\boldsymbol{\alpha}\in\Re^{N\times m}$). By substituting $\boldsymbol{\beta}=\mathbf{H}^{\mathrm{T}}\boldsymbol{\alpha}$ into (3), and multiplying both sides with $(\mathbf{H}\mathbf{H}^{\mathrm{T}})^{-1}\mathbf{H}$, we have

$$\boldsymbol{\alpha}^*-C\cdot\left(\mathbf{T}-\mathbf{H}\mathbf{H}^{\mathrm{T}}\boldsymbol{\alpha}^*\right)=\mathbf{0} \tag{5}$$

Then, we get the solution as

$$\boldsymbol{\beta}^*=\mathbf{H}^{\mathrm{T}}\boldsymbol{\alpha}=\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{H}^{\mathrm{T}}+\frac{\mathbf{I}_{N\times N}}{C}\right)^{-1}\mathbf{T} \tag{6}$$

Discriminative Extreme Learning Machine

Conventional ELM assumes that the hidden layer output $\mathbf{h}(\boldsymbol{x}_i)(i=1,2\ldots N)$ can be exactly transformed into strict label matrix as in (2). However, the previous assumption may be too rigid. We relax the strict label matrix into a slack label matrix by introducing a nonnegative relaxation matrix $\mathbf{M}$, which can not

Table 1 A case for ε-dragging

Hidden layer output	Category	**T**	**T** after ε-dragging	Constraints
$\mathbf{h}(\boldsymbol{x}_1)$	1	$[1, -1, -1]$	$[1+\varepsilon_{11}, -1-\varepsilon_{12}, -1-\varepsilon_{13}]$	$\varepsilon_{11}, \varepsilon_{12}, \varepsilon_{13} \geq 0$
$\mathbf{h}(\boldsymbol{x}_2)$	2	$[-1, -1, 1]$	$[-1-\varepsilon_{21}, -1-\varepsilon_{22}, 1+\varepsilon_{23}]$	$\varepsilon_{21}, \varepsilon_{22}, \varepsilon_{23} \geq 0$
$\mathbf{h}(\boldsymbol{x}_3)$	3	$[-1, 1, -1]$	$[-1-\varepsilon_{31}, 1+\varepsilon_{32}, -1-\varepsilon_{33}]$	$\varepsilon_{31}, \varepsilon_{32}, \varepsilon_{33} \geq 0$

only provide more freedom for $\boldsymbol{\beta}$ but also enlarge the distances between different classes as much as possible.

In implementation, we push these +1/−1 label outputs far away along two opposite directions. Specifically, with a positive slack variable ε_i, we hope the output will become $1+\varepsilon_i$ for the sample grouped into "1" and $-1-\varepsilon_i$ for the sample grouped into "−1." In this way, the distance between data points from different classes will be enlarged. By introducing ε-dragging term into the optimization function, the distances between different classes are expected to be enlarged. The following Table 1 illustrates an example of this case.

Before ε-dragging, the maximal distance between the first and third hidden layer output (randomly projected feature) is $\sqrt{(1-(-1))^2+(-1-1)^2+(-1-(-1))^2}=2\sqrt{2}$. While after ε-dragging, the distance becomes

$$\sqrt{((1+\varepsilon_{11})-(-1-\varepsilon_{31}))^2+((-1-\varepsilon_{12})-(1+\varepsilon_{32}))^2+((-1-\varepsilon_{13})-(-1-\varepsilon_{33}))^2}\geq 2\sqrt{2}$$

It can be seen that the distance between the first and third randomly projected feature becomes larger after ε-dragging. This shows that the use of the nonnegative label relaxation matrix allows margins between different classes to be enlarged. Concretely, we introduce an auxiliary matrix **B** that is defined as follows. If $\mathbf{T}_{ij}=1$, $\mathbf{B}_{ij}=+1$, and it indicates the positive dragging direction. If $\mathbf{T}_{ij}=-1$, $\mathbf{B}_{ij}=-1$, which means the negative dragging direction. We record nonnegative learnable dragging value εs in matrix $\mathbf{M}\in\mathfrak{R}^{N\times c}$ and get the relaxation label matrix as $\mathbf{T}^{*}=\mathbf{T}+\mathbf{B}\odot\mathbf{M}$, where $\odot$ is a Hadamard product operator of matrices. By substituting $\mathbf{T}^{*}$ into (2), we obtain the following discriminative ELM (DELM) model

$$\min_{\boldsymbol{\beta}\in\mathfrak{R}^{L\times c},\mathbf{M}} \frac{1}{2}\|\boldsymbol{\beta}\|^2+C\cdot\frac{1}{2}\|\mathbf{H}\boldsymbol{\beta}-\mathbf{T}-\mathbf{B}\odot\mathbf{M}\|^2 \quad s.t.\ \ \mathbf{M}\geq\mathbf{0} \tag{7}$$

Compared with (2), a ε-dragging-related term $\mathbf{B}\odot\mathbf{M}$ is integrated into (7) to enlarge the distances between different classes in label space. When solving (7), we can update each variable by fixing another iteratively. An iterative optimization method to solve problem (7) is presented as follows. Given **M**, problem (7) becomes

$$\min_{\boldsymbol{\beta}\in\mathfrak{R}^{L\times c}} \frac{1}{2}\|\boldsymbol{\beta}\|^2+C\cdot\frac{1}{2}\|\mathbf{H}\boldsymbol{\beta}-\mathbf{T}-\mathbf{B}\odot\mathbf{M}\|^2 \tag{8}$$

Let $\mathbf{Q}=\mathbf{T}+\mathbf{B}\odot\mathbf{M}$, and denote the objective function as $\ell(\boldsymbol{\beta})=\frac{1}{2}\|\boldsymbol{\beta}\|^2+C\cdot\frac{1}{2}\|\mathbf{H}\boldsymbol{\beta}-\mathbf{Q}\|^2$. The solution can be achieved by setting $\frac{\partial\ell(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}}=\mathbf{0}$, then

$$\boldsymbol{\beta}^{*}=\left(\mathbf{H}^{\mathrm{T}}\mathbf{H}+\frac{\mathbf{I}_{L\times L}}{C}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{Q} \tag{9}$$

Given $\boldsymbol{\beta}$, problem (7) becomes

$$\min_{\mathbf{M}} \frac{C}{2}\|\mathbf{H}\boldsymbol{\beta}-\mathbf{T}-\mathbf{B}\odot\mathbf{M}\|^2 \quad s.t.\ \ \mathbf{M}\geq\mathbf{0} \tag{10}$$

Let $\mathbf{R}=\mathbf{H}\boldsymbol{\beta}-\mathbf{T}$, we have

$$\min_{\mathbf{M}} \frac{C}{2}\|\mathbf{R}-\mathbf{B}\odot\mathbf{M}\|^2 \quad s.t.\ \ \mathbf{M}\geq\mathbf{0} \tag{11}$$

Due to the fact that the squared Frobenius norm of matrix can be decoupled element by element, (11) can be decoupled equivalently into $N\times c$ subproblems. For the ith row and jth column element of **M**, we have

$$\min_{\mathbf{M}_{ij}} \left(\mathbf{R}_{ij}-\mathbf{B}_{ij}\mathbf{M}_{ij}\right)^2 \quad s.t.\ \ \mathbf{M}_{ij}\geq 0 \tag{12}$$

where $\mathbf{R}_{ij}$ and $\mathbf{B}_{ij}$ are the ith row and jth elements of **R** and **B**, respectively. Note that $\mathbf{B}_{ij}^2=1$, we obtain $(\mathbf{R}_{ij}-\mathbf{B}_{ij}\mathbf{M}_{ij})^2=(\mathbf{B}_{ij}\mathbf{R}_{ij}-\mathbf{M}_{ij})^2$; thus, we can get

$$\mathbf{M}_{ij}=\max\left(\mathbf{B}_{ij}\mathbf{R}_{ij},0\right) \tag{13}$$

Based on the nonnegative constraint about $\mathbf{M}_{ij}$, **M** can be finally got as

$$\mathbf{M}=\max(\mathbf{B}\odot\mathbf{R},\mathbf{0}) \tag{14}$$

The complete algorithm for solving the optimization problem (7) is described in Algorithm 1.

Algorithm 1 Discriminative Extreme Learning Machine

Input:

Training dataset $\boldsymbol{X}=\left[\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_N\right]\in\Re^{d\times N}$ and their corresponding label matrix $\mathbf{T}=\left[\boldsymbol{t}_1,\boldsymbol{t}_2,...,\boldsymbol{t}_N\right]\in\Re^{c\times N}$, $\mathbf{M}_0=\mathbf{0}$, $\boldsymbol{\beta}_0=\mathbf{0}$, $\delta=10^{-4}$. $\mathbf{B}$ is initialized as $\mathbf{T}$.

1: Calculate the hidden layer output matrix $\mathbf{H}$ with randomly generated hidden neurons.

2: **While** not converged **do**

3: $\mathbf{Q}=\mathbf{T}+\mathbf{B}\ominus\mathbf{M}$, $\boldsymbol{\beta}_{k+1}=\left(\mathbf{H}^{\mathrm{T}}\mathbf{H}+\frac{\mathbf{I}}{C}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{Q}$

4: $\mathbf{R}=\mathbf{H}\boldsymbol{\beta}_{k+1}-\mathbf{T}$, $\mathbf{M}_{k+1}=\max\left(\mathbf{R}\ominus\mathbf{M},\mathbf{0}\right)$

5: **if** $\left(\left\|\mathbf{M}_{k+1}-\mathbf{M}_k\right\|_F^2+\left\|\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_k\right\|_F^2<\delta\right)$, **then**

6: break

7: **end if**

8: $\mathbf{M}=\mathbf{M}_{k+1}$, $\boldsymbol{\beta}=\boldsymbol{\beta}_{k+1}$, $k=k+1$

9: **end while**

10: **Output**: $\mathbf{M}$ and $\boldsymbol{\beta}$.

Once the optimal $\boldsymbol{\beta}$ and $\mathbf{M}$ obtained, we have $\mathbf{Q}=\mathbf{T}+\mathbf{B}\odot\mathbf{M}$, and the predicted output of a new test sample $\boldsymbol{z}$ can be computed as

$$\boldsymbol{y}=\mathbf{h}(\boldsymbol{z})\boldsymbol{\beta}=\mathbf{h}(\boldsymbol{z})\cdot\left(\mathbf{H}^{\mathrm{T}}\mathbf{H}+\frac{\mathbf{I}_{L\times L}}{C}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{Q} \tag{15}$$

Neuron Pruning-Inspired Discriminative Extreme Learning Machine

In classic ELM, the number of hidden neurons is always determined with human intervention in a trial-and-error way. It is tedious to select the suitable number of hidden neurons manually. Besides, an overlarge network also brings about longer prediction responses and unnecessary requirement for large memory as well as high cost in hardware resource. An alternative way is to train a network larger than necessary and then prune the unnecessary neurons. In this section, we will devise and present a novel neuron pruning-inspired discriminative ELM based on structured sparse model, which has been widely studied in pattern recognition and machine learning [26, 27, 42].

Model Formulation

Suppose that we are given N training samples $\{(\boldsymbol{x}_i,\boldsymbol{t}_i)\}$, $i=1,2,\ldots,N$, which belong to c ($\geq$2) classes. Here, $\boldsymbol{x}_i\in\Re^m$ is a

data point and $\boldsymbol{t}_i \in \mathfrak{R}^c$ is its label vector. $\mathbf{h}(\boldsymbol{x}_i) \in \mathfrak{R}^L$, as the ELM feature mapping, maps the sample $\boldsymbol{x}_i$ from m-dimensional input space to the L-dimensional hidden-layer feature space, which is called ELM feature space. Suppose that ELM can approximate the data label. The relation between the estimated outputs and the actual outputs is

$$\boldsymbol{t}_i = \beta_1 \mathrm{h}_1(\boldsymbol{x}_i) + \beta_2 \mathrm{h}_2(\boldsymbol{x}_i) + \ldots + \beta_L \mathrm{h}_L(\boldsymbol{x}_i) = \textstyle\sum_j^L \boldsymbol{\beta}_j \mathrm{h}_j(\boldsymbol{x}_i) \quad (i = 1, 2 \ldots N) \tag{16}$$

where $\boldsymbol{\beta}_j$ is the jth row of output weight matrix $\boldsymbol{\beta}$. The previous N equations can be compactly written as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{17}$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\boldsymbol{x}_1) \\ \vdots \\ \mathbf{h}(\boldsymbol{x}_N) \end{bmatrix} = \begin{bmatrix} \mathrm{h}_1(\boldsymbol{x}_1) & \ldots & \mathrm{h}_L(\boldsymbol{x}_1) \\ \vdots & \vdots & \vdots \\ \mathrm{h}_1(\boldsymbol{x}_N) & \cdots & \mathrm{h}_L(\boldsymbol{x}_N) \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\beta}_L^{\mathrm{T}} \end{bmatrix} \quad \mathbf{T} = \begin{bmatrix} \boldsymbol{t}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{t}_L^{\mathrm{T}} \end{bmatrix}$$

As shown in (16), each neuron will has its own response for the sample. The responses of all the neurons will result in the final label prediction $\mathbf{t}_i$ depending on the output weight matrix $\boldsymbol{\beta}$. If some rows of $\boldsymbol{\beta}$, i.e., $\boldsymbol{\beta}_j$ ($j = 1, 2 \ldots L$) are equal to zero, the corresponding neuron response will have no contribution on the estimated output. As a result, these irrelevant hidden neurons can be removed; i.e., a pruning of neurons can be conducted to get rid of useless neurons. In this way, we endow the output weights $\boldsymbol{\beta}$ the function of neuron selection by considering the relevance of hidden neurons with the class labels. Specifically, a new vector $\overline{\boldsymbol{\beta}}$, which collects the L_2 norms of the row vectors of $\boldsymbol{\beta}$, can be constructed as

$$\overline{\boldsymbol{\beta}} = \left[\|\boldsymbol{\beta}_1\|_2, \|\boldsymbol{\beta}_2\|_2, \ldots, \|\boldsymbol{\beta}_L\|_2\right]^{\mathrm{T}} \in \mathfrak{R}^L \tag{18}$$

where $\left\|\boldsymbol{\beta}_j\right\|_2 = \sqrt{\sum_{k=1}^c \beta_{jk}^2}, j = 1, 2 \ldots L$. Constructing d nonzero rows in $\boldsymbol{\beta}$ is just equivalent to pushing the number of nonzero entities in $\overline{\boldsymbol{\beta}}$ equal to be d

$$\left\|\overline{\boldsymbol{\beta}}\right\|_0 = d \tag{19}$$

Nevertheless, solving the problem with L_0 norm constraint is a NP-hard problem. Alternatively, we approximate L_0 norm with L_1 norm and adopt the following $\mathrm{L}_{2,1}$ norm of matrix $\boldsymbol{\beta}$

$$\left\|\overline{\boldsymbol{\beta}}\right\|_1 = \textstyle\sum_{j=1}^L \sqrt{\sum_{k=1}^c \beta_{jk}^2} = \|\boldsymbol{\beta}\|_{2,1} \tag{20}$$

Therefore, our pruning discriminative ELM (P-DELM) model is formulated as follows:

$$\min_{\boldsymbol{\beta},\mathbf{M}} \|\boldsymbol{\beta}\|_{2,1} + \frac{C}{2}\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T} - \mathbf{B} \odot \mathbf{M}\|_F^2 \ \ s.t. \ \ \mathbf{M} \geq \mathbf{0} \tag{21}$$

The errors between target output and actual output are taken into consideration, which provides more freedom for $\boldsymbol{\beta}$.

Optimization for Pruning Discriminative Extreme Learning Machine

Our P-DELM model could be optimized in a similar way as solving DELM. Given $\mathbf{M}$, let $\mathbf{Q} = \mathbf{T} + \mathbf{B} \odot \mathbf{M}$.The optimization problem becomes

$$\min_{\boldsymbol{\beta},\mathbf{M}} \|\boldsymbol{\beta}\|_{2,1} + \frac{C}{2}\|\mathbf{H}\boldsymbol{\beta} - \mathbf{Q}\|_F^2 \tag{22}$$

Obviously, the objection function is differentiable to $\boldsymbol{\beta}$ [28]. First, we consider the derivative of the term $\|\boldsymbol{\beta}\|_{2,1}$ w.r.t $\boldsymbol{\beta}$. According to the definition of $\mathrm{L}_{2,1}$-norm in (20), the derivative of $\|\boldsymbol{\beta}\|_{2,1}$ about the entity β_{jk} can be calculated as

$$\frac{\partial \|\boldsymbol{\beta}\|_{2,1}}{\partial \beta_{jk}} = \beta_{jk}\left(\textstyle\sum_{l=1}^c \beta_{jl}^2\right)^{-1/2} = \frac{\beta_{jk}}{\left\|\boldsymbol{\beta}_j\right\|_2} \tag{23}$$

Then, we get the derivative of $\|\boldsymbol{\beta}\|_{2,1}$ w.r.t $\boldsymbol{\beta}$as

$$\frac{\partial \|\boldsymbol{\beta}\|_{2,1}}{\partial \boldsymbol{\beta}} = \textstyle\sum \boldsymbol{\beta} \tag{24}$$

where $\sum$ is a diagonal matrix in $\mathfrak{R}^{L \times L}$ with the jth diagonal component computed as

$$\textstyle\sum_{jj} = \frac{1}{\left\|\boldsymbol{\beta}_j\right\|_2} \tag{25}$$

This shows that $\|\boldsymbol{\beta}\|_{2,1}$ can be written as $\|\boldsymbol{\beta}\|_{2,1} = \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\beta}^{\mathrm{T}} \textstyle\sum \boldsymbol{\beta}\right)$, where $\sum$ is defined in (25). By setting the derivation of the objection function (6) w.r.t $\boldsymbol{\beta}$ to $\mathbf{0}$, we have

$$\textstyle\sum \boldsymbol{\beta} + C \cdot \mathbf{H}^{\mathrm{T}}(\mathbf{H}\boldsymbol{\beta} - \mathbf{Q}) = \mathbf{0} \tag{26}$$

We further obtain the expression of $\boldsymbol{\beta}$ as follows:

$$\boldsymbol{\beta}^* = C \cdot \left(\textstyle\sum + C \cdot \mathbf{H}^{\mathrm{T}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{Q} \tag{27}$$

Note that $\sum$ depends on $\boldsymbol{\beta}$, which can be iteratively determined using $\boldsymbol{\beta}$ from the previous optimization step. Then, we fix $\boldsymbol{\beta}$ and solve the following problem to update$\mathbf{M}$.

$$\min_{\mathbf{M}} \frac{C}{2} \cdot \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T} - \mathbf{B} \odot \mathbf{M}\|_F^2 \ \ s.t. \ \ \mathbf{M} \geq \mathbf{0} \tag{28}$$

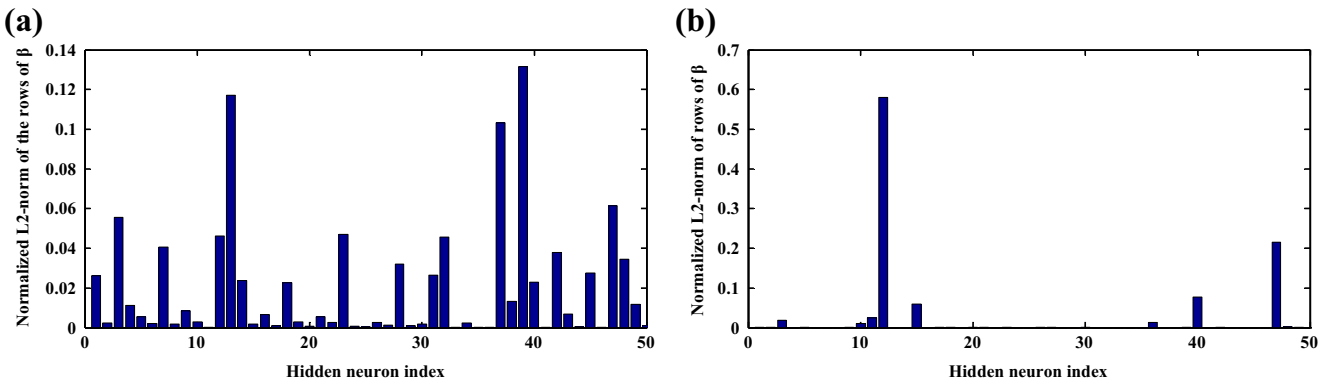

Fig. 2 The normalized L_2 norm of rows of β, namely $\overline{\beta}$ defined in (18). **a** is obtained by the classical ELM, and **b** is obtained by our model shown in (21)

Let $\mathbf{R} = \mathbf{H}\beta - \mathbf{T}$, we have

$$\min_{\mathbf{M}} \frac{C}{2} \cdot \|\mathbf{R} - \mathbf{B} \odot \mathbf{M}\|_F^2 \quad s.t. \quad \mathbf{M} \geq \mathbf{0} \tag{29}$$

Similarly, $\mathbf{M}$ can be got as

$$\mathbf{M} = \max(\mathbf{B} \odot \mathbf{R}, \mathbf{0}) \tag{30}$$

Figure 2 shows the difference between the output weight matrix obtained by the original ELM and our method. Figure 2a shows the normalized L_2-norm of rows of β, i.e., $\overline{\beta}$ defined in (18), obtained from the original ELM. Most of its entities are nonzero with the Frobenius norm constraint, which could only enforce β to be small. Contrastively, the $L_{2,1}$-norm in our model could get a row-sparse β as illustrated in Fig. 2b and distinguish the importance of different hidden neurons. In our P-DELM method, a few hidden neurons can undertake the task of information transmission from input space to the ELM feature space.

After the optimal β is obtained, d neurons can be selected from the L original neurons. We next present a pruning method to get suitable number of valuable hidden neurons. We first normalize $\overline{\beta}$ by dividing it by the sum of all the entities in $\overline{\beta}$ and then sort the entities in $\overline{\beta}$ from large to small. The descending sorted $\overline{\beta}$ is denoted as $\overline{\overline{\beta}} = \left[\overline{\overline{\beta}}_1, \overline{\overline{\beta}}_2, \ldots, \overline{\overline{\beta}}_L\right]$. Then, we select the hidden neurons by a threshold η. The ratio of the sum of the first d entities to the sum of all entities in $\overline{\overline{\beta}}$ is formulated as

$$\eta_d = \frac{\sum_{q=1}^{d} \overline{\overline{\beta}}_q}{\sum_{q=1}^{L} \overline{\overline{\beta}}_q} \tag{31}$$

Obviously, the denominator of (31) is 1, and $\eta_d = \sum_{q=1}^{d} \overline{\overline{\beta}}_q$. Given a threshold η_d in (0, 1), the number of valuable hidden neurons can be got as

$$d = \min\{d | \eta_d \geq \eta\} \tag{32}$$

Once d selected hidden neurons determined, the corresponding hidden layer output matrix $\tilde{\mathbf{H}}$ is utilized to update the output weight matrix as

$$\tilde{\beta} = \left(\tilde{\mathbf{H}}^{\mathrm{T}} \tilde{\mathbf{H}} + \frac{\mathbf{I}_{d \times d}}{C}\right)^{-1} \tilde{\mathbf{H}}^{\mathrm{T}} \mathbf{T} \tag{33}$$

The structure of the proposed P-DELM is shown in Fig. 3. We first adopt the P-DELM to get the valuable

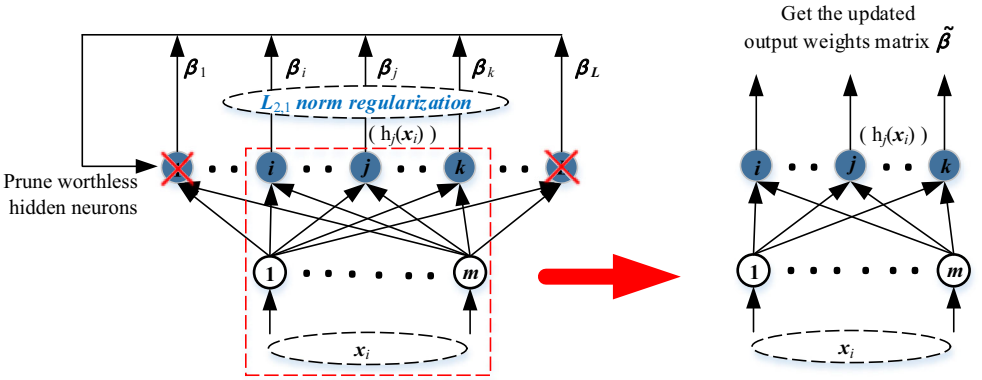

Fig. 3 The structure of proposed pruning DELM (P-DELM)

hidden neurons and then update the output weight matrix exploiting remaining hidden layer output matrix as in (33). The complete optimization algorithm for P-DELM is described in Algorithm 2.

Algorithm 2 Pruning DELM (P-DELM)

Input: the hidden layer output matrix $\mathbf{H}$, label matrix $\mathbf{T}$,

parameter C, η, δ, $\mathbf{M}^0 = \mathbf{0}$, $\boldsymbol{\beta}^0 = \mathbf{0}$, $\mathbf{B}$ is initialized as $\mathbf{T}$.

1: **While** not converged **do**

2: $\mathbf{Q} = \mathbf{T} + \mathbf{B} \ominus \mathbf{M}$. $\sum = \mathbf{I}_L$, $\mathbf{I}_L$ is the identity matrix with size L.

3: $\boldsymbol{\beta}^{k+1} = C \cdot \left(\sum + C \cdot \mathbf{H}^{\mathrm{T}} \mathbf{H} \right)^{-1} \mathbf{H}^{\mathrm{T}} \mathbf{Q}$.

4: Update $\sum_{jj} = 1 \Big/ \left\| \boldsymbol{\beta}_j^{k+1} \right\|_2$ j=1, 2…L.

5: $\mathbf{R} = \mathbf{H}\boldsymbol{\beta}^{k+1} - \mathbf{T}$, $\mathbf{M}^{k+1} = \max(\mathbf{B} \ominus \mathbf{R}, \mathbf{0})$

6: **If** $\left(\left\| \mathbf{M}^{k+1} - \mathbf{M}^k \right\|_F^2 + \left\| \boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k \right\|_F^2 < \delta \right)$

7: **Then** break

8: **End if**

9: $\mathbf{M} = \mathbf{M}^{k+1}$, $\boldsymbol{\beta} = \boldsymbol{\beta}^{k+1}$, $k = k + 1$

10: **End while**

11: Calculate $\overline{\boldsymbol{\beta}}$ according to (18), and get hidden neurons and the corresponding hidden layer output matrix $\overset{\circ}{\mathbf{H}}$. Update the output weights matrix $\overset{\circ}{\boldsymbol{\beta}}$ as in (33).

Experiments

Experimental Results for Discriminative Extreme Learning Machine

Face recognition (FR) is one of the classical problems in computer vision [29]. Facial images have big within-class scatter and small between-class scatter, which poses great difficulties on FR. In this section, four popular face databases, i.e., ORL [30], Extended Yale B [31], CMU PIE [32], and AR [33] databases, are employed to evaluate the performance of different methods. The *ORL face database* contains 400 images from 40 subjects. Each subject has ten images acquired at different times. The size of face image on ORL database is 32 × 32 pixels. The *Extended*

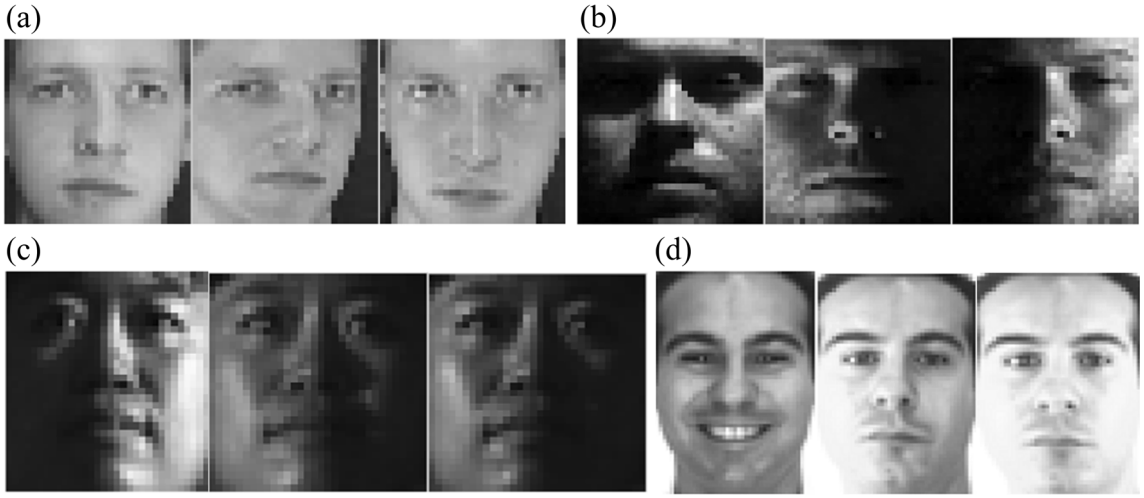

Fig. 4 Some example images used in our experiments. **a** The ORL database. **b** The Extended Yale B database. **c** The CMU PIE database. **d** The AR database

Yale B database consists of 2414 frontal facial images of 38 individuals. Each individual contains about 64 images, taken under various laboratory-controlled lighting conditions. In our experiments, each image is manually cropped and resized to 32 × 32 pixels. The *AR face database* consists of more than 4000 color images of 126 subjects. The *CMU PIE database* contains over 40,000 facial images of 68 individuals. Images of each individual were acquired across 13 different poses, under 43 different illumination conditions, and with 4 different expressions. Figure 4a–d shows several example images of one subject in ORL database, Extended Yale B database, CMU PIE database, and AR database, respectively.

We compare our algorithm with the least squares regression (LSR) [25], discriminative least squares regression (DLSR) [25], SVM [34], LSSVM [35], and the classic ELM on the eigenface feature [36]. For SVM and LSSVM, the Libsvm-3.12 and LSSVM-1.7 toolbox were used, respectively. For LSR and DLSR, the Matlab codes are provided by the authors [25]. The optimum linear transformation is first learned and the facial images under this transformation are employed for classification by a 1-NN classifier.

For the ORL face database, we randomly select l (= 4, 5, 6) images per subject for training and the reminder for testing. For each given l, we independently perform all the methods 10 times and report the average results. Two dimensions of eigenface feature, i.e., 50 and 100, are tested. Table 2 lists the recognition results of different approaches.

For the Extended Yale B database, we randomly select l (=10, 20, 30) images per subject for training and the reminder for testing. For each given l, we independently perform all the methods 10 times and report the average recognition results. Two dimensions of eigenface feature, i.e., 50 and 100, are tested. Table 3 shows the recognition results of different methods.

For the CMU PIE database, we use a near frontal pose subject, namely C07, for experiments, which contains 1629 images of 68 individuals. Each individual has about 24 images. A random subset with l (= 8, 10, 12) images for each individual is selected for training and the rest for

Table 2 Recognition results of different methods on ORL database

No. of training samples per person	4		5		6	
Feature dimension	50	100	50	100	50	100
NN	81.29 ± 2.52	81.29 ± 2.05	85.45 ± 1.74	85.65 ± 2.43	87.69 ± 2.90	87.25 ± 1.94
SVM	82.92 ± 3.58	83.75 ± 2.52	87.25 ± 1.65	89.75 ± 2.32	91.88 ± 2.80	92.56 ± 1.92
LSSVM	82.63 ± 3.38	84.83 ± 2.52	87.40 ± 1.43	88.85 ± 2.88	89.50 ± 3.09	90.13 ± 2.57
LSR	89.25 ± 2.31	91.25 ± 1.86	92.35 ± 1.93	92.95 ± 1.50	94.44 ± 2.79	94.75 ± 1.84
DLSR	89.96 ± 2.12	*93.38 ± 1.50*	93.25 ± 1.75	93.85 ± 1.80	93.19 ± 2.47	93.88 ± 1.64
ELM	88.38 ± 2.48	92.46 ± 1.99	91.80 ± 1.60	93.70 ± 1.67	95.06 ± 2.25	94.56 ± 1.79
DELM	*90.58 ± 1.88*	93.21 ± 2.00	*93.75 ± 1.75*	*93.90 ± 1.71*	*95.44 ± 1.35*	*96.06 ± 1.56*

Italic data are best recognition results (Recognition Rate ± Standard Deviation)

Table 3 Recognition results of different methods on Extended Yale B database

No. of training samples per person	10		20		30	
Feature dimension	50	100	50	100	50	100
NN	37.60 ± 1.24	47.48 ± 1.39	49.58 ± 1.12	61.07 ± 1.11	55.53 ± 0.75	68.27 ± 1.05
SVM	72.28 ± 1.64	76.13 ± 0.86	87.28 ± 0.97	90.57 ± 0.77	92.32 ± 0.90	94.67 ± 0.54
LSSVM	66.21 ± 2.67	68.98 ± 3.99	83.80 ± 0.90	85.87 ± 1.19	89.68 ± 0.53	91.24 ± 0.83
LSR	76.69 ± 1.71	79.98 ± 1.23	88.75 ± 1.05	92.29 ± 0.48	93.05 ± 0.54	95.24 ± 0.52
DLSR	77.08 ± 1.71	81.78 ± 1.17	88.17 ± 0.94	92.31 ± 0.72	92.45 ± 0.84	95.13 ± 0.73
ELM	77.21 ± 1.61	82.41 ± 1.12	89.01 ± 0.61	92.32 ± 0.66	93.57 ± 0.51	95.53 ± 0.63
DELM	*78.42 ± 1.39*	*83.08 ± 1.06*	*89.15 ± 0.72*	*92.94 ± 0.94*	*93.84 ± 0.51*	*95.97 ± 0.63*

Italic data are best recognition results (Recognition Rate ± Standard Deviation)

testing. For each given l, we independently perform all the methods 10 times and report the average recognition rates. Two dimensions of eigenface feature, i.e., 50 and 100, are tested. Table 4 lists the recognition accuracy together with the standard deviation obtained by different methods.

For the AR face database, a subset that contains 50 male subjects and 50 female subjects is chosen in our experiments. For each subject, seven images from session 1 are used for training, with other seven images from session 2 for testing. The size of image is 60 × 43. The recognition results of different methods are given in Table 5.

From Tables 2, 3, 4, and 5, one can conclude that the proposed DELM method can achieve promising performance. Moreover, with the increase of training samples per class and dimension of eigenface feature, all the methods tend to achieve higher recognition accuracy. DELM outperforms all the compared methods on most of the dimensions under different training sets. In comparison with DLSR, the proposed DELM performs better as a whole, which reveals the effect of executing an explicit mapping from the input space to a higher-dimensional ELM feature space. The proposed DELM also outperforms classical ELM. The gain mainly benefits from the enlarged margin between different classes by introducing a nonnegative label relaxation matrix.

Parameter Analysis for Discriminative Extreme Learning Machine

Similar to ELM, the proposed DELM algorithm has two key parameters, namely the number of hidden neurons L and the penalty constant C in (7). We further conduct experiments to investigate the effect of these parameters on the final recognition accuracy. Eleven different values of C (0.001, 0.01, 1, 100, 200, 500, 1000, 2000, 3000, 4000, and 5000) and seven different values of L (100, 500, 1000, 2000, 3000, 4000, and 5000) have been tried, resulting in 77 different pairs in total. The experiments on the previously mentioned databases for parameter analysis are performed, respectively.

Figure 5 shows the relationship between the recognition rate and the parameter pair (L, C). From Fig. 2, one can see that the recognition accuracy tends to increase with the increase of L and C. DELM is not especially sensitive to the

Table 4 Recognition results of different methods on CMU PIE database

No. of training samples per person	8		10		12	
Feature dimension	50	100	50	100	50	100
NN	68.69 ± 1.87	73.96 ± 1.01	76.06 ± 1.58	79.60 ± 1.14	80.06 ± 0.74	85.50 ± 0.96
SVM	88.76 ± 1.00	89.37 ± 1.30	91.96 ± 0.72	92.22 ± 0.99	92.51 ± 0.71	94.18 ± 1.02
LSSVM	79.31 ± 1.81	79.65 ± 2.56	84.88 ± 2.18	84.43 ± 1.85	85.67 ± 1.06	88.39 ± 1.61
LSR	*93.90 ± 0.61*	94.52 ± 0.52	94.40 ± 0.71	95.24 ± 0.50	94.71 ± 0.95	95.90 ± 0.75
DLSR	92.66 ± 0.76	93.30 ± 0.69	93.69 ± 0.72	94.39 ± 0.77	94.19 ± 0.73	95.45 ± 0.77
ELM	93.46 ± 0.79	94.27 ± 0.86	94.69 ± 0.60	94.95 ± 0.74	94.82 ± 0.92	*96.26 ± 0.96*
DELM	93.58 ± 0.44	*94.84 ± 0.90*	*94.95 ± 0.56*	*95.76 ± 0.66*	*95.83 ± 0.78*	95.93 ± 0.54

Italic data are best recognition results (Recognition Rate ± Standard Deviation)

Table 5 Recognition results of different methods on AR database

No. of training samples per person	7			
Feature dimension	50	100	200	300
NN	67.67	70.39	70.96	71.24
SVM	61.66	66.24	68.38	68.96
LSSVM	66.52	69.67	70.24	71.82
LSR	76.54	80.26	83.12	83.69
DLSR	78.97	85.55	87.12	88.27
ELM	82.55	87.84	91.42	*92.13*
DELM	*83.83*	*88.41*	*91.56*	*92.13*

Italic data are best recognition results (Recognition Rate ± Standard Deviation)

change of (L, C) in a large range, and it performs stable when L and C are assigned relatively large values.

Experimental Results for Pruning Discriminative Extreme Learning Machine

The Selection of Parameter η

In this section, we will study the characteristic of the key parameter η in P-DELM, and the experiments are carried out using the Diabetes dataset from the University of California at Irvine (UCI) Machine Learning Repository [37]. With the initialized 50

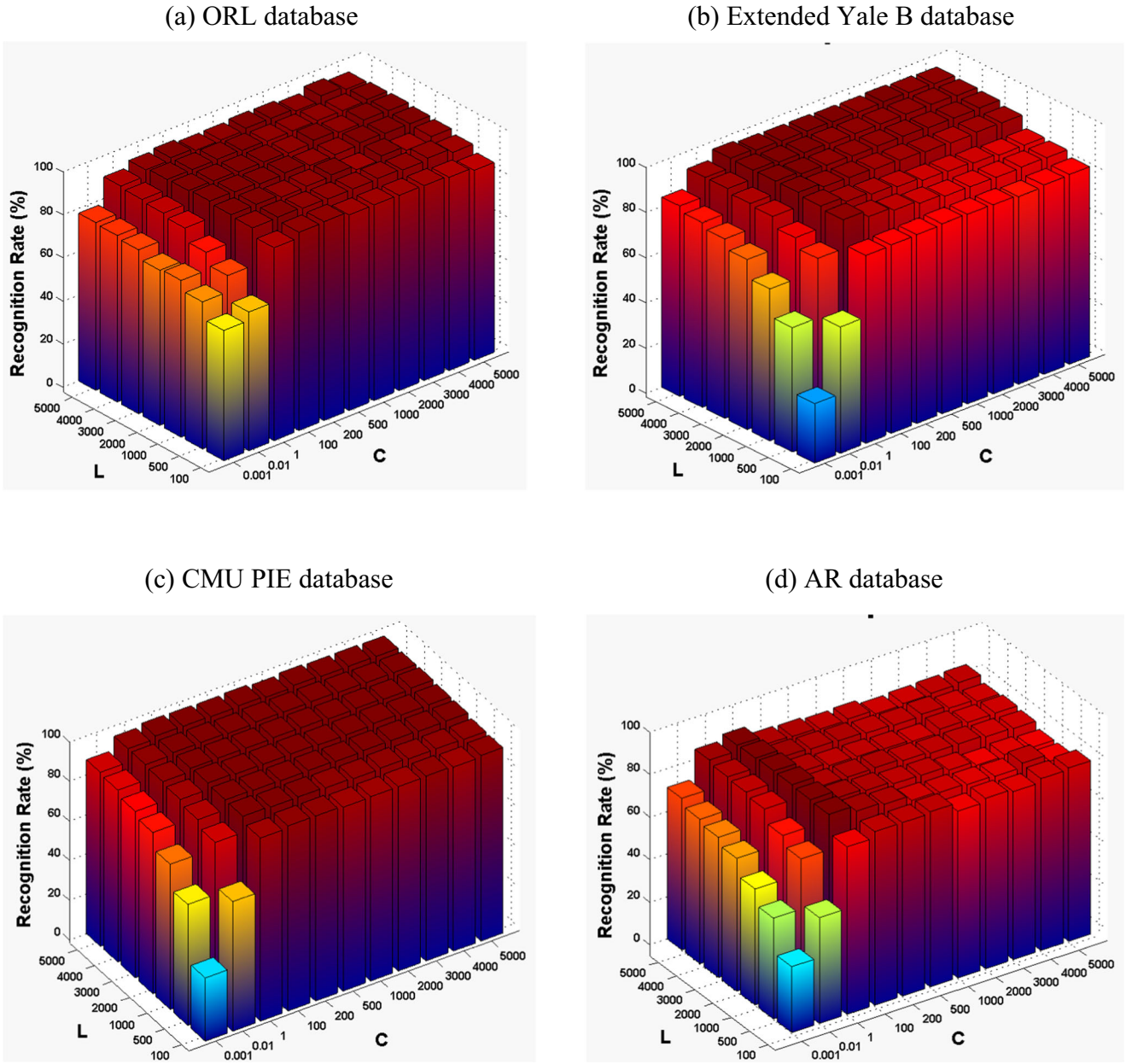

Fig. 5 The influence of tunable parameter (L, C) for DELM on different face database. **a** The ORL database, **b** the Extended Yale B database, **c** the CMU PIE database, and **d** the AR database

Table 6 Specification of classification benchmark problems

Dataset	No. of samples		No. of attributes	No. of classes
	Training	Testing		
Wine	100	78	13	3
Iris	102	48	4	3
Liver disorder	228	117	6	2
Glass	120	94	9	6
Segmentation	140	70	19	7
Diabetes	500	268	8	2

hidden neurons, we record the number of selected hidden neurons and corresponding testing accuracy of P-DELM. The results are acquired from 100 repeated experiments and are shown in Fig. 6. From the results, we observe that the number of selected hidden neurons and testing accuracy raise with the increase of threshold η. When η approaches 1, the number of selected hidden neurons and the testing accuracy tend to reach a plateau. Noteworthy, when $\eta = 0.9999$, only about $(14.56/50) \times 100\,\% = 29.12\%$ of the original hidden neurons are selected. These observations indicate that the introduction of $L_{2,1}$-norm regularization could result in a quite sparse $\overline{\beta}$, which could distinguish the importance of different hidden neurons in information transmission. As a result, we empirically set $\eta = 0.9999$ in the method, which can guarantee a good performance as the following experimental results demonstrate.

The Performance of Pruning Discriminative Extreme Learning Machine for Pattern Classification

In this section, the performance of P-DELM is evaluated on public benchmark datasets for classification problem comparing with the original ELM, DELM, and OP-ELM on several datasets from the UCI Machine Learning Repository [37]. The information and characteristic of the datasets are summarized in Table 6. $L_{2,1}$-DELM denotes an $L_{2,1}$-norm regularized DELM model without a neuron pruning process.

The results are averaged on 100 repeated experiments. We report the mean number of hidden neurons used, testing accuracy/STD, and the CPU time for training and testing in Table 7. From the results, one can conclude that our DELM, $L_{2,1}$-DELM, and P-DELM could always achieve a higher testing accuracy than ELM does. Meanwhile, P-DELM is faster in testing with a comparative or even better performance with fewer hidden neurons. In general, OP-ELM does not perform well in comparison with P-DELM with lower testing accuracy and more hidden neurons.

The Performance of Pruning Discriminative Extreme Learning Machine for Image Classification

We further conduct experiments to validate the performance of DELM, $L_{2,1}$-ELM, and P-DELM on image datasets, including COIL20 and Caltech256. The COIL20 database has 20 objects, and each object has 72 images which are obtained by the rotation of the object through 360° in 51 steps (1440 images in total) [38]. The size of each image is 32 × 32 pixels on COIL20. A subset of Caltech256 database [39, 40], which has 20 classes with 100 samples per category, is used in our experiment. In the experiments, we directly use grayscale image as the feature on COIL20 database, while 2048-dimensional PiCoDes [41] is adopted to represent the images in Caltech256 databases. We randomly select half of the images per class for training and the rest for testing. The experimental results are reported in Table 8. The experiments on these image datasets show promising results. We also note that our methods consume more time for training. A future work should reduce the computational complexity.

Conclusions

In this paper, we have presented a framework of DELM for pattern classification. DELM aims to enlarge the distances between different classes as much as possible by learning a nonnegative label relaxation matrix. The performance of DELM is compared with several state-of-the-art methods on

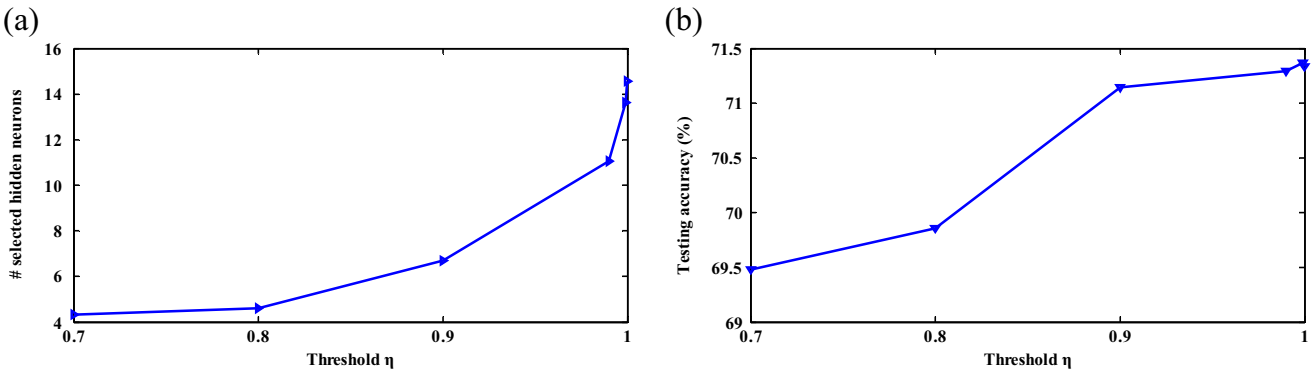

Fig. 6 Effect of the threshold η on the testing accuracy and number of selected hidden neurons. **a** Threshold η versus number of reserved hidden neurons. **b** Threshold η versus testing accuracy

Table 7 The performance comparison between ELM, DELM, $L_{2,1}$-DELM, and P-DELM on UCI datasets

Datasets	Methods	Average number of hidden neurons	Testing		Training time	Testing time
			Accuracy (%)	STD		
Wine	ELM	50	92.7308	2.9159	0.0058	0.0088
	DELM	50	92.6410	2.7188	0.0194	0.0037
	OP-ELM	34.86	81.5385	6.8908	0.1720	0.0036
	$L_{2,1}$-DELM	50	91.9487	3.3997	0.0398	0.0033
	P-DELM	15.72	93.4359	2.9285	0.0502	0.0031
Iris	ELM	80	74.0417	3.9884	0.0056	0.0105
	DELM	80	73.6667	3.2508	0.0716	0.0037
	OP-ELM	20.37	66.5625	14.9306	0.2923	0.0022
	$L_{2,1}$-DELM	80	76.9375	8.4340	0.0742	0.0041
	P-DELM	13.27	74.6458	4.4323	0.0864	0.0028
Liver disorder	ELM	50	64.6068	5.2801	0.0183	0.0077
	DELM	50	65.6667	4.8685	0.0413	0.0033
	OP-ELM	14.59	49.2906	7.2045	0.1862	0.0015
	$L_{2,1}$-DELM	50	62.5983	4.4255	0.0525	0.0044
	P-DELM	9.37	65.5812	4.6081	0.0845	0.0033
Glass	ELM	100	57.3830	4.2896	0.0088	0.0095
	DELM	100	59.2766	4.8293	0.1159	0.0063
	OP-ELM	38.75	19.7553	13.6750	0.6414	0.0032
	$L_{2,1}$-DELM	100	59.1809	5.3431	0.1259	0.0042
	P-DELM	29.11	57.9149	4.2048	0.1330	0.0031
Segmentation	ELM	100	74.6714	9.8093	0.0098	0.0109
	DELM	100	77.6714	9.9071	0.1280	0.0039
	OP-ELM	48.90	28.5429	14.5813	0.9460	0.0044
	$L_{2,1}$-DELM	100	75.6000	9.6735	0.1397	0.0047
	P-DELM	63.60	74.7286	9.8026	0.1441	0.0042
Diabetes	ELM	50	70.1791	2.5009	0.1016	0.0078
	DELM	50	70.4813	2.6633	0.1116	0.0047
	OP-ELM	19.28	54.7500	3.0000	1.3899	0.0029
	$L_{2,1}$-DELM	50	68.7724	1.6911	0.1038	0.0050
	P-DELM	11.67	71.3470	2.7595	0.1891	0.0036

Table 8 The performance comparison between ELM, DELM, $L_{2,1}$-DELM, and P-DELM on COIL20 database and Caltech256 database

Datasets	Methods	Average number of hidden neurons	Testing		Training time	Testing time
			Accuracy (%)	STD		
COIL20	ELM	5000	97.6250	0.7787	1.1497	0.4945
	DELM	5000	97.7361	1.0080	175.6368	0.5132
	L2,1-DELM	5000	97.5000	0.6898	190.8891	0.4930
	P-DELM	3092.7	97.6111	0.6954	192.6160	0.3198
Caltech256	ELM	10,000	52.8900	1.0535	4.4351	2.2324
	DELM	10,000	53.3700	2.0128	39.0736	2.2542
	$L_{2,1}$-DELM	10,000	53.0100	1.2485	40.8660	2.2293
	P-DELM	9971.1	53.4600	1.5064	75.3454	2.2542

public face databases under different experimental settings. The results demonstrate the effectiveness of DELM for FR when there are posture, facial expression, and illumination variations. In addition, we develop a novel method for the problem of architectural design of ELM network by introducing $L_{2,1}$-norm regularization into the DELM model. The obtained P-DELM model can distinguish the importance of different hidden neurons. Worthless neurons are then pruned for a more compact network. Experimental results show that P-DELM can achieve promising performance for pattern classification with fewer hidden neurons and less prediction time.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 61571069, 61401048), Chongqing University Postgraduates' Innovation Project (No. CYB15030), and in part by the Fundamental Research Funds for the Central Universities.

Compliance with Ethical Standards

Funding This study was funded by the National Natural Science Foundation of China (grant number 61571069 and 61401048), Chongqing University Postgraduates' Innovation Project (grant number CYB15030), and in part by the Fundamental Research Funds for the Central Universities.

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with animals performed by any of the authors.

Informed Consent Informed consent was obtained from all individual participants included in the study.

References

1. Taylor JG. Cognitive computation. Cogn Comput. 2009;1(1):4–16.
2. Clark A. Mindware: an introduction to the philosophy of cognitive science. New York: Oxford University Press; 2001.
3. Luo B, Hussain A, Mahmud M, et al. Advances in brain-inspired cognitive systems. Cogn Comput. 2016;8(5):795–6.
4. Zhang HY, Ji P, Wang JQ, et al. A neutrosophic normal cloud and its application in decision-making. Cogn Comput. 2016;8(4):1–21.
5. Gepperth A, Karaoguz C. A bio-inspired incremental learning architecture for applied perceptual problems. Cogn Comput. 2016;8(5):924–34.
6. Rumelhart DE, Hinton GE, Williams RJ. Learning representation by backpropagating errors. Nature. 1986;323(6088):533–6.
7. Zhang L, Zhang D, Tian F. SVM and ELM: who wins? Object recognition with deep convolutional features from ImageNet. Comput Sci. 2015.
8. Huang G-B, Zhou H, Ding X, Zhang R. Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man CybernB Cybern. 2012;42(2):513–29.
9. Huang GB. What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. Cogn Comput. 2015;7(3):263–78.
10. Huang GB, Chen L, Siew CK. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw. 2006;17(4):879–92.
11. Huang GB, Li MB, Chen L, Siew CK. Incremental extreme learning machine with fully complex hidden nodes. Neurocomputing. 2008;71(4–6):576–83.
12. Huang GB. An insight into extreme learning machines: random neurons, random features and kernels. Cogn Comput. 2014;6(3):376–90.
13. Fusi S, Miller EK, Rigotti M. Why neurons mix: high dimensionality for higher cognition. Curr Opin Neurobiol. 2016;37:66.
14. Huang G, Song S, Gupta JND, et al. Semi-supervised and unsupervised extreme learning machines. IEEE Trans Cybern. 2014;44(12):2405–17.
15. Tang J, Deng C, Huang GB. Extreme learning machine for multilayer perceptron. IEEE Trans Neural Netw Learn Syst. 2016;27(4):809–21.
16. Lekamalage CL, Yang Y, Huang GB, et al. Dimension reduction with extreme learning machine. IEEE Trans Image Process. 2016;25(8):3906–18.
17. Liu H, Sun F, Yu Y. Multitask extreme learning machine for visual tracking. Cogn Comput. 2014;6(3):391–404.
18. Liu H, Qin J, Sun F, et al. Extreme kernel sparse learning for tactile object recognition. IEEE Trans Cybern, 2016.
19. Zhang L, Zhang D. Domain adaptation extreme learning machines for drift compensation in E-nose systems. IEEE Trans Instrum Meas. 2015;64:1790–801.
20. Zhang L, Zhang D. Robust visual knowledge transfer via extreme learning machine based domain adaptation. IEEE Trans Image Process. 2016;25(10):4959–73.
21. Rong HJ, Ong YS, Tan AH, et al. A fast pruned-extreme learning machine for classification problem. Neurocomputing. 2008;72(1–3):359–66.
22. Lan Y, Soh YC, Huang GB. Constructive hidden nodes selection of extreme learning machine for regression. Neurocomputing. 2010;73(16–18):3191–9.
23. Miche Y, Sorjamaa A, Bas P, et al. OP-ELM: optimally pruned extreme learning machine. IEEE Trans Neural Netw. 2010;21(1):158–62.
24. He B, Sun T, Yan T, et al. A pruning ensemble model of extreme learning machine with L1/2 regularizer. Multidimension Syst Signal Process. 2016:1–19.
25. Xiang S, Nie F, Meng G, et al. Discriminative least squares regression for multiclass classification and feature selection. IEEE Trans Neural Netw Learn Syst. 2012;23(11):1738–54.
26. Liu H. Robust exemplar extraction using structured sparse coding. IEEE Trans Neural Netw Learn Syst. 2015;26(8):1816–21.
27. Liu H, Guo D, Sun F. Object recognition using tactile measurements: kernel sparse coding methods. IEEE Trans Instrum Meas. 2016;65(3):656–65.
28. Nie F, Huang H, Cai X, et al. Efficient and robust feature selection via joint ℓ2, 1-norms minimization. Adv Neural Info Process Syst. 2010:1813–1821.
29. Zhao W, Chellappa R, Phillips J, Rosenfeld A. Face recognition: a literature survey. ACM Comput Surv. 2003;35(4):399–458.
30. Samaria F S, Harter A C. Harter, A.: Parameterisation of a stochastic model for human face identification[C]// Applications of computer vision, 1994. Proceedings of the Second IEEE Workshop on. 1995:138–142.
31. Rizon M, Hashim MF, Saad P, et al. Face recognition using eigenfaces and neural networks. Am J Appl Sci. 2006;3(6):586–91.
32. Georghiades AS, Belhumeur PN, Kriegman DJ. From few to many: illumination cone models for face recognition under variable

lighting and pose. IEEE Trans Pattern Anal Mach Intell. 2001;23(6):643–60.
33. Martinez, A. M. The AR face database. Cvc Technical Report, 2010, 24.
34. Vapnik V. Statistical learning theory. New York: Wiley; 1998.
35. Suykens JAK, Vandewalle J. Least squares support vector machine classifiers. Neural Process Lett. 1999;9(3):293–300.
36. M. Turk, A. Pentland, Face recognition using eigenfaces. In: Proceeding of the CVPR, 1991.
37. C. Blake, C. Merz, UCI repository of machine learning databases [online], in (http://www.ics.uci.edu/mlearn/MLRepository.html), Department of Information and Computer Sciences, University of California, Irvine, USA, 1998.
38. S. A. Nene, S. K. Nayar, H. Murase, et al., Columbia object image library (COIL-20), Technical report, technical report CUCS-005-96, 1996.
39. Yu J, Tao D, Wang M, Yu J, Tao D, Wang M. Adaptive hypergraph learning and its application in image classification. IEEE Trans Image Process. 2012;21(7):3262–72.
40. G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset, Technical Report, California Institute of Technology, 2007.
41. Bergamo A, Torresani L, Fitzgibbon A. PICODES: learning a compact code for novel-category recognition. Adv Neural Info Process Syst. 2011:2088–2096.
42. Liu H, Yu Y, Sun F, et al. Visual-tactile fusion for object recognition. IEEE Trans Automation Sci Eng. 2016:1–13.