Intelligent Automation And Soft Computing, 2020
Vol. 26, no. 3, 421–427
DOI: 10.32604/iasc.2020.013918

Extreme Learning Machine with Elastic Net Regularization

Lihua Guo

School of Electronic and information Engineering, South China University of Technology, Guangzhou, China, 510641.

ABSTRACT

Compared with deep neural learning, the extreme learning machine (ELM) can be quickly converged without iteratively tuning hidden nodes. Inspired by this merit, an extreme learning machine with elastic net regularization (ELM-EN) is proposed in this paper. The elastic net is a regularization method that combines LASSO and ridge penalties. This regularization can keep a balance between system stability and solution's sparsity. Moreover, an excellent optimization method, i.e., accelerated proximal gradient, is used to find the minimum of the system optimization function. Various datasets from UCI repository and two facial expression image datasets are used to validate the efficiency of our system. Final experimental results indicate that our ELM-EN requires less training time than multi-layer perceptron, and can achieve higher recognition accuracy than ELM and sparse ELM.

KEY WORDS: Extreme learning machine; elastic net; regularized regression.

1 INTRODUCTION

DEEP neural network (DNN) attempts to create an explicit model to learn the data's representation from large-scale data, which is inspired by advances in neuroscience. Various architectures such as convolutional deep neural networks, deep belief networks and recurrent neural networks have been applied to solve many computer vision problems, e.g., hand-written digit recognition, object recognition, image classification, etc.

Two issues, i.e., over-fitting and computation time, should be considered carefully when applying DNN into real applications. DNN is prone to over-fitting because of large hidden layers, which allow to model rare dependencies in the training data. Regularization methods such as Ivakhnenko's unit pruning, weight decay (ℓ_2 -regularization) or sparsity (ℓ_1 -regularization) can be applied to help combat over-fitting. A more recent regularization method applied to DNN is the dropout regularization. In dropout, some units are randomly omitted from the hidden layers during training. This helps to exclude rare dependencies that occur in the training data.

The dominant method for training these deep learning structures is the back-propagation with gradient descent due to easier implementation. However, this method costs much computation time, especially for DNN. There are many carefully-designed parameters in DNN, such as the network structure parameters, learning rate and neural node's weights. Searching most optimal parameters may not be feasible due to computational resources and time costs.

For overcoming these issues of the deep learning framework, especially computation time, Huang, et. al. (2006) initially proposed an extreme learning machine (ELM). ELM was a feed-forward neural network for classification or regression with single layer feed-forward hidden nodes, where all weights connecting the inputs to hidden nodes were randomly assigned and learned in a single step. The experimental results showed that ELM was able to produce better generalization performances, and learned thousands of times faster than DNN that was trained by the back-propagation. After ELM, lots of research had been done to improve it. Huang (2014) discussed ELM with random neurons, random features and kernels, and gave a theory analysis why ELM could outperform the support vector machine (SVM). Huang (2015) further extended the shallow architecture of ELM into a hierarchical learning framework and proposed a locally connected ELM. Yang, et. al. (2015) proposed a nonlinear predictive control strategy based on ELM to address the path-tracking control problem of wheeled mobile robots. Miao, et. al. (2017) proposed a prediction model to improve the learning ability and its prediction

CONTACT Lihua Guo guolihua@scut.edu.cn

precision, which combined PCA and ELM. Huang, et. al. (2012) proposed a unified ELM, where both kernels and random hidden nodes could work for the feature mapping. It provided a unified framework to simplify and unify different learning methods, including SVM, feed forward neural networks, etc. Huang, et. al. (2010) further studied ELM for classification with regard to the standard optimization method, and extended ELM to a specific type of "generalized" support vector network. However, the sparsity was lost as equality constraints. Bai, et. al. (2014) proposed a sparse ELM as an alternative solution for classification, and it could reduce storage space and testing time.

The sparse regularization has a limitation, i.e., in the "large p, small n" case (high-dimensional data with few examples), the system with sparse regularization only selects at most n variables before it saturates. If there is a group of highly correlated variables, the system with a sparse regularization tends to select one variable from a group and ignores the others. To overcome this limitation, an extreme learning machine with elastic net regularization (ELM-EN) is proposed in this paper. The elastic net is a regularization method that linearly combines ℓ_1 and ℓ_2 penalties, which are used in LASSO and ridge regression methods respectively. Since the elastic net is a linear combination between ℓ_1 and ℓ_2 norm in the system optimization function, each term of loss function is strictly convex. An excellent optimization method, i.e., accelerated proximal gradient (APG), is used to find the minimum of the system optimization function.

The organization of the rest of this paper is as follows. Section II reviews the basic principle of ELM method. The technical details of our ELM-EN are given in Section III. Experiment and comparative analysis are discussed in Section IV. Conclusions are drawn in Section V.

2 REVIEW OF THE ELM METHOD

ELM firstly transformed the inputs into the hidden layer through ELM features mapping, then the outputs were generated through ELM learning, which included classification, regression, clustering, etc.

1) ELM feature mapping: the output function of ELM was as follows,

$$f(x) = \sum_i \beta_i h_i(x) = h(x)\beta \quad (1)$$

$\beta = [\beta_1, \ldots, \beta_L]^T$ was the vector of the output weights between the hidden layer with L nodes and the output layer with m nodes, and $H = [h_1, \ldots, h_L]$ was the vector of the hidden layer. Different activation functions were used in different hidden neurons. In real applications, $h_i(x)$ was as follows,

$$h_i(x) = G(w_i, b_i, x), w_i \in R^d, b_i \in R \quad (2)$$

$G(w_i, b_i, x)$ was a nonlinear piecewise continuous function and (w_i, b_i) were the i-th hidden node parameters. Some commonly used activation functions were the Sigmoid function, Fourier function and Gaussian function. In contrast with conventional artificial neural networks theories, hidden neurons did not need to tune in ELM, and which were randomly assigned. $h(x)$ actually mapped the data from the d-dimensional input space to the L-dimensional hidden layer random feature space, which was also called ELM feature mapping.

2) ELM learning: the solution of ELM aimed to reach the smallest training error as follows,

$$E = min_\beta \|H\beta - T\| = min_\beta \|G(w_i, b_i, x)\beta - T\| \quad (3)$$

$T = [t_1, \ldots, t_m]$ depicted the output target of the final neurons. This optimization function was ℓ_2 norm, which was a square error between $H\beta$ and T.

Training ELM was simply equivalent to find a least-squares solution of the linear system $H\beta = T$. The smallest norm least squares solution of Eq. (3) was as follows,

$$\beta = (HH^T)^{-1}HT \quad (4)$$

3 OUR ELM WITH ELASTIC NET REGULARIZATION

BAI, et. al. (2014) proposed a sparse ELM (SELM) to reduce storage space and testing time, and the optimization function of SELM is as follows,

$$E = min_\beta \|H\beta - T\| + \|\beta\|_1 \quad (5)$$

When the system handles the high-dimensional data with few examples, the sparse regularization may meet a large-dimension-few-example problem, i.e., the system only selects one variable from one group, and ignores the others. Moreover, the square error in ELM is an effective loss function when there is a regression problem. However, it is not an effective loss function when there is a classification problem. For overcoming these problems, we propose an ELM with elastic net regularization as follows,

$$E = min_\beta Loss(H, \beta, T) + \lambda_1 \|\beta\|_1 + (1 - \lambda_1)\|\beta\|_2 \quad (6)$$

$Loss(H, \beta, T)$ is the loss function. The loss function is the square loss for the regression, and it is the cross-entropy loss for the classification (logistic loss is the special case of cross entropy loss in the binary classification).} The parameter λ_1 is a balance variable between LASSO and ridge penalties. When $\lambda_1 = 1$, our system degrades into SELM. Meanwhile, when $\lambda_1 = 0$, our system becomes the traditional ELM method.

There are several advantages using this elastic net regularizer. First, ℓ_2 norm regularizer helps to remove the limitation on the number of output weight

coefficients; second, it encourages the grouping effect, which makes all output weight coefficients maintain a consistency; third, it also stabilizes the ℓ_1 norm regularization.

In ELM, the optimization function is strictly convex, and it can be solved using the Moore-Penrose generalized inverse. The elastic net is a linear combination between ℓ_1 and ℓ_2 norm in the system optimization function, where each term of loss function is strictly convex.

For efficiently solving Eq. (6) in our ELM-EN, APG (Huan, 2015) is used in our system. APG is an excellent optimization method not only for convex programming problem, but also for non-convex programming problem. In our ELM-EN, each term of Eq. (6) is convex. Therefore, APG can be guaranteed to find the minimum solution from Eq. (6). During real implementation, APG comprises alternately updating a weight matrix sequence $\{\beta^t\}$ and an aggregation matrix sequence $\{\Psi^t\}$. Each iteration consists of two steps:

a) A generalized gradient mapping step to update matrix β^{t+1} with current aggregation matrix Ψ^t. Given the current matrix Ψ^t, we update β^{t+1} as follows,

$$\beta^{t+1} = \Psi^t - \eta\nabla^t \qquad (7)$$

η is the step size parameter, ∇^t is the gradient of Eq. (6) in step t. The optimization function (Eq. (6)) can separate into three terms. The first term is $Loss(H, \beta, T)$. When system's application is a regression problem, the loss function is a square error between $H\beta$ and T, and the gradient is $(H\beta - T)H^T$ and. When the system's application is a classification problem, the loss function is a cross entropy loss, and the gradient is $-Hlog(H\beta) - H$. The second term is $\|\beta\|_1$, whose gradient is $sign(\beta)$. The third term is $\|\beta\|_2$, whose gradient is β.

b) An aggregation forward step to update Ψ^{t+1} by linearly combining β^{t+1} and β^t. We construct a linear combination of β^{t+1} and β^t to update Ψ^{t+1} according to the APGmethod (Yuan 2012),

$$\Psi^{t+1} = \beta^{t+1} + \frac{a_{t+1}(1-a_t)}{a_t}(\beta^{t+1} - \beta^t) \qquad (8)$$

a_t is conventionally set as $a_t = \frac{2}{t+2}$.

4 EXPERIMENTAL RESULTS AND DISCUSSION

OUR ELM-EN is compared with four currently representative methods, i.e., MLP (Rumelhart 1986), SVM (Cortes 1995), ELM (Huang 2014) and SELM (Bai 2014). All datasets are evaluated with MATLAB R2012b running in a personal computer with a 3.1GHz CPU and an 8GB RAM memory. Kernel methods in MATLAB Toolbox is used to implement SVM. The evaluation metrics include four aspects: training accuracy, testing accuracy, training time and testing time.

Initially, we analyze the performance between ELM-EN with ELM and SELM with different numbers of hidden network nodes. The Arcene dataset from the UCI repository is chosen as the evaluation dataset, whose task is to distinguish cancer versus normal patterns from mass-spectrometric data. This dataset is a binary classification problem. The data's dimension is 10000, which is larger than the number of data (900). Figure 1 and 2 give three method's training and testing accuracies with different numbers of hidden network nodes. Figure 1 and 2 indicate that ELM meets over-fitting when the number of hidden network nodes is over 100. Training accuracy of SELM is worst among three methods because SELM meets the "large attribute and small instance" problem. ELM-EN achieves the best performance when the number of hidden network nodes is 170, which outperforms ELM and SELM.

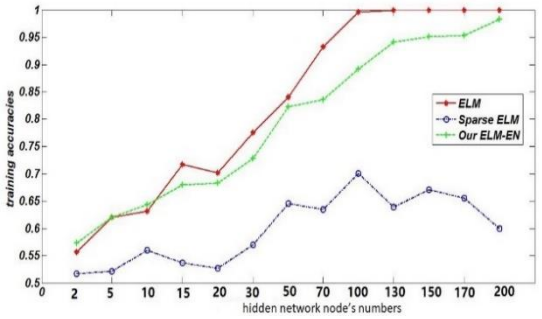

Figure 1. Training Accuracies of Three Methods with the Hidden Network Node Numbers *L*.

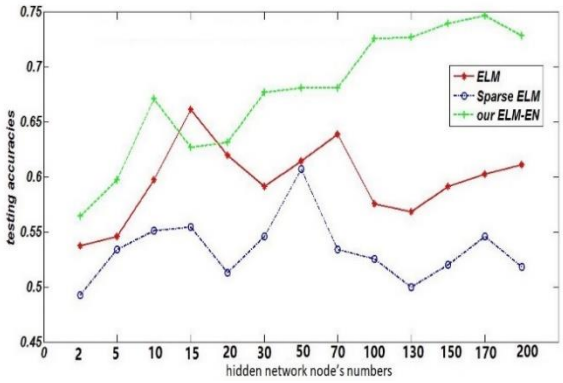

Figure 2. Testing Accuracies of the Three Methods with the Hidden Network Node Numbers *L*.

We provide the sensitivity studies for parameters $\lambda_1$and L. We set λ_1 =[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1] and L =[2, 5, 10, 15, 20, 30, 50, 70, 100, 130, 150, 170, 200]. Figure 3 and 4 give training and testing accuracies in different values of two parameters λ_1 and L. With increasing L, the system more easily meets over-fitting. Therefore, L cannot be infinite in real implementation. When L is small, the hidden layer's space cannot efficiently represent the

information of data, and ELM meets the under-fitting problem. Therefore, the regularization using variation of λ_1 has little impact on the system performance. When L is large (e.g., L is over 100), ELM meets the over-fitting problem. Regularization is an efficient method to overcome the over-fitting problem. Therefore, regularization using variation of λ has large impact on system performance. The parameter λ_1 keeps a balance between ℓ_1 and ℓ_2, and it needs to be tuned in different datasets. In this Arcene dataset, the system can achieve best performance when L is 170, and λ_1 is 0.2.

Furthermore, we evaluate the performance of ELM-EN when testing the high-dimensional data with few examples, therefore, three datasets with high feature dimension are chosen, which are DOROTHEA, GISETTE, and DEXTER from the UCI Repository. DOROTHEA is a drug discovery dataset, whose feature dimension is 100000, and the number of data is only 1950. GISETTE is a handwritten digit recognition problem, whose feature dimension is 5000, and only ten percent of data is chosen, whose number of data is 1350. DEXTER is a text classification problem, whose feature dimension is 20000, and the number of data is 2600. The common point of these three datasets is that the feature dimension is larger than the number of data, and all of them are over 5000. Figures 5-7 give the training and testing accuracies during testing three datasets. When testing DOROTHEA, the training accuracy of ELM has been 100%, but the testing accuracy of ELM is deteriorated when the number of neural nodes is between 600 and 1000. It is obvious that the system meets the over-fitting problem. This phenomenon also exists when testing GISETTE and DEXTER. The experimental results indicate that SELM cannot solve this problem. The accuracy curve of ELM-EN is more stable than those of ELM and SELM, and the testing accuracies of ELM-EN are higher than those of ELM and SELM. Performance of these three datasets further proves that ELM-EN has more robustness than ELM and SELM when evaluating the high-dimensional data with few examples.

Table 1. **Dataset of Binary and Multi-class Classification.**

Datasets	#Train	#Test	Features	Classes
Binary Classification				
Breast	342	341	10	2
Diabetes	384	384	8	2
Mushroom	4062	4062	22	2
Magic	9510	9510	11	2
Spambase	2301	2301	57	2
Arcene	450	450	10000	2
Multi class Classification				
Iris	75	75	4	3
Wine	89	89	13	3
Segment	1155	1155	19	7
Satimage	4435	2000	36	6

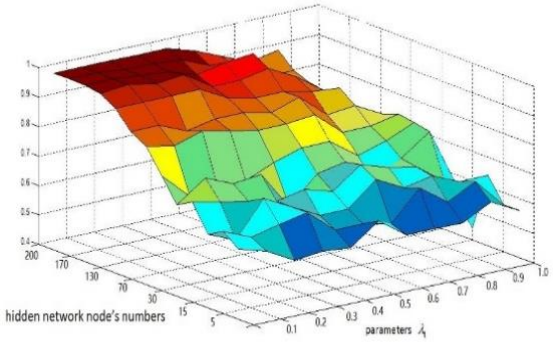

Figure 3. **Training Accuracies in Different Values of Two Parameters λ_1 and L.**

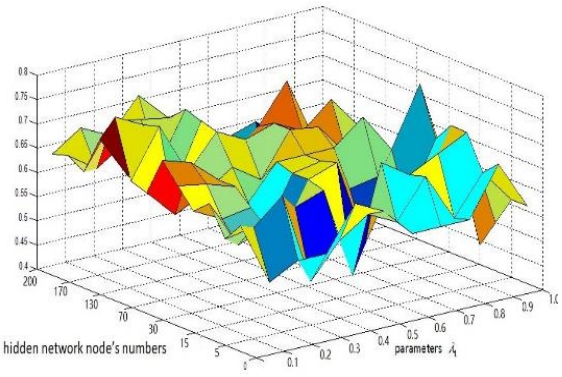

Figure 4. **Test Accuracies in Different Values of Two Parameters λ_1 and L.**

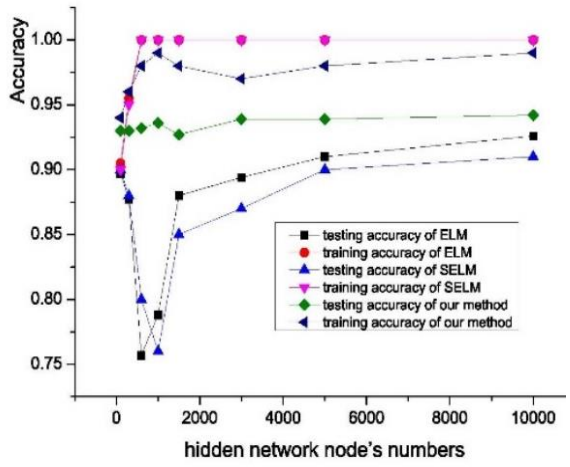

Figure 5. **Training and Testing Accuracies of DOROTHEA in Different Numbers of Hidden Network Nodes.**

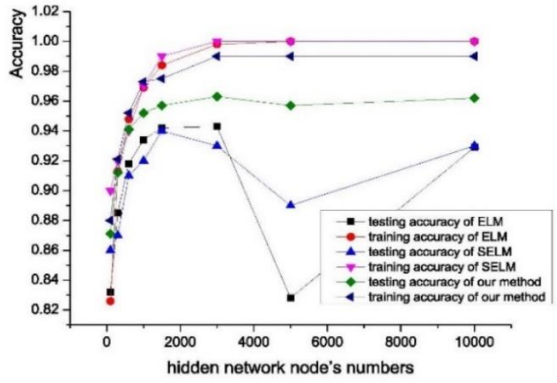

Figure 6. Training and Testing Accuracies of GISETTE in Different Numbers of Hidden Network Nodes.

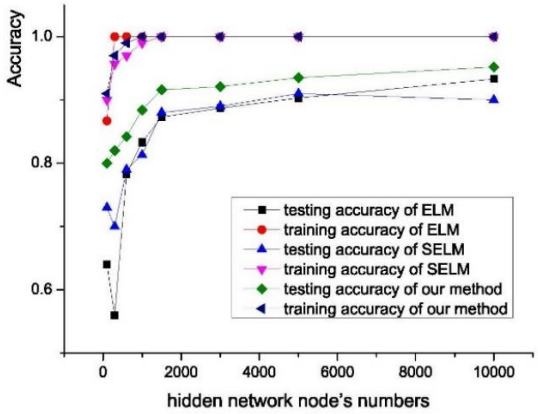

Figure 7. Training and Testing Accuracies of DEXTER in Different Numbers of Hidden Network Nodes.

Additionally, ten datasets are chosen from the UCI repository to evaluate the performance of ELM-EN. They are six binary classification datasets and four multi-class classification datasets. Details are summarized in Table 2. During the experiment, the label is either ***1*** or ***-1*** for binary classification, and the label is *1*, *2*,..., *N* for multi-class classification, where N is the number of classes. In order to reduce human involvement, the five-fold cross validation method is used to find the optimal parameters, i.e., the length scale of Gaussian function σ, the number of hidden network nodes L and λ_1. As shown in Table 3, the optimal parameters of L and λ_1 are specified for each dataset. Table 4 gives the system performance during testing six binary class and four multi-class datasets. When testing datasets with high dimension features, i.e., Arcene and Spambase, the computer cannot train MLP, and it meets the shortage of memory in MATLAB. Therefore, the performance of MLP is NULL during testing Arcene and Spambase. ELM, SELM and our ELM-EN have an analytical solution, therefore, these three methods almost have less training time than MLP. About the testing accuracy, our ELM-EN can achieve higher accuracy than SVM in six datasets, ELM in five datasets, and SELM in six datasets. However, in the Arcene dataset, SVM outperforms ELM, SELM and our ELM-EN. The main reason is that the dimension of input feature is far larger than the number of data, and the random feature mapping suffers the curse of dimensionality when lacking the feature selection. However, SVM seeks the largest margin from support vectors, and seems to be a sort of feature selection. About training time, our ELM-EN has less time than SVM in nine datasets, and ELM in three datasets. The training time of our ELM-EN is roughly equal with that of SELM.

Finally, two facial expression image datasets are chosen to evaluate the performance of ELM-EN, namely the COHN-KANADE (Kannade 2000) and JAFFE (Lyons 1998) datasets. COHN-KANADE consists of facial images depicting 210 persons, and the facial expression includes seven kinds, i.e., anger, disgust, fear, happiness, sadness, surprise and neutral. JAFFE consists of 210 facial images from 10 Japanese female persons, and each person has 3 images of facial expression. Table 5 illustrates the performance of ELM, S-ELM and ELM-EN for different numbers L of hidden layer nodes. During testing COHN-KANADE and JAFFE, ELM-EN generally provides enhanced performance when compared to both ELM and S-ELM algorithms.

5 CONCLUSION

AN extreme learning machine with elastic net regularization (ELM-EN) was proposed in this paper. The elastic net regularization was used to linearly combine LASSO and ridge penalties. Various datasets from the UCI Repository and two facial expression datasets were used to evaluate our system. Experimental results showed that our ELM-EN had less training time than MLP, and outperformed ELM and SELM methods. The parameters λ_1 was used to keep a balance between the ℓ_1 and ℓ_2 regularizers. In our method, this parameter λ_1 was chosen by the cross validation, and the optimal parameter λ_1 was different for different datasets. In the future, an automatic parameter selection method needs further research.

6 ACKNOWLEDGMENT

THIS work was supported by This work is supported by Natural Science Foundation of Guangdong Province (No.2015A030313210), Science and Technology Program of Guangzhou (No.201707010141).

Table 2. **Parameter Specification.**

	MLP	SVM		ELM		Sparse ELM		Our ELM-EN		
	Sigmoid	Gaussian kernel		Gaussian kernel		Gaussian kernel		Gaussian kernel		
	L	C	σ	L	σ	L	σ	L	σ	λ_1
Breast	100	2	1	2	1	200	1	200	1	0.1
Diabetes	100	10	5	10	5	200	1	200	5	0.2
Mushroom	100	1	1	1	1	0.2	0.5	100	1	0.2
Magic	100	2	1	200	1	50	0.5	200	1	0.3
Arcene	1000	50	500	500	1000	500	1000	1000	500	0.2
Spambase	1000	5	0.5	10	1	2	0.5	100	0.5	0.2
Iris	1000	10	1	500	2	1	0.5	200	2	0.1
Wine	1000	5	1	1	2	5	0.5	50	2	0.2
Segment	1000	1000	0.2	1	0.1	1	0.1	2000	0.1	0.2
Satimage	1000	500	1	1	0.2	1	0.1	2000	0.2	0.2

Table 3. **System Performance of Ten Datasets. A: Training Accuracy, B: Testing Accuracy, C: Training Time, D: Testing Time. NULL Means that the MLP Cannot Train the Model of the Shortage of the Memory in MATLAB.**

		Breast	Diabetes	Mushroom	Magic	Arcene	Spambase	Iris	Wine	Segment	Satimage
MLP	A(%)	93.72	**90.36**	96.92	**89.84**	NULL	NULL	82.68	88.76	68.78	73.42
	B(%)	89.01	64.06	78.15	85.87	NULL	NULL	61.33	38.33	78.17	63.84
	C(s)	8.14	268.05	5372.8	4423.8	NULL	NULL	37.29	376.89	7513.5	1469.6
	D(s)	0.094	0.048	0.36	0.49	NULL	NULL	0.14	0.34	0.34	**0.27**
SVM	A(%)	98.25	78.65	100	84.29	**100**	**96.61**	**100**	**100**	**100**	**100**
	B(%)	97.36	73.96	100	85.73	**82.94**	92.83	93.33	97.75	91.43	**90.55**
	C(s)	0.055	0.13	41.48	311.77	**0.25**	9.7	0.025	0.03	5.13	11.59
	D(s)	0.001	0.0026	0.32	2.23	**0.03**	0.17	0.0009	0.0011	0.34	0.41
ELM	A(%)	**99.12**	83.33	100	88.46	100	95.13	100	100	100	100
	B(%)	98.24	74.48	100	**86.88**	67	**93.7**	97.33	**98.89**	96.1	90.95
	C(s)	0.0092	**0.012**	2.38	24.1	1.35	0.57	0.0029	0.0027	**0.23**	2.66
	D(s)	0.0039	0.0062	0.64	4.5	0.41	0.28	0.0008	0.0008	**0.064**	0.35
SELM	A(%)	99.05	84.92	100	87.47	69.3	95.1	98.4	100	100	99.85
	B(%)	98.21	74.67	100	86.2	61.5	93	97.27	97.92	95.77	90.08
	C(s)	**0.0075**	0.016	**0.82**	**5.11**	1.01	0.32	**0.0028**	**0.006**	0.24	**2.41**
	D(s)	**0.0009**	**0.003**	0.058	**1.44**	0.31	0.095	**0.0007**	**0.0013**	0.23	0.55
Our ELM-EN	A(%)	99.01	85.73	**100**	88.65	97.1	96.4	99.43	**100**	**100**	**100**
	B(%)	**98.25**	**78.53**	**100**	86.2	74.9	93.5	**97.35**	97.8	**96.12**	90.35
	C(s)	0.0083	0.015	0.83	5.13	1.21	**0.3**	**0.0028**	**0.006**	0.24	2.51
	D(s)	0.001	**0.003**	**0.058**	**1.44**	.30	**0.095**	**0.0007**	**0.0013**	0.23	0.56

Table 4. **Recognition Accuracies (%) of Two Facial Expression Image Datasets.**

	L	ELM	S-ELM	ELM-EN
COHN-KANADE (Kannade 2000)	50	54.98	42.33	**55.15**
	100	**59.55**	52.24	59.42
	250	63.88	58.41	**64.05**
	500	66.65	62.16	**67.98**
	1000	68.69	63.51	**69.16**
JAFFE (Lyons 1998)	50	52.1	36.95	**54.18**
	100	62.81	48.05	**67.31**
	250	73.62	63.81	**75.84**
	500	79.57	73.33	**82.17**
	1000	83.38	80.24	**85.25**

7 DISCLOSURE STATEMENT

NO potential conflict of interest was reported by the authors.

8 REFERENCES

C. Cortes, and V Vapnik (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.

G. B. Huang (2014). An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels, *Cognitive Computation*, 6, 376-390.

G. B. Huang (2015). Local receptive fields based extreme learning machine, *IEEE Computational Intelligence Magazine*, 10(2), 18-29.

G. B. Huang, H. Zhou, X. Ding, and R. Zhang (2012). Extreme learning machine for regression and multiclass classification, *IEEE Transactions on Cybernetics*, 42(2), 513-529.

G. B. Huang, Q. Y. Zhu and C. K. Siew (2006). Extreme Learning Machine: Theory and Applications, *Neurocomputing*, 70, 489-501.

G. B. Huang, X. Ding, and H. Zhou (2010). Optimization method based extreme learning

machine for classification, *Neurocomputing*, 74, 155-163.

L. Huan, and Z. Lin (2015). Accelerated Proximal Gradient Methods for Nonconvex Programming, *Advances in Neural Information Processing Systems*.

M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba (1998). Coding facial expressions with Gabor wavelets. *In Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 200-205.

T. Kanade, Y. Tian, and J. Cohn (2000). Comprehensive database for facial expression analysis. *In Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 46-53.

X. T. Yuan, X. Liu, S. Yan (2012). Visual classification with multitask joint sparse representation, *IEEE Transactions on Image Processing*, 21(10), 4349-4360.

Y. Yang, X. Lin, Z. Miao. (2015). Predictive Control Strategy Based on Extreme Learning Machine for Path-Tracking of Autonomous Mobile Robot, *Intelligent Automation and Soft Computing*, 21(1), 1-19.

Y. Z. Miao, X. P. Ma, and S. P. Bu. (2017). Research on the Learning Method Based on PCA-ELM, *Intelligent Automation and Soft Computing*, 23(4), 1-6.

Z. Bai, G. B. Huang, D. Wang, H. Wang and M. B. Westover (2014). Sparse Extreme Learning Machine for Classification, *IEEE Transactions on Cybernetics*, 44(10), 1858-1870.

9 NOTES ON CONTRIBUTORS

Lihua Guo received B.S. and M.S. degrees at Nanjing University of Posts and Telecommunications (NUPT) in 1999, and 2002, and PH.D. degree at Shanghai Jiao-Tong University in 2005. He is a associate professor at the South China University of Technology. His research interests are image understanding and pattern recognition.