Filomat 34:15 (2020), 4985–4996
https://doi.org/10.2298/FIL2015985L

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Toward Deep Neural Networks: Mirror Extreme Learning Machines for Pattern Classification

Bolin Liao[a], Chuan Ma[a], Meiling Liao[a], Shuai Li[b], Zhiguan Huang[c]

[a]*School of Information Science and Engineering, Jishou University, Jishou, 416000, Hunan, China*
[b]*Department of Electrical and Electronic Engineering, College of Engineering, Swansea University, UK*
[c]*Guangdong Provincial Engineering Technology Research Center for Sports Assistive Devices, Guangzhou Sport University, Guangzhou, 510500, China*

Abstract. In this paper, a novel type of feed-forward neural network with a simple structure is proposed and investigated for pattern classification. Because the novel type of forward neural network's parameter setting is mirrored with those of the Extreme Learning Machine (ELM), it is termed the mirror extreme learning machine (MELM). For the MELM, the input weights are determined by the pseudoinverse method analytically, while the output weights are generated randomly, which are completely different from the conventional ELM. Besides, a growing method is adopted to obtain the optimal hidden-layer structure. Finally, to evaluate the performance of the proposed MELM, abundant comparative experiments based on different real-world classification datasets are performed. Experimental results validate the high classification accuracy and good generalization performance of the proposed neural network with a simple structure in pattern classification.

1. Introduction

Pattern classification, being one of the most crucial areas of artificial intelligence, is the construction of a classification function or classification model to map a dataset to a given category [1–7]. It has widely applied in various scientific and engineering fields. Therefore, a variety of pattern classification methods are developed by scholars and researchers [2, 3, 5–8, 11]. At present, artificial neural networks have become superior methods for pattern classification because of its extraordinary system modeling characteristics, self-learning and self-adaptive ability [3, 7, 9, 10]. The common neural networks used for pattern classification include BP [12, 13], SVM [11, 14], RBF [15] neural networks and so on.

Remarkably, Huang *el at.* proposed a single hidden layer feed-forward neural network in 2004, called extreme learning machine (ELM), which is a simple and effective learning algorithm [16, 17]. Traditional neural network learning algorithms (such as BP algorithm) need to set a large number of network training parameters, and they are easy to produce a local optimal solution [18–20]. ELM only needs to set the number of hidden-layer neurons in the network. It does not need to adjust the input weights of the network, which are generated randomly, and the offset of

2010 *Mathematics Subject Classification*. 68T07.
Keywords. Mirror extreme learning machine (MELM), Weights determination, Pseudoinverse, Pattern classification, Classification datasets.
Received: 07 October 2018; Revised: 22 October 2018; Accepted: 09 November 2018
Communicated by Predrag Stanimirović
Research supported in part by the National Natural Science Foundation of China (with number 61563017), in part by the Hunan Natural Science Foundation of China (with number 2017JJ3258), and by the Research Foundation of Education Bureau of Hunan Province, China (with number 20A396).
Email addresses: mulinliao8184@163.com (Bolin Liao), machuan15@126.com (Chuan Ma), 18867217967@163.com (Meiling Liao), shuai.li@swansea.ac.uk (Shuai Li), 63649658@qq.com (Zhiguan Huang)

the hidden neurons in the process of executing the algorithm. In addition, the output weights of ELM are determined directly by the following formula [16]:

$$\hat{\beta} = \mathbf{H}^{+}\mathbf{T},$$

where the $\mathbf{H}^{+}$ denotes the pseudoinverse of matrix $\mathbf{H}$. Therefore, ELM has the advantages of fast learning and well generalization performance and has been extensively applied in many fields [16–29]. Huang *et al*. first proposed the primary concept and theory of ELM in [17]. In [24], Huang *et al*. applied ELM to the regression and multi-class classification. Wen *et al*. proposed a hybrid structure-adaptive radial basis function-extreme learning machine (HSARBF-ELM) network in [25]. Based on acoustic feature transfer learning, Deng *et al*. used ELM for recognizing emotions from whispered speech [26]. Akusok *et al*. presented and constructed a high-performance ELM toolbox for big data [27]. Wang and Han proposed a online sequential extreme learning machine with kernels (OS-ELMK) for predicting nonstationary time series [28]. Based on sample entropy, a dynamic ensemble extreme learning machine was proposed to overcome the problems of instability and over-fitting, and increase the prediction accuracy [29] Notably, the so called WASD (weights and structure-determination) neural networks are also ELM-like neural networks [3, 30, 31].

It is important to highlight the fact that the aforementioned ELM and ELM-like neural networks all generate randomly the input weights and analytically the output weights [16–31]. As a consequence, one has a question whether it is possible to exchange the determination methods of the input weights and the output weights. However, to the best of the author's knowledge, this issue has not been answered and investigated in any literature. In this paper, a new type of neural network is proposed and investigated, of which the input weights are determined by the pseudo-inverse method and the output weights are generated randomly. Because the new type of neural network happens to be a mirror image of the ELM, we term it the mirror ELM (MELM). The strict theoretical deduction is also provided to prove the feasibility of this algorithm. In addition, to obtain the optimal hidden-layer structure, we adopt the growing method. That is, the number of hidden-layer neurons increases one by one and stops increasing when the classification accuracy is not changed. Finally, abundant comparative experiments are performed to validate the performance of MELM.

The remainder of this paper is organized as follows. In Section 2, the detailed design process of MELM, including the model, the weight determination method and the structure determination method, is proposed and investigated. In Section 3, comparative experiments based on different real-world classification datasets are conducted to valuate the performance of MELM. Section 4 presents the discussions and conclusions.

2. Design of Mirror Extreme Learning Machine

In this section, the MELM model is first given. Afterwards, the formula for input weights determination is derived in theory. The growing method for structure determination is also presented in this section. Finally, the detailed description of MELM algorithm is given.

2.1. MELM Model

The generalized MELM model, which consists of multiple inputs and multiple outputs, is presented for pattern classification and shown in Fig. 1. As displayed in Fig. 1, the MELM model is similar to the traditional three-layer-structure ELM, constructed by the input layer, hidden layer and output layer. In this paper, we assume that the MELM model has J inputs and K outputs. That is, there are J neurons in the input layer and K neurons in the output layer, which are activated by a simple linear activation function. The hidden layer neurons, of which the number is M, are activated by a monotonous nonlinear activation function $f(\cdot)$.

In addition, the connection weight of the mth (with $m = 1, 2, ..., M$) hidden layer neuron to the kth (with $k = 1, 2, ..., K$) output layer neuron, which is called the output weight, is denoted by u_{mk} and randomly generated within a interval $[a1, a2]$. The connection weight of the jth (with $j = 1, 2, ..., J$) input layer neuron to the mth hidden layer neuron, which is called the input weight, is denoted by w_{jm} and determined by the pseudo-inverse method in the next section. Furthermore, the bias b_m of the mth hidden layer neuron is randomly generated in a interval $[a3, a4]$, and the

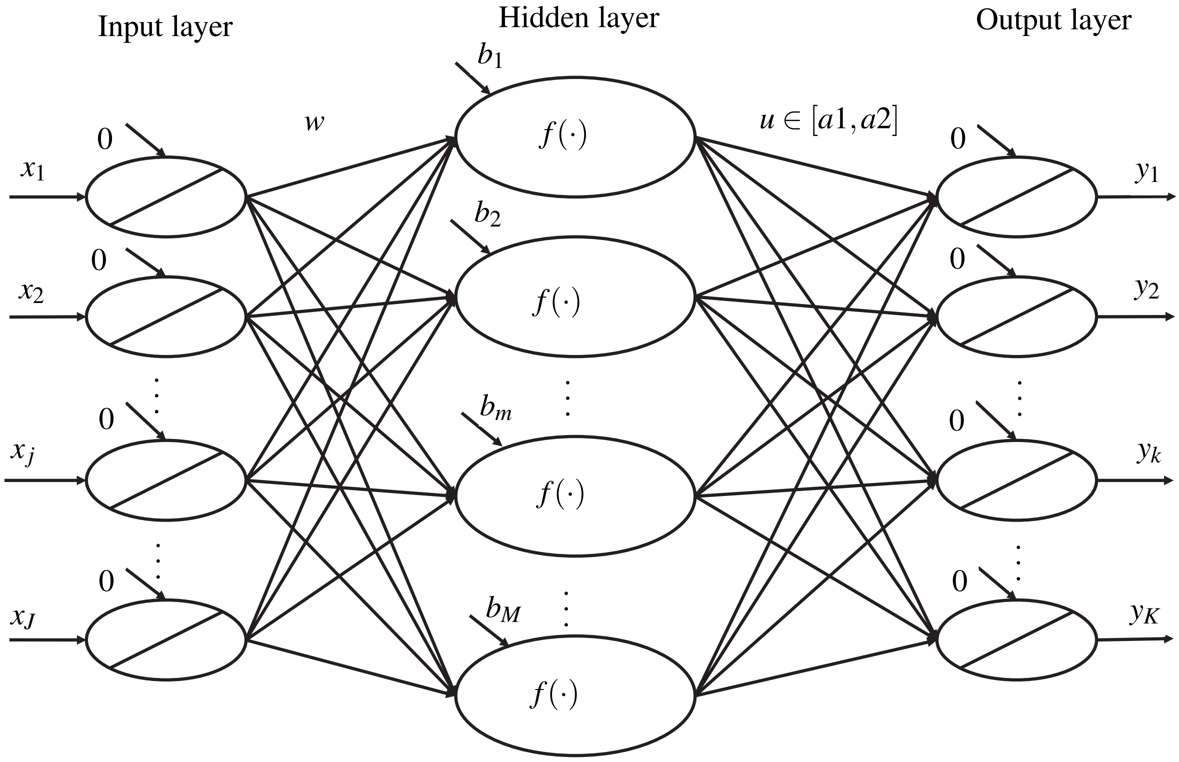

Figure 1: Model of MELM

biases of the input layer and output layer neurons can be set to zero in theory. Therefore, the output of the kth output layer neuron can be obtained as

$$y_k = \sum_{m=1}^{M} u_{km} f(\sum_{j=1}^{J} w_{mj} x_j - b_m), \tag{1}$$

where x_j corresponds to the input of the jth input-layer neuron. The compact matrix form of Eq. (1) is

$$\mathbf{y} = \mathbf{U} f(\mathbf{W}\mathbf{x} - \mathbf{b}), \tag{2}$$

where $\mathbf{y} = [y_1, \ y_2, \ \cdots, \ y_K]^{\mathrm{T}} \in \mathbb{R}^{K\times 1}$, $\mathbf{x} = [x_1, \ x_2, \ \cdots, \ x_J]^{\mathrm{T}} \in \mathbb{R}^{J\times 1}$, $\mathbf{b} = [b_1, \ b_2, \ \cdots, \ b_M]^{\mathrm{T}} \in \mathbb{R}^{M\times 1}$, $\mathbf{U}$ and $\mathbf{W}$ are output weight matrix and input weight matrix respectively. Thereinto,

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1M} \\ u_{21} & u_{22} & \cdots & u_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ u_{K1} & u_{K2} & \cdots & u_{KM} \end{bmatrix} \in \mathbb{R}^{K\times M},$$

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1J} \\ w_{21} & w_{22} & \cdots & w_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{MJ} \end{bmatrix} \in \mathbb{R}^{M\times J}.$$

Table 1: Features of different real-world classification datasets

Dataset	No. of Attributes	No. of Classes	No. of Samaples
Iris	4	3	150
Liver Disorders (LD)	6	2	345
Pima Indians Diabetes (PID)	8	2	768
Wine	13	3	178
Ionosphere	34	2	351
Glass	9	7	214
Zoo	16	7	100
WFRN	24	4	5456
SL	35	19	186

2.2. Weight Determination of MELM

In this paper, we assume that the number of distinct samples is N. Therefore, one could obtain a matrix-form output as follows:

$$\mathbf{Y} = \mathbf{U}f(\mathbf{W}\mathbf{X} - \mathbf{B}), \tag{3}$$

where the matrix-form outputs $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N] \in \mathbb{R}^{K\times N}$, the matrix-form inputs $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{J\times N}$, and the matrix-form offsets of the hidden-layer neurons $\mathbf{B} = [\mathbf{b}, \mathbf{b}, \cdots, \mathbf{b}] \in \mathbb{R}^{M\times N}$. To obtain the optimal input weights, we have the following theorem.

Theorem 2.1. *Assume that activation function $f(\cdot)$ is strictly monotonous. When output weights $\mathbf{U}$ and bias $\mathbf{B}$ are chosen from $[a1, a2]$ and $[a3, a4]$ respectively, the optimal input weights are*

$$\mathbf{W} = (f^{-1}(\mathbf{U}^{+}\mathbf{Y}) + \mathbf{B})\mathbf{X}^{+}, \tag{4}$$

where $f^{-1}(\cdot)$ denotes the unique inverse function of $f(\cdot)$.

Proof: Left multiplying $\mathbf{U}^{+}$ in both sides of Eq. (3), one can obtain

$$\mathbf{U}^{+}\mathbf{Y} = \mathbf{U}^{+}\mathbf{U}f(\mathbf{W}\mathbf{X} - \mathbf{B}) = f(\mathbf{W}\mathbf{X} - \mathbf{B}).$$

Then, solving the inverse function of the above equation, we have

$$f^{-1}(\mathbf{U}^{+}\mathbf{Y}) = \mathbf{W}\mathbf{X} - \mathbf{B}.$$

The above equation can be rewritten as

$$\mathbf{W}\mathbf{X} = f^{-1}(\mathbf{U}^{+}\mathbf{Y}) + \mathbf{B}.$$

Right multiplying $\mathbf{X}^{+}$ in both sides of the above equation, we finally obtain

$$\mathbf{W}\mathbf{X}\mathbf{X}^{+} = (f^{-1}(\mathbf{U}^{+}\mathbf{Y}) + \mathbf{B})\mathbf{X}^{+},$$

that is,

$$\mathbf{W} = (f^{-1}(\mathbf{U}^{+}\mathbf{Y}) + \mathbf{B})\mathbf{X}^{+}.$$

The proof is thus completed. ■

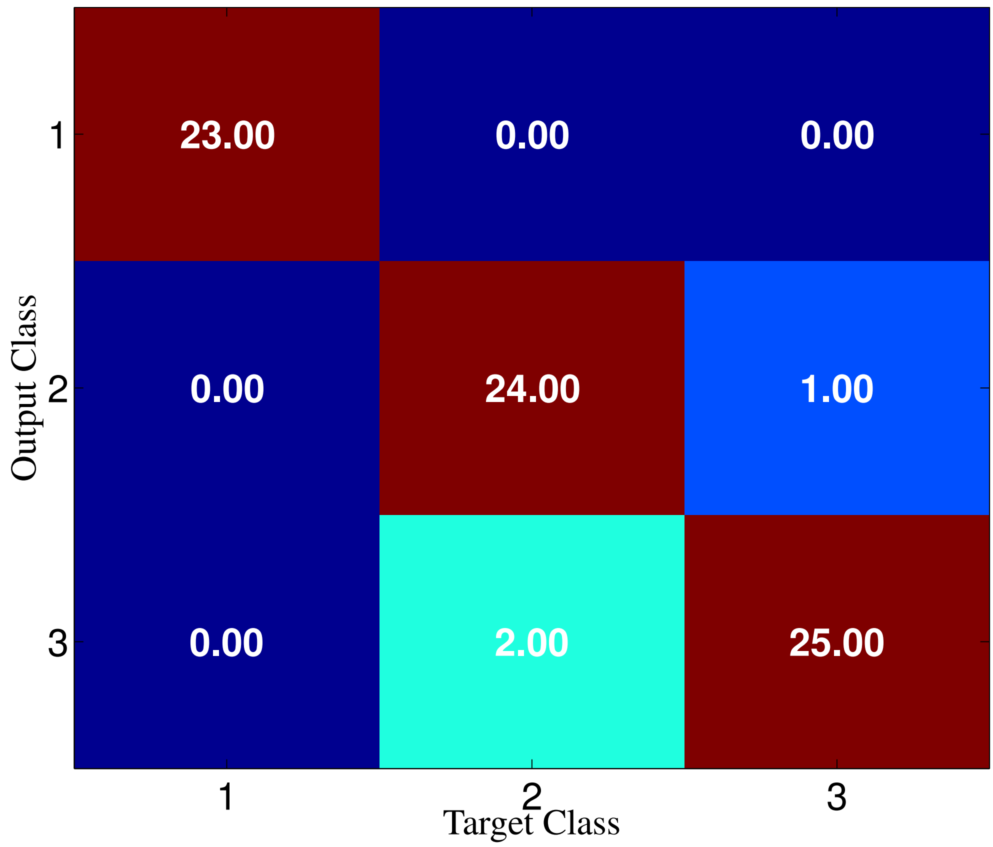

Figure 2: Training Confusion Matrix of Iris dataset.

2.3. Structure Determination of MELM

For illustrative purposes, we first define the classification accuracy CA as

$$CA = \frac{N_{\text{sc}}}{N_{\text{tc}}}$$

where N_{sc} is the number of successful classification samples, and N_{tc} is the number of total classification samples.

For the structure determination of MELM, we adopt a growing method. First, the hidden-layer neurons are increased one by one. When the classification accuracy CA no longer changes, we continue to increase one hidden-layer neuron. If the classification accuracy CA still stay the same, it is deemed the optimal CA and the training is completed. The number of hidden-layer neurons that first reached the optimal CA is the optimal number. Therefore, the structure of MELM is determined.

Based on Theorem 1 and the growing method, a simple learning algorithm for the MELM can be summarized as below.

Description of MELM:

Definition:

− J The number of input-layer neurons
− M The current number of hidden-layer neurons
− M_{opt} The optimal number of hidden-layer neurons
− CA The current classification accuracy of MELM

– $CA_{\max}$ The maximal CA found

Initialization

– Initialize $M = 1$ and $CA_{\max} = 0$

– Generate the output-layer weights $\mathbf{u}_1$ randomly within interval $[a1, a2]$

– Generate the hidden-layer bias (i.e., b_1) randomly within interval $[a3, a4]$

– Compute input-layer weight $\mathbf{w}_1$ using Eq. (4), and initialize counter $c = 0$

Step 1 Let $M \leftarrow M + 1$

Step 2 Randomly generate the output-layer weights and hidden-layer biases corresponding to the new neuron

Step 3 Compute input-layer weights using $\mathbf{W} = (f^{-1}(\mathbf{U}^{+}\mathbf{Y}) + \mathbf{B})\mathbf{X}^{+}$ and obtain the current CA of MELM

Step 4 If $CA_{\max} < CA$ let $CA_{\max} \leftarrow CA$, $M_{\text{opt}} \leftarrow M$ and return to *Step* 1. Otherwise, let $c \leftarrow c + 1$, and proceed to the next step

Step 5 If $c < 3$, return to *Step* 1. Otherwise, proceed to the next step

Step 6 Stop the procedure, let $M_{\text{opt}} \leftarrow M - 2$, and delete the corresponding output-layer weights, input-layer weights and hidden-layer biases.

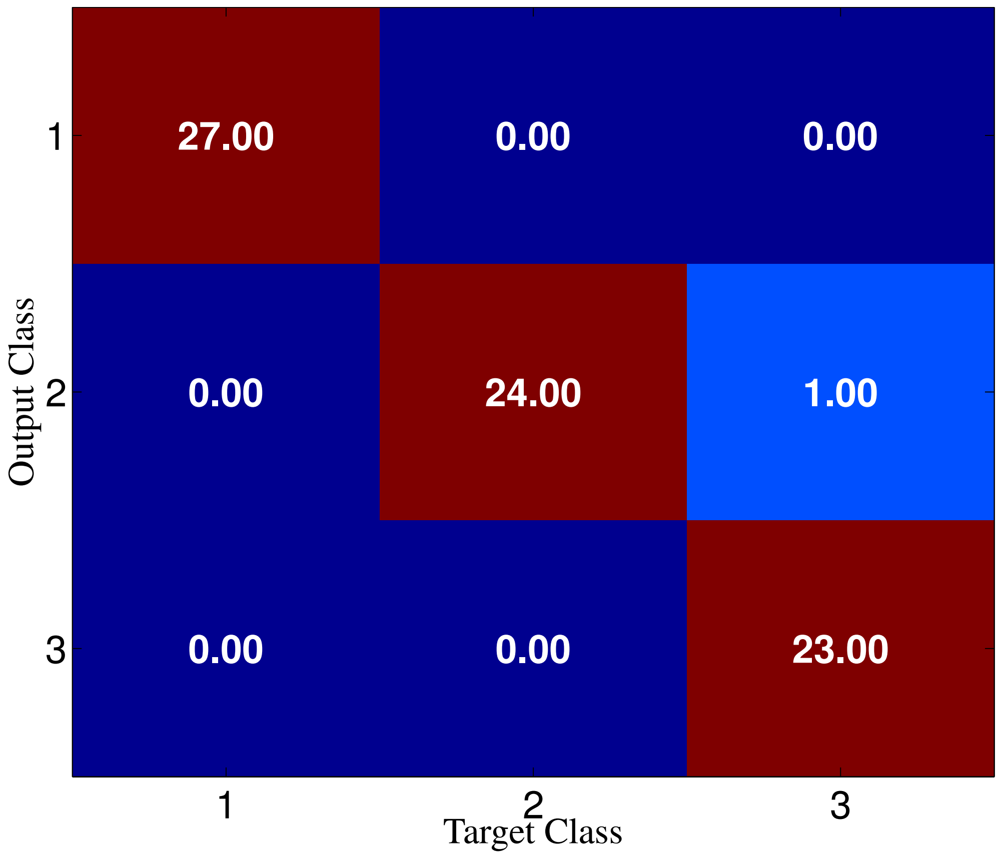

Figure 3: Test Confusion Matrix of Iris dataset.

Besides, for a better understanding, we also present the pseudo code of MELM as follows.

Algorithm 2.2 (MELM Algorithm). *1:* ***procedure*** MYPROCEDURE

2: $\mathbf{U}, \mathbf{W}, \mathbf{B} \leftarrow$ *Weight and bias of MELM*

3: $M \leftarrow$ *Number of hidden-layer neurons of MELM*

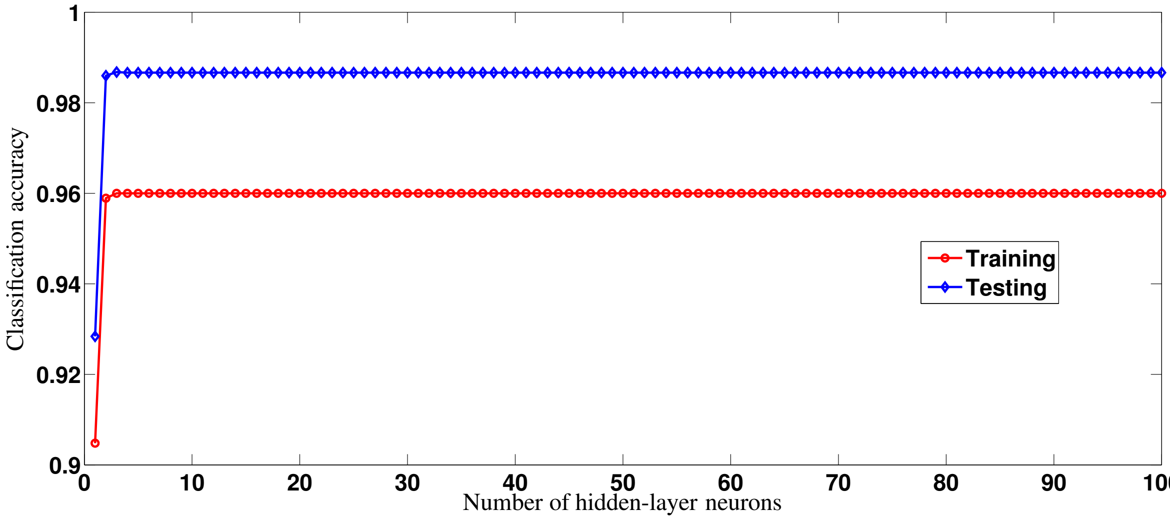

Figure 4: MELM classification accuracy for Iris dataset.

top:

  if $CA(i) <= CA_{max}$ *and* $CA(i-1) <= CA_{max}$ *and* $CA(i-2) <= CA_{max}$ ***then return*** $\mathbf{W},\ \mathbf{U},\ \mathbf{B}$

  $M \leftarrow i-2$

loop:

  $CA(i) = N_{sc}/N_{tc}$.

  if $CA(i) > CA_{max}$ ***then***

    $CA_{max} \leftarrow CA(i)$.

    $c \leftarrow 0$.

    $\mathbf{U} = randi([a1\ a2], K, i)$.

    $\mathbf{B} = randi([a3\ a4], i, N)$.

    $\mathbf{W} = (f^{-1}(\mathbf{U}^{+}\mathbf{Y}) + \mathbf{B})\mathbf{X}^{+}$.

  else if $CA(i) <= CA_{max}$ *and* $c < 3$ ***then***

    $c = c + 1$.

    $i = i - 1$.

  else

    goto top.

    close*;*

  $i \leftarrow i + 1$.

  goto loop.

3. Pattern Classification Experiments

In this section, comparative experiments for pattern classification are performed to evaluate the performance of the proposed MELM. Experimental datasets are obtained from the UCI machine learning library [32]. The features of the involved datasets are shown in Table 1. For comparative purpose, in the following experiments, half of the dataset is selected randomly for training, and the rest of the dataset is used for testing, which is the same as literature [3]. For simplicity's sake and with out loss of generality, the biases $\mathbf{B}$ of hidden-layer neurons are set as $\mathbf{0}$, the output weights $\mathbf{U}$ are randomly assigned in $[-1, 1]$. In addition, the arctan function is used as the activation function, and its inverse function is the tan function.

Table 2: Performance comparison of MELM with the number of hidden-layer neurons fixed and with the growing method for pattern classification based on the Iris dataset

MELM	Accuracy Rate (%)		
	Training	Testing	Neurons
	90.48	92.84	1
	95.89	98.60	2
	96.00	98.68	3
	96.00	98.68	4
Fixed	96.00	98.67	5
Number	96.00	98.67	10
	96.00	98.67	20
	96.00	98.67	50
	96.00	98.67	100
Growing method	96.00	**98.68**	3

Table 3: Comparison of Testing Classification Accuracy of MELM, SOCPNN-W, MOCPNN-W, MLP-ELM, MLP-LM as well as Regularized RBFNN and SVM for Pattern Classification

Dataset	Average Classification Accuracy (%) / Rank of Testing Classification Accuracy						
	MELM	SOCPNN-W [3]	MOCPNN-W [3]	MLP-ELM [3]	MLP-LM [3]	RBFNN [3]	SVM [3]
Iris	**98.68/1**	97.08/2	95.35/4	94.38/6	95.33/5	94.16/7	96.56/3
LD	**69.77/1**	66.78/2	66.78/2	58.93/5	60.34/4	57.64/7	58.90/6
PID	75.71/3	**76.29/1**	**76.29/1**	64.56/6	69.02/4	60.66/7	65.10/5
Wine	**98.89/1**	92.03/4	97.76/2	50.12/6	94.41/3	87.39/5	41.78/7
Ionosphere	86.86/2	86.30/3	86.30/3	82.90/7	85.24/6	86.19/5	**91.41/1**
Glass	**73.83/1**	47.21/6	62.09/4	45.14/7	62.89/2	57.87/5	67.76/3
Zoo	90.00/4	93.37/2	92.48/3	86.31/7	89.66/5	**95.00/1**	87.54/6
WFRN	52.38/6	46.5/7	75.73/4	67.68/5	**92.00/1**	85.88/3	88.25/2
SL	79.57/2	29.97/7	78.80/3	63.86/6	75.12/4	**84.47/1**	74.91/5
Avg. Rank	**2.33**	3.78	2.89	6.11	3.78	4.56	4.22

3.1. Pattern Classification on Iris dataset

In this subsection, the Iris dataset, of which the feature is shown in Table 1, is first utilized to check the effectiveness of MELM. The corresponding results are displayed in Fig. 2 through Fig. 4 and Table 2. Figures 2 and 3 are the confusion-matrix graphs. In Figs.2 and 3, the numbers on the diagonal of the matrix represent the number of samples classified correctly, while the numbers at other locations represent the number of samples that are misclassified. Clearly, it can be observed that the MELM accomplishes the classification task well either in testing or in training. In addition, as seen from Fig. 4, the optimal classification accuracy is around 98% and 96% in testing and in training, respectively. Note that, Fig. 4 reveals a fact that the generalization performance of MELM is very stable starting from a small number of hidden-layer neurons. This fact is also verified in Table 2. Specifically, when the number of hidden-layer neurons is 3, the classification accuracy achieves optimal and holds steady. It is worth pointing out that it is also true to other cases. Regarding Table 2, one more thing needs to be explained. Herein, to test the presented growing method, the number of hidden-layer neurons of MELM is manually tuned by an interval of 1 or automatically tuned by the growing method. As shown in the table, the presented growing method could find the optimal structure of MELM. As a consequence, the effectiveness of MELM for pattern classification is preliminary confirmed.

3.2. Performance Comparison on Different Datasets

In this subsection, the performance comparison of the proposed MELM and the other existing neural networks (including the SOCPNN-W, MOCPNN-W, MLP-ELM, MLP-LM, RBFNN and SVM) is conducted for all of the real-

Table 4: Comparison of Network Structures of MELM, SOCPNN-W, MOCPNN-W, MLP-ELM, MLP-LM as well as Regularized RBFNN and SVM for Pattern Classification

Dataset	Number of Hidden-Layer Neurons (Average Value of All Trials)/Ranking						
	MELM	SOCPNN-W [3]	MOCPNN-W [3]	MLP-ELM [3]	MLP-LM [3]	RBFNN [3]	SVM [3]
Iris	**3.00/1**	12.06/3	47.99/6	15.17/4	10.73/2	75.00/7	30.34/5
LD	**4.00/1**	11.80/2	23.60/5	13.58/3	14.08/4	173.00/7	172.88/6
PID	**6.00/1**	12.44/2	24.88/5	15.78/4	12.78/3	384.00/6	384.00/6
Wine	**3.00/1**	16.68/3	48.67/5	20.48/4	12.54/2	90.00/6	90.00/6
Ionosphere	**4.00/1**	11.30/3	22.60/4	24.76/5	10.95/2	176.00/7	89.35/6
glass	**9.00/1**	10.01/2	70.12/5	18.85/4	15.92/3	109.00/7	99.59/6
Zoo	14.00/2	15.91/3	76.51/7	18.16/4	**10.82/1**	52.00/6	39.15/5
WFRN	**22.00/1**	74.36/4	399.70/5	57.16/3	22.74/2	2729.00/7	1562.43/6
SL	**9.00/1**	33.24/4	471.37/7	22.20/3	14.89/2	97.00/6	95.83/5
Avg. Rank	**1.11**	2.89	5.44	3.78	2.33	6.56	5.67

world classification datasets shown in Table 1 [3]. Meanwhile, to avoid the influence generated by the randomness in the setting process of initial parameters, 100 trials are performed for all the algorithms and the average results are displayed in Tables 3 and 4.

As seen from Table 3, one can find that the MELM achieves the highest or second highest testing classification accuracy for most of the datasets. To comprehensive rank the generalization performances of these seven neural networks, the average-rank ranking method is utilized [33], in which a smaller number means better generalization performance. It is seen from Table 3 that the average rank of MELM is 2.00 (a smallest number), which implies that the MELM performs the best in pattern classification with respect to all datasets among the seven neural networks.

In addition, the detailed numbers of hidden-layer neurons are displayed in Table 4. Table 4 illustrates an important fact: the number of hidden-layer neurons of MELM is the least among the seven neural networks. That is say, the MELM has the simplest structure, which also means that the computational complexities of MELM is lower than the other six neural networks.

In summary, the above experimental results demonstrate the fantastic generalization performance and the extremely simple structure of the proposed MELM on pattern classification.

3.3. Regularized MELM

To improve the generalization performance of ELM and to make the solution more robust, Huang *et al.* proposed a regularized ELM [34]. After adding a regularization term, the output weight of ELM is determined by the following formula [34]:

$$\beta^* = (\frac{\mathbf{I}}{c} + \mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{T},$$

where $\mathbf{I}$ denotes an identity matrix and c is a constant that needs to be set by the users. The formula and related theories have been fully proved to be effective for the ELM regularization [34]. Same as above, we derive the expression for the regularization input weight of MELM as follows:

$$\mathbf{W} = (f^{-1}(\mathbf{U}^*\mathbf{Y}) + \mathbf{B})\mathbf{X}^*,$$

and

$$\mathbf{U}^* = (\frac{\mathbf{I}_1}{c} + \mathbf{U}^{\mathrm{T}}\mathbf{U})^{-1}\mathbf{U}^{\mathrm{T}},$$

$$\mathbf{X}^* = (\frac{\mathbf{I}_2}{c} + \mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}},$$

Table 5: Performance of regularized MELM at different constant c values.

Datasets		Classification Accuracy (%)			
		c=1	c=10	c=100	c=1000
Iris	Test	82.67	94.67	98.67	98.67
	Train	88.00	96.00	96.00	96.00
LD	Test	62.21	69.77	69.77	69.77
	Train	59.54	71.68	68.79	67.33
PID	Test	74.18	76.04	75.78	75.78
	Train	71.35	75.78	75.00	75.00
Wine	Test	38.2	91.01	96.63	98.88
	Train	41.57	87.64	89.89	91.00
Lonsphere	Test	87.47	87.38	86.86	86.89
	Train	90.65	90.30	90.34	89.76
Glass	Test	64.69	71.79	74.77	73.83
	Train	60.75	66.36	67.29	67.29
Zoo	Test	46.00	90.00	90.00	90.00
	Train	50.98	86.27	88.24	88.24
WFRN	Test	52.16	52.31	52.38	52.38
	Train	51.39	51.25	51.28	51.39
SL	Test	26.88	79.57	79.57	79.57
	Train	21.51	87.10	91.40	91.40

where $\mathbf{I}_1$ is a $M \times M$ identity matrix, $\mathbf{I}_2$ is a $J \times J$ identity matrix. The effects of regularized MELM are shown in the Table 5. As seen from Table 5, when the constant c is large enough, the regularized MELM can achieve remarkable classification accuracy with small number of hidden-layer neurons, which further reflects that the regularized MELM has superior generalization performance and good stability.

4. Discussions and Conclusions

In this paper, a novel neural network, called MELM, has been proposed and investigated for pattern classification. Compared with the existing neural networks, the MELM has several interesting and important characteristics, which are summarized as below.

1) The MELM adopts a novel idea for the weight determination. That is, the input weights are tuned analytically by the pseudoinverse method, and the output weights are assigned randomly. This is not only a complement to the ELM in mathematics, but it also brings some other advantageous performance.
2) The growing method is employed for the structure determination. It together with the weight determination method leads to a more simple structure of MELM than the existing neural networks.
3) The MELM does not degrade the generalization performance of feed-forward neural network. On the contrary, the MELM possesses better classification accuracy for pattern classification.

The weight determination formula has been derived and proved in theory. Besides, the learning algorithm for the MELM has been provided. Abundant experimental results based on various real-world classification datasets have demonstrated that the MELM possesses the superior generalization performance and a simple structure. If MELM is applied to other problems such as regression fitting, there may be surprising results. In addition, it is worth pointing out that systematic investigations on the computational complexity and numerical stability of MELM can be a future research direction of the work.

Acknowledgment

Kindly note that Shuai Li is the corresponding author. The authors also thank the editors and reviewers for the time and effort they spent in providing constructive comments to improve the quality of this paper.

References

[1] R. Duda, P. Hart, and D. Stork, Pattern Classification, John Wiley & Sons, New York, 2012.
[2] G. Ou and Y.L. Murphey, Multi-class pattern classification using neural networks, Pattern Recognit., 40 (1) (2007) 4–18.
[3] Y. Zhang, Y. Yin, D. Guo, X. Yu, and L. Xiao, Cross-validation based weights and structure determination of Chebyshev-polynomial neural networks for pattern classificationg, Pattern Recognit., 47 (10) (2014) 3414–3428.
[4] Á. Morera, Á. Sánchez, J.F. Vélez, and A.B. Moreno, Gender and Handedness Prediction from Offline Handwriting Using Convolutional Neural Networks, Complexity, 2018 (2018) 1–14.
[5] J. Vitola, F. Pozo, D. A. Tibaduiza, and M. Anaya., A Sensor data fusion system based on k-nearest neighbor pattern classification for structural health monitoring applications, Sensors, 17 (2) (2017) 1–26.
[6] H. Nakayama and N. Kagaku, Pattern classification by linear goal programming and its extensions, J. Global Optim., 12 (2) (1998) 111–126.
[7] H. Zhang, J. Liu, D. Ma, and Z. Wang, Data-core-based fuzzy min-max neural network for pattern classification, IEEE Trans. Neural Netw., 22 (12) (2011) 2339–2352.
[8] G. Parthasarathy and B. N. Chatterji, A class of new KNN method for low sample problems, IEEE Trans. Syst., Man, Cybern., 20 (3) (1990) 715–718.
[9] B. Du, Design of exponential state estimators for neutral-type neural networks with mixed time delays, Filomat, 30 (13) (2016) 3435–3449.
[10] X. Tao, Q. Niu, M. Jackson, M. Ratcliffe, Classification of video lecture learners' cognitive and negative emotional states using a Bayesian belief network, Filomat, 32 (5) (2018) (In press).
[11] C. Cortes and V. Vapnik, Support vector networks, Mach. Learn., 20 (3) (1995) 273–297.
[12] D. E. Rumelhart and J. L. McClelland, Parallel Distributed Processing, MIT Press, Cambridge, 1986.
[13] W. Zhou, Verification of the nonparametric characteristics of backpropagation neural networks for image classification, IEEE Trans. Geosci. Remote Sens., 37 (2) (1999) 771–779.
[14] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn., 46 (1-3) (2002) 389–422.
[15] A.G. Bors and M. Gabbouj, Minimal topology for a radial basis functions neural network for pattern classification, Digit. Signal Process., 4 (3) (1994) 173–188.
[16] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, Extreme learning machine: A new learning scheme of feedforward neural networks, Proc. IEEE Int. Joint Conf. Neural Netw., 2 (2004) 985–990.
[17] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, Extreme learning machine: Theory and appications, Neurocomuting, 70 (1-3) (2006) 489–501.
[18] G. Feng, G.-B. Huang, Q. Lin, and R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, IEEE Trans. Neural Netw., 20 (8) (2009) 1352–1357.
[19] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw., 17 (6) (2006) 1411–1423.
[20] S. Li, Z.-H. You, H. Guo, X. Luo, and Z.-Q. Zhao, Inverse-free extreme learning machine with optimal information updating, IEEE Trans. Cybern., 46 (5) (2015) 1229–1241.
[21] Z. Huang, Y. Yu, J. Gu, and H. Liu, An efficient method for traffic sign recognition based on extreme learning machine, IEEE Trans. Cybern., 47 (4) (2017) 920–933.
[22] S. Ding, H. Zhao, Y. Zhang, X. Xu, and R. Nie, Extreme learning machine: algorithm, theory and applications, Artif. Intell. Rev., 44 (1) (2015) 103–115.
[23] J. Cao, Z. Lin, and G.-B. Huang, Composite function wavelet neural networks with extreme learning machine, Neurocomputing, 73 (7-9) (2010) 1405–1416.
[24] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst., Man, Cybern. B, Cybern., 42 (2) (2012) 513–529.
[25] H. Wen, H. Fan, W. Xie, and J. Pei, Hybrid structure-adaptive RBF-ELM network classifier, IEEE Access, 5 (2017) 16539–16554.
[26] J. Deng, S. Frühhplz, Z. Zhang, and A. B. Schuller, Recognizing emotions from whispered speech based on acoustic feature transfer learning, IEEE Access, 5 (2017) 5235–5246.
[27] A. Akusok, K.-M. Björk, Y. Miche, and A. Lendasse, High-performance extreme learning machines: a complete toolbox for big data applications, IEEE Access, 3 (2015) 1011–1025.
[28] X. Wang and M. Han, Online sequential extreme learning machine with kernels for nonstationary time series prediction, Neurocomputing, 145 (2014) 90–97.
[29] J.-H. Zhai, H.-Y. Xu, and X.-Z. Wang, Dynamic ensemble extreme learning machine based on sample entropy, Soft Comput., 16 (9) (2012) 1493–1502.
[30] Y. Zhang, D. Guo, Z. Luo, K. Zhai and H. Tan, CP-activated WASD neuronet approach to Asian population prediction with abundant experimental verification, Neurocomputing, 198 (2016) 48–57.
[31] Y. Zhang, X. Yu, D. Guo, Y. Ying and Z. Zhang, Weights and structure determination of multiple-input feedforward neural network activated by Chebyshev polynomials of Class 2 via cross-validation, Neural Comput. Applic., 25 (7-8) (2014) 1761–1770.
[32] C. L. Blake and C. J. Merz, UCI Repository of Machine Learning Databases, Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, 1998. [Online] Available: http://www.ics.uci.edu/ mlearn/MLRepository.html

[33] P. B. Brazdil and C. Soares, A comparison of ranking methods for classification algorithm selection, Proc. Eur. Conf. Mach. learn., (2000) 63–75.
[34] G. Huang, G.-B. Huang, S. Song and Y. You, Trends in extreme learning machines: A review, Neural Netw., 61 (2015) 32–48.