IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 2, 2019, accepted March 11, 2019, date of publication March 14, 2019, date of current version April 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2905077

Random Search Enhancement of Incremental Regularized Multiple Hidden Layers ELM

JINGYI LIU[1], XINXIN LIU[2], CHONGMIN LIU[2], BA TUAN LE[2,3], AND DONG XIAO[2]

[1]College of Sciences, Northeastern University, Shenyang 110819, China
[2]Information Science and Engineering School, Northeastern University, Shenyang 110819, China
[3]NTT Hi-Tech Institute, Nguyen Tat Thanh University, Ho Chi Minh City 700000, Vietnam

Corresponding author: Dong Xiao (xiaodong@ise.neu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0304100, in part by the National Natural Science Foundation of China under Grant 71672032, and in part by the Fundamental Research Funds for Central University under Grant N180404012 and Grant N182608003.

ABSTRACT The extreme learning machine (ELM) represents one of the most successful approaches in the field of machine learning recently, especially in solving classification and regression problems. A key advantage of the multiple hidden layers' ELM (MELM) is that the computational time required to train the neural network is significantly lower because it uses random selection and analytical solution, respectively, to determine the weights of the hidden nodes and output nodes. However, due to the use of too many or too few hidden nodes during the training process, the phenomenon of over-fitting or under-fitting may occur in the prediction process. Aiming at the design of MELM neural network architecture, this paper applies the enhanced random search method to the MELM network model and proposes an incremental MELM training algorithm based on the Cholesky decomposition, namely, random search enhancement of incremental regularized MELM (EIR-MELM). The algorithm automatically determines the optimal MELM network structure by increasing the hidden nodes one by one and calculates the output weights by flexibly adopting the Cholesky decomposition method, which effectively reduces the computation burden caused by the incremental process of the hidden layer neurons. However, some hidden nodes added to the network may only have a weak influence on the final output of the network. Adding randomly generated nodes directly to the network only increases the complexity of the neural network structure. Therefore, in the process of adding hidden nodes, EIR-MELM adds a selection phase. According to the principle of structural risk minimization, the optimal node is selected from multiple randomly generated nodes to be added to the network, so that EIR-MELM has a more compact network structure. The experimental researches on the benchmark datasets for classification problems show that EIR-MELM can effectively determine the optimal MELM network structure automatically with high calculation efficiency.

INDEX TERMS Extreme learning machine, multiple hidden layers, incremental learning procedures, Cholesky decomposition, random search enhancement.

NOMENCLATURE

N	The total number of training samples.
X	Set of input samples.
T	Set of labeled samples.
M	The number of hidden layers.
$L_{\min}$	The minimum number of hidden nodes
$L_{\max}$	The maximum number of hidden nodes.
L	The number of used hidden nodes, which is used as a subscript in the following symbols to indicate there are L hidden nodes in the corresponding hidden layer.
$B_{1,L}$	The bias matrix of the first hidden layer having L hidden nodes.
$B_{2,L}$	The bias matrix of the second hidden layer having L hidden nodes.
$B_{3,L}$	The bias matrix of the third hidden layer having L hidden nodes.

The associate editor coordinating the review of this manuscript and approving it for publication was Huanqing Wang.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

$W_{1,L}$ The input weight matrix that links the input layer to the first hidden layer having L hidden nodes.

$W_{2,L}$ The matrix of connection weights between the first hidden layer and the second hidden layer having L hidden nodes.

$W_{3,L}$ The connection weight matrix between the second hidden layer and the third hidden layer having L hidden nodes.

$w_{1,j}$ The weight vector that links the input nodes and the jth node in the first hidden layer.

b_{1j} The bias of the jth node in the first hidden layer.

$\beta_{1,L}$ The connection weight matrix between the first hidden layer and the output layer having L hidden nodes.

$\beta_{1,L+1}$ The connection weight matrix between the first hidden layer and the output layer having $L+1$ hidden nodes.

$\beta_{2,L}$ The connection weight matrix between the second hidden layer and output layer having L hidden nodes.

$\beta_{3,L}$ The connection weight matrix between the third hidden layer and the output layer having L hidden nodes.

$\beta_{M,L}$ The connection weight matrix between the Mth hidden layer and the output layer having L hidden nodes.

$(\beta_{1,L})_j$ The j-th row of the matrix $\beta_{1,L}$, which indicates the connection weights between the jth node in the first hidden layer and the output nodes.

$\beta_{1,L}^{+}$ The Moore-Penrose (MP) generalized inverse of the connection weight matrix $\beta_{1,L}$.

$\beta_{2,L}^{+}$ The MP generalized inverse of the connection weight matrix $\beta_{2,L}$.

$H_{1,L}$ The output matrix of the first hidden layer having L hidden nodes.

$H_{1,L+1}$ The output matrix of the first hidden layer having $L+1$ hidden nodes.

$H_{2,L}$ The prediction output matrix of the second hidden layer having L hidden nodes.

$H_{3,L}$ The prediction output matrix of the third hidden layer having L hidden nodes.

$H_{M,L}$ The prediction output of the Mthhidden layer having L hidden nodes.

$H_{1,L}^{+}$ The MP generalized inverse of the matrix $H_{1,L}$.

$H_{2,L}^{+}$ The MP generalized inverse of the matrix $H_{2,L}$.

$H_{2^*,L}$ The expected output matrix of the second hidden layer having L hidden nodes.

$H_{3^*,L}$ The expected output matrix of the third hidden layer having L hidden nodes.

$W_{2HE,L}$ The learning parameter of the second hidden layer having L hidden nodes.

$W_{3HE,L}$ The learning parameter of the third hidden layer having L hidden nodes.

$W_{MHE,L}$ The learning parameter of the Mth hidden layer having L hidden nodes.

I_L The L-order identity matrix, where L is the number of hidden nodes.

I_{L+1} The L+1-order identity matrix, where L is the number of hidden nodes.

ξ_L The expected learning accuracy used to determine the optimal number of hidden nodes, where L is the number of hidden nodes.

ξ_M The expected learning accuracy used to determine the optimal number of hidden layers, where M is the number of hidden layers.

$R_{1,L}$ The cost function of the EIR-MELM prediction model having L hidden nodes and one hidden layer.

$R_{M,L}$ The cost function of the EIR-MELM prediction model having M hidden layers and L hidden nodes.

$t(x)$ The final output of EIR-MELM neural network.

I. INTRODUCTION

Neural networks have been extensively used in many fields due to their capabilities to approximate complex nonlinear mappings directly from the input samples. There are many different kinds of common network model, such as BP neural network [1], RBF neural network [2] and Hopfield neural network [3]. The strong learning abilities of the neural networks are achieved through the propagation of information between neurons [4]. Seen from the viewpoint of the direction of the neural network internal information transfer, two main neural networks have been investigated: feedforward type neural network and feedback type neural network. Extreme Learning Machine (ELM) described in this paper is a novel single hidden layer feedforward neural networks (SLFNs) algorithm [5]–[7]. According to the network architectures, SLFN network architectures can be divided into two categories: the SLFNs with additive hidden nodes and radial basis function (RBF) networks [8], [9] which use RBF nodes in the hidden layer.

Extreme Learning Machine (ELM) is an ideal regression and classification algorithm due to its fast training speed and better generalization performance. It mainly has the following characteristics. First of all, in the process of training and testing, ELM is able to approximate the tag variable of arbitrarily complex and nonlinear small set [10], [11], and the error is almost close to zero. Secondly, ELM can generate a unique optimal solution to avoid falling into the local optimality during the training and testing of the neural network models. Finally, a key advantage of ELM is that the computational time required for training the neural networks is significantly lower, because it uses random selection and analytical solutions to determine the weights of hidden nodes and output nodes, respectively. In order to further improve the performance of ELM, the regularization extreme learning machine (RELM) is put forward in [12] from the principle

of structural risk minimization in statistical learning theory. By introducing regularization parameters to weigh structural risks and empirical risks, it allows ELM to have better generalization ability [13], [14].

However, how to obtain the appropriate number of hidden layer neuron nodes remains a challenging task. If too many or too few hidden nodes are used in the training process, it may lead to the phenomenon of over-fitting or under-fitting during the prediction process. To design the neural network architecture of ELM, the incremental Extreme Learning Machine (I-ELM) is proposed in [15]. Different from the traditional neural network theory, this algorithm adopts an incremental form to add hidden nodes to the network one by one. The function of SLFNs as a universal approximation can be maintained by simply selecting the number of hidden nodes and properly adjusting the connection weight matrix between the hidden layer and the output layer. During the execution of this improved ELM, the establishment of the network model is completely automatic, and users do not need to intervene in the learning process by manually tuning control parameters. However, when the new hidden nodes are added to existing networks, retraining the network model will take a significant amount of training time. To solve the above problems, Feng *et al.* [16] proposed a simple and effective method to automatically determine the number of hidden nodes of ELM, which is called error minimized extreme learning machine (EM-ELM). This method can add randomly generated hidden nodes to ELM one by one or group by group. For the added group, its size can be arbitrarily changed. In the growth of ELM network structure, the connection weight matrix between the hidden layer and the output layer is gradually updated in a progressive incremental manner. However, recent research has shown that some hidden nodes added to the network may only have a weak impact on the final network output, and adding randomly generated nodes directly to the model only increases the complexity of the structure. Therefore, Lan *et al.* [17] proposed an enhanced EM-ELM based on the random search method, namely EEM-ELM. The algorithm adds a selection phase in the process of adding hidden nodes. According to the principle of error minimization, the optimal node is selected from a number of randomly generated nodes and added to the network. Compared to EM-ELM, EEM-ELM has a more compact network structure. However, EEM-ELM still has two problems in need of solution. Firstly, the initial hidden layer output matrix may not be a full rank matrix, which will affect the accuracy of the calculation results. Secondly, due to over-fitting, EM-ELM method cannot always maintain better generalization performance. In order to effectively avoid the above problems, an improved EEM-ELM is proposed in [18] based on the regularization method, called the enhancement of incremental regularized extreme learning machine (EIR-ELM). In each step of the learning process to update the network structure, multiple hidden nodes are randomly generated primarily, and then the optimal hidden nodes are selected and added to the existing network according to the principle of error minimization. When new hidden nodes are added one after another, EIR-ELM is always able to recursively update the connection weight matrix utilizing a fast computation format. The number of hidden nodes determines the learning accuracy and generalization ability of the regularization method, and it is also a key factor that must be confirmed in advance when designing the RELM network structure. In order to avoid the disadvantages and difficulties in artificially selecting the number of hidden nodes, Zhang and Wang [19] proposed an incremental RELM training algorithm based on Cholesky decomposition (CF-RELM), which can automatically seek the optimal number of hidden nodes. The algorithm calculates the output weight by utilizing Cholesky decomposition approach, and has the advantages of high prediction accuracy and fast calculation speed, which is applicable to chaotic time series prediction. In order to further expand the applicable scope of the regularization method, an improved incremental RELM (II-RELM) is put forward in [20]. The algorithm can automatically search the optimal network structure by gradually adding new hidden nodes one by one, and update the connection weight matrix with less calculation cost and higher accuracy. The neural network generalized inverse (NNGI) based on II-RELM is applied to two-motor synchronous decoupling control. The simulation indicates that the proposed algorithm has excellent performance in predictive control. It realizes the decoupling control between velocity and tension.

When dealing with input data with complex noise signals and high-dimensional information, or with more categories, the accuracy of the model established by traditional ELM is greatly declines. The literature [21] starts from the improvement of its network structure. On the basis of the traditional ELM three-layer structure, the number of hidden layers is increased to form a neural network with one input layer, multiple hidden layers and one output layer, namely multiple hidden layers ELM (MELM). MELM inherits the idea that ELM randomly initializes the weight matrix and bias vector of the first hidden layer, and calculates the parameters of the newly added hidden layer by forcing the actual output of the hidden layer to be as close as possible to the expected output, thus build a neural network model with multiple hidden layers. Compared with the traditional ELM model, MELM can effectively improve the prediction accuracy by optimizing the transmission of network parameters layer by layer between different hidden layers, and has the advantages of proper approximate accuracy, high convergence speed and not easily falling into local optimization. Despite the way in which MELM randomly initializes the first hidden layer parameters avoids the fact that the algorithm falls into local optimum and over-fitting, but it also leads to the failure of some hidden nodes in the model or the reduction of their effect on the neural network, resulting in a large number of redundant neuron nodes in the hidden layer. In addition, due to the complexity of various training samples actually applied in the training of MELM prediction model, it is difficult to accurately give the optimal number of hidden nodes based on

experience alone, so that MELM prediction model not only has enough hidden nodes to ensure its learning accuracy, but also has as few hidden nodes as possible to maintain its simple network structure, which often requires more reasonable methods and theories for the selection of hidden nodes.

In order to realize the effective design of the MELM network structure, simplify the calculation process and achieve the desired accuracy requirements while reasonably selecting the number of hidden nodes, the enhanced random search method is applied to the MELM network model. In this paper, a recursive solution method for connection weight matrix based on Cholesky decomposition is proposed, namely random search enhancement of incremental regularized multiple hidden layers ELM (EIR-MELM). The algorithm can adjust the number of hidden nodes in the network and determine the optimal network structure adaptively according to the prediction data. In the process of increasing the number of hidden nodes, a selection phase is added. According to the principle of structural risk minimization, the optimal nodes are selected from the randomly generated multiple hidden nodes and added to the network, so that the EIR-MELM has a more compact network structure. The benchmark datasets of classification problems for different activation functions of hidden nodes are used for empirical research. The results show that compared with traditional MELM and other popular machine learning methods, EIR-MELM can produce rapid response and robust prediction accuracy on a variety of complex training datasets.

The rest of this paper is organized as follows: Section 2 presents a brief review of the basic concepts and related work of multiple hidden layers ELM, Section 3 describes the proposed EIR-MELM technique, Section 4 reports and analyzes the experimental results, and finally, Section 5 summarizes key conclusions of the present study.

II. BRIEF REVIEW OF MULTILAYER EXTREME LEARNING MACHINE (MELM)

MELM tries to find a mapping relationship that makes the output predicted by the extreme learning machine neural network with multiple hidden layers infinitely close to the actual given result. This mapping relationship will be specifically reflected in the solution process of weight matrix and bias vector of the hidden layer. In the training process of network parameters, the number of hidden layers of MELM neural network needs to be selected according to the change of predicted data. In order to ensure that the final hidden layer output is close to the expected hidden layer output, except that the parameters of the first hidden layer are randomly initialized, the parameters training process start from the second hidden layer until all the parameters in the network are calculated. In addition, during the process of solving the model, the weight matrix and the bias matrix of each hidden layer are acquired and recorded to calculate the final predicted output result of the MELM neural network. In the following algorithm flow, the solving process of the network parameters will be explained in detail.

Consider N arbitrary distinct training samples $\{X, T\} = \{x_i, t_i\}$ $(i = 1, 2, \cdots, N)$, there is an input sample $X = [x_1, x_2, \cdots, x_N]^T$ and a desired matrix $T = [t_1, t_2, \cdots, t_N]^T$ composed of labeled samples, where $x_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^T \in R^n$ and $t_i = [t_{i1}, t_{i2}, \cdots, t_{im}]^T \in R^m$. Let L denote the number of hidden nodes with activation function $g(x)$. Meanwhile, it is assumed that all hidden layers in MELM model contain the same number of hidden nodes. During the execution of the MELM algorithm, the multiple hidden layers in the network are first treated as a single hidden layer, and then the parameters of the hidden layer in MELM network containing only a single hidden layer are randomly initialized, that is, the input weight matrix $W_{1,L} = [w_{1,1}, w_{1,2}, \cdots, w_{1,L}]^T \in R^{L\times n}$ that links the input layer to the first hidden layer, and the bias vector $B_{1,L} = [b_{11}, b_{12}, \cdots, b_{1L}]^T \in R^{L\times N}$ of the first hidden layer nodes. So $H_{1,L} \in R^{N\times L}$ is the output matrix of the first hidden layer, it can be calculated as follows,

$$\begin{aligned} H_{1,L} &= g(W_{1,L}X + B_{1,L}) \\ &= \begin{bmatrix} g(w_{1,1}x_1 + b_{11}) & \cdots & g(w_{1,L}x_1 + b_{1L}) \\ \vdots & & \vdots \\ g(w_{1,1}x_N + b_{11}) & \cdots & g(w_{1,L}x_N + b_{1L}) \end{bmatrix} \\ &= [\, h_{1,1} \cdots h_{1,L} \,] \end{aligned} \tag{1}$$

and whose scalar entries $(h_{1,j})_i = g(w_{1,j}x_i + b_{1j})$, $(i = 1, 2, \cdots, N, j = 1, 2, \cdots, L)$ may be interpreted as the output of the jth node in the first hidden layer with respect to x_i, where $h_{1,j} = \left[g(w_{1,j}x_1 + b_{1j}) \cdots g(w_{1,j}x_N + b_{1j}) \right]^T$, $(j = 1, 2, \cdots, L)$,$w_{1,j} = [(w_{1,j})_1, (w_{1,j})_2, \cdots, (w_{1,j})_n]^T$ is the matrix of connection weights between n input nodes and the jth nodes in the first hidden layer, and where b_{1j} is the bias of the jth node in the first hidden layer. Finally, the matrix-vector product $w_{1,j}x_i$ should be interpreted as the inner product between matrix $w_{1,j}$ and vector x_i.

The unique parameter to be calculated in the MELM is the connection weight matrix $\beta_{1,L}$ between the first hidden layer and the output layer, and which is detailed as $\beta_{1,L} = [(\beta_{1,L})_1, (\beta_{1,L})_2, \cdots, (\beta_{1,L})_L]^T \in R^{L\times m}$, with the vector components $(\beta_{1,L})_j = [(\beta_{1,L})_{j1}, (\beta_{1,L})_{j2}, \cdots, (\beta_{1,L})_{jm}]^T$ $(j = 1, 2, \cdots, L)$ that represent the connection weight vector between the jth node in the first hidden layer and m output nodes. Utilizing the least-squares method it follows that

$$\beta_{1,L} = H_{1,L}^{+}T \tag{2}$$

where $H_{1,L}^{+}$ is the Moore-Penrose (MP) generalized inverse of the matrix $H_{1,L}$, which can be obtained using the orthogonal projection method. That is, if $(H_{1,L})^T H_{1,L}$ is nonsingular, then $H_{1,L}^{+} = (H_{1,L}^T H_{1,L})^{-1} H_{1,L}^T$, otherwise $H_{1,L}^{+} = H_{1,L}^T (H_{1,L} H_{1,L}^T)^{-1}$ when $H_{1,L} H_{1,L}^T$ is nonsingular. A advantage of adopting the MP method in the solution process is that the above formula gains the solution matrix $\beta_{1,L}$ of the least two-norm if $(H_{1,L})^T H_{1,L}$ is nonsingular, a valuable benefit when realizing that smaller weights may lead to better generalization performance.

Now the second hidden layer is added to the MELM neural network, restoring the structure of neural network with two hidden layers, and the two hidden layers are fully connected. The prediction output matrix $H_{2,L}$ of the second hidden layer can be obtained as

$$H_{2,L} = g(W_{2,L}H_{1,L} + B_{2,L}) \tag{3}$$

where $W_{2,L}$ is the matrix of connection weights between the first hidden layer and the second hidden layer. Since it is assumed that the first hidden layer and the second hidden layer contain the same number of neuron nodes, $W_{2,L}$ is a square matrix. After the second hidden layer is added, the notation $H_{1,L}$ denotes the output of the first hidden layer with respect to all L hidden nodes. The matrix $B_{2,L}$ is the bias of the second hidden layer.

According to the algorithm flow of MELM, it follows that the expected output matrix $H_{2^*,L}$ of the second hidden layer can be calculated as,

$$H_{2^*,L} = T\beta^+_{1,L} \tag{4}$$

where $(\beta_{1,L})^+$ is the MP generalized inverse of the connection weight matrix $\beta_{1,L}$. The calculation method of $(\beta_{1,L})^+$ is the same as the previous discussion for $H^+_{1,L}$. In order to make the predicted output of the second hidden layer in the MELM neural network infinitely close to the expected output, it can be assumed that $H_{2,L} = H_{2^*,L}$.

Subsequently we give the definition of the augmented matrix $W_{2HE,L} = [\,B_{2,L}\ W_{2,L}\,]$, which is the learning parameter of the second hidden layer and can be calculated as follow

$$W_{2HE,L} = g^{-1}(H_{2^*,L})H^+_{2E,L} \tag{5}$$

where $H^+_{2E,L}$ is the MP generalized inverse of the matrix $H_{2E,L} = [\,1\ H_{1,L}\,]^T$, $\mathbf{1}$ represents a one-column vector of size N, and whose elements are the scalar unit 1. The symbol $g^{-1}(x)$ denotes the inverse of activation function $g(x)$ of hidden nodes. The method to calculate $H^+_{2E,L}$ is discussed before.

In order to test the performance of MELM algorithm, the experiments involved different activation functions for classification and regression problems is conducted, and the widely used Logistic sigmoid function $g(x)$=1/(1 + e^{-x}) is adopted.

When the connection weight matrix $W_{2,L}$ between the first hidden layer and the second hidden layer and the bias matrix $B_{2,L}$ of the second hidden layer are all solved, we can update the predicted output matrix $H_{2,L}$ of the second hidden layer to be

$$H_{2,L} = g(W_{2,L}H_{1,L} + B_{2,L}) = g(W_{2HE,L}H_{2E,L}) \tag{6}$$

and then the connection weight matrix $\beta_{2,L}$ between the second hidden layer and output layer is calculated as

$$\beta_{2,L} = H^+_{2,L}T \tag{7}$$

where $H^+_{2,L}$ is the MP generalized inverse of the matrix $H_{2,L}$, obtained using the approach described before.

Based on the above algorithm principle, we continue to treat all the hidden layers in MELM neural network as two hidden layers, that is, the first hidden layer represents one hidden layer independently, and the subsequent hidden layers are considered as one hidden layer. The parameters of the first hidden layer including the weight matrix and the bias matrix are randomly initialized. According to the calculation process shown in the above formulas (1)-(7), the parameters and the output matrix of the second hidden layer can be obtained.

Now the third hidden layer is added to the MELM network, which is restored to the neural network structure containing three hidden layers. Since the neurons between each hidden layer are all connected together, the prediction output matrix $H_{3,L}$ of the third hidden layer can be calculated as

$$H_{3,L} = g(W_{3,L}H_{2,L} + B_{3,L}) \tag{8}$$

where $W_{3,L}$ is the connection weight matrix between the second hidden layer and the third hidden layer, and $B_{3,L}$ is the bias matrix of the third hidden layer. After the addition of the third hidden layer, $H_{2,L}$ is considered as the predicted output matrix of the second hidden layer.

When continuing the workflow of MELM algorithm, the expected output matrix $H_{3^*,L}$ of the third hidden layer can be calculated as follows

$$H_{3^*,L} = T\beta^+_{2,L} \tag{9}$$

where $\beta^+_{2,L}$ is the MP generalized inverse of the connection weight matrix $\beta_{2,L}$, which can be obtained according to the method discussed above. In order to meet the requirement that the predicted output of the third hidden layer is infinitely close to the expected output in the implementation of MELM, it can be assumed that $H_{3,L} = H_{3^*,L}$.

Given the augmented matrix $W_{3HE,L} = [\,B_{3,L}\ W_{3,L}\,]$, it is the learning parameter of the third hidden layer, where the weight matrix $W_{3,L}$ and the bias matrix $B_{3,L}$ of the third hidden layer can be solved according to the following formula.

$$W_{3HE,L} = g^{-1}(H_{3^*,L})H^+_{3E,L} \tag{10}$$

where $H^+_{3E,L}$ is the MP generalized inverse of the matrix $H_{3E,L} = [\,1\ H_{2,L}\,]^T$, which is still obtained by the method described above. $\mathbf{1}$ represents a one-column vector with N elements, where each element is the scalar unit 1. The notation $g^{-1}(x)$ is the inverse of the activation function $g(x)$ of hidden nodes. Following the above calculation process, after all the parameters of the hidden layer are solved, we can update the prediction output matrix $H_{3,L}$ of the third hidden layer as follows.

$$H_{3,L} = g(W_{3,L}H_{2,L} + B_{3,L}) = g(W_{3HE,L}H_{3E,L}) \tag{11}$$

Therefore, the connection weight matrix $\beta_{3,L}$ between the third hidden layer and the output layer can be calculated according to the formula (12).

$$\beta_{3,L} = H^+_{3,L}T \tag{12}$$

Finally, the final output $t(x)$ of MELM neural network with three hidden layers can be expressed as

$$t(x) = H_{3,L}\beta_{3,L} \tag{13}$$

In the MELM neural network, if the number of hidden layers $M \geq 3$, an iterative format can be adopted to implement the calculation process, that is, formula (3)-(7) is iteratively performed for $M - 3$ times until all hidden layer parameters are solved. Finally, in order to better improve the generalization ability of MELM neural network model and make the network prediction output more stable, it should be emphasized that the algorithm does not add all hidden layers to the network at one time, nor does it calculate all hidden layer parameters at one time, but adds one hidden layer after another hidden layer to the network. Each time a new hidden layer is added, the parameters of the newly added hidden layer, including the weight matrix and the bias matrix, are calculated immediately to prepare for the calculation of the hidden layer parameters to be added next time.

III. RANDOM SEARCH ENHANCEMENT OF INCREMENTAL REGULARIZED MULTIPLE HIDDEN LAYERS ELM (EIR-MELM)

On the basis of the training process of MELM shown in equations (1) to (13), its essence is to solve the connection weight matrix $\beta_{M,L}$ between the hidden layer and the output layer, where the subscript M is the number of hidden layers and the subscript L is the number of hidden nodes. In view of equation (2), the solution of the connection weight matrix given in [21] involves the inverse operation of the higher-order matrix, the main drawback of this approach lies in the use of a pseudo-inverse in the calculation (in the Moore-Penrose sense), which can lead to numerical instabilities if the effective training data set is not full rank. However, this is unfortunately very often the case, with real-world datasets. At the same time, during the training process of optimizing the network structure, if he number of hidden nodes L changes, it will take a large amount of computing time to retain the network, thus the modeling efficiency of MELM prediction model will be greatly reduced. The following approach proposes three improvements on the computation of the original MELM: random search enhancement, Tikhonov regularization and fast matrix calculations based on Cholesly decomposition.

A. THE SOLUTIONS OF MELM BY CHOLESKY DECOMPOSITION

Tikhonov [22] proposed a new method for solving ill-posed problems, namely regularization method. Since then, regularization theory has always been the core thought of many neural networks and machine learning algorithms. Deng *et al.* [12] successfully applied the regularization method to ELM, and further pointed out that ELM is established based on the principle of empirical risk minimization (ERM). When the sample size in the datasets is too small, ELM is prone to over-fitting. According to the theory of statistical learning [23], [24], the structural risk minimization principle (SRM), which is equivalent to regularization, is a strategy proposed to prevent over-fitting. On the basis of ERM, It adds a regularization term to control the complexity of the model. Therefore, the learning machine algorithm with good generalization performance should consider using SRM to replace ERM, and establish a model with better prediction for both training data and unknown test data. It has been proved in [25] that when the norm of the connection weight matrix $\beta_{1,L}$ is small, the network tends to have better generalization performance. Without loss of generality, we assume that the number of hidden layers $M = 1$, and the objective function $L(\beta_{1,L})$ can be minimized as follows

$$\begin{aligned}\min L(\beta_{1,L}) &= \left\|H_{1,L}\beta_{1,L} - T\right\|^2 + C\left\|\beta_{1,L}\right\|^2 \\ &= (H_{1,L}\beta_{1,L} - T)^T(H_{1,L}\beta_{1,L} - T) + C\beta_{1,L}^T\beta_{1,L}\end{aligned} \tag{14}$$

where $C > 0$ is the tradeoff parameter between $\left\|\beta_{1,L}\right\|^2$ and $\left\|H_{1,L}\beta_{1,L} - T\right\|^2$. The partial derivative of the objective function $L(\beta_{1,L})$ with respect to the variable $\beta_{1,L}$ can be obtained

$$\frac{\partial L(\beta_{1,L})}{\partial \beta_{1,L}} = -2H_{1,L}^T(T - H_{1,L}\beta_{1,L}) + 2C\beta_{1,L} \tag{15}$$

And set the partial derivative equal to zero and we get

$$2H_{1,L}^T H_{1,L}\beta_{1,L} - 2H_{1,L}^T T + 2C\beta_{1,L} = 0 \tag{16}$$

By further solving equation (16), we can obtain

$$(CI_L + H_{1,L}^T H_{1,L})\beta_{1,L} = H_{1,L}^T T \tag{17}$$

where $I_L \in R^L$ is the L-order identity matrix. When $C > 0$, CI_L is a positive definite matrix. It is easy to prove that $H_{1,L}^T H_{1,L}$ is a semi-positive definite matrix, as a result that $CI_L + H_{1,L}^T H_{1,L}$ is a positive definite matrix. And because of $CI_L + H_{1,L}^T H_{1,L}$ contains the item of CI_L, it is also a non-singular matrix. Therefore, we can get

$$\beta_{1,L} = (CI_L + H_{1,L}^T H_{1,L})^{-1} H_{1,L}^T T \tag{18}$$

The above regularization method adds L_2 penalty term to the cost function, namely L_2 regularization (Tikhonov regularization). Obviously, the traditional ELM method is just the special case of the RELM method when $C \to 0$.

On the basis of formula (17), let $A_L = CI_L + H_{1,L}^T H_{1,L}$, $B_L = H_{1,L}^T T$, then (17) can be rewritten as

$$A_L\beta_{1,L} = B_L \tag{19}$$

Consequently, the process of solving $\beta_{1,L}$ can be transformed into solving linear equations in the form of equation (19). As the premise of applying Cholesky decomposition to solving the linear equations is that its coefficient matrix must be a symmetric positive definite matrix. The above analysis process indicates that the matrix A_L satisfies the requirements of symmetry and positivity, thus the solving process of the connection weight matrix $\beta_{1,L}$ based on Cholesky decomposition can be designed as follows.

We first calculate the Cholesky decomposition result of the matrixA_L

$$A_L = U_L U_L^T \tag{20}$$

where U_L is a lower triangular matrix with positive diagonal elements. The non-zero element $(u_L)_{ij}$ in U_L can be calculated by the element $(a_L)_{ij}$ of A_L according to equation (21).

$$(u_L)_{ij} = \begin{cases} ((a_L)_{ii} - \sum_{n=1}^{i-1} (u_L)_{in}^2)^{\frac{1}{2}} & i = j, \\ ((a_L)_{ij} - \sum_{n=1}^{j-1} (u_L)_{in}(u_L)_{jn})/(u_L)_{jj} & i > j, \end{cases} \tag{21}$$

where $i = 1, \cdots, L, j = 1, \cdots, L$. By substituting equation (20) into equation (19) and multiplying both sides of the equation by U_L^{-1}, we can get the following results

$$U_L^T \beta_{1,L} = F_L \tag{22}$$

where $F_L = U_L^{-1} B_L$. So the process of solving $\beta_{1,L}$ is equivalent to solving equation (22). Since $F_L = U_L^{-1} B_L$ is equivalent to$U_L F_L = B_L$, by comparing the elements on both sides of the equation, the calculation formula of the element $(f_L)_i$ in F_L can be obtained as

$$(f_L)_i = \begin{cases} (b_L)_i/(u_L)_{ii} & i = 1, \\ ((b_L)_i - \sum_{n=1}^{i-1} (u_L)_{in}(f_L)_n)/(u_L)_{ii} & i > 1, \end{cases} \tag{23}$$

where $i = 1, \cdots, L$, $(b_L)_i$ is the element at the corresponding position of B_L. Finally, on the basis of obtaining U_L and F_L, the elements of the connection weight matrix $\beta_{1,L}$ can be calculated by the elements of U_L and F_L.

$$(\beta_{1,L})_i = \begin{cases} (f_L)_i/(u_L)_{ii} & i = L, \\ ((f_L)_i - \sum_{n=1}^{L-i} (u_L)_{i+n,i}(\beta_{1,L})_{i+n})/(u_L)_{ii} & i < L, \end{cases} \tag{24}$$

B. INCREMENTAL LEARNING PROCEDURES OF EIR-MELM

Suppose the initial number of the nodes in the first hidden layer is $L = 1$, the output matrix of the first hidden layer is $H_{1,L}$, which has L hidden nodes. And the connection weight matrix between the first hidden layer and the output layer is $\beta_{1,L}$. Compared with the solution method of $\beta_{1,L}$ shown in (8), the calculation format of $\beta_{1,L}$ based on Cholesky decomposition does not involve the inversion of the higher-order matrix, and it can be achieved only by using simple matrix four arithmetic operations. More importantly, when the number of hidden nodes in EIR-MELM increases from L to $L + 1$, the output matrix of the first hidden layer changes from $H_{1,L} \in R^{N \times L}$ to $H_{1,L+1} \in R^{N \times (L+1)}$, which can be specifically expressed as follows

$$H_{1,L+1} = \begin{bmatrix} H_{1,L} & \vdots & h_{1,L+1} \end{bmatrix} = \begin{bmatrix} h_{1,1} & \cdots & h_{1,L} & \vdots & h_{1,L+1} \end{bmatrix} \tag{25}$$

where $h_{1,j} = \begin{bmatrix} g(w_{1,j}x_1 + b_{1j}) & \cdots & g(w_{1,j}x_N + b_{1j}) \end{bmatrix}^T$, $(j = 1, 2, \cdots, L + 1)$, then we can get

$$\begin{aligned} A_{L+1} &= CI_{L+1} + H_{1,L+1}^T H_{1,L+1} \\ &= C \begin{bmatrix} I_L & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} H_{1,L}^T \\ h_{1,L+1}^T \end{bmatrix} \begin{bmatrix} H_{1,L} & h_{1,L+1} \end{bmatrix} \\ &= \begin{bmatrix} CI_L + H_{1,L}^T H_{1,L} & H_{1,L}^T h_{1,L+1} \\ h_{1,L+1}^T H_{1,L} & C + h_{1,L+1}^T h_{1,L+1} \end{bmatrix} \\ &= \begin{bmatrix} A_L & Q_{L+1} \\ Q_{L+1}^T & P_{L+1} \end{bmatrix} \end{aligned} \tag{26}$$

where $I_{L+1} \in R^{L+1}$ is the $L+1$-order identity matrix, $Q_{L+1} = \begin{bmatrix} h_{1,L+1}^T h_{1,1} & \cdots & h_{1,L+1}^T h_{1,L} \end{bmatrix}^T$, $P_{L+1} = C + h_{1,L+1}^T h_{1,L+1}$. Note that the purpose of introducing identity matrix I_{L+1} is to get a positive definite matrix which is more stable in the numerical calculation. Given the relationship between A_{L+1} and A_L as shown in formula (26), we can make the following judgment in accordance with the Cholesky decomposition process shown in (21), the $L(L + 1)/2$ non-zero elements $(u_{L+1})_{11}, (u_{L+1})_{21}, \cdots (u_{L+1})_{LL}$ are equal to the non-zero elements of U_L in the Cholesky decomposition result U_{L+1} of A_{L+1}, so there is no need to reevaluate. The matrix U_{L+1} can be obtained by calculating the $L + 1$ non-zero elements from $(u_{L+1})_{(L+1)1}$ to $(u_{L+1})_{(L+1)(L+1)}$, and the specific calculation process is as follows.

$$U_{L+1} = \begin{bmatrix} U_L & 0 \\ \bar{u}_{L+1} & (u_{L+1})_{(L+1)(L+1)} \end{bmatrix} \tag{27}$$

where $\bar{u}_{L+1} = \begin{bmatrix} (u_{L+1})_{(L+1)1} & \cdots & (u_{L+1})_{(L+1)L} \end{bmatrix}$, according to the solution methods of formula (20) and formula (26), the following conclusions can be drawn

$$\begin{aligned} &(u_{L+1})_{(L+1)j} \\ &= ((a_{L+1})_{(L+1)j} - \sum_{n=1}^{j-1} (u_{L+1})_{(L+1)n}(u_{L+1})_{jn})/(u_{L+1})_{jj} \\ &\quad \times (j = 1, 2, \cdots, L) \end{aligned} \tag{28}$$

$$\begin{aligned} &(u_{L+1})_{(L+1)(L+1)} \\ &= ((a_{L+1})_{(L+1)(L+1)} - \sum_{n=1}^{L} (u_{L+1})_{(L+1)n}^2)^{\frac{1}{2}} \end{aligned} \tag{29}$$

where $((a_{L+1})_{(L+1)j}$ $(j = 1, 2, \cdots, L + 1)$ represents the corresponding element in A_{L+1}. At the same time we can get

$$\begin{aligned} B_{L+1} &= H_{1,L+1}^T T \\ &= \begin{bmatrix} H_{1,L} & \vdots & h_{1,L+1} \end{bmatrix}^T T \\ &= \begin{bmatrix} H_{1,L}^T T \\ h_{1,L+1}^T T \end{bmatrix} = \begin{bmatrix} B_L \\ h_{1,L+1}^T T \end{bmatrix} \end{aligned} \tag{30}$$

Therefore, it can be known from the calculation method of F_L shown in equation (23)

$$F_{L+1} = \begin{bmatrix} F_L \\ f_{L+1} \end{bmatrix} \tag{31}$$

And since

$$U_{L+1}F_{L+1} = \begin{bmatrix} U_L & 0 \\ \bar{u}_{L+1} & (u_{L+1})_{(L+1)(L+1)} \end{bmatrix} \begin{bmatrix} F_L \\ f_{L+1} \end{bmatrix} = \begin{bmatrix} U_L F_L \\ \bar{u}_{L+1}F_L + (u_{L+1})_{(L+1)(L+1)} f_{L+1} \end{bmatrix} = B_{L+1} = \begin{bmatrix} B_L \\ h_{1,L+1}^T T \end{bmatrix} \tag{32}$$

we can also figure out

$$f_{L+1} = (h_{1,L+1}^T T - \sum_{n=1}^{L} (u_{L+1})_{(L+1)n}(f_L)_n)/(u_{L+1})_{(L+1)(L+1)} \tag{33}$$

Finally, $\beta_{1,L+1}$ is calculated according to equation (24).

From the calculation process described in equations (27) to (33), it can be seen that EIR-MELM adopts the fast calculation format, which can realize the incremental updating of the connection weight matrix. In conclusion, F_{L+1} can be obtained only by calculating f_{L+1}, and there is no need to recalculate from $(f_L)_1$ to $(f_L)_L$. The solution method of $\beta_{1,L+1}$ based on Cholesky decomposition makes full use of the information stored in the calculation of $\beta_{1,L}$, so that U_{L+1} and F_{L+1} can be obtained on the basis of U_L and F_L respectively. Therefore, when the number of hidden nodes increases one by one, the calculation of $\beta_{1,L+1}$ can be carried out on the basis of the calculation of $\beta_{1,L}$, and it can be quickly achieved only through simple four arithmetic operations. Under this condition, if the method shown in formula (2) is used for calculation, $\beta_{1,L+1}$ needs to be recalculated in the way of higher-order matrix inversion operation, and the solution can not be obtained on the basis of the calculation of $\beta_{1,L+1}$. Therefore, EIR-MELM can further improve the training speed while ensuring the learning accuracy. The incremental training process of EIR-MELM based on Cholesky decomposition is as follow.

Step1: The minimum number of hidden nodes in EIR-MELM is set as $L_{\min}$, and the maximum number of hidden nodes is set as $L_{\max}$. The expected learning accuracy used to determine the optimal number of hidden nodes is ξ_L. Let the number of hidden nodes $L = L_{\min}$, and calculate $A(k)_L$ and $B(k)_L$.

Step2: The Cholesky decomposition result U_L of A_L is calculated in accordance with formula (21). Besides, U_L and B_L are used to calculate F_L according to formula (23).

Step3: According to formula (24), U_L and F_L are utilized to calculate $\beta_{1,L}$. On the basis of $\beta_{1,L}$, the EIR-MELM prediction model with L hidden nodes and one hidden layer is established.

Step4: Calculate the cost function $R_{1,L}$ of the EIR-MELM prediction model as follow.

$$R_{1,L} = \left\|H_{1,L}\beta_{1,L} - T\right\|^2 + C\left\|\beta_{1,L}\right\|^2 = (H_{1,L}\beta_{1,L} - T)^T(H_{1,L}\beta_{1,L} - T) + C\beta_{1,L}^T\beta_{1,L} \tag{34}$$

Step5: Let $L = L + 1$, according to the principle shown in formula (27)-(33), calculate U_L and F_L on the basis of U_{L-1} and F_{L-1}, and then go to Step 3. Determine from the beginning of $L = L_{\min} + 4$ whether the following condition is satisfied simultaneously

$$\left|\frac{R_{1,L-i} - R_{1,L-i-1}}{R_{1,\max}}\right| \le \xi_L \tag{35}$$

where $R_{1,\max}$ is the maximum value of $R_{1,L_{\min}}, \cdots, R(k)_{1,L}$, $i = 0, \cdots, 3$. If the formula (35) is satisfied, the training process is completed, L is determined to be the optimal number of hidden nodes, and the corresponding EIR-MELM prediction model is established. Otherwise, when $L < L_{\max}$, keep increasing L until the condition $L = L_{\max}$ is satisfied.

In the incremental learning process of EIR-MELM, the number of hidden nodes increases successively from the initial value, and the expansion stops when $R_{1,L}$ is no longer significantly reduced. At this point, even if the number of hidden nodes continues to increase, the $R_{1,L}$ representing the learning accuracy and generalization ability of the EIR-MELM will not be significantly improved. Instead, it will result in a large number of redundant hidden nodes in the EIR-MELM. Therefore, the number of hidden nodes of EIR-MELM is optimal at this time.

C. PROPOSED EIR-MELM

When the number of hidden layer nodes included in the EIR-MELM network model increases one by one, the network structure of the model has changed, a simple and direct method to establish the model is to retrain the network model using the entire training data in the training dataset $\aleph$. However, such a program will inevitably lead to a serious waste of training time. Therefore, an efficient and necessary alternative method is to make full use of the information acquired in the training process of the EIR-MELM network model, the updated network parameters are directly calculated according to the network parameters obtained in the previous training process.

Consider using the training dataset $\aleph$ to complete the prediction process of data Z. First, a EIR-MELM network structure with multiple hidden layers, including one input layer, M hidden layers and one output layer, is presented. And each hidden layer contains the same number of hidden nodes, so the output of EIR-MELM network with L hidden nodes can be described as the following functional form

$$t_L(x) = \sum_{j=1}^{L} (\beta_{M,L})_j g(w_{M,j}, b_{Mj}, x), x \in R^n \tag{36}$$

Assuming that the number of training data N contained in the given training dataset $\{X, T\}$ is far greater than the number of hidden nodes L. The initial number of hidden nodes is set as $L_{\min}$, the maximum number of hidden nodes is set as $L_{\max}$, and the initial number of hidden layers is set as 1. The expected learning accuracy used to determine the optimal number of hidden nodes is set as ξ_L, and the expected learning

accuracy used to determine the optimal number of hidden layers is set as ξ_M.

Step1: Initialize the neural network phase:

1) Let $L = L_{\min}$, $M = 1$. Assign the connection weight and the bias $(w_{1,j}, b_{1j})$, $(j = 1, \cdots, L)$ of the nodes in the first hidden layer randomly.
2) Calculate the output matrix $H_{1,L}$ of first hidden layer. $H_{1,L} = H(w_{1,1}, \cdots, w_{1,L}, b_{11}, \cdots, b_{1L}, x_1, \cdots, x_N)$.
3) Calculate A_L and B_L according to equation (19).
4) Calculate the Cholesky decomposition result U_L of A_L according to the formula (21), and calculate F_L following the formula (23) using U_L and B_L.
5) Calculate $\beta_{1,L}$ according to equation (24) using U_L and F_L, and establish the EIR-MELM prediction model with L hidden nodes and one hidden layer on the basis of $\beta_{1,L}$.
6) Calculate the sum of the empirical risk and the structural risk $R_{1,L}$ in the EIR-MELM prediction model according to (34).

Step2: Update the network recursively by incremental learning procedures. While $L \leq L_{\max}$

1) Let $L = L + 1$. For each $i = 1, 2, \cdots, r$, generate a new hidden node $(w^{(1,i)}, b^{(1i)})$ randomly and add it to the existing network, where $w^{(1,i)}$ and $b^{(1i)}$ represent the connection weight and the bias of the $i-th$ hidden node, respectively. Calculate $U_L^{(i)}$ and $F_L^{(i)}$ on the basis of U_{L-1} and F_{L-1} according to the principle shown in formula (26) -(33).
2) According to formula (24), update the output weight matrix $\beta_{1,L}^{(i)}$ using $U_L^{(i)}$ and $F_L^{(i)}$.
3) Calculate the cost function $R_{1,L}^{(i)}$ of the EIR-MELM prediction model according to the formula (34). $R_{1,L}^{(i)} = \left\| H_{1,L}^{(i)} \beta_{1,L}^{(i)} - T \right\|^2 + C \left\| \beta_{1,L}^{(i)} \right\|^2$
4) Let $i^* = \left\{ i | \min\limits_{1 \leq i \leq r} R_{1,L}^{(i)} \right\}$, choose the hidden node $(w^{(1,i^*)}, b^{(1i^*)})$ that has a smallest cost function and add it to the existing network, then $w_{1,L} = w^{(1,i^*)}$, $b_{1L} = b^{(1i^*)}$, $U_L = U_L^{(i^*)}$, $F_L = F_L^{(i^*)}$, $\beta_{1,L} = \beta_{1,L}^{(i^*)}$, $R_{1,L} = R_{1,L}^{(i^*)}$. Go to Step2 (1). Start from $L = L_{\min} + 4$ to determine whether the formula (35) is satisfied. If the termination condition is met, the training process is completed and L is determined to be the optimal number of hidden nodes. Otherwise, when $L < L_{\max}$, the number of hidden nodes L continues to increase until the condition $L = L_{\max}$ is satisfied.
5) Let $M = M + 1$. Calculate the expected output matrix $H_{M*,L}$ of the Mth hidden layer according to formula (11). $H_{M*,L} = T\beta_{M-1,L}^{+}$.
6) Calculate the learning parameter $W_{MHE,L}$ of the Mth hidden layer according to the equation (13). $W_{MHE,L} = g^{-1}(H_{M*,L})H_{ME,L}^{+}$.
7) Calculate the prediction output $H_{M,L}$ of the Mth hidden layer according to formula (14). $H_{M,L} = g(W_{MHE,L}H_{ME,L})$.
8) Updated the connection weight matrix $\beta_{M,L}$ between the Mth hidden layer and the output layer according to the solving method shown in the formula (18)-(24).
9) Calculate the cost function $R_{M,L}$ in the EIR-MELM prediction model according to the formula (34), and then go to Step2 (5). Start from $M = 5$ to determine whether the following conditions are satisfied simultaneously
$$\left| \frac{R_{M-i,L} - R_{M-i-1,L}}{R_{\max,L}} \right| \leq \xi_M \tag{37}$$
where $R_{1,\max}$ is the maximum value of $R_{1,L_{\min}}, \cdots, R(k)_{1,L}$, $i = 0, \cdots, 3$. If the formula (37) is met, the training process is completed, and M is determined to be the optimal number of hidden layers. Otherwise, the number of hidden layers M is continuously increased until the condition is satisfied.
10) According to the newly obtained connection weight matrix $\beta_{M,L}$, establish the EIR-MELM prediction model with the number of hidden layers M and the number of hidden nodes L. Calculate the final output results of the neural network $t_L(x) = \sum_{j=1}^{L} (\beta_{M,L})_j g(w_{M,j}, b_{Mj}, x)$ according to formula (36). Therefore, the predicted value of data Z is $t_L(Z)$.

To sum up, we have achieved the establishment of the EIR-MELM neural network model through the training dataset, and completed the prediction of the data Z with this model.

IV. EXPERIMENT AND DISCUSSION

In this section, experiments of the proposed EIR-MELM are conducted on benchmark datasets and coal spectral data for classification problem. To explore the improvement of classification accuracy and training time of our method, original ELM, EIR-ELM are also evaluated. All simulations in this experiment are performed under Windows 10; Intel(R) Core(TM) i5-7500 CPU @3.40GHZ, 16GB RAM, MATLAB 2016b. The activation function is the sigmoid function.

A. CLASSIFICATIONS ACCURACY EVALUATION OF EIR-MELM MODEL

We conduct the experiments from the following aspects. Given a definite classification accuracy and the maximum number of hidden nodes, how many hidden nodes are needed to meet the expect classification accuracy, and how does the classification accuracy change, when the hidden node is added into the network one by one.

Figure 1 shows the classification accuracy of the EIR-MELM model with different number of hidden nodes and hidden layers on Diabetic Retinopathy Debrecen dataset [26]. From Fig. 1(a) it can be seen that with the increasing of the number of hidden nodes, the classification accuracy of the model greatly improves. The model enters a stable state when the number of hidden nodes is about 25.

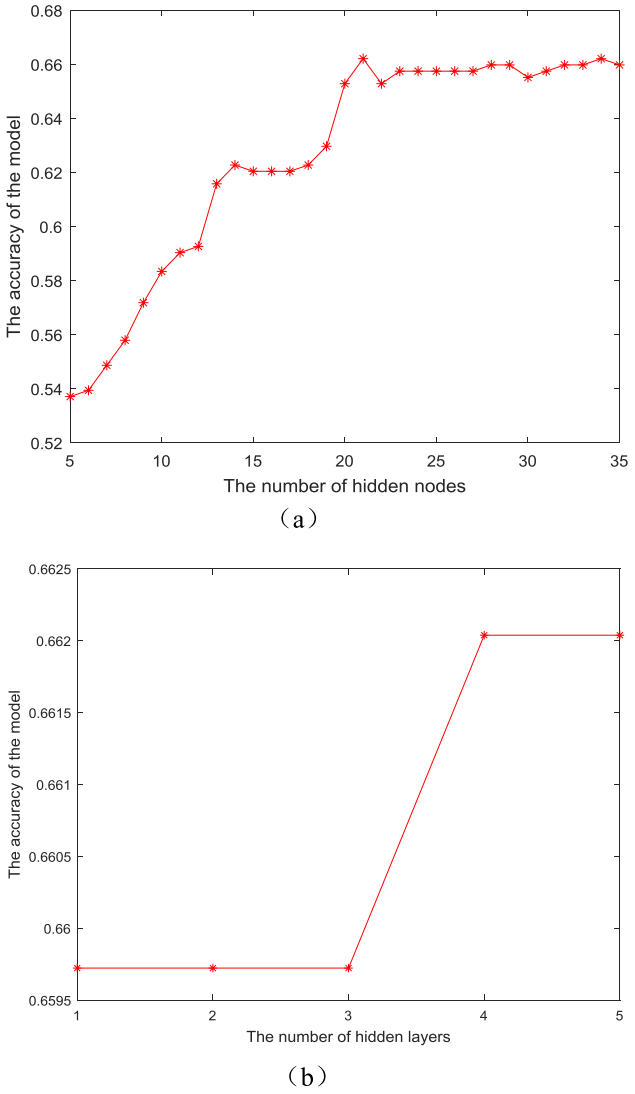


FIGURE 1. Classification accuracy of EIR-MELM with different number of hidden nodes and hidden layers on Diabetic Retinopathy Debrecen dataset.

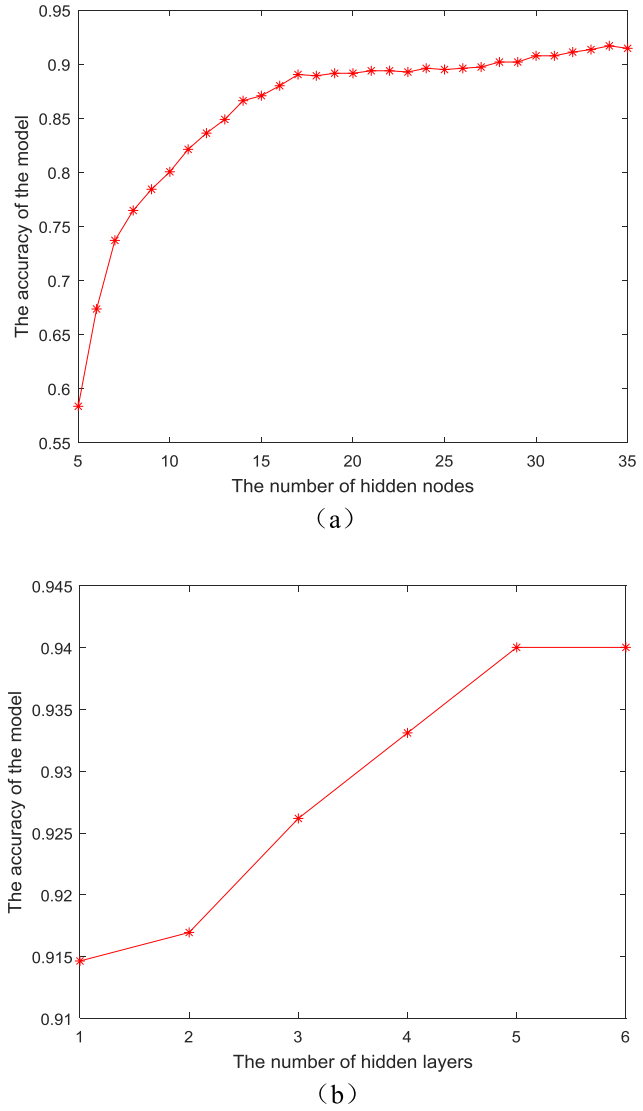


FIGURE 2. Classification accuracy of EIR-MELM with different number of hidden nodes and hidden layers on image segmentation dataset.

According to Fig. 1(b), it can be concluded that the classification accuracy of the model is also dramatically boost as the number of hidden layers increases.

Figure 2 illustrate the classification accuracy of the EIR-MELM model with different number of hidden nodes and hidden layers on Image Segmentation dataset. From Fig. 2(a) we can see that when the number of hidden nodes increases, the classification accuracy of the model dramatically promotes. When the number of hidden nodes is about 35, the model enters a stable state. According to Fig. 2(b), we can conclude that the classification accuracy of the model is extremely enhanced through the number of hidden layers increases.

The analysis now focuses on the coal spectral data as an example. The results are reported in Figure 3, which presents the classification accuracy of the EIR-MELM model with different number of hidden nodes and hidden layers on coal spectral data. Form Fig. 3(a), it can be readily seen from the figure that as the increasing of the number of hidden nodes, the classification accuracy of the model extremely enhances. When the number of hidden nodes reaches about 25, the model can enter a stable state. According to Fig. 3(b), it can be readily concluded from the figure that the classification accuracy of the model is greatly improved with the increase of the number of hidden layers.

According to the above analysis, we can conclude that the EIR-MLEM algorithm can always achieve good classification effect for benchmark datasets and coal spectral data.

Next, we investigate the improvement of classification accuracy of the EIR-MELM algorithm in comparison with original ELM and EIR-ELM.

For the classification problem, the classification accuracy of different models is listed in Table 1. From these results we can see that in general the EIR-MELM can always get higher

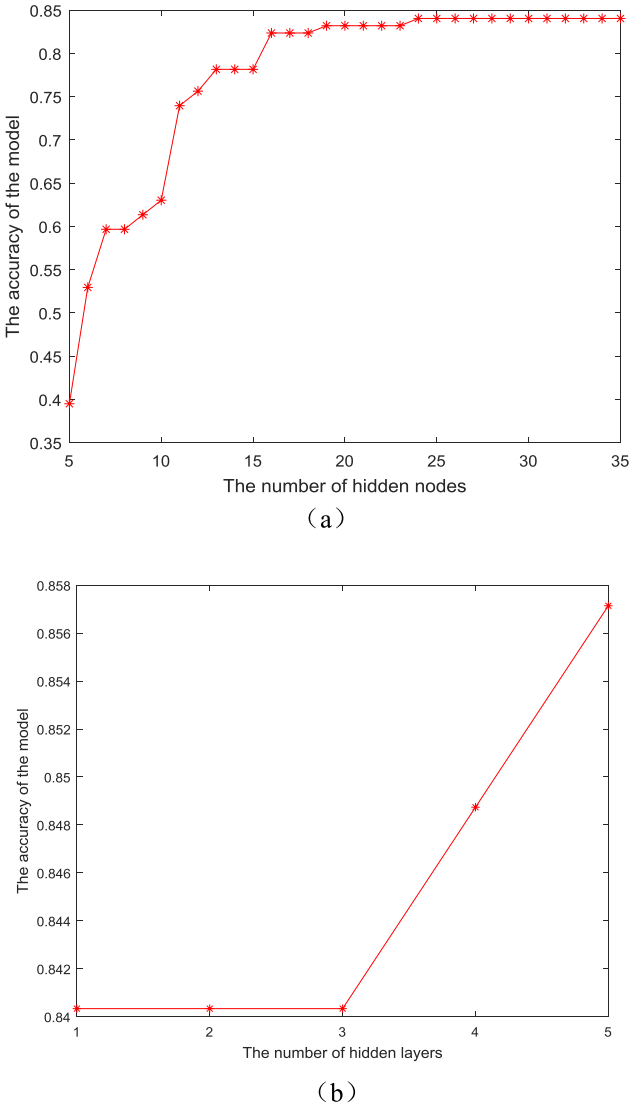


FIGURE 3. Classification accuracy of EIR-MELM with different number of hidden nodes and hidden layers on coal spectral data.

TABLE 1. Classification accuracy of different models.

Data	ELM	EIR-ELM	EIR-MELM
Diabetic Retinopathy Debrecen	0.5730	0.6597	0.6620
Image Segmentation	0.5836	0.9146	0.9400
Coal spectral data	0.3950	0.8403	0.8571

classification accuracy than original ELM and EIR-ELM for different datasets.

B. COMPUTATIONAL COMPLEXITY ANALYSIS OF EIR-MELM MODEL

In the following simulation, we compare the computation complexity of the original ELM and EIR-ELM on benchmark datasets and coal spectral data. While the hidden nodes are added into the model one by one, and we need to retain the network for original ELM and update the network output weight matrix recursively for EIR-ELM each time.

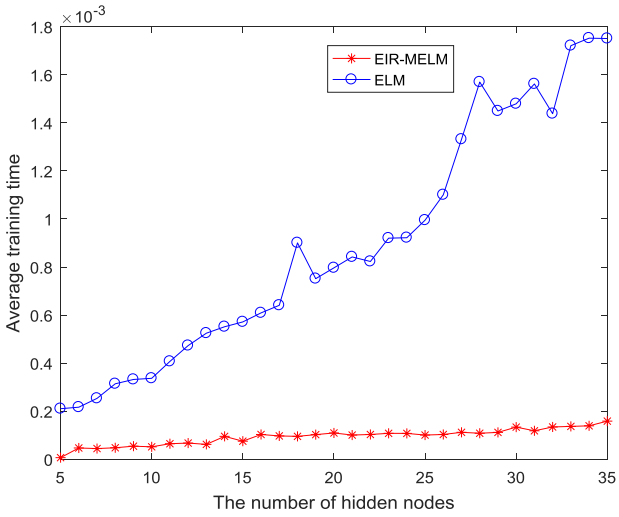


FIGURE 4. The result comparison of training time of EIR-MELM and ELM on Diabetic Retinopathy Debrecen dataset.

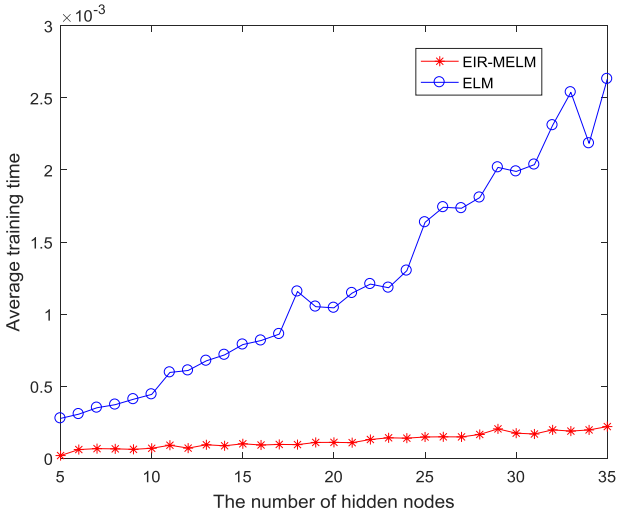


FIGURE 5. The result comparison of training time of EIR-MELM and ELM on image segmentation dataset.

Figure 4 shows the result comparison of training time of the EIR-MELM classification model and the original ELM classification model based on the Diabetic Retinopathy Debrecen dataset. Figure 5 shows the result comparison of the training time of the EIR-MELM classification model and the original ELM classification model based on the Image Segmentation dataset. Figure 6 shows the result comparison of the training time of the EIR-MELM classification model and the traditional ELM classification model on coal spectral data. From the simulation results, we can see that the original ELM spends more time retraining the network and recalculating the output weight matrix compared with EIR-MELM. And the gap is even greater if the network has more hidden nodes.

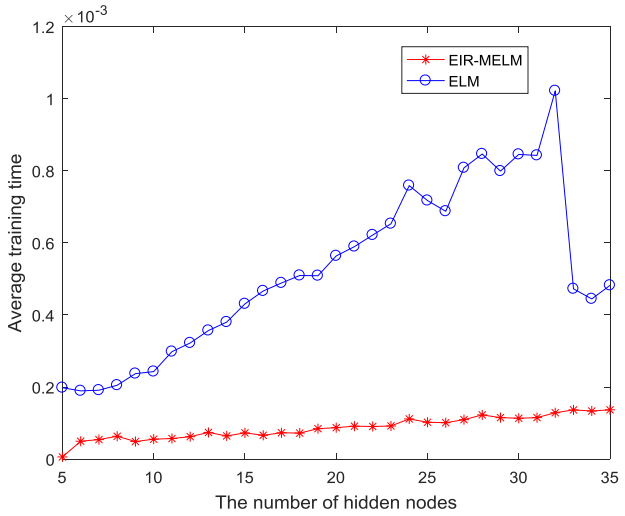


FIGURE 6. The result comparison of training time of EIR-MELM and ELM on coal spectral data.

Experiments also illuminate that the EIR-MELM has greatly improved the training speed of the model, and with the increase of the number of hidden nodes, the advantage of EIR-MELM is more obvious.

In conclusion, the EIR-MLEM model is an incremental MELM training algorithm based on Cholesky decomposition. By updating the output weights in an incremental manner, the training time of the model can be effectively reduced.

V. CONCLUSIONS AND DISCUSSION

(1) EIR-MELM weighs MELM's structural risk and empirical risk by introducing regularization parameters. As an improved model that can automatically determine the optimal network structure, its generalization ability is significantly improved compared with MELM. Experimental studies on benchmark datasets for regression and classification problem show that EIR-MELM can effectively determine the optimal network structure of MELM, and has the advantages of high prediction accuracy and fast calculation speed.

(2) EIR-MELM utilizes the Cholesky decomposition method to effectively reduce the computational complexity to solve the output weight matrix once. In addition, during the process of increasing the number of hidden nodes, the calculation of the output weight matrix can also be performed on the basis of the previous calculation result, and it can further reduce the amount of the calculation. Therefore, the calculation efficiency of EIR-MELM is higher than that of MELM.

(3) EIR-MELM adds a selection phase in the increasing process of hidden nodes. In each step of learning process, multiple hidden nodes are randomly generated at first. According to the principle of structural risk minimization, we select the optimal nodes form multiple randomly generated nodes and add it to the existing network. When new hidden layer neuron nodes are added one by one, the Cholesky decomposition method is adopted to recursively update the output weight of the network, so that EIR-MELM can have a more compact network structure while ensuring numerical stability.

(4) Future directions should include in the scope of the study the design of an adaptive strategy to prune the redundant neuron nodes in the hidden layers. For time-varying or non-stationary systems, using EIR-MELM to develop accurate prediction approaches less influenced by model parameters is also worth studying in the future.

REFERENCES

[1] J. Lyu and J. Zhang, "BP neural network prediction model for suicide attempt among Chinese rural residents," *J. Affect. Disorders*, vol. 246, pp. 465–473, Mar. 2019.

[2] B. Niu, D. Wang, N. D. Alotaibi, and F. E. Alsaadi, "Adaptive neural state-feedback tracking control of stochastic nonlinear switched systems: An average dwell-time method," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published. doi: 10.1109/TNNLS.2018.2860944.

[3] M. Kobayashi, "Twin-multistate commutative quaternion Hopfield neural networks," *Neurocomputing*, vol. 320, pp. 150–156, Dec. 2018.

[4] H. Wang, P. X. Liu, S. Li, and D. Wang, "Adaptive neural output-feedback control for a class of nonlower triangular nonlinear systems with unmodeled dynamics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3658–3668, Aug. 2018.

[5] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[6] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Channel equalization using complex extreme learning machine with RBF kernels," in *Advances in Neural Networks—ISNN* (Lecture Notes in Computer Science), vol. 3973. Berlin, Germany: Springer-Verlag, 2006, pp. 114–119.

[7] G.-B. Huang, M.-B. Li, L. Chen, and C.-K. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, nos. 4–6, pp. 576–583, 2008.

[8] H. Wang, W. Sun, and P. X. Liu, "Adaptive intelligent control of nonaffine nonlinear time-delay systems with dynamic uncertainties," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1474–1485, Jul. 2017.

[9] B. Niu, H. Li, Z. Zhang, J. Li, T. Hayat, and F. E. Alsaadi, "Adaptive neural-network-based dynamic surface control for stochastic interconnected nonlinear nonstrict-feedback systems with dead zone," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published. doi: 10.1109/TSMC.2018.2866519.

[10] D. Sovilj, K. M. Björk, and A. Lendasse, "Comparison of combining methods using extreme learning machines under small sample scenario," *Neurocomputing*, vol. 174, pp. 4–17, Jan. 2016.

[11] Y. Yin, Y. Zhao, M. Li, and B. Zhang, "An enhanced extreme learning machine for efficient small sample classification," in *Proceedings of ELM-2015* (Proceedings in Adaptation, Learning and Optimization) vol. 6. Cham, Switzerland: Springer, Jan. 2016, pp. 501–509.

[12] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *Proc. IEEE Symp. Comput. Intell. Data Mining*, Xi' an, China, Mar./Apr. 2009, pp. 389–395.

[13] W. Y. Deng and L. Chen, "Color image watermarking using regularized extreme learning machine," *Neural Netw. World*, vol. 20, no. 3, pp. 317–330, May 2010.

[14] J. M. Martínez-Martínez, P. Escandell–Montero, E. Soria-Olivas, J. D. Martín-Guerrero, R. Magdalena-Benedito, and J. Gómez-Sanchis, "Regularized extreme learning machine for regression problems," *Neurocomputing*, vol. 74, no. 17, pp. 3716–3721, Oct. 2011.

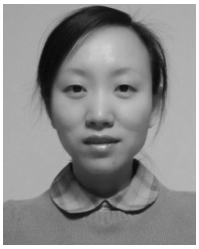
[15] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, nos. 16–18, pp. 3056–3062, 2007.

[16] 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.*, vol. 20, no. 8, pp. 1352–1357, Aug. 2009.

[17] Y. Lan, Y. C. Soh, and G. B. Huang, "Random search enhancement of error minimized extreme learning machine," in *Proc. Eur. Symp. Artif. Neural Netw.-Comput. Intell. Mach. Learn.*, Bruges, Belgium, 2010, pp. 327–332.

[18] Z. Xu, M. Yao, Z. Wu, and W. Dai, "Incremental regularized extreme learning machine and it's enhancement," *Neurocomputing*, vol. 174, pp. 134–142, Jan. 2016.

[19] X. Zhang and H.-L. Wang, "Incremental regularized extreme learning machine based on Cholesky factorization and its application to time series prediction," *Acta Phys. Sinica*, vol. 60, no. 11, Nov. 2011, Art. no. 110201.
[20] J.-L. Ding, F. Wang, H. Sun, and L. Shang, "Improved incremental regularized extreme learning machine algorithm and its application in two-motor decoupling control," *Neurocomputing*, vol. 149, pp. 215–223, Feb. 2015.
[21] D. Xiao, B. Li, and S. Zhang, "An online sequential multiple hidden layers extreme learning machine method with forgetting mechanism," *Chemometrics Intell. Lab. Syst.*, vol. 176, pp. 126–133, May 2018.
[22] A. N. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Sov. Math. Doklady*, vol. 4, no. 4, pp. 1035–1038, 1963.
[23] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer-Verlag, 1995.
[24] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.
[25] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 525–536, Mar. 1998.
[26] *UCI Machine Learning Repository*. Accessed: Jan. 10, 2019. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html



JINGYI LIU received the B.S. and M.S. degrees from Northeastern University, in 2003 and 2008, respectively, where she is currently a Lecturer. Her research interests include engineering numerical calculation and applied mathematics.

XINXIN LIU received the bachelor's degree in measurement and control and instrumentation from the Hebei University of Engineering, Handan, China, in 2017. She is currently pursuing the master's degree with the College of Information Science and Engineering, Northeastern University, Shenyang, China. Her research direction is based on machine learning and data processing. Her research interests include neural networks, extreme learning machine algorithm, and machine learning.

CHONGMIN LIU received the bachelor's degree in automation and electrical engineering from Jinan University, Jinan, China, in 2017. She is currently pursuing the master's degree with the College of Information Science and Engineering, Northeastern University. Her research direction is based on machine learning and spectral analysis technique. Her research interests include neural networks, extreme learning machine algorithm, and machine learning.

BA TUAN LE was born in Viet Tri, Phu Tho, Vietnam. He received the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China. His areas of interests include intelligence information process, remote-sensing exploration, and spectral analysis technique.

DONG XIAO received the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2009. Since 2006, he has been a Professor with the College of Information Science and Engineering, Northeastern University. His research interests include neural networks, extreme learning machine algorithm, partial least-squares algorithm, and modified median minimum distance algorithm.