Improved Ensemble Extreme Learning Machine Regression Algorithm

Meiyi Li, Weibiao Cai, Xingwang Liu

HAL Id: hal-02197774
https://inria.hal.science/hal-02197774

Submitted on 30 Jul 2019

Improved Ensemble Extreme Learning Machine Regression Algorithm

Meiyi Li[1], Weibiao Cai[2], Xingwang Liu[3]

[1] Xiangtan University College of Information Engineering,
Xiangtan University, Xiangtan, China,
`1356695709@qq.com`
[2] Xiangtan University College of Information Engineering,
Xiangtan University, Xiangtan, China,
`2481596161@qq.com`
[3] Information Center of Jianglu mechanical & electrical group Co., Ltd.

ABSTRACT. Compared with other traditional neural network algorithms, the Extreme Learning Machine (ELM) has the advantages of simple structure, fast learning speed and good generalization performance. However, there are still some shortages that restrict the further development of ELM. For example, the randomly generated input weights, biases and the ill-conditioned appearance of the hidden layer design matrix all affect the generalization performance and robustness of the ELM algorithm model. In order to overcome the adverse affects of both, an improved ensemble extreme learning machine regression algorithm(ECV-ELM) is proposed in this paper. The method first generates multiple sub CV-ELM model through AdaBoost..RT method, and the selects the best set of submodels to integrate. The ECV-ELM algorithm makes use of ensemble learning method to complement each other among sub models, so that the generalization performance and robustness of the algorithm are better than that of the sub model. The results of regression experiments on multiple data sets show that the ECV_ELM algorithm can effectively reduce the influence of the ill conditioned matrix, the random input weight and bias, and has good generalization performance and robustness.

KEYWORDS: extreme learning machine, ensemble learning, robustness, CV-ELM algorithm

1   Introduction

Artificial neural network (ANN) has been widely used in various fields due to its good learning ability and high speed optimization capability [1-2]. However, the artificial neural network is used to calculate the parameters of the model by the gradient descent method. The gradient descent method will increase the time complexity of the algorithm and cause the algorithm to fall into the local optimal solution easily. Due to these shortcomings, it may take more time to train the network and not necessarily guarantee the best solution. Therefore, searching for high

efficiency and high real-time neural network has become the main research direction of many scholars.

The Extreme Learning Machine (ELM) [3] is a new single-hidden-layer feedforward neural networks (SLFNs) proposed by G.B. Huang et al. in 2006. The method can randomly generate input weights and hidden layer node thresholds, and calculate the output layer weights without the need for iterative operations, so its computation speed is much better than the traditional neural network algorithm. When constructing the model, the traditional neural network algorithm is different from the traditional neural network algorithm. The limit learning function gets the minimum norm of the weight of the output layer while reaching the minimum training error. According to the neural network theory: For feed-forward neural networks, the smaller the training error and norm of the weight of the LM model, the stronger its network generalization ability [4]. Therefore, we can theoretically demonstrate the generalization ability of ELM algorithm. In recent years, ELM has been applied to various fields of various disciplines, such as face recognition [5], classified, regression, image processing [6], ground reconstruction [7], and so on, because of the ability to effectively improve the defects of traditional neural networks.

Although ELM has fast learning speed and good generalization performance, when some column vectors in the hidden layer design array are approximated to linear correlation, that is, the hidden layer design array has multiple collinearity or ill posed. It can result in poor generalization performance and stability by using ordinary least square method to estimate the solution of the ill conditioned matrix. To this end, many scholars have improved the ELM model, for example, M.Y.Li et al. Proposed an improved limit learning machine regression algorithm (CV-ELM) based on conditional index and variance decomposition ratio (CV-ELM) [8], Ceng Lin and others proposed the limit learning machine (PC-ELM) [9] based on the principal component estimation.

The CV-ELM algorithm improves the generalization performance of the extreme learning machine to a certain extent and can ensure good algorithm robustness. However, this algorithm still has defects in some cases, so that the model can not achieve the minimum error. Our analysis considers that there are two main reasons: First, high-dimensional data may obscure the noise components in the data, making the proposed method unable to completely isolate the relevant variables, that is, the algorithm can not reduce the noise completely. Secondly, some initial parameters of CV-ELM, such as input weight and hidden layer bias, are generated randomly, which makes the generalization ability and robustness of the model affected by them. Therefore, this paper introduces the ensemble learning method [10] in the CV-ELM algorithm. This method can train a number of similar learners at the same time, and then extract the best set of learners from these neural networks to integrate [11]. Through regression experiments of multiple data sets, it is proved that this method can achieve good generalization performance and stability.

2   Review of ELM and CV-ELM

In this section, we mainly review ELM[3] and CV-ELM[10].

2.1 Extreme learning machine

For N arbitrary distinct samples (x_i, y_i), where $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in R^n$ is an n-dimensional feature of the ith sample, and $y_i = [y_{i1}, y_{i2}, ..., y_{im}]^T \in R^m$, Then, the output of a feedforward neural network with L hidden nodes and excitation function G (x) can be expressed as

$$f_L(x) = \sum_{i=1}^{L} \beta_i\, G(a_i \cdot x_i + b_i),\ a_i \in R^n, \beta_i \in R^m, \qquad [1]$$

where $a_i = [a_{i1}, a_{i2}, ..., a_{in}]^T$ is the weight connecting the i-th hidden node and the input nodes, and b_i is threshold of the i-th hidden nodes, $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T$ is the weight connecting the i-th hidden node and the output nodes, $a_i \cdot x_i$ denote the inner product of a_i and x_i. The excitation function G(x) can choose "Sigmoid", "Sine" or "RBF" and so on.

If this feed forward neural network with L hidden layer nodes and M output layer nodes can approximate this N samples with zero error, then the above N equations can be written compactly as

$$f_L(x) = \sum_{i=1}^{L} \beta_i\, G(a_i . x_i + b_i) = y_i, i = 1, 2, \cdots, L, \qquad [2]$$

(2) can be simplified as

$$H\beta = Y \qquad [3]$$

where

$$H = \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ G(a_1, b_1, x_N) & \cdots & G(a_L, b_L, x_N) \end{bmatrix}_{N \times L} \qquad [4]$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times M} \qquad Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times M} \qquad [5]$$

H is called the hidden layer output matrix of the network, and ELM training can be transformed into a problem of solving the least squares solution of output weights. The output weight matrix $\hat{\beta}$ can be obtained from (6)

$$\hat{\beta} = (H^T H)^{-1} H^T Y = H^+ Y \qquad [6]$$

Where H^+ represents the Moore-penrose generalized inverse of the hidden layer output matrix H.

2.2 CV-ELM

The hidden layer matrix H is calculated according to the ELM model, and then the H matrix is separated according to the condition number and variance decomposition ratio to obtain

$$H = (H1, H2) \qquad [7]$$

where H1 is the non-interfereence data column in the hidden layer matrix H and H2 is the interference data columns.

According to the LS principle, the output weight matrix $\hat{\beta}$ can be obtained from (8)

$$\hat{\beta} = (H^TH)^{-1}H^TY$$

$$= ([H1, H2]^T[H1, H2])^{-1}[H1, H2]Y$$

$$= \begin{bmatrix} H1^TH1 & H1^TH2 \\ H2^TH1 & H2^TH2 \end{bmatrix}^{-1} [H1, H2]Y \qquad [8]$$

In order to enhance the generalization performance and stability of the model without destroying the authenticity of the data of the non-interference data, and add a small constant to the diagonal elements of the data matrix of the interfering data. The output weight matrix $\hat{\beta}$ can be obtained from (9)

$$\hat{\beta} = \begin{bmatrix} H1^TH1 & H1^TH2 \\ H2^TH1 & H2^TH2 + kI \end{bmatrix}^{-1} [H1, H2]Y \qquad [9]$$

where k is a small constant and I is the unit matrix.

CVELM algorithm is described as follows:

Known training samples (x_i, y_i), $i = 1, \cdots, N$, the number of hidden nodes is L, and the excitation function is $G(x)$.

(1) Random setting of input weights a_i and bias b_i, $i = 1, \cdots, L$

(2) Computing hidden layer output matrix H

(3) Through the condition number and variance decomposition machine decomposition matrix H, and get (H1, H2)

(4) Determining the ridge parameter k

(5) The output matrix $\hat{\beta}$ is calculated by Equation (10)

3 Improved Ensemble Extreme learning Machine

The CV-ELM algorithm overcomes the situation that the generalized performance and robustness of the algorithm deteriorate when the hidden layer design array is ill-conditioned. However, the random generation of input weights and incomplete cancellation of noise under high-dimensional data can cause generalization performance and robustness to be poorly processed. Therefore, this paper proposes a CV-ELM regression algorithm based on ensemble learning (ECV-ELM), which makes use of the complementarity between multiple learners, thus making the

ensemble better performance. ECV-ELM overcomes the shortcomings of poor model stability due to the random generation of input weight, bias and incomplete cancellation of CV-ELM noise. It combines the ensemble learning method with the CV-ELM regression algorithm and uses some common methods to selet the appropriate sub CV-ELM model, which can further improve the performance of the entire CV-ELM.

It is assumed that the training set and test set are $G = \left\{\left(x_i, \ y_i\right)|i = 1, \ 2, \right.$ $\left. \cdots, \ 1\right\}$, $G' = \left\{\left(x_i, \ y_i\right)|i = 1, \ 2, \ \cdots, \ 1\right\}$, and x_i is the model input, and the y_i is the output of the model. First, according to the training set G come into the different training sub set $G = \left\{G_1, \ \cdots, \ G_T\right\}$, several different sub CV-ELM models are generated by different training subsets. Then, a part of the excellent sub CV-ELM model is selected according to the training results. Finally, the results are made by means of the average method.

In summary, the proposed ECV-ELM integrated regression algorithm can be summarized as follows:

Input: Training sample set T

Output: Integrated CV-ELM regression model

(1) Using training set G to randomly generate T intersecting data sub sets $G = \left\{G_1, \ G_2, \ \cdots, \ G_T\right\}$, set the activation function of all sub models to g(x), and the number of hidden layer neurons is L;

(2) Initialization t=1;

(3) Determine whether it reaches the number of iterations, that is, t<=T; if yes, execute step (4); otherwise, execute step (7);

(4) Using the random function to generate the input weight a and the hidden layer offset b;

(5) The t-th sub CV-ELM model is trained using the randomly generated a, b, and t-th data subsets;

(6) Perform step (3);

(7) Calculate the MSE values of all the submodels. According to the size of the MSE values, select the k best sub CV-ELM models；

$$\text{MSE} = \frac{1}{n\left(Y - H\hat{\beta}\right)^T\left(Y - H\hat{\beta}\right)} \tag{10}$$

(8) using the simple average method to integrate K sub CV-ELM and get the final model, that is, the ECV-ELM model.

4   Experiment and Analysis

This section analyzes the CV-ELM regression method (ECV-ELM) based on ensemble learning proposed in the previous section. In order to better carry out experimental analysis, the time and prediction results of standard ELM algorithm,

CV-ELM algorithm and ECV-ELM algorithm are compared. The experiment uses 5 regression analysis data sets from UCI database and LIACC, which are Balloon data set, California House data set, Cloud dataset, Strike data set, Bodyfat data set. There are huge differences between the input attributes and the number of samples in these five data sets, which can better analyze the performance of the algorithm, as shown in Table 1. The number of hidden layer nodes required for each dataset in the ECV-ELM neutron model is shown in table 2. All experiments in this chapter are run on Windows7 64 bit operating system and Matlab 2016 environment in 3.30GHz i5-4590 CPU, 4G RAM.

Table 1. Regression Analysis Dataset

Data set	Attributes	samples	
		Training	Testing
Cloud	9	599	1499
Strike	6	416	209
Balloon	1	1334	667
Bodyfat	14	168	84
House	8	8000	12640

Table 2. Number of required hidden layer nodes in each dataset

Data set	Cloud	Strike	Balloon	Bodyfat	House
Hidden nodes	80	110	100	30	100

Table 3 shows the comparison of test time and training time of ELM, ECV-ELM and CV-ELM on multiple data sets. Table 4 shows the comparison of RMSE of ELM, ECV-ELM and CV-ELM on multiple data sets. Table 5 shows the comparison of DEV of ELM, ECV-ELM and CV-ELM on multiple data sets. Among them, the activation functions of the standard ELM, the CV-ELM model and the ECV-ELM model all use the Sigmoid function. In the ECV-ELM model, the number of training sub models is $T = 20$, the number of integrated sub models is $k = 10$, and the subtraining set selected by the submodel is three quarters of the randomly selected training set.

Table 3. Comparison of training and testing time of ELM,ECV-ELM,CV-ELM

Data set		Cloud	Strike	Balloon	Bodyfat	House
ELM	Training	0.0115	0.0156	0.0293	0.0037	0.0836
	Testing	0.0193	0.0107	0.0168	0.0034	0.0755
ECV-ELM	Training	0.4368	0.4368	0.6240	0.3705	2.002
	Testing	0.1832	0.1248	0.1014	0.0732	0.9844
CV-ELM	Training	0.0375	0.0502	0.0506	0.0250	0.1732
	Testing	0.0502	0.0310	0.0380	0.0299	0.1114

Table 4. Comparison of testing RMSE of ELM,ECV-ELM,CV-ELM

Data set	RMSE		
	ELM	ECV-ELM	CV-ELM
Cloud	0.0793	**0.0666**	0.0685
Strike	0.4950	**0.4511**	0.4640
Balloon	0.0939	**0.0897**	0.0898
Bodyfat	0.0385	**0.0265**	0.0316
House	0.2626	**0.2530**	0.2542

Table 5. Comparison of testing DEV of ELM,ECV-ELM,CV-ELM

Data set	Dev		
	ELM	ECV-ELM	CV-ELM
Cloud	0.0067	**8.3245E-04**	0.0054
Strike	0.1403	**4.3562E-03**	0.0334
Balloon	0.0168	**9.6536E-06**	1.0863E-04
Bodyfat	0.0230	**2.8888E-04**	0.0163
House	0.0099	**7.9302E-05**	0.0037

5 Conclusions

In this paper, a CV-ELM regression algorithm based on ensemble learning is proposed. This algorithm uses ensemble learning method to overcome the disadvantage of poor model stability caused by CV-ELM random input weight, bias and incomplete noise elimination. Through different regression data sets, the performance of ECV-ELM algorithm is analyzed. It is concluded that although the time cost of ECV-ELM algorithm is longer than that of CV-ELM, the generalization performance and robustness of the algorithm have been greatly improved.

Compared to the CV-ELM algorithm, the ECV-ELM algorithm proposed in this section is trained by training multiple CV-ELM sub models, and then using ensemble learning methods to make multiple CV-ELM learners complementation, thus improving the generalization ability and robustness of the algorithm.

6 References

1. Han F, Ling Q H, Huang D S. An improved approximation approach incorporating particle swarm optimization and a priori information into neural networks[J]. Neural Computing & Appli- cations, 2010, 19(2):255-261.
2. Han F, Ling Q H. A New Learning Algorithm for Function Approximation By Incorporating A Priori Information Into Feedforward Neural Networks[J]. Neural Computing & Applica- tions, 2008, 17(5-6):433-439.
3. Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: Theory and applications[J]. Neurocomputing, 2006, 70(1–3):489-501.
4. Zhang Min, Zeng Xinmiao, Ma Changchun. An online learning algorithm based on cluster-based extreme learning machine[J]. Computer Engineering and Applications, 2014, 50(11): 188-191.
5. He B, Xu D, Rui N, et al. Fast Face Recognition Via Sparse Coding and Extreme Learning Machine[J]. Cognitive Computation, 2014, 6(2):264-277.
6. An L, Yang S, Bhanu B. Efficient smile detection by Extreme Learning Machine[J]. Neurocomputing, 2015, 149(PA):354-363.
7. Zhou Z H, Zhao J W, Cao F L. Surface reconstruction based on extreme learning machine[J]. Neural Computing & Applications, 2013, 23(2):283-292.
8. Mei-yi LI, Wei-biao CAI, Qing-shuai SUN. Extreme Learning Machine For Regression Based on Condition Number and Variance Decomposition Ratio[C]// 2018 International Conference on Mathematics and Artificial Intelligence.
9. ZENG Lin,ZHANG Xiaogang,BU Zhanyu et al. Extreme learning machine based on principal components estimation[J]. CEA, 2016, 52(4): 110-114.
10. Hansen L K, Liisberg C, Salamon P. Ensemble methods for handwritten digit recognition[C]// Neural Networks for Signal Processing. IEEE, 2002:333-342.
11. Tang W, Zhou Z H. Bagging-Based Selective Clusterer Ensemble[J]. Journal of Software, 2005, 16(4).