Extreme Learning Machine Based on Calculating the Output Weight of Partial Robust M-regression

Guanghua Yao¹, Gaitang Wang²*

¹Hydrology and Water Resources Bureau of Henan Province, Zhengzhou, Henan Province, 45003, China
²Xi'an Institute of Modern Control Technology, Xi'an, Shaanxi Province, 710065, China

*Corresponding author’s e-mail:wgt0104@163.com

Abstract. In order to improve the nonlinear mapping capability and learning performance of extreme learning machine (ELM), a new learning algorithm called extreme learning machine based on calculating the output weight of partial robust M-regression is proposed. This algorithm introduces the partial robust M-regression into the extreme learning machine algorithm. Firstly, the hidden layer output matrix \( H \) is calculated by extreme learning machine. Secondly, the matrix \( H \) and vector \( Y \) are weighted by weighted strategy. Then, PRM algorithm is used to establish the regression model between the weighted matrix \( H_w \) and vector \( Y_w \), and calculate its regression coefficient, namely output weight of the ELM Algorithm. The proposed method predicts the Mackay’s robot arm regression and sediment concentration of the Yellow River Basin to verify the effectiveness of the method. The simulation results show that the proposed PRMELM algorithm is superior to the original extreme learning machine algorithm in prediction accuracy and generalization performance.

1. Introduction

In 2004, Professor Huang guangbin proposed the extreme learning machine (ELM) algorithm, which is a new single hidden layer forward artificial neural networks(SLFNs). In the process of network learning, compared with the traditional BP neural network, the extreme learning machine algorithm does not need to adjust any learning parameters (such as input weight and threshold), only needs to calculate the output layer weight through the generalized inverse matrix of the hidden layer output matrix, which greatly improves the learning speed of the network. Due to its high learning efficiency and generalization ability, ELM is widely used in classification, pattern recognition [1][2]. Based on ELM algorithm, many improved algorithms are proposed, such as: M-ELM [3], PELM[4], Incremental extreme learning machine(I-ELM)[5], EMD-ELM[6], Multiple timing-driven based ELM[7], OS-ELM[8], EOS-ELM[9], CI-ELM[10].

However, when solving practical engineering problems, there may be multicollinearity between sample data, and using least squares or MP generalized inverse to solve the output weights is easy to lead to ill-conditioned solution. In order to overcome or solve this problem, the extreme learning machine ridge regression learning Algorithm (ELMRR) is proposed in reference [11] and [12]. Although the training time and test time of ELMRR is longer than that of ELM, the performance of ELMRR is better than ELM. In view of the above problems, this paper applies Partial Robust M-regression (PRM) algorithm to extreme learning machine, and an extreme learning machine based on calculating the output...
weight of partial robust M-regression (PRMELM) is proposed. The method is used to predict the sediment concentration in the Yellow River basin. The experimental results verify the accuracy and effectiveness of the method.

2. Review of ELM

Suppose that the number of hidden layer nodes in SLFNs network is L, and g(x) is the activation function of the network. For N, arbitrary distinct samples \((x_i, y_i) \in \mathbb{R}^n \times \mathbb{R}^m\). If the SLFNs with \(N\) hidden nodes can approximate \(N\) samples with zero error, it then implies that there exist \(w_i, b_i\) such that

\[
 f_N(x_j) = \sum_{i=1}^{N} \beta_i G(w_i, b_i, x_j) = y_j, \quad j = 1, 2, \cdots N
\]

The Eq.(1) can be written compactly as

\[
 H\beta = Y
\]

where

\[
 H(w_1, w_2, \cdots, w_N, b_1, b_2, \cdots, b_N, x_1, x_2, \cdots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_N \cdot x_1 + b_N) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_N \cdot x_N + b_N) \end{bmatrix}_{\times \times N}
\]

\[
 \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{bmatrix}, \quad Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}
\]

where \(w_i = [w_{i1}, w_{i2}, \cdots, w_{in}] \in \mathbb{R}^n\) is the connection weight between the \(i\)th node of the hidden layer and the input layer, \(\beta = [\beta_1, \beta_2, \cdots, \beta_N] \) is the connection weight between the \(i\)th node of the hidden layer and the output layer, \(H \) is the output matrix of hidden layer, \(b_i\) represents the threshold of the \(i\)th node of the hidden layer, \(w_i \cdot x_j\) represents the dot product of \(w_i\) and \(x_j\).

Typically, the output weights of ELM networks are calculated using the least-squares(LS) solution:

\[
 \hat{\beta} = H'Y
\]

where \(H'\) is the Moore-Penrose generalized inverse of the hidden layer output matrix \(H\).

3. Improved extreme learning machine

The basic idea of using partial robust M-regression algorithm to calculate the output weight is to establish the PRM regression model between the hidden layer output matrix \(H\) and the output vector \(Y\). Next, firstly, the hidden layer output matrix \(H\) and output vector \(Y\) are weighted, and then the single dependent variable PLS model between \(Hw\) and \(Yw\) is built.

3.1. Output weight calculation of PRM

Assuming the size of matrix \(H\) is \(n \times m\), the residual and leverage weights of \(i\)th row of matrix \(H\) are expressed as follows[13]:

Residual weight \(W'_i\) :

\[
 W'_i = f\left(\frac{r_i}{\hat{r}}, c\right)
\]

\[
 f(z, c) = \frac{1}{1 + (\frac{z}{c})^2}
\]

where \(r_i\) is the residual value between the predicted value and the actual value, and \(\hat{r}\) represents the robust scale estimate.
Leverage weight $W_i^*$:

$$W_i^* = f\left(\frac{||x_i - \text{median}(T)||}{\text{median}(||x_i - \text{median}(T)||)}\right), i = 1, 2, \cdots, n$$  \hspace{1cm} (8)

Where median represents the median value of Euclidean distance, $t_i$ is the $i$th row of the score vector matrix, and $\text{med}_{i1}$ is the median value of L1 of the score vector.

From the residual weight and the leverage weight, the weight $W_i$ of the $i$th row of the matrix H can be calculated, which is expressed as follows:

$$W_i = \sqrt{W_i^* W_i^*}$$  \hspace{1cm} (9)

After weighting matrix $H$ and output vector $Y$ with equation (9), the weighted expression is as follows:

$$H_W = H \times W_i$$  \hspace{1cm} (10)

$$Y_W = Y \times W_i$$  \hspace{1cm} (11)

The PLS model of the single dependent variable between $H_W$ and $Y_W$ can be expressed as:

$$H_W = TP^T + E$$  \hspace{1cm} (12)

$$Y_W = TQ + F + M \beta + F$$  \hspace{1cm} (13)

where $P$ is the load matrix, $Q$ is the regression coefficient of the score vector, $E$ is the residual matrix, $F$ is the residual vector, $\beta$ is the regression coefficient of the single dependent variable PLS model.

3.2. Algorithm steps

The detailed steps of PRMELM algorithm can be described as follows:

Step 1: Select training sample data $(x_i, y_i)$, where $i = 1, 2, \ldots, n$, the input matrix of the training sample is $X$, and the output vector is $Y$;

Step 2: Randomly assign input weight $w_i$ and bias $b$;

Step 3: The hidden layer output matrix $H$ is calculated by formula (3);

Step 4: The hidden layer output matrix $H$ is taken as the input data sample of PRM algorithm, and the output vector $y$ is taken as the output data sample;

Step 5: The weighted $W_i$ is initialized by using the equation (6), equation (7) and equation (8), and the threshold value of the relative error of the score vector regression coefficient is set;

Step 6: The input and output data (i.e. matrix $H$ and vector $Y$) are weighted according to the weighting strategy, and a single dependent variable PLS model is established for the weighted $H_W$ and $Y_W$, and the score vector is updated;

Step 7: The residual error $r_i$ between the predicted value and the actual value of each sample data is calculated, and the weight $W_i$ is updated by using equation (6);

Step 8: Judge whether the relative difference of the score vector regression coefficient is less than the given threshold, if it is less than the given threshold, then carry out step 9, otherwise carry out step 5;

Step 9: The regression coefficient of PRM is calculated, which is the output weight $\beta$ of ELM algorithm.

4. Simulation research

4.1. Regression experiment

In order to verify the feasibility and effectiveness of the PRMELM algorithm, Mackay’s robot arm regression function is used to verify it. Mackay’s robot arm problem is a verification function of Gaussian process regression analysis by C. E. Rasmussen and C. K. I. Williams in 2006 [14] [15]. The function has eight input variables, and the specific meaning of each variable is shown in reference [15]. The sample data of this experiment is from the reference [15]. The training samples and the test samples
have 200 groups of data respectively, and the target value of the training samples increases the Gaussian noise with the variance of 0.0025.

In the experiment, using radial basis function as activation function of hidden layer in ELM algorithm and PRMELM algorithm, and using the root mean square error (RMSE) as the performance index, and the simulation results are shown in Fig. 1 and table 1.

(a) ELM algorithm
(b) PRMELM algorithm
Fig.1 Prediction results and errors of PRMELM and ELM

| Algorithms | Time(s) | RMSE | #nodes |
|------------|---------|------|--------|
| ELM        | 0.1250  | 0.0018 | 0.0498 | 0.9569 | 200     |
| PRMELM     | 0.2349  | 0.0018 | 0.0308 | 0.4691 | 200     |

It can be seen from Fig. 1 and table 1 that the prediction results of PRMELM algorithm for Mackay’s robot arm regression are obviously better than ELM algorithm. The prediction errors of PRMELM almost all fall between [-1, 1], and only a few point error exceeds 1. From the running results of ELM algorithm, although the running time of the proposed algorithm is sacrificed, the test accuracy is greatly improved. Therefore, the PRMELM algorithm is effective and feasible.

4.2. Sediment concentrations prediction of Yellow River Basin
Taking Longmen (Ma wang miao II) station as an example, the sediment content(kg/m³) of the Yellow River basin is predicted. Water level(m), starting distance(m), water depth(m), depth(m), velocity(m/s) and water temperature(C) are selected as the influencing factors of the sediment content of the Yellow River basin. The PRMELM algorithm is used to predict the sediment content of the Yellow River basin.

A total of 100 sets of sample data were collected. In order to compensate for the error in recording data, 0.001~0.009 random numbers were added to each set of sample data. During the experiment, the number of samples is normalized first, then 50 sets of data after normalization are randomly selected as training samples, the remaining 50 sets of data are used as test samples, and the root mean square error (RMSE) is used as the performance index to evaluate the algorithm. Fig. 2 shows the predicted value and error of PRMELM and ELM algorithm. The performance comparison of PRMELM and ELM algorithms is shown in Table 2.
From Fig. 2 and table 2, it can be concluded that PRMELM algorithm is superior to ELM algorithm in the prediction accuracy of sediment concentration of the Yellow River Basin. In most cases, the test error of PRMELM is between [-2,2], and only the absolute value of prediction error of individual points is greater than 2, while the prediction error of ELM algorithm is obviously inferior to that of PRMELM algorithm.

5. Conclusion

Based on the Extreme Learning Machine (ELM), a new algorithm for Extreme Learning Machine (ELM) is presented, that is, the ELM based on calculating the output weight of partial robust M-regression. This algorithm uses PRM regression instead of the original least squares method or MP generalized inverse to compute the output weights of ELM networks. When solving practical problems, the algorithm not only solves the ill-conditioned problem when using least squares or MP generalized inverse to solve output weights, but also improves the non-linear mapping ability of the network. The prediction results of Mackay's robot arm regression function and sediment content of the Yellow River basin show that although the algorithm has high test accuracy and good generalization ability, there are still some shortcomings, such as the selection of the number of hidden layer nodes.

References

[1] Guang-Bin Huang. Extreme Learning Machines [EB/OL].http://www.ntu.edu.sg/home/egbhuang/pdf/HGB-ELM-Software-Beijing2016.pdf, 2016-12-22.

[2] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: Theory and applications[J]. Neurocomputing, 2006, 70(1): 489-501.

[3] ZHANG Wen-bo,JI Hong-bing, WANG Lei, ZHU Ming-zhe. Multiple hidden layer output matrices extreme learning machine[J]. Systems Engineering and Electronics,2014,36(8): 1656-1659.

[4] ZHANG Wen-bo,JI Hong-bing. Fusion of Extreme Learning Machines[J]. Journal of Electronics & Information Technology,2013,35(11):2728-2732.
[5] SONG Shaojian, XIANG Weikang, LIN Xiaofeng. Improvement Algorithm of an Incremental Extreme Learning Machine [J]. Information and control, 2016, 45(6): 735-741+758.

[6] YAN Qi-sheng, WANG Shi-tong, ZHANG Yan-fei, DING Mu-hua. Uranium resource price prediction based on empirical mode decomposition and extreme learning machine [J]. Control and Decision, 2014, 29(7): 1187-1192.

[7] XU Yuan, LU Yushuai, CAI Yi. Multiple timing-driven based extreme learning machine whole process fault prediction and its application [J]. CIESC Journal, 2015, 66(1): 351-356.

[8] N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan. A fast and accurate online sequential learning algorithm for feedforward networks [J]. IEEE Transactions on Neural Networks, 2006, 17(6): 1411-1423.

[9] Yuan Lan, Yeng Chai Soh, Guang-Bin Huang. Ensemble of online sequential extreme learning machine [J], Neurocomputing, 2009, 72: 3391–3395.

[10] Guang-Bin Huang, Lei Chen. Convex incremental extreme learning machine, Neurocomputing, 2007, 70: 3056–3062.

[11] WANG Gaitang, LI Ping, SU Chengli. ELM Ridge Regression Learning Algorithm of Ridge Parameter Optimization [J]. Information and control, 2011, 40(4): 497-500+506.

[12] WANG Gaitang, LI Ping, SU Chengli. Soft sensor modeling method of extreme learning machine ridge regression [J], JOURNAL OF HEFEI UNIVERSITY OF TECHNOLOGY, 2011, 34(8):1150-1154.

[13] Kruger U, Zhou Y, Wang X, Rooney D, Thompson J. Robust partial least squares regression: Part I, algorithmic developments [J]. Journal of Chemometrics, 2008, 22(1): 1-13.

[14] C. E. Rasmussen, C. K. I. Williams. Gaussian Processes for Machine Learning [M], The MIT Press, 2006.

[15] C. E. Rasmussen, C. K. I. Williams. Gaussian Processes for Machine Learning [EB/OL]. http://www.gaussianprocess.org/gpml/2010-06-29.