IOP Conf. Series: Materials Science and Engineering **320** (2018) 012015 doi:10.1088/1757-899X/320/1/012015

Improved Extreme Learning Machine based on the Sensitivity Analysis

Licheng Cui[1,2,3], Huawei Zhai[3], Benchao Wang[1] and Zengtang Qu[1]

[1] Liaoning Police College, Dalian, 116036, China
[2] Faculty of electronic information and electrical engineering, Dalian University of Technology, Dalian, 116024, China
[3]Information Science and Technology, Dalian Maritime University, Dalian, 116026, China

Abstract. Extreme learning machine and its improved ones is weak in some points, such as computing complex, learning error and so on. After deeply analyzing, referencing the importance of hidden nodes in SVM, an novel analyzing method of the sensitivity is proposed which meets people's cognitive habits. Based on these, an improved ELM is proposed, it could remove hidden nodes before meeting the learning error, and it can efficiently manage the number of hidden nodes, so as to improve the its performance. After comparing tests, it is better in learning time, accuracy and so on.
Keywords: Sensitivity analysis; SVM; Extreme learning machine; Importance

1. Introduction

Extreme learning machine [1](ELM) is a special algorithm of single-hidden layer feed-forward neural networks (SLFNs), all parameters can be selected randomly except the two, the number of hidden nodes and output weight.

Compared with other algorithms, ELM is simple, training time is shorter, but it is also weak in computational complexity and generation ability. So, it needs to be improved. Literature[2] introduced the wavelet transform and proposed a combination predicting method, it uses the wavelet to decompose the data and then predict, this method does not modified the ELM itself, the performances of ELM is not improved remarkably. Literatures[3, 4] introduced the concept of sensitivity of hidden nodes, sorted each hidden node based on the sensitivity, and finally removed several hidden nodes whose sensitivity are least, these resolved design problem of the structure of SLFNs. Literatures[5-8] referred from the idea of sequential algorithms, proposed variable-structure extreme learning machine, which could be adjust the number of hidden nodes with the new inputting samples.

Referencing the idea of the importance in pruning algorithms, this paper modifies the computing method of the sensitivity, and proposed an improved ELM (IPS-ELM for short), which could remove hidden nodes based on their sensitivity.

2. ELM

Based on the ELM, the output of SLFNs is

$$f_M(x)=\sum_{i=1}^{M}\beta_i G(a_i,b_i,x) \tag{1}$$

where, M is the number of hidden nodes, $\beta_i \in R^m$ is the output weight of the ith hidden node, $G(a_i,b_i,x)$ is the activation function of the ith hidden node, $a_i \in R^n$ is the input weight and $b_i \in R$ is the threshold of the ith hidden node. $x \in R^n$ is the input vector.

For given training samples (x_i, y_i), $i = 1,...,N$, x_i is n×1 input vector, y_i is m×1 expected output vector. So, the real output y_i' is

$$y_i' = \sum_{i=1}^{M} \beta_i G(a_i, b_i, x_i),\ i = 1, 2, \cdots N \tag{2}$$

After being transformed, the equation(2) can be deployed as $H\beta = Y'$, β is the output weight, $\beta = [\beta_1 \cdots \beta_M]^T$, Y' is the real output, $Y' = [y_1' \cdots y_N']^T$, H is the output matrix of hidden nodes,

$$H = \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_M, b_M, x_1) \\ \vdots & \ddots & \vdots \\ G(a_1, b_1, x_N) & \vdots & G(a_M, b_M, x_N) \end{bmatrix} \tag{3}$$

In ELM, the parameters a_i and b_i are generated randomly, and the output weight β needs to be regulated as the new sample inputs, its optimal solution is

$$\beta = H^{\dagger} Y \tag{4}$$

where, Y is the real output, $H^{\dagger}$ is Moore-Penrose generalized inverse, if $rank(H) = M$, the equation(4) could be transformed to

$$\beta = (H^T H)^{-1} H^T Y \tag{5}$$

3. Improve ELM

3.1 The sensitivity of hidden nodes

Literatures[3, 4] gives the concept of the sensitivity, according to the function space, the sensitivity of hidden nodes is the space distance between the output results before and after removing hidden nodes.

$$E(l,i) = \| f_{bef} - f_{aft} \| = |h_{li}| \|\beta_l\| \tag{6}$$

$E(l,i) \in (0, +\infty)$, the larger $E(l,i)$ is, the greater the sensitivity is, otherwise, the less it is. $E(l,i)$ is only based on one sample, and not enough to measure the real state, so it needs large amount of samples, so as to they can fully show their change law and trends, so the equation could be transformed to

$$E_{avg}(l) = \frac{1}{N} \sum_{i=1}^{N} |h_{li}| \|\beta_l\| \tag{7}$$

in which, $h_{li} = G(a_l, b_l, x_i)$, $E_{avg}(l) \in (0, +\infty)$, $E_{avg}(l) \to +\infty$ shows the sensitivity is greatest, otherwise the sensitivity is less. This leads to be another problem, for different real applications, it can not judge the degree of the sensitivity only by its value, because, in some applications, its value is very great, but in others, it is very little, there is no single criterion. So, another solution is given by the paper.

Given training samples (x_i, y_i), $i = 1,...,N$, the sensitivity based on each sample is $E(l,1), E(l,2),..., E(l,N)$. So, the sensitivity of the hidden node could be expressed by a point Se, $Se = (E(l,1), E(l,2),..., E(l,N))$, in space. In real applications, Se can not measure it directly, so, the Murkowski distance between Se and space original point O can be used when $q = 2$, it is the Euclidean distance.

$$d(Se, O) = \left(\sum_{i=1}^{N} |E(l,i) - O_i|^q \right)^{1/q} = \left(\sum_{i=1}^{N} |E(l,i)|^q \right)^{1/q} \tag{8}$$

$d(Se, O) \in (0, +\infty)$, so it is similar to $E_{avg}(l)$ and not consistence with people's cognitive habits, so it need to be transformed, its final result is

$$Sen(l) = e^{\frac{-\alpha}{d(Se,O)}} = e^{-\alpha \|\beta_l\|_q^{-1} \left(\sum_{i=1}^{N} |h_{li}|^q \right)^{-1/q}} \tag{9}$$

$Sen(l) \in (0,1)$, $Sen(l) \to 1$ shows the sensitivity is greater, otherwise it is less, which fully meets people's cognitive habits. Besides, it could measure the degree of sensitivity in different applications in one criterion. α is adjusting factor, which is used to manage the distribution of the sensitivity.

IOP Conf. Series: Materials Science and Engineering **320** (2018) 012015 doi:10.1088/1757-899X/320/1/012015

3.2 The output weight

The output weight of hidden nodes is the only parameter to be regulated during training, it is vital for ELM and its improved algorithms. After removing hidden nodes, the important task is to update the output weight, according to equation(5), matrix H and its inverse matrix H^{-1} should be obtained first. Set the s^{th} hidden node is the removing hidden node, so the s^{th} row and column of the matrix H is transformed to the N^{th} row and n^{th} column, and the result matrix H_1' is further changed to

$$H_1' = \begin{bmatrix} H_1 & u \\ g & k \end{bmatrix} \tag{10}$$

in which, $u = [G(a_s,b_s,x_1)\cdots G(a_s,b_s,x_{s-1})\,G(a_s,b_s,x_{s+1})\cdots G(a_s,b_s,x_N)]^T$ is (N-1)×1 vector, $k = G(a_k,b_k,x_k)$ is a constant, $g = [G(a_1,b_1,x_s)\cdots G(a_{s-1},b_{s-1},x_s)\,G(a_{s+1},b_{s+1},x_s)\cdots G(a_M,b_M,x_s)]^T$ is 1×(N-1) vector. Referencing the literature[9], the inverse matrix of H_1' can be deployed as

$$H_1'^{-1} = \begin{bmatrix} \left(H_1 - ukg\right)^{-1} & -H_1^{-1}u\left(k - gH_1^{-1}u\right)^{-1} \\ -kg\left(H_1 - kug\right)^{-1} & \left(k - gH_1^{-1}u\right)^{-1} \end{bmatrix} \tag{11}$$

set $G = \left(H_1 - ukg\right)^{-1}$, so

$$H_1 = G^{-1} + kug \tag{12}$$

Referencing the literature[10], the inverse matrix of H_1 is

$$H_1^{-1} = G - \frac{GkugG}{1 + gGu} \tag{13}$$

So, output matrix H_1 and its inverse matrix H_1^{-1} is known, the output weight is,

$$\beta_1 = \left(H_1^T H_1\right)^{-1} H_1 Y_1 \tag{14}$$

where, $Y_1 = \left[y_1,...,y_{s-1},y_{s+1},...,y_N\right]^T$ 。

3.3 Description of the improved ELM

After detail analysis, the description of the improved ELM is

S1: Based on the specific problems, initial the threshold parameters ε and ε_{se}, and then it is to establish the SLFNs with enough hidden nodes by ELM, the number of hidden nodes M is less than the size M of training samples, M<N. Finally, according to the equation(5), the output weight β and output matrix H of hidden nodes is obtained.

S2: For each input training sample, the sensitivity $E(s,i)$ of each hidden node is obtained by equation(6) and then the sensitivity $Sen(l)$ can be computed by equation(9).

S3: For current inputting sample x_{N+1}, the output result y_{N+1}' of SLFNs could be obtained, for the given threshold ε, if the error of real output result y_{N+1}' and expected output result y_{N+1} meets $SSE(y_{N+1} - y_{N+1}') < \varepsilon$, it could remove some hidden node with less sensitivity, go to Step S4, otherwise, Ready for the prediction or classification based on the next new sample.

S4: For the given threshold ε_{se}, if the sensitivity of the s^{th} hidden node meets $Sen(s) < \varepsilon_{se}$, s<M, this hidden node could be removed.

S5: After removing hidden nodes, update the output weight by equation(14), and it continues to analyze if there is hidden nodes with less sensitivity, go to Step S3.

4. Tests

The performance of IPS-ELM is tested on the benchmark problems, inlcuding two regression problems (Auto-MPG and Abalone) and two classification problems (Image segment and DNA), SA-ELM[4] and ImSAP-ELM[3] are selected to be compared with IPS-ELM.

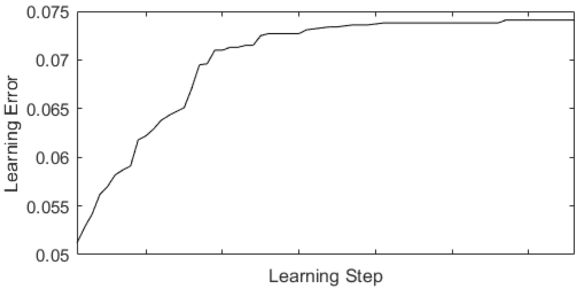

Figure 1. the variation curve of learning error.

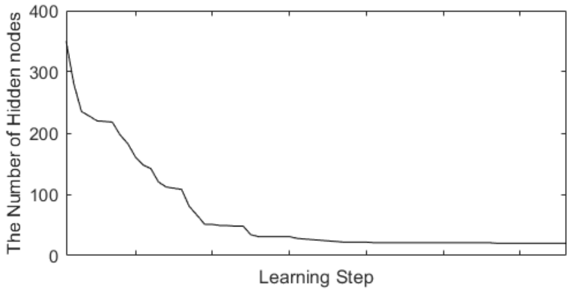

Figure 2. the variation curve of the number of hidden nodes.

Figure 1 and Figure 2 show the variation of learning error and the number of hidden nodes during the training respectively on Auto-MPG. From figures, the learning error is larger and larger with the number of hidden nodes decreasing, and the learning error finally reaches 0.073, it is fully acceptable.

Table 1. Comparison between IPS-ELM and other algorithms on regression problems

Algorithm	Auto-MPG			Abalone		
	Number of hidden node	Training time /s	error	Number of hidden node	Training time/s	error
IPS-ELM	21	1.6195	0.0008	12	0.1931	0.0007
SA-ELM	21	1.6287	0.0009	14	0.2161	0.0006
ImSAP-EM	23	2.0853	0.0021	20	0.2836	0.0012
ELM	52	1.9837	0.0027	25	0.0187	0.0037

Table1 is the testing results of the two regression problems, the finally error of the four algorithms is a little difference, but they are fully acceptable. The max error is 0.0037 of ELM on Abalone problem, and the minimum error is 0.0006 of SA-ELM on Abalone problem. So, compared with ELM, the accuracy of IPS-ELM is improved by 70% and 81% respectively. Compared with the improved ELM, it is improved by 74% and 50% respectively. The number of hidden nodes is smaller, can reach 21, so the SLFNs is more simple, learning time is shorter and can reach 1.6s and 0.19s.

Table 2. Comparison between IPS-ELM and other algorithms on classification problems

Algorithm	Image segment			DNA		
	Number of hidden node	Training time /s	error	Number of hidden node	Training time/s	error
IPS-ELM	49	0.1583	0.00322	156	0.4182	0.0051
SA-ELM	52	0.1975	0.00348	172	0.4837	0.0051
ImSAP-ELM	70	5.2183	0.00329	201	4.9235	0.0047
ELM	200	1.4	0.78	347	8.79	1.25

Table 2 is the testing results of the two classification, finally error (standard error) of the four algorithms is a little difference, in some applications, the accuracy of ELM is not acceptable, whose max error reaches 1.25. The classification accuracy of IPS-ELM is fully acceptable, the max is only 0.0051, equals with SA-ELM and only a little bigger than ImSAP-ELM by 0.0004, which is negligible. The number of hidden nodes by IPS-ELM is smaller, it is more simple, learning time is shorter and could reaches 0.15s and 0.41s respectively.

5. Conclusion

After deeply analyzing ELM and related algorithms, there are some weak points in their performances, such as the learning error, the number of hidden nodes and so on, so, based on these, a novel algorithm is proposed, it modified the analyzing method of the sensitivity of hidden nodes, which fully meets people's cognitive habits, based on the sensitivity, it removes the hidden nodes with less sensitivity on the promise of meeting the accuracy. After performance tests, IPS-ELM is better and acceptable.

6. ACKNOWLEDGEMENTS

This research is partially supported by the Science and Technology Funds of Dalian under Grant No. 2015A11GX010 and by Natural Science Foundation of Liaoning Province of China under Grant No.20170520196.

7. REFERENCE

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

[2] Haipeng C, Xiaohang T, Yu W, et al. Short-Term Parking Space Prediction Based on Wavelet-ELM Neural Networks[J]. Journal of Jilin University(Science Edition). 2017, 55(2): 388-392.

[3] DU Zhan-long, Xiao-Min L, Lei-Ping X, et al. Improved sensitivity-analysis based pruning extreme learning machine[J]. Control and Decision. 2016, 31(2): 249-255.

[4] Fan-Jun L, Hong-Gui H, Jun-Fei Q. Pruning algorithm for extreme learning machine based on sensitivity analysis[J]. Control and Decision. 2014, 29(6): 1003-1008.

[5] Li G, Liu M, Dong M. A new online learning algorithm for structure-adjustable extreme learning machine[J]. Computers & Mathematics with Applications. 2010, 60(3): 377-389.

[6] Alencar A S C, Rocha Neto A R, Gomes J P P. A new pruning method for extreme learning machines via genetic algorithms[J]. Applied Soft Computing. 2016, 44: 101-107.

[7] Yin J. A variable-structure online sequential extreme learning machine for time-varying system prediction[J]. Neurocomputing. 2017, 261(Supplement C): 115-125.

[8] Shao Z, Er M J. An online sequential learning algorithm for regularized Extreme Learning Machine[J]. Neurocomputing. 2016, 173(2016): 778-788.

[9] Peking U. Advanced Algebra (Second Edition)[M]. Beijing: Higher Education Press, 1987.

[10] Wenyu S, Yaxiang Y. Optimization theory and methods[M]. Beijing: Science Press, 1997.