

Extreme learning machine based on improved genetic algorithm

Hai LIU^{1,a}, Bin JIAO^{2,b}, Long PENG^{3,c}, Ting ZHANG^{4,d}

¹Student of Shanghai Dlanji University

²Teacher of Shanghai Dianji University

³Student of Shanghai Dlanji University

⁴Student of Shanghai Dlanji University

^aemail: liuhaiariel@qq.com, ^bemail: jiaob@sdju.edu.cn, ^cemail: 375696898@qq.com, ^demail: 1013753534@qq.com

Keywords: Improvedgeneticalgorithm; Extreme learning machine; Function approximation

Abstract. This paper puts forward a novel algorithm called extreme learning machine (ELM) which is optimized by improved genetic algorithm(IGA), and points out the weaknesses of ELM. The input weights and thresholds randomly generated by ELM are optimized by IGA. After it, ELM can get the more effective input weights and thresholds and be better applied in function approximation. The results of simulation shows that the optimized algorithm has a high approximation accuracy and faster convergence speed.

Introduction

The principle of classic function approximation has the rigorous theoretical analysis and mature system, but it results in many algorithms with some common faults such as a large amount of calculation, poorer adaptability, higher demand for the model and data, strong dependence, etc. The superiority of neural network applied in function approximation can be shown in many cases. For example , the mode characteristics of the data is not very clear and the data is fuzzy, nonlinear or with more noise, etc. The research of function approximation by neural network has a very good theoretical value[1].

Huang et al., in 2004, put forward a novel feed-forward neural network[2]called extreme learning machine(ELM). In ELM, the input weights and thresholds in the hidden neurons are randomly assigned, and the output weight matrix is calculated by Moore-Penrose(MP). Compared to the traditional feed-forward neural network, ELM has the features of faster learning speed, higher precision, simple parameter adjustment and so on. However, although the randomly generated input weights and thresholds avoid the problems of local optimal value and over-fitting, in ELM it results in the failure of some nodes in the hidden layer.

Therefore, the purpose of this paper is to use IGA to optimize ELM and get the more effective input weights and thresholds which can improve the effectiveness of nodes in the hidden layer in ELM and then improve the function approximation accuracy of ELM as a whole.

The principle of function approximation

Function approximation[3][4] is an important part of function theory and important in the numerical calculation

The problem of how to approximate the function is a basic problem of function approximation. Using simple function $g(x)$ to replace function $f(x)$ approximatively is one of the most basic concepts and methods in computational mathematics. Approximate replace is also known as approximation. Function $f(x)$ is described as approximated function and function $g(x)$ as approximate function. The difference between the two is

$$R(x) = f(x) - g(x) \quad (1)$$

It is called as the error or remainder of approximation.

The problem solved by function approximation is how to solve the approximate equation with the small amount of calculation under the given precision. It is described as: for functions A , a class of function $f(x)$ is required to find a function $g(x)$ in functions B ($B \subset A$) ,which is simple and easy to calculate, to make the difference between $g(x)$ and $f(x)$ reach the minimum in a way.

Generally speaking, the most common functions A are the continuous functions in the interval of $[a,b]$. The most commonly used functions B are rational fraction, trigonometric polynomial, algebraic polynomial, etc.

Extreme learning machine

ELM [5] [6] is a novel algorithm for training single-layer feed-forward neural network [4]. Supposing having N learning samples, its set is described as $N = \{(x_n, y_n) | n = 1, 2, \dots, N; x_n \in R^{d_1}; y_n \in R^{d_2}\}$. If its structure has K nodes in the hidden layer, where $g(x)$ is the activation function, the input and output of the ELM network is described as:

$$T_q = \sum_{k=1}^K \beta_k g(\omega_k * x_q + b_k) \quad (2)$$

$$g(X) = \frac{1}{1 + e^{-X}} \quad (3)$$

Where $\omega_k = [\omega_{k1}, \omega_{k2}, \dots, \omega_{kd_1}]^T$ or $\beta_k = [\beta_{k1}, \beta_{k2}, \dots, \beta_{kd_2}]^T$ represents the connection weights between the input layer or the output layer and k th node in the hidden layer ; b_k is the bias of the k th hidden node.

Formula (2) can be written as the following form:

$$H\beta = T \quad (4)$$

Where H represents

$$H = \begin{bmatrix} g(\omega_1 \cdot x_1 + b_1) & \cdots & g(\omega_K \cdot x_1 + b_K) \\ \vdots & \ddots & \vdots \\ g(\omega_1 \cdot x_Q + b_1) & \cdots & g(\omega_K \cdot x_Q + b_K) \end{bmatrix}$$

$$\beta = [\beta_1^T, \beta_2^T, \dots, \beta_K^T]^T_{K \times d_1}$$

$$T = [T_1^T, T_2^T, \dots, T_K^T]^T_{K \times d_1}$$

So using the least square solution of minimum 2-norm and equation (4) to solve the weight vector is

$$\hat{\beta} = H^+ T \quad (5)$$

where H^+ represents Moore-Penrose of H .

Therefore, after the training samples set N ,activation function $g(x)$ and K nodes in the hidden layer given, the steps of the ELM algorithms are as follows:

(1)randomly assign the input weights and thresholds: ω_k and b_k .

(2)calculate the output matrix H in the hidden layer.

(3)calculate the output weights β in the hidden layer.

ELM optimized by IGA

GA is an evolutional algorithm and its principle is to follow the law:'Evolution, the theory goes, guarantees survival to the fittest' in the nature. It uses a population of strings to encode the initial candidate solutions. GA then employs genetic operators (selection, mutation, crossover) to generate new populations based on the initial population, and gradually evolves towards the best solution.

The ELM network optimized by IGA is to use IGA to optimize the initial weights and thresholds randomly assigned by the ELM network[7], then gains the optimal weights and thresholds to predict the output of the network. The algorithm mainly includes three important parts: making sure the structure of the ELM network, weights and thresholds optimized by IGA, the training and prediction of the ELM network.

1) create the ELM network: finish the structure of the ELM network according to the number of the input and output parameters.

2) the optimization of IGA: when optimizing the weights and thresholds of the ELM network, IGA calculates the individual fitness through (6); where Y represents the individual fitness; E_i and F_i represent the expected output or predicted output of the i th node; k represents the node of the output; l represents the coefficient.

$$Y = l \sum_{i=1}^k \text{abs}(E_i - F_i) \quad (6)$$

GA[6] employs genetic operators (selection, crossover and mutation) to find out the fitness value of the best individual. The selection operator generally employs the proportional selection method based on fitness. The choice probability of each individual is shown in (7).

$$p_i = \frac{l}{Y_i \sum_{i=1}^N l} \quad (7)$$

where Y_i represents the fitness value of individual i ; N represents the number of individual in populations. If the greater the individual fitness is, the bigger the choice of it selected is. To choose the mating individual, the selection operator requires repeated selections. A number is randomly generated in [0,1] every selection. The random number is treated as the optional pointer to determine the selected individual. Due to the random operation, the selection error of this selection method is relatively large and sometimes even the higher fitness can't be chosen.

So, in order to reduce the selection error, we employ the championship selection operator to improve the selection operator [8]. When the operator is choosing, it chooses k individuals from the population and finds the individual with the best fitness as the best individual from these k individuals. The best individual is one of the next generation population. The process produces a new population by being repeated n times. The advantages of this method are with no requirements for positive or negative value of the individual fitness and with a larger probability to guarantee the best individual selected or the worst individual eliminated.

The crossover operator employs the cross method of real. The r th chromosome a_r and the s th chromosome a_s are at the j th cross. In (8) and (9), c represents a random number between 0 and 1.

$$a_{rj} = a_{rj}(1-c) + a_{sj}c \quad (8)$$

$$a_{sj} = a_{sj}(1-c) + a_{rj}c \quad (9)$$

The mutation operator is shown in (10); where a_{ij} represents the j th gene of the i th individual at the mutation.

$$a_{ij} = \begin{cases} a_{ij} + \frac{m(g_{\max} - g)}{g_{\max}}(g_{\min} - g_{ij}) & 0 \leq m < 0.5 \\ a_{ij} + \frac{m(g_{\max} - g)}{g_{\max}}(g_{ij} - g_{\max}) & 0.5 \leq m < 1 \end{cases} \quad (10)$$

Where g represents the current number of iterations, g_{\max} represents the biggest number of iterations; a_{\max} or a_{\min} represents the upper or lower of gene a_{ij} .

3) the prediction of the ELM network: the initial weights and thresholds are assigned by the optimized best individual. Last, traingthe network predicts the output[9] [10].

The flowchart of the ELM network optimized by IGA is shown in Fig.1.

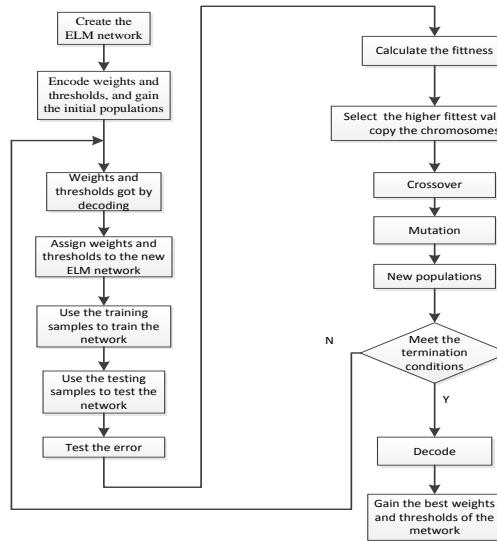


Fig.1. The flowchart of IGAELM

Simulation

We choose function $y = \sin(2\pi x)$, $x \in [0, 3000]$, a simple nonlinear continuous function as an example. ELM, GAELM or IGAELM is applied to function approximation by operating the software of MATLAB. Fig.2 shows the approximation curve of ELM; in the ELM, ‘sig’ represents the activation function. Fig.3 shows the approximation curve of GAELM; in the GA, MAGEN is equal to 100. Fig.4 shows the approximation curve of GAELM; in the GA, MAGEN is equal to 100. curve of ELMcurve of GAELM curve of IGAELM

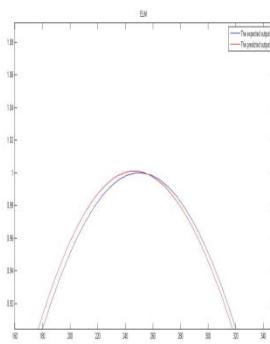


Fig.2. The approximation

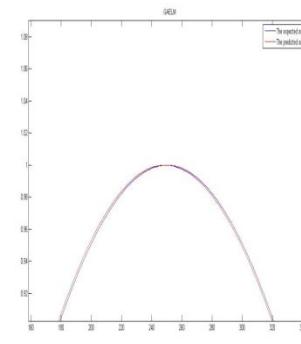


Fig.3. The approximation

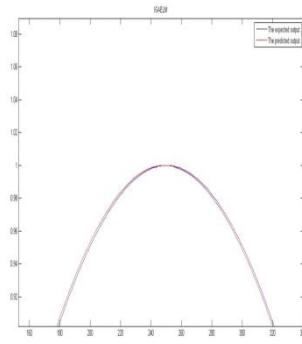


Fig.4. The approximation

The approximation curve of ELM is on the left and the curve of the original function on the right in Fig.2. Fig.3 and Fig.4 are the same as Fig.2. However, the error drawings of GAELM and IGAELM for function approximation are shown in Fig.5 and Fig.6. The approximation error curve

of GAELM is under the approximation error curve of ELM in Fig.5. Fig.6 is the same as Fig.5.

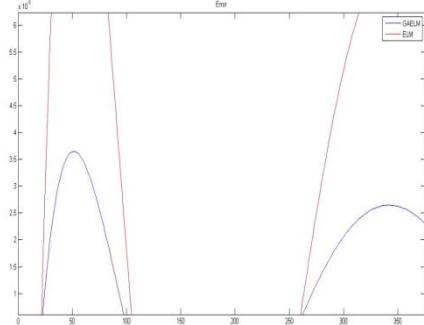


Fig.5. The approximation error drawing of GAELM

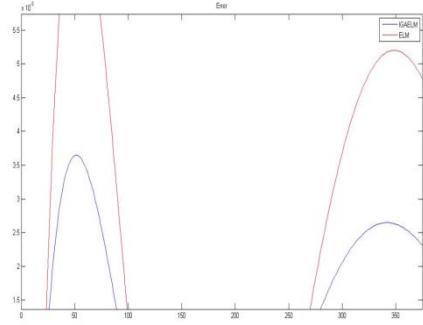


Fig.6. The approximation error drawing of IGAELM

Conclusion

This paper puts forward the ELM network based on IGA. ELM gains the best weights and thresholds based on the optimized initial weights and thresholds, and then predicts the output of the network. The algorithm is not easy to fall into the local minimum point and has the good approximation for function. The results of simulation show that the approximation precision of this algorithm is better than ELM and GAELM. At the same time, the approximation error of IGAELM is smaller than GAELM. In the future, this algorithm should possibly be used into a more wide area.

Acknowledgement

In this paper, the research was supported by the project of Shanghai Science and Technology Commission (Project No.13DZ0511300) .

References

- [1] Rui Dong. The method of function approximation based on Neural Network [D]. Northeast Normal University, 2011.
- [2] Huang Guangbin, Zhu Qinyu, Siew Cheekheong. Extreme learning machine: theory and applications [J]. Neurocomputing, 2006, 70 (1/2/3) : 489-501.
- [3] Guo-qiang Zhu, Shi-rong Liu, Jin-shou Yu. Support vector machine and its application in function approximation [J]. Journal of East China University of Science and Technology, 2002, 05:555-559+568.
- [4] Park J.Sandberg I W.Univerval Approximation Using Radial Basis Function Networks. Neural Computation, 1991, 3 (2):246-257.
- [5] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
- [6] Long-yi Chen, Yan-lin He, Xiao-yong Lin. The research and application of extreme learning machine based on informationentropy [J].Computer and Applied Chemistry, 2014, 08:982-986.
- [7] Xiao-hong Ren, Wei-dong Xu, Li-xin Liu, Tian-peng Zhou, Ying-gao Le. BP neural network applied the thermal error compensation of Computer numerical machine tools and based on genetic algorithm [J]. Manufacturing Automation, 2011, 09:41-43
- [8] Chong-sheng Shen. In genetic algorithm the selection operator is achieved by MATLAB [J]. Journal of Shanghai Institute of Technology (Natural Science), 2003 01: 199-202.
- [9] Jie Wang, Hao-yang Bi. A kind of extreme learning machine based on particle swarm

optimization [J].Journal of Zhengzhou University (Science Edition), 2013, 01:100-104.

[10] Lu Gan.The research and application of extreme learning machine [D]. Xidian University, 2014.