Evaluation model of enterprise operation based on BP neural network optimization algorithm

Yan Zhang 1*, Ziwei Hu 2, Li Ji 3, Nanlin Sun 4, Yuluo Lin 5

1 Tongling University, Tongling, Anhui, 244000, China
2 Beijing Normal University - Hong Kong Baptist University United International College, Zhuhai, Guangdong, 519000, China
3,4 University of Chinese Academy of Social Sciences, Beijing, Beijing, 102488, China
5 Chengdu University of Technology, ChengDu,Sichuan, 610000,China

Corresponding author’s e-mail: emilyzhang919@gmail.com

Abstract. The parameter selection of the traditional BP neural network (BPNN) has randomness, which makes the network prone to local extreme values during the calculation process. In order to solve this problem, this paper introduces the bat algorithm (BA) to optimize the parameter selection process of the BPNN and apply the algorithm to evaluate the enterprises’ operating condition, a corresponding evaluation model of the enterprises’ operating condition is established, and the evaluation model is applied to the prediction of the enterprises’ future operating condition and compared with the prediction effect of the traditional BPNN model. The prediction accuracy of the BPNN optimization algorithm is higher than the prediction accuracy of the traditional BPNN. The established enterprise operation evaluation model can effectively predict the future operation of the enterprise.

1. Introduction

There are many methods for evaluating the operating condition of enterprises, and most of them are completed under the model classification. Since traditional pattern classification methods are difficult to achieve the desired effect in practical applications, artificial neural network technology has gradually become an effective tool for pattern classification. The BP neural network (BPNN) algorithm is a more commonly used classification method, but the BPNN algorithm has some unavoidable defects: such as network over-learning, easy to fall into local extreme values, and poor generalization ability, especially if the BPNN is initialized. The poor selection of weights and thresholds will make it difficult for the BPNN to converge, and thus the prediction effect is poor. In recent years, some scholars have combined many intelligent algorithms with BPNN to optimize the network parameters of BPNN [1].

The bat algorithm (BA) is an excellent method to obtain the global optimal solution. The algorithm uses a speed-displacement search model, which is easy to operate and has low computational complexity. Through the global search results, the optimal solution can be guaranteed with a large probability. Taking into account the advantages of the BA algorithm, the article uses the bat algorithm to search for the better initial weights as well as thresholds of the BPNN to compensate for the random defect in the selection of the weights and thresholds of the BPNN connection, so as to achieve complementary advantages and improve the stability of the algorithm and the purpose of global search capabilities, and apply improved algorithms to the evaluation of enterprises conditions.
2. BPNN optimization algorithm

2.1 BPNN

BPNN is actually based on the error back propagation as the main working form. This form has the ability to approximate any continuous function, and the most obvious property is nonlinear mapping. The central theoretical basis of BP neural network is that input the required learning samples accurately, and carry out multiple adjustment training of network weights and deviations by means of back propagation, so that the network error square sum can be minimized, and the condition for the end of training is the threshold $\theta$ is greater than or equal to the sum of squared errors, and the weights and deviations of the network are completely saved. To set various parameters, here we define the number of neurons in the input layer as $m(i = 1, 2, \ldots, m)$, and the number of neurons in the hidden layer as $n(j = 1, 2, \ldots, n)$, the number of neurons in the output layer is $l(k = 1, 2, \ldots, l)$, and the weight between the neurons in the input layer and the neurons in the hidden layer is defined as $v_{ij}$. The weights between neurons in the output layer are defined as $w_{jk}$. According to the above settings, we construct a 3-layer BPNN, as shown in Figure 1 [2-3].

![Figure 1. 3-layer BPNN](image)

Among them, input layer input is defined as $x_i$; hidden layer output is defined as $h_j$; output layer output is defined as $y_k$; expected output is defined as $d_k$, threshold value is $\theta$, output error is defined as $E$, prediction error is defined as $e_k = y_k - d_k$, and error accuracy is defined as $\mu$. The maximum number of learning times is defined as $N$, the learning rate is defined as $\eta \in (0, 1)$, and the error signal is defined as $\delta$.

About updating weights:

(1) The BP neural network is initialized. According to the input and output of the whole network, the relevant parameters are determined, including the number of nodes in the input layer, the hidden layer and the output layer, the weight, the threshold and the learning rate.

(2) Calculate the output of hidden layer.

$$y_i = f\left(\sum_{j=1}^{m} v_{ij} - \theta_i\right)$$  \hspace{1cm} (1)

(3) Perform output layer output calculation.

$$e_k = y_k - d_k$$  \hspace{1cm} (2)

(4) Correction of weights and thresholds

The correction of the weight is:
\[
\Delta w_j = -\eta \frac{\partial E}{\partial w_{jk}} = \eta (d_k - y_k) y_j (1- y_k) y_j \\
\Delta v_j = -\eta \frac{\partial E}{\partial v_{jk}} = \eta \sum_{k=1}^{l} \delta_k w_{jk} y_j (1 - y_j) x_i
\]

The threshold value is modified to:

\[
\Delta \theta_k = \epsilon_k, \quad k = 1, 2, \ldots, l \\
\Delta \theta_j = \eta_i (1 - y_j) \sum_{k=1}^{l} w_{jk} \epsilon_k, \quad j = 1, 2, \ldots, n
\]

(5) If the condition is met, the algorithm stops; if the end condition is not met, return to execution (2).

The transfer functions \(f(x)\) are all sigmoid functions, and the sigmoid function is a continuously derivable function (where \(e\) is a natural constant) \[4\].

\[f(x) = 1/(1 + e^{-x})\] (5)

2.2 Bat algorithm

The essence of the bat algorithm (BA) is the deduction of group evolution. The first step of this algorithm is to realize the random initialization of the entire population in the feasible solution space, so as to determine the initial position and initial velocity of the individual within the group. In this process, the position is used In order to characterize the problem solution, and then evaluate the entire group, finally determine the best position in the group. Then, update the individual flight speed and position according to equations (6) and (7) respectively \[5\]:

\[F_i = F_{\text{min}} + (F_{\text{max}} - F_{\text{min}}) \beta\] (6)

\[V_i^{t+1} = V_i^t + (P^t_i - P_{\text{*}}) \cdot F_i\] (7)

\[P_i^{t+1} = P_i^t + V_i^{t+1}\] (8)

Where \(F_i\) is the pulse frequency of the \(i\)-th bat, \(F_{\text{min}}\) and \(F_{\text{max}}\) are the minimum and maximum values of the pulse frequency, \(\beta\) is a random number obeying uniform distribution on \([0, 1]\), \(V_i^{t+1}\) and \(V_i^t\) are flight speed of the \(i\)-th bats in generations \(t+1\) and \(t\); \(P_i^t\) is the position of the \(i\)-th bat in generation \(t\); \(P_{\text{*}}\) is the optimal position of the bat in the current group; \(P_i^{t+1}\) is the \(i\)-th bat in \(t+1\) generation position.

When the algorithm performs a local search operation, once a solution is selected from the existing optimal solutions, then each bat in the group follows the principle of random walk to generate a local new solution\[6\]

\[P_{\text{new}} = P_{\text{old}} + \epsilon A^t\] (9)

Where \(\epsilon \in [-1, 1]\) is a random number; \(A^t\) is the average volume of all bats in the same time period. When searching for prey, the ultrasonic pulse sound emitted by the bat in the initial stage is large, but at the same time its frequency is low, which can effectively achieve wide-area search. When the bat detects the prey, it gradually reduces the former intensity Increase the frequency of the latter to achieve precise positioning of prey, and use equations (10) and (11) to simulate this search feature \[7\].

\[A_i^{t+1} = \alpha \cdot A_i^t\] (10)

\[R_i^{t+1} = R_0 (1 - e^{\gamma t})\] (11)

Where \(R_i^{t+1}\) is the pulse frequency of the \(i\)-th bat at \(t+1\) generation; \(R_0\) is the maximum pulse frequency of the \(i\)-th bat; \(\gamma > 0\) is the pulse frequency increase factor; \(A_i^{t+1}\) and \(A_i^t\) are Is the sound
intensity of the \( i \)-th bat transmitting pulses at generations \( t+1 \) and \( t \); \( \alpha \in [-1,1] \) is the attenuation coefficient of the pulse sound intensity.

2.3 Algorithm flow

The algorithm flow of this paper is as follows:

(1) Encoding. The individual coding method of the bat algorithm is real coding, and the parameters of the BP NN are coded as a whole parameter. Each individual contains all the weights and thresholds of BPNN and can represent a BPNN structure.

(2) Randomly generate \( n \) bats to form the initial bat population, and randomly initialize the bat's position, velocity, volume, and pulse incidence in the search space.

(3) Connect the position of each individual bat to fitness.

(4) Determine whether the algorithm meets the termination condition, that is, whether the maximum number of iterations is reached. If the termination condition is met, the algorithm ends and the optimal solution is output. Otherwise, go to step (5).

(5) Update the speed and position of the bat according to equation (6), (7) and (8).

(6) Generate a random number \( \text{Rand1} \). If we find that \( \text{Rand1} > R_i \) (\( R_i \) is the pulse frequency of the \( i \)-th bat), use equation (9) to get \( P_{\text{new}} \).

(7) Evaluate the location.

(8) Generate a random number \( \text{Rand2} \), if we find that \( \text{Rand1} < A_i \) (\( A_i \) is the pulse intensity of the \( i \)-th bat) and \( f(x_i) < f(x_*) \), then update the individual bats and reduce \( A_i \) by equation (5), increase \( R_i \) by equation (6).

(9) If the termination condition is met, the target value is output and the program execution is terminated, otherwise go to (3).

(10) Decode the individual optimized by BA as the initial connection weight and threshold of BPNN.

3. Establishment and simulation of evaluation model

Taking the 600 listed enterprises announced by the two exchanges in Shenzhen and Shanghai in 2017 as the training sample, there are 300 enterprises with normal operations, 300 enterprises with poor operations, and a total of 20 enterprises in the test sample. There are 10 enterprises with normal operations and 10 enterprises with poor operations.

3.1 Evaluation model

Using the Mann-Whitney U test method to test the correlation between the 11 selected initial financial indicators and the enterprises' operating conditions by SPSS software. Finally, 9 indexes with \( \text{sig. value} < 0.01 \) were selected, these indexes are earnings per share, net assets per share, net asset yield, net asset growth rate, total asset growth rate, total asset turnover rate, net profit rate, operating income growth rate, fixed asset growth rate. A three-layer BPNN is established, and neurons in the input layer only play a connecting role, and do not perform signal transformation. The selected 9 financial indicators are used as the input of the BP neural network, so the input layer node is 9; use "0" to represent poorly operating enterprises, and "1" to represent normal operating enterprises, so the output layer node is 1; The number of nodes is selected as 15 by trial and error. The optimal initial weights and thresholds of the BPNN obtained by using the BA algorithm are
Therefore, the BPNN-based evaluation model of enterprise operating conditions is

$$\mathbf{B}_1 = [-1.4557 \ 1.4772 \ 0.3188 \ 0.1994 \ -1.4202 \ 1.4121 \ 0.4882 \ -0.5962]$$

$$\mathbf{V}_1 = [-0.3309 \ -1.8014 \ 1.6109 \ 1.7791 \ -0.0365 \ -0.0430 \ -0.6491 \ 1.6002]$$

$$\mathbf{B}_2 = -1.4721$$

3.2 Network simulation

In order to test the prediction effect of the evaluation model, 20 test samples were predicted using the BPNN algorithm and the method in this paper, including 10 enterprises with normal operations, indicated by "1", and 10 enterprises with poor operations, using "0" Said. The simulation results of the two algorithms are shown in Figure 1, and the accurate recognition rate is shown in Table 1. In Figure 1, the output value of the test sample is equal to 1 or greater than 0.5, the enterprise is considered to be operating normally; the output value is equal to 0 or less than 0.5, the enterprise is considered to be in poor operating condition. We can find out some conclusions from Figure 1 and Table 1: The average accurate recognition rate of BPNN for test samples is 70%, and the average accurate recognition rate of optimized BPNN for test samples is 95%, which shows that the prediction accuracy of BPNN proposed in this paper is obvious Better than traditional BPNN.

$$y = f[\sum_{j=1}^{15} v_j g(\sum_{i=1}^{9} w_{ji} x_i + B_{ij}) + B_2]$$

The transfer function from the input layer to the hidden layer is \textit{tansig}, and the hidden layer and the transfer function to the output layer is \textit{logsig}.
Figure 2. Identification of test samples

### Table 1. The average accurate recognition rate

| Method       | Index                        | Enterprises with normal operations | Enterprises with poor operations | Total |
|--------------|------------------------------|-----------------------------------|---------------------------------|-------|
|              | Number of test samples       | 10                                | 10                              | 20    |
| BPNN         | Identification number        | 6                                 | 8                               | 14    |
|              | Accurate recognition rate(%) | 60                                | 80                              | 70    |
| This paper   | Number of test samples       | 10                                | 10                              | 20    |
| method       | Identification number        | 9                                 | 10                              | 19    |
|              | Accurate recognition rate(%) | 90                                | 100                             | 95    |

### 4. Conclusion

In order to make up for the shortcomings of BPNN and improve the accuracy of BPNN algorithm prediction, the BA algorithm is used to optimize the initial weights and thresholds of BPNN and apply them to the evaluation of the operating condition of listed enterprises. Forecasting accuracy, so the evaluation model of business operation based on BPNN optimization algorithm proposed in this paper is a more effective model and has better application value.

### References

[1] Yi, J.Q., Wang, Q., Zhao, D.B., et al. (2007) BPNN prediction-based variable-period sampling approach for networked control systems. Applied Mathematics & Computation., 185:976-988.

[2] Mozaffari, S. (2018) Parallel image encryption with bitplane decomposition and genetic algorithm. Multimedia Tools & Applications, 77:1-21.

[3] Zhao, Y., Nan, J., Cui, F.Y., et al. (2007) Water Quality Forecast Through Application of BPNN at Yuqiao Reservoir. Journal of Zhejiang University-Science A(Appplied Physics & Engineering), 8:1482-1487.

[4] Guo, Z.H, Wu, J., Lu, H.Y., et al. (2011) A case study on a hybrid wind speed forecasting method using BPNN. KNOWLEDGE-BASED SYSTEMS, 24:1048-1056.

[5] He, X. S., Ding, W. J, Yang, X. S. (2014) Bat algorithm based on simulated annealing and Gaussian perturbations. Neural Computing & Applications, 25:459-468.
[6] Rajeswari, C., Sathiyabhama, B., Devendiran, S., et al. (2015) Diagnostics of gear faults using ensemble empirical mode decomposition, hybrid binary bat algorithm and machine learning algorithms. Journal of Vibroengineering, 17: 1169-1187.

[7] Baziar, A., Rostami, M. A., Akbari-Zadeh, M. R. (2014) An intelligent approach based on bat algorithm for solving economic dispatch with practical constraints. Journal of Intelligent & Fuzzy Systems, 27:1601-1607.