Eng Int Syst (2021) 1: 45–53
© 2021 CRL Publishing Ltd

Engineering
Intelligent Systems

Interactive Genetic Algorithm Based on the BP Neural Network Proxy Model

Yingchun Zhao*

Library of Criminal Investigation Police University of China, Shenyang 110035, China

A proxy model based on the BP neural network is designed, and interactive genetic methods are used to solve the problem of input parameter selection difficulty within the BP neural network. The global optimization capacity of the interactive genetic algorithm is used to address the issue that the BP neural network is prone to falling into the trap of using a local minimum due to the random selection of the initial weight. The results of the experiment verified that the BP neural network proxy model can perform self-learning of expert experience and can make intelligent predictions of occupational competence.

Keywords: Prediction of Competence; BP Neural Network; Interactive Genetic Algorithm; Proxy Model

1. INTRODUCTION

An artificial neural network is an abstract computing model of the human brain and a type of Artificial Intelligence (AI). Attributed to its highly nonlinear mapping learning and induction capacity, it has been extensively used in various prediction models [1–2]. The BP neural network is an artificial neural network with back propagation of errors, and it is one of the most widely used models since the introduction of neural network models. Approximately 70% to 80% of the neural network models are BP neural networks or their variants [3–5]. The BP neural network model is not without its defects. There is no effective method for the selection of input parameters. When faced with an input data set with multiple features there can be a learning instability [6–7]. At the same time, the BP neural network is also highly sensitive to the initial weight. If the initial weight is not selected correctly, it can lead to the neural network falling into the local minimum, and thus the global optimal solution cannot be obtained [8–10]. The prediction of occupational competence is highly demanding for the accuracy of the prediction results, and prediction models with higher accuracy must be established [10–13]. In this paper, the advantages of the classification model based on the neural network are leveraged, and the information gain algorithm is used to carry out effective selection and verify the optimal feature combination, which accounts for the defects of the BP neural network where the input parameters cannot be determined [14–15]. The excellent global optimal performance of the interactive genetic algorithm is used to perform initialization on the weights of the BP neural network. The three are combined to establish a BP neural network prediction model. The results of the experiment verify that the proxy model has an outstanding performance and can accomplish prediction tasks successfully.

2. FRAMEWORK OF THE BP NEURAL NETWORK PROXY MODEL

The BP neural network model is mainly composed of the classification contribution priority interactive genetic method

*E-mail of Corresponding Author: 1322051922@qq.com

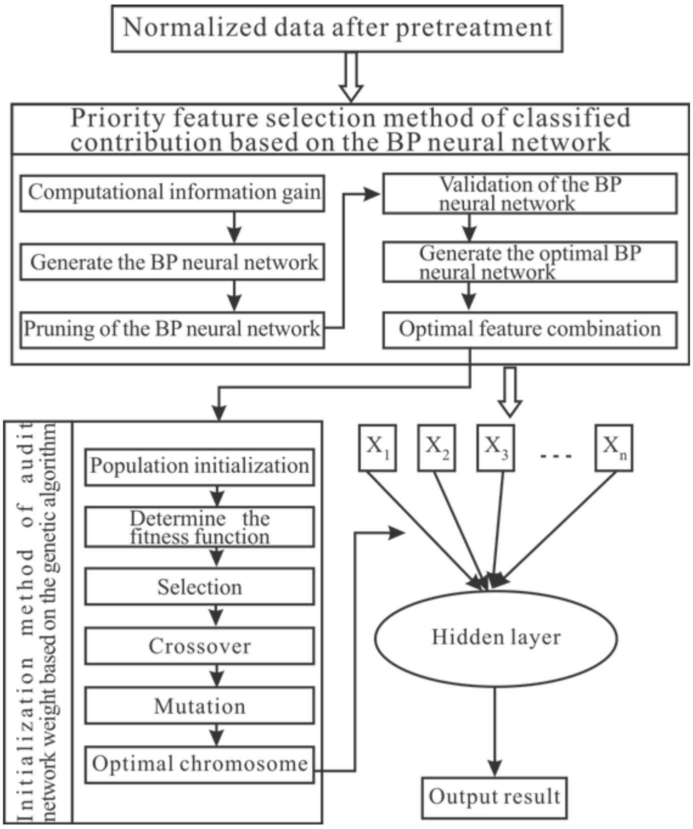
Normalized data after pretreatment
Priority feature selection method of classified contribution based on the BP neural network
Computational information gain
Generate the BP neural network
Pruning of the BP neural network
Validation of the BP neural network
Generate the optimal BP neural network
Optimal feature combination
Initialization method of audit network weight based on the genetic algorithm
Population initialization
Determine the fitness function
Selection
Crossover
Mutation
Optimal chromosome
X1
X2
X3
...
Xn
Hidden layer
Output result

Figure 1 Proxy Model Process Framework.

based on the neural network, the neural network weight initialization method based on the genetic algorithm, and the BP neural network for the x prediction of competence. The specific process framework of the BP neural network is shown in Figure 1.

From Figure 1, it can be seen that the underlying data is preprocessed into the normalized data that can be directly used by the proxy model. The interactive genetic method can be used to extract the optimal feature combination and use it as the input parameters of the BP neural network. Through the improved genetic algorithm, the optimal initial weight can be selected for the optimal feature combination. Finally, the output result can be obtained based on the BP neural network.

3. INTERACTIVE GENETIC ALGORITHM BASED ON THE BP NEURAL NETWORK

Training data usually contains many variables, in which some variables are not correlated with or have very little effect on the competence of the target. When the number of variables is excessively large, it is difficult for the neural network to run appropriately, which can increase the possibility of overfitting. Hence, before the data is input into the BP neural network for training, it is necessary to reduce the number of variables according to the competence of the target, select the appropriate feature variables, and determine the input parameters of the BP neural network.

The interactive genetic method for the classification contribution priority is mainly based on the information gain of features. The basic idea is to calculate the training data set, compare the information gain of each feature, and then select the features with a relatively large information gain.

It is assumed that the training data set is D, $|D|$ stands for the sample capacity of the data set, that is, the number of all samples contained in the data set. Assuming that there are K classes, $C_k, k = 1, 2, \ldots, K$, $|C_k|$ stands for the number of samples contained in the class, $\sum_{k-1}^{k} |C_k| = |D|$. It is assumed that the feature A has n different values $\{a_1, a_2, \ldots, a_n\}$, and the data set D can be divided into n subsets $D_1, D_2, \ldots, D_n$ based on the different values of feature A, in which $\sum_{i-1}^{n} |D_i| = |D|$. If the set of samples in the subset D_i that falls into the category C_k is D_{ik}, that is, $D_{ik} = D_i \cap C_k$, in which $|D_k|$ stands for the number of samples of D_{ik}, the information gain is calculated according to the following method.

Input: Information gain $g(D, A_j)$ of each feature A_j for the training data set D;

Output: Information gain of each feature in the training data set D.

Step 1: Calculate the empirical entropy $H(D)$ of the data set D as the following:

$$H(D) = -\sum_{k-1}^{k} \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{D} \tag{1}$$

Step 2: Traverse the feature set and calculate the conditional entropy $H(D|A_j)$ of each feature in the data set D in turn as the following:

$$H(D|A_j) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{k} \frac{|D_{ik}|}{|D|} \log_2 \frac{D_{ik}}{|D_i|} \tag{2}$$

Step 3: Calculate the information gain $g(D, A_j)$ as the following:

$$g(D, A) = H(D) - H(D|A_j) \tag{3}$$

Step 4: Repeat Step 2 and Step 3 until the calculation of the information gain of all features in the feature set is completed.

The information gains of all the features of the data set obtained by calculation are sorted in descending order, and the features with the highest rank are selected as the optimal feature combination based on the specific application requirements.

4. WEIGHT INITIALIZATION METHOD OF THE NEURAL NETWORK BASED ON THE BP GENETIC ALGORITHM

The standard genetic algorithm and the BP neural network form complementary advantages, which have been successfully applied to the classification and prediction of competence. However, the algorithm still has some defects, primarily a slow convergence, low search efficiency, and that it is prone to falling into closed competition during the early search phase. To make the genetic algorithm meet the requirements of neural network weight initialization better and play a better role in the proxy model, it is necessary to make improvements to the key links of the genetic algorithm.

4.1 Improved Selection Strategy

In the standard genetic algorithm, the selection strategy is mainly based on the fitness value of individuals to determine the probability of being selected and included in the next generation group. Hence, the probability of the individual with a low fitness value to be selected is tiny, which can easily lead to the emergence of super individuals and make the entire algorithm stagnate without moving forward. To ensure that the algorithm can converge as quickly as possible, and that the algorithm converges to the optimal global solution with a probability of 1, the determination criterion for the selection method is improved.

When the next generation group is selected, the individual with the highest fitness value in the current group enters the next generation unconditionally, that is, the current optimal solution can be retained, and the backward fluctuation of the algorithm can be prevented. Individuals i and j are randomly selected in the parent and offspring generations, and the probability of individuals i and j being selected and included in the next generation is as follows, respectively:

$$p(i) = \begin{cases} 1, f(i) \geq f(j) \\ \exp\left(\frac{f(i)-f(j)}{T}\right), \text{ otherwise} \end{cases} \tag{4}$$

$$p(j) = \begin{cases} 0, f(i) \geq f(j) \\ 1 - \exp\left(\frac{f(i)-f(j)}{T}\right), \text{ otherwise} \end{cases} \tag{5}$$

In the above equations, $f(i)$ and $f(j)$ stand for the fitness values of individuals i and j, respectively; T stands for the control parameter. After the selection of each generation, the T value decreases proportionally. With the increase in the number of iterations, the T value decreases continuously. When $T \to 0$, the following can be obtained

$$\exp\left(\frac{f(i) - f(j)}{T}\right) \to 0;$$

At this point, the probability that individuals i and j are selected is turned into the following:

$$p(i) = \begin{cases} 1, & f(i) \geq f(j) \\ 0, & \text{otherwise} \end{cases}$$

$$p(j) = \begin{cases} 0, & f(i) \geq f(j) \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

The improved selection method is shown as the following.

ReSelect ()

```
1. The individual with the highest fitness value is selected and included in the next generation directly.
2. Select $i$ and $j$ in the parent and offspring groups randomly
3. If $f(i) >= f(j)$
4. Individual $i$ is selected and included in the next generation
5. $T = T \times \alpha$
6. Else
7. If
8. Individual $i$ is selected and included in the next generation
9. Else
10. Individual $j$ is selected and included in the next generation
11. $T = T \times \alpha$
12. End
13. End
```

4.2 Interactive Genetic Model

In the standard genetic algorithm, the excellent genes of the superior parents are inherited, which allows the offspring to have a relatively high adaptive value. Given that the parents themselves can increase their adaptive value through self-adjustment, the performance of the offspring can be improved accordingly through genetics. For the learning model, it is necessary to introduce the learning operators, and the following definitions are provided first.

Definition 1 Let x be the bit strong of the group $P(K)$, where $f(x)$ is the fitness value function of the bit strong x, and f is the mean of the fitness value of the group. If $f(x) > f$, x is referred to as an excellent bit strong.

Definition 2 Let x_{im} be the m-th bit of the bit strong x_i. If $x_{im} = x_{jm} \forall i, j \in \{1, 2, \ldots, k\}$ and $i \neq j$, the bit m is referred to as the excellent bit of K excellent bit strings $x_1, x_2, \ldots, x_k$.

Definition 3 It is assumed that $\hat{H}$ is a random mode in the group, $\bar{f}\hat{H}$ is the average adaptive value of the mode H. If $\bar{f}\hat{H} > \bar{f}(\hat{H})$, H is referred to as an excellent mode of the group.

Let x be a bit strong with a relatively low fitness value in the group, p_1 is the learning rate, then the learning operator S can be obtained as the following:

$$S \ (x, H) \rightarrow y.$$
$$y_i = \begin{cases} x_i, \ \text{rand } (0, 1) > p_1; \\ H_i, \ \text{rand } (0, 1) \leq p_1 \end{cases} \tag{7}$$

In the above equation, rand(0,1) stands for a random number between 0 and 1. The bit strong x learning mode H becomes y, and the bit strong y enters the excellent mode H at the probability of p_1. Hence, the performance of the entire algorithm is improved, and the specific steps of the algorithm are shown as the following.

The improved learning mode algorithm is shown as the following.

```
SeStudy ()
1. Initialize parameters $P_c, P_m, P_1, N$;
2. Initialize the population $P(K)$, $K = 0$
3. Calculate the fitness value of the bit strong in $P(k)$
4. Let solution = the maximum fitness value in $P(K)$
5. while (the termination conditions are not met)
6. Search the excellent mode H in $P(K)$
7. f (presence of H)
8. Correct the bit strong with the low fitness value using the
   learning operator
9. Conduct the selection, crossover and mutation operations
   on $P(K)$
10. K = K + 1
11. Calculate the fitness of the bit strong in $P(K)$
12. if (solution < maximum fitness value in $P(K)$)
13. solution = maximum fitness value in $P(K)$
14. return solution.
```

5. CONSTRUCTION METHOD OF BP NEURAL NETWORK FOR COMPETENCE PREDICTION

5.1 Design of the BP Neural Network Structure

1) Determination of the number of hidden layers.

   The BP neural network can contain one or more hidden layers. However, in theory, it has been proven that for most prediction applications, one hidden layer can already meet the functional requirements as any non-linear mapping can be implemented by increasing the number of nodes in the hidden layer.

2) Determination of the number of nodes in the hidden layer.

   The number of nodes included in the hidden layer has a significant effect on the performance of the neural network. If the hidden layer contains a relatively high number of nodes, the performance of the neural network will be relatively good. However, the training time will be increased accordingly, and the convergence speed will be slower. If there are a small number of nodes in the hidden layer, the prediction accuracy of the neural network will be reduced. In general, the number of nodes in the hidden layer can be determined as the following.

   (1) $\sum_{i-0}^{n} C_M > k$, in which k stands for the number of training samples, n stands for the number of nodes in the input layer, and M stands for the number of nodes in the hidden layer.

   (2) $M = \sqrt{n+m} + a$, in which m and n stand for the number of nodes in the input layer and the number of nodes in the output layer, respectively.

   (3) $M = \log_2 n$, in which n stands for the number of nodes in the input layer.

3) Number of neurons in the output layer.

   The number of neurons in the output layer is determined according to the specific issue. In the occupational competence prediction model, for the classification prediction when the target attribute is nominal data, one output node can meet the requirements. For the data fitting issue where the target attribute is a continuous value, it is only necessary to set the corresponding output nodes based on the type of competency and the number as required.

5.2 Improved Learning Algorithm

The adjustment of weights is the core of the neural network learning algorithm. The technology of weight adjustment mainly involves two critical parameters: momentum and learning rate [7]. The so-called momentum refers to the tendency of the weight in each unit to change in a certain direction. Each weight remembers whether it has become larger or smaller, and the momentum manages to keep it going in the same direction. If the value of the momentum is relatively large, the neural network will respond slowly to those training samples that reverse the direction of the weight. If the value of the momentum is relatively small, the neural network allows the weight value to oscillate back and forth more freely. The learning rate controls the speed at which the weights change. In general, the optimal way to increase the learning rate is to start at a relatively high rate and decrease gradually with the progress of the neural network training process. In this way, when the neural network gets closer and closer to the optimal solution, fine-tuning can be used to achieve the optimal weight. The steepest descent method used by the standard BP neural networks has a relatively slow convergence speed, and the convergence speed can be increased by setting the momentum and changing the learning rate.

The change in the learning rate can be determined by the increase or decrease of the error e [8]. When the error approaches the target in a relatively small manner, it suggests that the direction of correction is appropriate, which is an increase in the learning rate. When the error increases beyond a certain range, it suggests that the correction in the previous step was not carried out correctly. At this point, the step size is reduced, and the correction process in the previous step is canceled. The equation for the increase and decrease in the learning rate is as follows:

$$\eta(n+1) = \begin{cases} k_{inc}\eta(n), e(n+1) < e(n); \\ k_{dec}\eta(n), e(n+1) > e(n) \end{cases} \tag{8}$$

Based on the error back propagation, the momentum term is introduced to control the amount of change in the weight, which can obtain better performance than the original algorithm. Hence, some definitions are stipulated for the description of the algorithm.

It is assumed that the training sample data set is $D = \{(x^{(p)}, t^{(p)})\}_{p=1}^{N}$, in which $x^{(p)}$ stands for the input vector, $t^{(p)}$ stands for the output vector, N stands for the number of training samples and the sum of the squared errors is taken as the error function of the network:

$$E = \frac{1}{2}\sum_{p=1}^{k}\sum_{k=1}^{m}\left[t_k^{(p)} - y_k^{(p)}\right]^2 \tag{9}$$

In the above equation:

$y_k^{(p)} = y_k(x^{(p)})$.

From the chain differentiation rule, the following can be obtained:

$$\frac{\partial E}{\partial v_{kj}} = \sum_{p=1}^{N}\left[y_k^{(p)} - t_k^{(p)}\right]\varphi\left(\sum_{i=1}^{d}\omega_{jt}x_i + \omega_{j0}\right)$$

$$\frac{\partial E}{\partial v_{k0}} = \sum_{p=1}^{N}\left[y_k^{(p)} - t_k^{(p)}\right] \tag{10}$$

$$\frac{\partial E}{\partial \omega_{ji}} = \sum_{p=1}^{N}\sum_{k=1}^{m}\left[y_k^{(p)} - t_k^{(p)}\right]\varphi\left(\sum_{i=1}^{d}\omega_{ji}x_i + \omega_{j0}\right)$$
$$\left[1 - \varphi\left(\sum_{i=1}^{d}\omega_{ji}x_i + \omega_{j0}\right)\right]x_i \tag{11}$$

$$\frac{\partial E}{\partial \omega_{ji}} = \sum_{p=1}^{N}\sum_{k=1}^{m}\left[y_k^{(p)} - t_k^{(p)}\right]\varphi\left(\sum_{i=1}^{d}\omega_{ji}x_i + \omega_{j0}\right)$$
$$\left[1 - \varphi\left(\sum_{i=1}^{d}\omega_{jt}x_i + \omega_{j0}\right)\right] \tag{12}$$

It is assumed that η is the learning rate, and α is the momentum constant, then the change of the weight vector can be obtained as the following:

$$\Delta v_{kj} = -\eta\frac{\partial E}{\partial v_{kj}} + \alpha\Delta v_{kj}^{old} \tag{13}$$

$$\Delta v_{k0} = -\eta\frac{\partial E}{\partial v_{k0}} + \alpha\Delta v_{k0}^{old} \tag{14}$$

$$\Delta \omega_{ji} = -\eta\frac{\partial E}{\partial \omega_{ji}} + \alpha\Delta \omega_{ji}^{old} \tag{15}$$

$$\Delta \omega_{j0} = -\eta\frac{\partial E}{\partial \omega_{j0}} + \alpha\Delta \omega_{j0}^{old} \tag{16}$$

The steps of the improved algorithm to increase the momentum and learning rate are as follows.

Step 1: Initialize the weights and biases $v_{kj}, \omega_{ji}, v_{k0}$ and ω_{j0}, in which, $k = 1, 2, \ldots, m,\quad j = 1, 2, \ldots, n$, and $i = 1, 2, \ldots, d$. The learning rate is set as , and the momentum is set as α.

Step 2: For the input vector $x^{(p)}$, the output $y_k^{(p)}$ of the neural network is calculated, and its error is calculated according to the expected target output. If the error is greater than the set threshold, proceed to Step 3; otherwise, proceed to Step 4.

Step 3: If the number of iterations exceeds the upper limit, proceed to Step 4; otherwise, update the weights and biases $v_{kj}^{new} = v_{kj}^{old} + \Delta v_{kj}$, $v_{k0}^{new} = v_{k0}^{old} + \Delta v_{k0}$, $\omega_{kj}^{new} = \omega_{kj}^{old} + \Delta\omega_{kj}$, $\omega_{k0}^{new} = \omega_{k0}^{old} + \Delta\omega_{k0}$, and then go back to Step 2.

Step 4: The training is terminated.

6. EXPERIMENT ANALYSIS

For enterprises and institutions, to improve work and production efficiency, it is necessary not only to adopt a talent-oriented approach but also to make the best use of all available talent. With the continuous improvement of information technology, a huge amount of personnel basic data has been accumulated in various business system databases. How to convert this basic data into capacity data is an issue of great concern in the field of human resource management. The traditional method of acquiring competence data mainly depends on the evaluation of expert groups. In the circumstance of a large amount of basic data and high requirements for real-time competence data, it is impossible to provide real-time, accurate, and reliable competence information in a timely manner, which can influence employers and enterprises. Hence, it is necessary

Table 1 Classification and Prediction Table of Training Samples.

Classification of training samples	Prediction category	
	0	1
0	2786	1214
1	162	3838

Table 2 Classification and Prediction Table of Test Samples.

Classification of training samples	Prediction category	
	0	1
0	269	731
1	173	827

Table 3 Classification and Prediction Table of Test Samples.

Pruning degree	Accuracy rate of the training samples /%	Accuracy rate of the test samples /%
70	87.6	53.7
75	83.9	54.6
80	82.7	63.4
85	81.3	71.6
90	75.4	71.3
95	66.3	69.4

to establish a competence prediction model and transform the basic data into competence data automatically through the self-learning expert experience in a way that can provide support for the decision-making process in enterprises and institutions. The basic information and business performance data of employees of an enterprise are used as the experimental data. The goal of competence prediction is to determine whether the employees can complete the tasks assigned by the company. Due to the requirements of the employer data and information confidentiality, we cannot show the real basic data in the occupational data center in this paper, and the data set used has been declassified to a certain extent where the true feature field names are not used. The data items are normalized data that meet the input requirements of the BP neural network after data pre-processing. The data set includes 10,000 samples, and each sample has 24 attributes, which include the basic elements such as age, height, weight, and training-related competence parameters. The accuracy of each attribute value is retained to 10 decimal places. All the data can be divided into two categories, in which the target attribute of 1 indicates a positive sample of completed tasks, and the target attribute of 0 indicates a negative sample of uncompleted tasks. There are 5,000 positive and negative samples each, in which 80% of the data in each type of sample are used as training data, and 20% are used as the validation data to test the classification accuracy of the model.

The information gain of all fields is calculated, and the attribute with the largest information gain value is selected as the classification node in turn. The C4.5 algorithm is used to establish a neural network. The classification results of the training samples and test samples are shown in Table 1 and Table 2, respectively.

From Tables 1 and 2, it can be calculated that the uncorrected neural network model has a classification accuracy rate of 82.8% for the training samples and a classification accuracy rate of 54.8% for the test samples. From the prediction results, it can be seen that the classification of training samples based on the model has reached a relatively ideal level. However, for the testing samples, the classification error rate is relatively high. The reason is that the over-fitting has weakened the generalization capacity of the model, and the excessive memory training samples lead to the loss of a relatively good classification capacity. The pruning of the neural network is the key technology to avoid noise and improve classification accuracy. However, the degree of neural network pruning can also affect the prediction accuracy. The prediction based on the neural network models using different pruning levels is shown in Table 3.

From Table 3, it can be seen that the prediction accuracy rate of the training samples will continue to decrease with the increase in the degree of pruning. For the test samples, as the degree of pruning continues to increase from 70, the prediction accuracy rate continues to increase and reaches the highest level at about 85, which then starts to decline. Hence, it can be preliminarily determined that the optimal pruning degree is between 85 and 90. The specific pruning degree needs to be further broken down, and the specific situation is shown in Table 4 as the following.

According to the results, when the degree of pruning is 84, the classification accuracy of the test samples is the highest. At

Table 4 Classification and Prediction Table of Test Samples.

Pruning degree	Accuracy rate of the training samples /%	Accuracy rate of the test samples /%
82	82.1	69.7
83	82.1	70.6
84	81.7	72.4
85	81.3	71.6
86	79.4	71.3
87	78.3	71.4
88	76.8	71.1

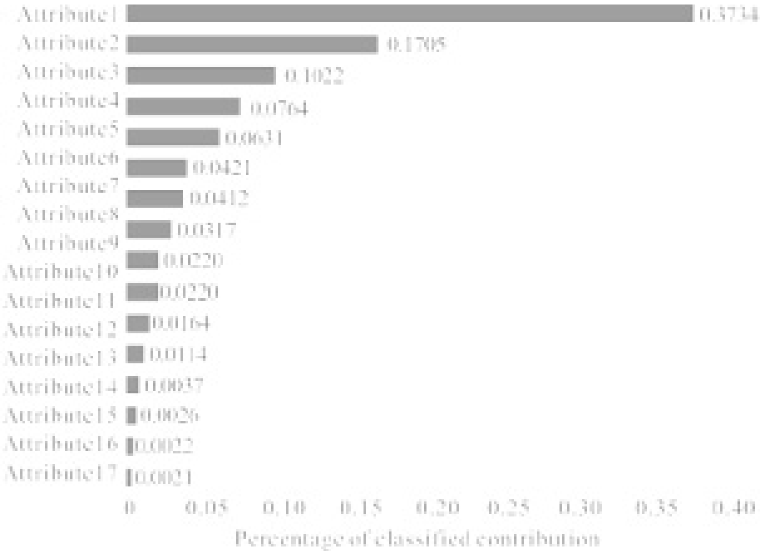

Figure 2 Sorting Diagram of Contribution Degrees of Feature Classification.

this point, the accuracy of the training sample is 81.7%, which is only 1% lower. Hence, the neural network thus generated is the optimal neural network. Based on the information gain ratio, the attributes are sorted according to the percentage of classification contribution, and the results are shown in Figure 2.

Twelve features with a classification contribution greater than 1% are selected as the optimal feature combination, and the attribute numbers are 1, 7, 3, 6, 4, 9, 21, 16, 2, 19, 14, and 8, respectively. From the subsequent proxy model, it can be known that these selected features will be used as the input variables for the neural network.

To verify the usability and performance advantages of the GA-BP neural network proxy model in the prediction of "Combat" capabilities, the basic training data set after treatment is used to verify the model and analyze the prediction results. Two main experiments are conducted as the following: In the first experiment, the prediction performance of the BP neural network model without the interactive genetic method is compared with the optimal feature combination as the model of the neural network input parameters. In the second experiment, the performance of the single BP neural network model and the single GA algorithm is compared with that of the GA-BPBP neural network model, and comparative analysis is carried out.

In the experiment, the data set shown in Figure 2 is used, and Matlab 2011a is applied to perform the programming. The CPU of the program is Intel Pentium Dual E2220, the main frequency is 2.4 GHz, and the memory is 1 GB.

According to the BP neural network algorithm, the optimal feature combination is adopted for the input variables. The number of neural nodes in the hidden layer is set to 9, the learning rate is set to 0.0023, the momentum factor is set to 0.00003, the maximum number of iterations is set to 2000, the error tolerance is set to 0.00001, the ratio of training samples to test samples is set to 8: 2, and the ratio of positive to negative samples is set to 1: 1. Hence, there are a total of 8000 training samples and 2000 test samples. The error curve of the single BP neural network training prediction is shown in Figure 3 as the following.

From Figure 3, it can be seen that after the number of iterations is higher than 400, the change rate of the error curve becomes smaller, and the entire neural network is close to convergence. At this point, the total prediction accuracy rate is 85.80%.

1) Comparative Experiments of the Interactive Genetic Method.

   Based on the experimental results shown in Figure 3, the input parameters of the BP neural network are set as the

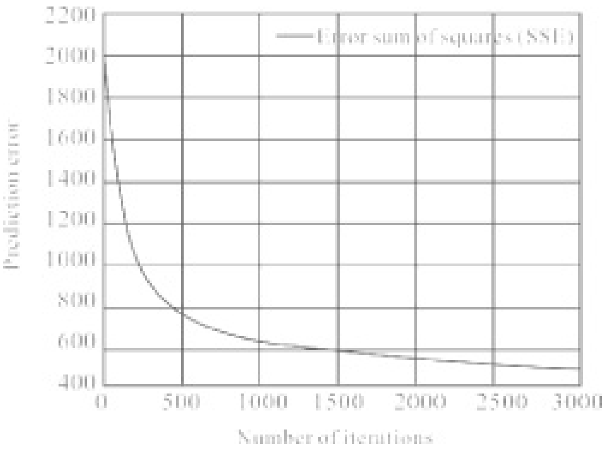

Figure 3 Training Error Curve of the BP Neural Network.

Table 5 Classification and prediction table of training samples.

Number of nodes in the hidden layer	Number of convergence iterations	Prediction accuracy rate /%
3	60	75.3
4	80	86.5
5	68	88.15
6	166	94.05
7	86	92.5
8	94	91.95
9	107	92.05
10	129	91.75

optimal feature combination that includes 12 features, which runs with the single BP neural network model 10 times each. The prediction error rate change curve based on the two models are plotted with the percentage of the unit of measurement, and the diagram is shown in Figure 4 as the following.

After the interactive genetic method is added, the prediction accuracy of the model is improved significantly. From the amplitude of the curve, it can be seen that the interactive genetic method can effectively eliminate the interference of noise and greatly improve the stability of prediction. Compared with the single BP neural network, its performance has been dramatically improved, and its robust performance in the prediction of competence allow the system to run in more stable manner.

2) Experiment of Performance Comparison between the Single Model and Proxy Model

Although the optimal solution can be obtained based on the single genetic algorithm, as it adopts an exhaustive search approach, when the volume of data is huge, the running of the algorithm takes a long time. Its iteration curve is shown in Figure 5 as the following.

Compared with the single BP neural network model, in the case that the same accuracy is achieved, it is necessary to perform the interation more than 1,000 times in the genetic algorithm, where the convergence rate is excessively slow, and the program running time is excessively long. The GA-BPBP neural network model draws on the strong points to make up for the respective weak point of the two models to achieve excellent results in both convergence speed and prediction accuracy. The number of iterations and the prediction errors of the proxy model when the number of nodes in the hidden layer is different are shown in Table 5 as the following.

From Table 5, it can be seen that when the number of nodes in the hidden layer is 6, the prediction accuracy rate of the proxy model is the highest. At this point, the number of convergence iterations is 166, which is much faster than that in the single BP neural network and the single genetic algorithm. The proxy model has a prediction accuracy rate of 94.05%, which is the highest among all the single models.

7. CONCLUSIONS

According to the requirements of enterprises and institutions for the acquisition of competence data, to implement the self-learning of expert experience and intelligent prediction of occupational competence, a proxy model based on the BP neural network is put forward in this paper. In the model, the advantages of the neural network in selecting the optimal features and the global optimization performance of the genetic algorithm are fully leveraged to makes up for the defects of the BP neural network and improve the learning convergence speed and prediction accuracy. The experiment

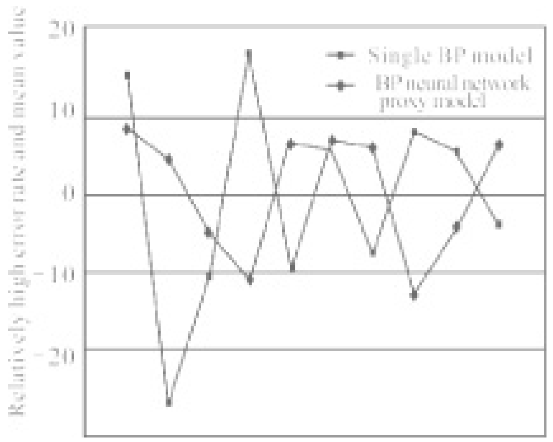

Figure 4 Comparison Diagram of the Error Rate and the Mean Deviation.

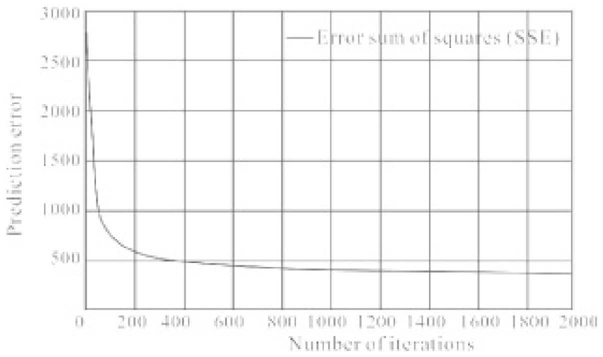

Figure 5 Error Curve of the Genetic Algorithm.

has verified that the proposed proxy model can deliver the function of competence prediction very well and has very excellent stability.

REFERENCES

1. Huang S. Optimization of combustion for pellet shaft furnace based on artificial neural network and genetic algorithm. Journal of Southeast University, 2012, 42(1), 88–93.
2. Ding SF, Su CY, Yu JZ. An optimizing BP neural network algorithm based on genetic algorithm. Artificial Intelligence Review, 2011, 36(2), 153–162.
3. Mohammad AEK, Hong J, Wang DW. Polishing of uneven surfaces using industrial robots based on neural network and genetic algorithm. International Journal of Advanced Manufacturing Technology, 2017, 93(1), 1–9.
4. Belciug S, Gorunescu F. A hybrid neural network/genetic algorithm applied to breast cancer detection and recurrence. Expert Systems, 2012, 30(3), 243–254.
5. Ge D Y, Yao X F, Xiang W J. Application of BP neural network and genetic algorithm in testing of micro-drill's rounded corner. Sensor Letters, 2011, 9(5), 1943–1947.
6. Zhou W H, Xiong S Q. Optimization of BP neural network classifier using genetic algorithm. Energy Procedia, 2011, 11, 578–584.
7. Miao S C, Yang J H, Wang X H. Blade pattern optimization of the hydraulic turbine based onneural network and genetic algorithm. Hangkong Dongli Xuebao/Journal of Aerospace Power, 2015, 30(8), 1918–1925.
8. Jarndal A. Genetic algorithm-based neural-network modeling approach applied to algan/gan devices. International Journal of RF and Microwave Computer-Aided Engineering, 2013, 23(2), 149–156.
9. Han L, Zhang Z Y. The application of immune genetic algorithm in main steam temperature of PID control of BP network. Physics Procedia, 2012, 24(1), 80–86.
10. Soares S, Antunes C H, Araújo R. Comparison of a genetic algorithm and simulated annealing for automatic neural network ensemble development. Neurocomputing, 2013, 121(18), 498–511.
11. Huang K, Liu B C, Peng J G, Feng D S, & Wang Y F. Intelligent back-analysis of tunnel surrounding rock displacement based on genetic algorithm and neural network. Journal of Central South University, 2011, 42(1), 213–219.
12. Ren C X, Wang C B, Yin C C. The prediction of short-term traffic flow based on the niche genetic algorithm and bp neural network. Lecture Notes in Electrical Engineering, 2013, 211, 775–781.
13. Zhang Y J, Xu J R, Fu X H. Hybrid algorithm combining genetic algorithm with back propagation neural network for extracting the characteristics of multi-peak Brillouin scattering spectrum. Frontiers of Optoelectronics, 2017, 10(1), 62–69.
14. Dou R, Zong C, Nan G. Multi-stage interactive genetic algorithm for collaborative product customization. Knowledge-Based Systems, 2016, 92, 43-54.
15. Yoon J, Lee S, Lee I, et al. Optimized image resizing using flow-guided seam carving and an interactive genetic algorithm. Multimedia Tools and Applications, 2014, 71(3), 1013–1031.