The Modeling of Genetic and Tabu Search Algorithm Based BP Neural Network in the Risk Analysis of Investment

Liu Yang, Ding Yunlong
School of Management
Harbin Institute of Technology, China
ly19800219@163.com, dingyunlong@hit.edu.cn

Journal of Digital
Information Management

ABSTRACT: *There are many parameters which affect the economic benefit indicators for evaluation and they are all uncertain. Because of the parameters' uncertainty, evaluation indicators values of economic benefit in project investment are also uncertain. In these years, more and more BP Neural network based models are established to evaluate the risk of investment for its characters of wide information distribution and strong fault tolerance. However, the internal defect of local minima problem in this model will greatly discount the effects of evaluation if the training data are not proposed selected. To solve this problem of traditional BP neural network, an optimization algorithm that combines the advantages of genetic algorithm (GA) and Tabu Search (TS) is proposed. The training process is divided into two phases in order to search promising parameters of the BP neural network in favor of better evaluation and analysis of the investment. In the first phase, the initial parameters are first searched by GA algorithm by taking advantage of its various searches of the solutions, and the best parameters are selected by Tabu search algorithm in the second phases. After calculation of practical data, it proves that the new algorithm has better convergence rate and predictive accuracy, which make the evaluation and risk analysis of the investment faster and more accurate.*

Subject Categories and Descriptors:
I.2.10 [BP Neural Network]: Risk Analysis; **I.4.10 [Risk Analysis of Investment]**

General Terms:
BP Neural Network, Risk Analysis of Investment

Keywords: BP Neural Network, Genetic Algorithm, Tabu Search Algorithm, Risk Analysis, Investment Evaluation

Received: 1 July 2013, Revised 12 August 2013, Accepted 18 August 2013

1. Introduction

With the development of society, investment becomes one of the main promotions of the economy. Along with the investment's prosperity is the extremely high risk. So how to control and evaluate investment risk has become one of the main topics for economic. Plenty of research about evaluation theories and applications has been carried out in decades, and many different models are established to evaluate the risk of investment. For example, integrate the neural network [1] and the grey system theory [2] to evaluate the investment risk.

There are many parameters which affect the economic benefit indicators for evaluation, what's worse, the parameters are uncertain for some point of view, which make the indexes to evaluate the investment uncertain. So it is unavoidable that there are some risks in the conclusion of investment evaluation. Quantitative analysis and a reasonable estimate of risk to the investment is the essential work in the economic evaluation. Before analysis the investment risk, responsible math model should be established, and we can get the optimal results to achieve the best solution. S. Ye [3] et al proposed the net-present-value-at-risk (NPV-at-risk) method which combines the weighted average cost of capital and dual risk return methods to evaluate investment risk in infrastructure project. Becerra-Fernandez [4] et al presents the insights gained from applying knowledge discovery in databases (KDD) processes for the purpose of developing intelligent models to classify a country's investing risk. The research of Benaroch, M [5] proves that if the operational traits of the investment are properly selected, the risk of investment can be predicted accurately. As the time goes, the environment of investment becomes more and more complicate, it is almost impossible for the researchers to consider all the factors that affect the risk of investment. All the solutions mentioned above get the optimal results by using ether analytic hierarchy process [6] or fuzzy theory [7] to establish nonlinear model. However, if the parameters which affect the investment, the resultsmake us to rebuild the model and redo the whole analysis procedure, which is really a big waste of time and resources. The development of neural network has provided the investment of risk new tools. Back propagation (BP) neural network is a multi-level feedback neural network. The BP neural network based evaluation method has the characters of fast speed, high efficiency, self-learning ability and wide adaption, which make it a better method to simulate the evaluation process of evaluation experts. And a lot a work has been done in the area. Zhu W.X. [8] introduces BP network into risk assessment of performance for expressway management corporations. Wu Y.N [9] et al proved that RBF neural network based on ant colony algorithm [10] has good performance in credit risk evaluation of construction enterprises.

Even though the BP neural network solution has many advantages, there are some unavoidable defects, General speaking, there are two main problems. The first one is the slow convergence rate. BP algorithm is the steepest descent method, the training period is difficult to grasp. And the train speed of the algorithm is very slow, what's more, the training stop is hard to define, if the step is too long, the accuracy of the algorithm cannot achieved, sometimes the algorithm will diverge if the training data is not proper selected. If the step is too small, the number of iterations will explode to slow the convergence rate. When the output is going to saturate, there will be some flat area on the error surface, the change of the weight has little effects to the variation of deviation. That means the time deviation will not descend no matter how long you have trained. The second defect of BP neural network is the local minima problems. As to the complicate neural network, the deviation surface is uneven; there are many local minima points on the surface. When search for the optimal results by using BP algorithm, the possibility of trapping into the local minima is very big. There are some other defects as well, for example, the stability of training problem. When approach the training upper limit, the classification ability of the algorithm will descend.

To overcome the defects of the BP neural network, in this paper, we introduce the genetic algorithm and the Tabu search algorithm into the BP neural network. In order to search a promising initial solution in favor of locating the best global solution, the training period is divided into two phases, first search a promising initial solution by using genetic algorithm, and then select the best solution from the initial solutions by using the Tabu search algorithm. The rest of the paper is organized as follows. In section 2, we will introduce the details about the optimization model of BP neural network. And in section 3, an experiment is done to test the advantage of the model. At last, we make a conclusion and finish the whole paper.

2. The optimization model of BP neural network

2.1 The Structure of BP Neural Network

BP neural network is a one-way propagation multi-layer feed-forward network, which adopts backward propagation algorithm. There are three layers in the BP neural network: input layer, hidden layer and output layer. The neural network described in figure 1 is the most commonly used network in the applications.

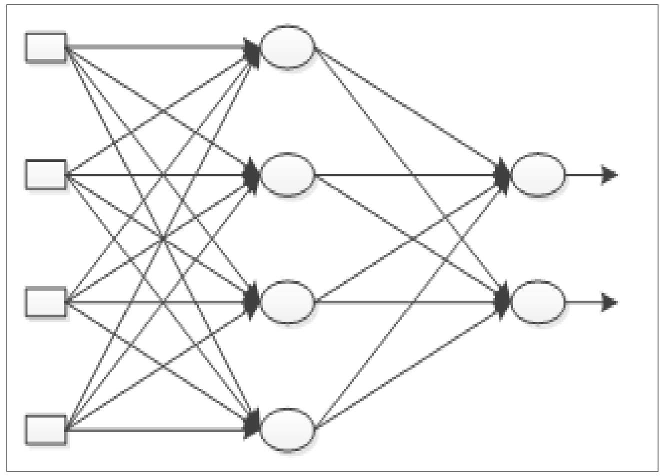

Figure 1. Three layers BP neural network

The learning process of the algorithm can divided into two phases. The first phase is the procedure of forward propagation, calculate the output value on the nodes from input layer to hidden layer, the node on the layer only accept the input from the previous layer, and the results are useful to the next layer. The second phase is the backward propagation process. If the expected results are not achieved on the output layer, then calculate the deviation between the expected results and the actual output values recursively layer by layer. After that, adjust the weight of previous layer to minimize the deviation according to the deviation values generated in the previous step. Adjust the weight of network and the threshold on the direction of decrease of the deviation function slope to approach the objective function gradually. Each adjust of the network weight are proportional to the deviation of the network.

In the follow part of the section, we will discuss some details about the three layers BP neural network.

Suppose the input-output pair of the BP network is (X_p, T_p), $p = 1, 2, ..., P$, where P is the number of the training sample, X_p is the input vector of $p-th$ sample, $X_p = (x_{p1}, ... x_{pM})$, M is the dimension of the input vector; T_p is the output vector of sample, $T_p = (T_{p1}, ... T_{pN})$, N is the dimension of the output vector. Neural network has only a single hidden layer, suppose the number of nodes in hidden layer is H. We can use w_{ij} as the weight of the network between the input layer and hidden layer as well as between the hidden layer and output layer. w_{ij} is the connection weight between the node i - th in the previous layer and j - th node in this layer. The propagation functions of the hidden layer and output layer in the neural network are sigmoid functions [11].

$$f(x) = 1 / (1 + e^{-x}) \qquad (1)$$

The deviation function is

$$E = \frac{1}{2}\sum_{k=1}^{N}(t_k - o_k)^2 \qquad (2)$$

The implementation of BP neural network has the following steps.

The output of the hidden layer is

$$y_j = f(net_j) = f(\sum_{i=1}^{M} \omega_{ij} x_i) \qquad (3)$$

x_i is the input of the $i-th$ node in the input layer, y_j is the output of the $j-th$ node in the hidden layer. The node of o_k in the output layer is

$$o_k = f(net_k) = f(\sum_{j=1}^{H} \omega_{jk} y_j) = f(\sum_{j=1}^{H} \omega_{jk}\, f(\sum_{i=1}^{M} \omega_{ij} x_i)) \qquad (4)$$

We can define the descent gradient of deviation as

$$\delta_j = -\frac{\partial E}{\partial net_j} = -\frac{\partial E}{\partial o_j}\frac{\partial E}{\partial net_j} = -\frac{\frac{1}{2}\sum_k (t_k - o_k)^2}{\partial o_j} f'(net_j)$$
$$= (t_j - o_j)^2 f'(net_j) \qquad (5)$$

The increase of output layer and hidden layer's weight is proportional with the descent of the gradient. The adjust function of the weight is

$$w_{ji}(t+1) - w_{ji}(t) = \eta \delta_i o_j \qquad (6)$$

η is the learning rate, which depends the quality of weight update.

2.2 Hidden layer and the performance of the neural network

General speaking, the more layers of the neural network has, the more accuracy of the classification. However, in fact, too much layers make the network not only too much complicate but also lower the accurate of the classification. It is proved that the three-layer BP neural network can achieve nonlinear continuous function approximation at any precision. Theoretically the deviation can be lower if appropriate number of layers is chosen. The accurate of the BP neural network can also be improved by selecting appropriate number of hidden layer nodes, what's more, the training results of this kind is easier to adjust. So the optimization of the neural network is the optimization of nodes in the hidden layer of the network, from some point of view.

The number of nodes in hidden layer is one of the main factors that affect the performance of the BP neural network. The less of the hidden nodes, the less information gained from the network, the lower of the accuracy, slower convergence rate of the network. So, there should be more hidden nodes in the hidden layer to pursue higher accuracy. But, too much nodes in the hidden layer make the topology of the neural network too complex, and the learning time can be very long, which make the network overtraining, and the worsen the error tolerance. And the relationship between the deviation and the number of the nodes in hidden layer is described in the following figure.

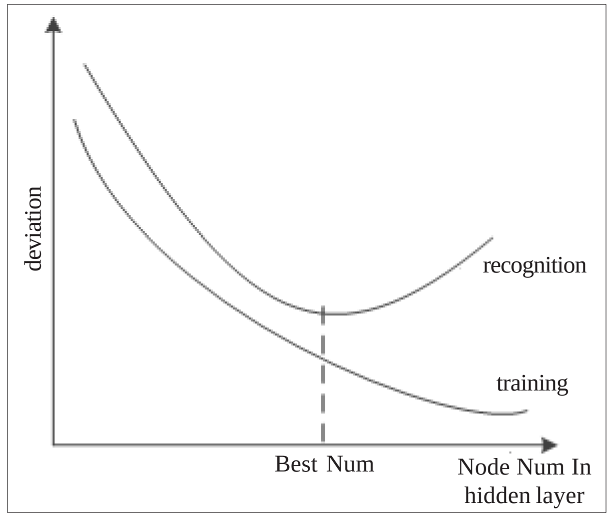

Figure 2. Relationship between the hidden nodes number and the deviation

2.3 Genetic and Tabu Search Algorithm Based BP Neural network

The main idea of the optimized algorithm is to adjust the weight of the connection in the neural network to achieve the demand training accuracy after the structure of BP the neural network is settled. The training process is divided into two phases in order to search promising parameters of the BP neural network in favor of better evaluation and analysis of the investment. In the first phase, the initial parameters are first searched by GA algorithm by taking advantage of its various searches of the solutions, and the best parameters are selected by Tabu search algorithm in the second phases.

To improve the training rate, we will train the weight of the BP neural network and the structure of the network separately. In the optimization of the weight of the BP neural network, we combine the genetic algorithm, Tabu forbidden search algorithm and BP neural network in this paper.

2.3.1 The fitness of Genetic Algorithm

In this paper, we will redefine the fitness of genetic algorithm and population fitness.

If population NET includes N individual (neural network), and net_i to represent the $i-th$ neural network of the population, then the fitness f_i of the $i-th$ individual and the fitness f_{sum} of the population can be defined as

$$\begin{cases} f_i = E_i \\ f_{sum} = \sum_{i=1}^{N} E_i \end{cases} \qquad (7)$$

E_i is the fitting deviation of the $i-th$ individual (neural network) of the population.

2.3.2 Operations of Genetic Algorithm

As to every net_i in the population, we calculate the fitting deviation E_i by using the test samples. And then the deviations are sorted in ascending order. And handle the individuals with the operation of selection, crossover and mutation. The individuals of the new generation are kept in the population tmp_NET. Then select the 30% of the best individuals of the new population to replace the 30% of the worst individuals of the last generation, and 70% percent of the individuals are kept into the next generation.

2.3.2.1 Selection

The individual selection of the population in the algorithm is based on roulette method [12]. The probability of the individual selection is

$$p_i = \frac{f_i}{\sum_{i=1}^{N} f_i} = \frac{f_i}{f_{sum}} \tag{8}$$

To select the individuals which are used in the crossover step, the selecting procedure goes as following.

1) Calculate p_i and f_{sum} by using formula (7) and formula (8);

2) Generate a random number p $(0 \le p < 1)$, if $p < p_s$ (p_s is the preset selection probability), then go to 3), else, finish the procedure;

3) Generate a random number r $(0 \le r < 1)$, find the minimize number k

$$\sum_{i=1}^{k} f_i \ge r \times f_{sum} \tag{9}$$

4) Select the $k-th$ individual (neural network), and then repeat step 2) until N individuals are selected.

2.3.2.2 Crossover

The procedure of crossover operation goes as following.

1) Generate a random number p $(0 \le p < 1)$, if $p > p_c$ (p_c is the preset crossover probability), then go to 2), else, finish the procedure;

2) Select the individual x_i and individual x_j by using the selection operation, then get the next generation individuals x_m and x_n by using multi-point crossover method [13] according to formula (10)

$$\begin{aligned} x_m &= r \times x_i + (1 - r) \times x_j \\ x_n &= r \times x_j + (1 - r) \times x_i \end{aligned} \tag{10}$$

3) Join the new individuals generated into the new population tmp_NET. Repeat step 2) until N individuals are generated.

2.3.2.3 Mutation

The mutation procedure goes as following.

1) Generate a random number p $(0 \le p < 1)$, if $p < p_m$ p_c (p_m is the preset mutation probability), then go to 2), else, finish the procedure;

2) Generate a random number k for every connection, if $k > r$ (r is a constant number, and $0 < r < 1$), the add the weight of the connection with a random number x $(-10 < x < 10)$.

2.3.3 Operations of Tabu Search Algorithm

Mark the local optima which is searched during the search procedure, and avoid re-searching the marked ones, the process is called Tabu search. The notion of neighborhood, Tabu list [14], aspiration criterion [15] and breaking level are involved in the Tabu search algorithm.

2.3.3.1 Neighborhood of candidate solutions

Suppose the initial solution x is given, some of the solutions in the neighborhood of x are called neighborhood candidate. And every candidate is a neural network, the variable of the solution is the weight of the connection and the nodes number in the hidden layer.

2.3.3.2 Tabu list

The Tabu list is used to record the local optima that searched recently, which is forbidden to access in the following recursive search. The length of the Tabu list is set when the Tabu list is created. The length of the Tabu list is set to T at the initial phase.

2.3.3.3 Breaking level function $A(x, s)$

The breaking level function $A(x, s)$ is the fitness value of the optima $A(x, s) = f(x_b)$. If the fitness value of some Tabu candidate is better than the optima, that's, $f(x_{new}) < A(x, s)$ take Tabu candidate as the current solution.

In fact, we can remove the solutions which are near the optima out of the Tabu list. If the current solution x_{new} satisfies the formula (11), we can take it as the current solution.

$$|f(x_{new}) - f(x_{best})| < TC \tag{11}$$

$f(x_{new})$ is the current solution, and $f(x_{best})$ is the corresponding fitness function of the optima, that's the fitting deviation of the neural network. TC is the preset scope variation.

2.3.3.4 Tabu search

To keep the constancy of the training, the candidate should be around the current solution net. The candidate solutions are usually generated by genetic algorithm, the mutation operation to solution net generates N neural networks. If the optima is better than net, then take it as the current solution, else, redo the genetic operation to get the solution.

The procedure of Tabu search algorithm goes as following.

1) Initiate the parameter TS, current solution x_{new} and Tabu list;

2) Check if the terminating condition is satisfied, if yes, terminate the procedure, else go to step 3);

3) Generate and select some neighborhood solution;

4) Check whether the breaking level is satisfied, if yes, replace the current solution x_{new} and the first member of the Tabu list with the optima x_b;

5) Check the Tabu attributes of the candidate, select the

solution which is not in the Tabu list as the current optima, and replace the first member in the Tabu list, then go to step 2);

6) Output the searched optima.

2.4 Tabu Search and Genetic Algorithm based weight optimization in neural network

In order to accelerate the evolution rate and prevent the premature of the genetic algorithm, we integrate the Tabu search algorithm into the genetic algorithm. It has proved that Tabu search algorithm has faster training rate the genetic algorithm. But Tabu search depends on too much about the initial solution, which is the main defect of this algorithm. What's more, the Tabu search algorithm can find only one solution at a time. If combine the two algorithms, we can get better search results to enhance the performance of the training algorithm. The procedure of the algorithm can be described in figure 3.

There are two parts in the optimization algorithm. Combine the genetic algorithm and the Tabu search algorithm; optimize the weight of the neural network. And then search the solution globally according to the genetic algorithm, which provides a searching center for accurate locating the optima in the next step.

The kernel part of the algorithm is the genetic algorithm ion, which prevents the defect of and Tabu search. Genetic algorithm provides various searches for the solut falling into local minimum point because of gradient descent. At the same time, jump out of the local minima by using the mutation operation. Tabu search can avoid the repeating search about the local minima. The samples used in the algorithm is normalized in the preprocess procedure. Then calculate the maximum and minimum values by using empirical formula, which determine the optimal number of nodes in the hidden layer. In the next phase, the optimal solution of the neural network can be achieved by using genetic algorithm and Tuba search algorithm.

This algorithm divides the training algorithm into several phases, the algorithm is as follows.

Algorithm: Genetic and Tabu Search based Optimization Algorithm

Input: Sample set, population scale, p_s, p_c, p_m

Output: Optimized neural network (structure and weight)

Steps:

1) Preprocess the data in the sample set;

2) Calculate the scale of the node number in the hidden layer by using empirical formula;

3) Select a node number in the hidden layer, initialize the neural network, and find the initial value of the neural

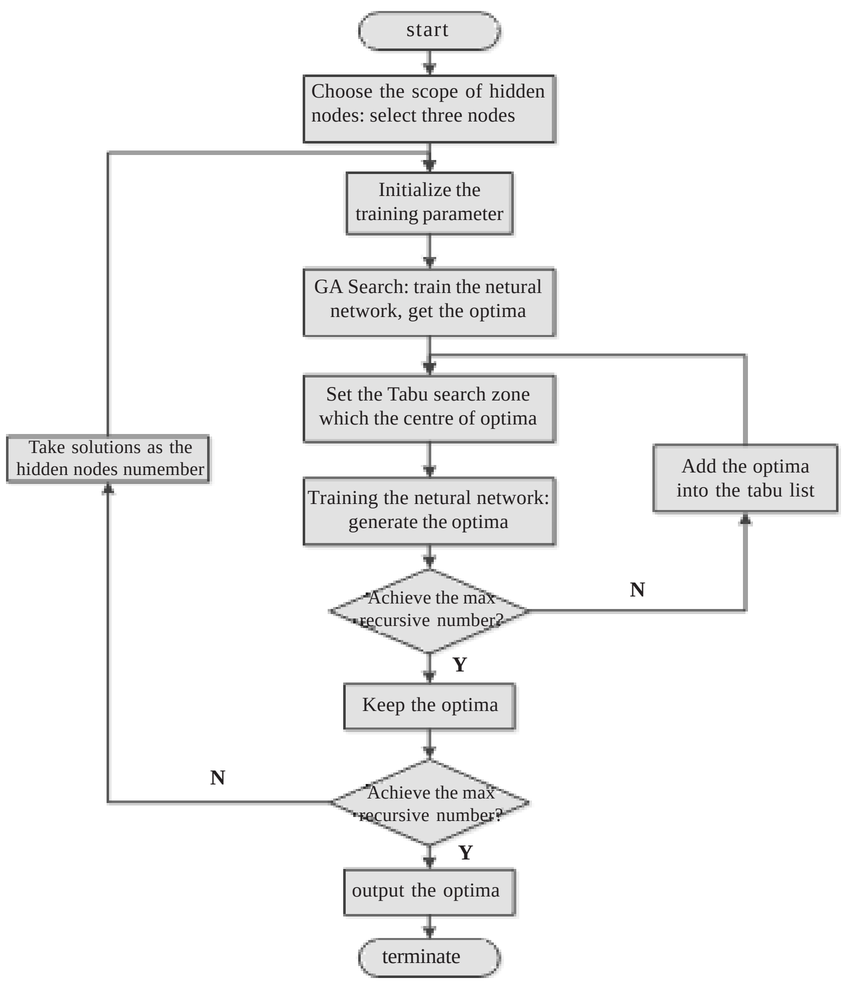

Figure 3. The structure of optimization algorithm

network by using genetic algorithm;

4) Optimize the weight of the network;

5) Check whether the deviation is under the pre-defined value, and whether the number of the node is the minima. If any of the two is satisfied, go to step 7), else go to step 6);

6) Execute the Tabu operation, go to step 4);

7) Save the node number of the hidden layer of the neural network, the initial value, the minima deviation, and the training times;

8) Set a node number of the hidden layer, if there are solutions, go to step 3), else go to step 9);

9) Select the neural network which achieves the optima, finish.

3. Risk Analysis of Investment by using Genetic and Tabu Search Algorithm based BP Neural Network

3.1 samples and quantitation

We collect all the information about 10 accomplished projects, all of which are similar in investment, the indexes that constrains the investment are: 1 investment, 2 consumption, 3 income, 4 cement, and 5 steel.

Because the integration evaluation of the project is a multi-target decision, one of the main characters is the uncommon of different target. It is difficult to compare and analysis the original value of the targets. Before processing with our algorithm, we should normalize the indexes of different projects, which is also a demand about our algorithm. Now suppose:

a) There are m decision schemes A_i $(1 \le i \le m)$ in the integrate evaluation of the project;

b) There are n evaluation indexes f_i $(1 \le j \le n)$;

c) A matrix $X = (x_{ij})_{m \times n}$ which includes m decision schemes and n evaluation indexes, x_{ij} is the $i-th$ decision scheme, $j-th$ evaluation index;

d) The normalized index matrix is $R = (R_{ij})_{m \times n}$

Normalize the indexes with linear scaling method.

a) Efficiency index is

$$x_j^* = \max_{1 \le i \le m} x_{ij} (1 \le i \le m, 1 \le j \le n) \quad (12)$$

Then we can define

$$R_{ij} = \frac{x_{ij}}{x_j^*} \quad (13)$$

b) The cost index is

$$x_j^\bullet = \min_{1 \le i \le m} x_{ij} (1 \le i \le m, 1 \le j \le n) \quad (14)$$

Then we can define

$$R_{ij} = \frac{x_j^\bullet}{x_{ij}} \quad (15)$$

However, different indexes of the project have different importance in the evaluation, so the weight is needed to adjust the importance of different indexes. In this paper, we calculate the weight of different indexes with the entropy value method [16].

Consider the multi-index evaluation with m decision schemes and n indexes.

$$P_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}} (1 \le i \le m, 1 \le j \le n) \quad (16)$$

According to the Information Theory [17], the entropy of index f_i is

$$E_j = -k \sum_{i=1}^{m} P_{ij} \ln (P_{ij}) (1 \le j \le n) \quad (17)$$

k is the constant, $k = (\ln m)^{-1}$.Because $0 \le P_{ij} \le 1$, then $0 \le \sum_{i=1}^{m} P_{ij} \ln (P_{ij}) \le \ln m$, $0 \le E_{ij} \le 1$ $(1 \le j \le n)$, the initial deviation of the index is

$$d_j = 1 - E_j \quad (18)$$

Then the initial weight of index can be defined as

$$\bar{\omega}_j = \frac{\lambda_j \omega_j}{\sum_{j=1}^{n} \lambda_j \omega_j} \quad (19)$$

λ_j is personal bias of different index.

The detail risk indexes of the quantitative results of the 10 projects collected in this paper show in table 1.

3.2 BP neural network construction

Before the evaluation of the project using BP neural network, we should construct the BP neural network. First, take the five indexes of the project as the input nodes of the BP neural network. Second, use the algorithm proposed in this paper to calculate the nodes in the hidden layer. To make comparison, we use both the traditional BP neural network and the Genetic and Tabu Search Algorithm based BP neural network to evaluate the effects of the predictions.

As to the data in table 1, the first 7 samples are used to train the BP neural network, and the last 3 samples are used to test. There are five indexes in the project, so the nodes number in input layer is 5. The nodes number in of the hidden layer can be calculated by using the method explained in section 2.2. The output of the evaluation is the integrated evaluation of the projects. So the nodes number of input layer, hidden layer and output layer of the neural network are 5, 11, and 1. And the learning rate is 0.1. The test results are shown in table 2.

From the above table, we can find that the algorithm in our paper has faster convergence rate and more accurate. Now consider the deviations of the two algorithms, which

Project	Weight of index					
	1	2	3	4	5	total
	0.342	0.151	0.107	0.204	0.196	
1	0.289	0.123	0.114	0.184	0.188	0.266
2	0.332	0.125	0.110	0.225	0.232	0.452
3	0.346	0.210	0.093	0.264	0.223	0.236
4	0.360	0.136	0.076	0.114	0.167	0.354
5	0.221	0.144	0.123	0.145	0.186	0.228
6	0.236	0.164	0.122	0.155	0.199	0.543
7	0.311	0.150	0.113	0.246	0.187	0.452
8	0.324	0.156	0.142	0.223	0.204	0.412
9	0.379	0.147	0.102	0.213	0.221	0.523
10	0.410	0.156	0.098	0.225	0.219	0.446

Table 1. Evaluation indexes of the investment

Cycle	Enhanced network		BP neural Network	
	training	recognition	training	recognition
4096	0.000322	0.001532	0.00898	0.012467
32768	0.000045	0.000311	0.000982	0.001035
131072	0.000023	0.000122	0.000336	0.000345
1048576	0.000002	0.000008	0.000103	0.000092
5000000	0.000001	0.000002	0.000002	0.000001

Table 2. The training accuracy of the samples with different initial values

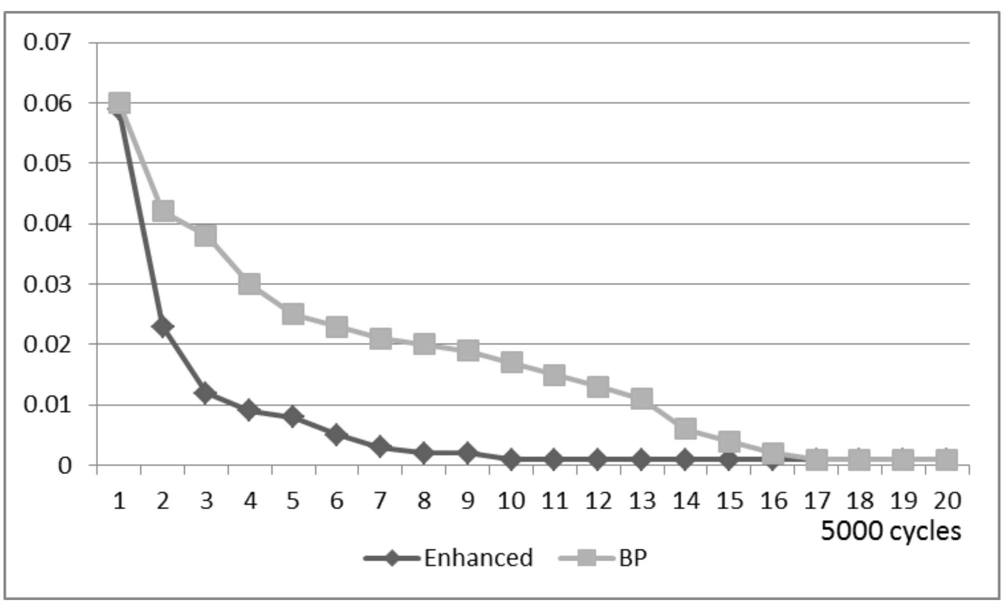

Figure 4. The deviation of different algorithm

is show in figure 4. In this experiment, we take the 3rd sample as an example.

From the above the table, we can find that the Genetic and Tabu search algorithm Based BP neural network has better performance in prediction the result. In fact, the total result prediction is very important, which can help

Project	Actual result	Result by enhanced	Result by traditional
8	0.412	0.404	0.332
9	0.523	0.532	0.501
10	0.446	0.453	0.425

Table 3. The test results

us to find the risk of the project, which can help the decision maker to evaluate the risk of the investment. Also, the model can also help the project manager to find which one of the factors has more importance to the whole project, which he can take more care about the aspect.

Now we will check the accuracy in the test. From the above test, we can set the accept deviation as 0.00002, and the recursive cycle as 100000, and we can get the following results.

4. Conclusion

The neural network is trained by the history data, and the performance of the risk of the investment depends on the quantity and quality of historical data by using BP neural network, the deviation of the predication will be smaller if the historical data is more. And the fault tolerance of the BP neural network will help to automatic correct the fault in the data.

We designed and implemented a genetic and Tabu search algorithm based BP neural network to simulate the project. And we take the different factors as the input of the BP neural network, and nodes number in the hidden layer decide the accuracy of the network, in this paper, we adopt genetic algorithm to find the suitable number in the hidden layer, to avoid re-search the nodes which has been visited, the Tabu search list is used to forbid the access to these node, the experiment proves that the algorithm in our paper is much better in the convergence rate, predict accuracy and deviation convergence rate.

5. Acknowledgment

The study described in this paper was supported by the National Nature Science Foundation of China (No. 71073038).

References

[1] McCulloch, Warren, Walter Pitts. (1990). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5 (4) 115–133.

[2] Deng Julong. (1989). Introduction to Grey System Theory. The Journal of Grey System. p. 1-24.

[3] Ye, S., Tiong, R. L. K. (2000). NPV-at-Risk Method in Infrastructure Project Investment Evaluation. *J. Constr. Eng. Manage*.126 (3) 227–233.

[4] Irma Becerra-Fernandeza, Stelios H. Zanakisa, Steven Walczakb. (2002). Knowledge discovery techniques for predicting country investment risk. *Computers & Industrial Engineering*. 43 (4) 787-800.

[5] Benaroch, M. (2001). Option-based management of technology investment risk. *Engineering Management.* 48 (4) 428-444.

[6] Saaty, Thomas, L. (2008). Relative Measurement and its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors – The Analytic Hierarchy/Network Process. *Review of the Royal Spanish Academy of Sciences, Series A, Mathematics*, 102 (2) 251–318.

[7] Francis Jeffry Pelletier. (2000). Review of Met mathematics of fuzzy logics in The Bulletin of Symbolic Logic, 6 (3) 342-346.

[8] Wenxi, Zhu . (2007). Applying an Improved BP Network to Risk Assessment of Performance for Expressway Management Corporations. Wireless Communications, Networking and Mobile Computing, WiCom 2007. *International Conference*. p. 4618- 4621.

[9] Yunna, Wu (2008). Application of RBF Neural Network Based on Ant Colony Algorithm in Credit Risk Evaluation of Construction Enterprises. Risk Management & Engineering Management, ICRMEM '08. *International Conference*.11, p. 653-658.

[10] Dorigo, M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. Evolutionary Computation, *IEEE Transactions on*. 1 (1) 53-66.

[11] Han, Jun, Morag, Claudio. (1995). The influence of the sigmoid function parameters on the speed of back propagation learning. In Mira, José; Sandoval, Francisco. *From Natural to Artificial Neural Computation*. p. 195–201.

[12] Billingsley, Patrick. Probability and Measure (2nd ed.). *John Wiley & Sons Inc*. 94.

[13] Eshelman, Larry J.. (1990). The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination, in Gregory J. E. Rawlins editor, *In*: Proceedings of the First Workshop on Foundations of Genetic Algorithms. p. 265-283.

[14] Glover, Fred . (1989). Tabu Search - Part 1. *ORSA Journal on Computing*, 1 (2) 190–206.

[15] Malek, M., Huruswamy, M., Owens, H., Pandya, M. (1989). Serial and parallel search techniques for the traveling salesman problem. Annals of OR: Linkages with Artificial Intelligence. 21(1) 59-84.

[16] Carnot, S. (1986). Reflexions on the Motive Power of Fire, translated and annotated by R. Fox, Manchester University Press, 26.

[17] Rando Allikmets, Wyeth, W., Wasserman, Amy Hutchinson, Philip Smallwood, Jeremy Nathans, Peter K. Rogan, Thomas D. Schneider, Michael Dean. (1998). Organization of the ABCR gene: analysis of promoter and splice junction sequences, *Gene* 215, 1, p. 111-122.