

Predicting Software Quality by Optimized BP Network Based on PSO

Kewen Li^{1,2}

¹School of Management, Tianjin University, Tianjin, China

²College of Computer and Communication Engineering,

China University of Petroleum, Dongying, China

Email: likw@upc.edu.cn

Jisong Kou¹, Lina Gong²

¹School of Management, Tianjin University, Tianjin, China

²College of Computer and Communication Engineering,

China University of Petroleum, Dongying, China

Email: gln88888@163.com

Abstract—The prediction model of software quality is the key technology in the software quality evaluation system, which can be used to evaluate software quality characteristics that users care about. Prediction models are often used to find the nonlinear relationship between metric data and quality factors. The paper predicted the relationship between metric data and quality factors with historical data by using the optimized BP network based on PSO. According to the algorithm, 28 groups of data are adopted in the experiment, and compared with the results by using BP network. Experiments show that the algorithm has a better performance than the BP network algorithm and perfectly solve the problem of slow convergence and easily getting into local minimum.

Index Terms—software quality, prediction model, software metrics, neural network, PSO

I. INTRODUCTION

Software quality is the natural attribute of software, which is determined when software is in the process of design and development. In the ISO/IEC9126 which is the international standard quality characteristics in the international standard organization (ISO), the software quality is defined as: the software quality is the characteristics which are the ability reflecting the software products to satisfy the prescribed requirements and potential requirements and the sum of characteristics^[1]. Software quality is the comprehensive embodiment of quality characteristics of the final product and software development process, which includes internal properties and external properties. The internal properties of software are the properties that can be measured according to the software itself, for example, the size of software, coupling and cohesion. The external properties of software are the ones that can not be measured depended on software itself, which is connected with some other entities in the environment, for example,

functionality, reliability, usability, efficiency, maintainability and portability. So, predicting and controlling the quality of software are usually to predict the external properties of software based on measuring the internal properties of software, which builds the reasonable relationship between the internal properties of software and the external properties of software.

Development and maintenance management of the software project need receiving the quality knowledge in order to obtain the essential quality rank early. The quality prediction model is the key technology in the software quality evaluation system, by which the objective and quantitative relationship is established between the software internal properties (metrics) and external properties (quality indicators). It can avoid the interference of subjective factors, and make the evaluation result be in keeping with the actual quality status of the evaluated software, and is helpful for management decisions^[2].

At present, there have been some research results in the field of software quality prediction model. The traditional software quality prediction model is based on the size and complexity of the software to predict the number of defects^[3], in which the regression method is used. In addition, there are some software quality prediction models based on test data^[4]. Professor Khoshgoftaar in the American Florida Atlantic University has done a lot of work in this field. Khoshgoftaar and Seliya^[5] proposed a method based on regression tree to predict the number of defects in the software modules, while Khoshgoftaar proposed a software quality prediction model which combines fuzzy clustering and module-order model^[6]. Professor Cai kaiyuan introduced the fuzzy method into the field of software quality and reliability, who proposed a fuzzy model for software reliability confirmation^[7,8]. But these methods have certain limitation. The models are some rough, which can

not better describe internal software defects and failures that software shows.

The artificial neural network is applied in software quality prediction model in 1992, which get good results. Karunanithi applied neural network in software quality prediction model, which was based on defect data in an actual project and used three kinds of neural network model to estimate the total number of defects. Experiments showed that neural network is better than statistical method^[9]. Nidhi Gupta and Manu Pratap Singh used the execution time for BP neural network's input to predict the number of errors^[10]. YuLiJun and lama used forward cascade neural network to predict software quality. By comparing with some statistical methods, they get better results^[11]. Liang Tian and Afzel Noore used genetic algorithm to optimize the number of neurons in input layer and hidden layers^{[12][13]}. K.K.Aggarwal adopted different mechanisms to train neural network and made a comparison of the results predicted by the different mechanisms^[14]. S.L.Ho, M.xie and T.N.Goh estimated the software quality by adjusting the parameters of neural network^[15].

In general, there are some good results in software quality prediction at home and abroad. Software quality prediction models are mainly based on the software test report and software metric, which use statistic methods and neural network, and experiments show that neural network is better than statistical method. Although people mostly use BP network, there are some problems like slow convergence speed and easily getting into local minimum, and for the different problems of the software quality, BP neural network is not totally applicable. In order to understand the connotation of software quality prediction model, the paper proposes the optimized BP neural network based on PSO, which builds the reasonable relationship between the internal properties of software and the external properties of software. This model used CK metric index which shows the internal properties of object-oriented software: CBO、DIT、NOC、RFC、WMC、LCOM for input of the prediction model.

The paper presents the optimized BP network algorithm based on PSO to predict software quality. In order to understand the connotation of software quality

prediction model, section II firstly introduces the general theory of software metrics and the relationship with the software quality prediction model. Secondly the neural network with strong ability in data analysis is introduced in detail. Then the improved algorithm is described in particular in section IV. In the fifth part, the improved algorithm is devoted to the prediction model of software quality, and 28 groups of data are adopted in the experiment, and compared with the results by BP network. At last, the conclusions and ideas for future research is made.

II. THE RELATIONSHIP BETWEEN SOFTWARE METRICS AND SOFTWARE QUALITY PREDICTION MODEL

Software quality metric is the quantitative measurement and metric method for properties affected on software quality^[16]. In IEEE "Standard for Software Quality Metrics Methodology, IEEE Std, 1061-1992, 1993", the metric is definition as follows: metric is a function, its input is software data, output is a single value. It can explain the influence extent which a given property effect on the software quality.

Metric is a mapping which is from a real or experiential world to the mathematics world. Metric usually includes the following four basic measures:

- (1) Determine the software properties which must be measured;
- (2) Build the experiential system of software properties;
- (3) Map the experiential system to the formal system through the criteria of metrics;
- (4) Evaluate the criteria of metrics.

Software quality metric is the quantitative measurement for software quality attributes. On the basis of quantitative quality, a prediction model is one of the effective methods to improve the software quality. It can evaluate the characteristics of software quality which users concern about. When it is unable to assess the external attributes of the software directly, we can get them indirectly through calculating the metrics of other properties within the software. The relationship among software quality characteristics^[17], metrics and quality prediction model is shown in Fig. 1.

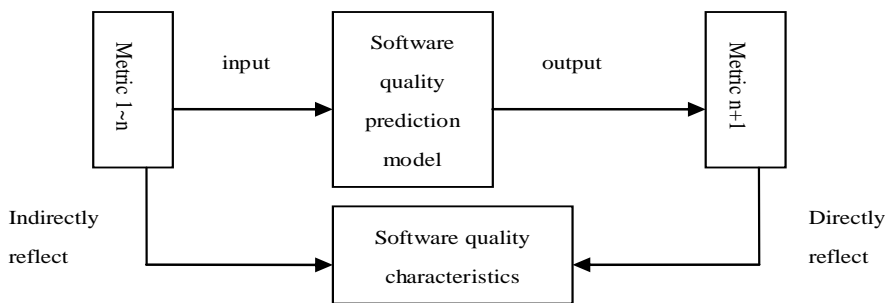


Figure 1. The relationship among software quality characteristics, metrics and quality prediction model.

In the figure, the n+1 metric can directly reflect the characteristics of software metric, but it can not be indirectly obtained. Therefore, the metric 1,..., metric n

can be regarded as input for software quality prediction model so as to get the prediction value of metric n+1

which can be used to reflect the software quality characteristics.

The software quality metric runs through the all process of the software engineering and the process after the software is delivered. The basic purpose of software metric is to change the software process by using metric which is for the need of management. The result of metric is a measurement which can be used for making decision and comparing, and measuring the known object to track and evaluate the unknown objects. So on the one hand, it can measure a final software quality, on the other hand, it can predict the possible quality of software in the development process of software, which can put forward the effective improvement.

It is because the relationship exists between software metrics and software quality that we can build software quality prediction model by looking for the relationship between software metrics and quality attributes from the historical data. It is a process of learning, and it is feasible that neural network with the ability of learning can be used to solve the prediction model.

III. BP NETWORK

A. The BP network introduction

Artificial neural network is inspired by the operation of the biological neural network function, which is an information processing mathematical model that is similar to the brain structures of synaptic connection.

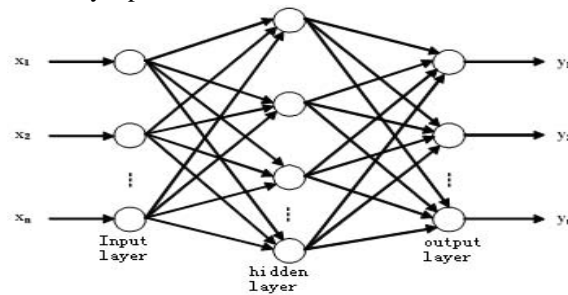


Figure 2. The structure of BP network.

The BP neural network divide the learning algorithm into two phases of forward propagation and back propagation. In the process of forward propagation, input information is processed from input layer to output layer through each hidden layer. The neurons' status of each layer only influences the neurons' status of next layer. If it cannot get the desired output in output layer, it turns into back propagation. The error signal returns through the original connecting passage. It makes error signal minimum by modifying the neurons' weight. Besides the nodes in the input layer, the input in the hidden layer and output layer is the weight sum of the output in the upper layer. The activated degree of each node is determined by its input signal, activation function and the threshold of node. The characteristic function in the node is usually S function, which is shown as formula (1).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

Neural network is structured by a large number of nodes (or neurons) and the connection along them. Each node represents a specified output function, which is called the activation function. Each connection between two nodes represents a weight, which is the memory of the artificial neural network. The outputs of network depend on the difference of the connections of network, weights and activation functions. Network itself is usually an approximation to nature algorithm or function, which may also be an expression of logical strategy. The advantage and feature of the artificial neural network is shown as follows:

(1) It can approximate fully any complicated nonlinear relation;

(2) All the qualitative information is storage in each neuron of network as potential distribution, so it has good robustness and fault tolerance.

(3) It uses the parallel distributed processing method, which makes the quick computation possible;

(4) It can handle quantitative and qualitative knowledge in the same time.

BP network is a kind of artificial neural network. BP network is an error back propagation algorithm of feed forward multilayer neural network, which is proposed by D. Ruvmelhar and McClelland in 1985. BP network is used in multilayer neural network. There is not only an input layer and an output layer, but also one or more hidden layers in the network. The model in Fig. 2 is the BP neural network that possesses one hidden layer.

B. The learning algorithm of BP network

The learning algorithm of BP network is shown as follows:

- (1) Initialize all the weights for the minimal random;
- (2) Provide training set;
- (3) Compute the output of the hidden layers and output layer to compute the actual output;
- (4) Computer the error between actual output and the expected output;
- (5) Adjust the weight of output layer;
- (6) Adjust the weight of hidden layers;
- (7) Return to (3) until the error meets the need.

BP network is widely used in feed forward multilayer neural network with multiple outputs, which can be used in nonlinear classification. So BP network is currently the most mature and widely used neural network in the fields of the information processing and pattern recognition.

But its defect lies in that the training time is longer and it easily gets into local minimum.

IV. OPTIMIZED BP NETWORK BASED ON PSO

A. Particle Swarm Optimizer (PSO)

Particle Swarm Optimizer (PSO) is an evolution algorithm based on groups. It is proposed by Jim Kennedy in 1995 and it is successfully used in function optimization. PSO is the intelligent optimization technique based on groups, which can search the solution space to get the optimal solution by the interaction between particles. PSO algorithm is one of the evolutionary algorithms, which is similar to genetic algorithm and uses fitness function to evaluate the quality of solution. But PSO has the advantages such as simple, less parameters, quick convergence speed and easily realization, so PSO is more and more applied in the fields of function optimization, neural network training, pattern classification and traditional optimization algorithm.

In PSO, each potential solution for the optimized problem is a bird in searching space, which is called "a particle". That is to say that the position of each particle is a potential solution. Firstly PSO initializes a group of particles randomly, and each particle moves in the solution space. A vector decides the moving direction and displacement of the particle, and the corresponding fitness is calculated by a function to judge whether the target is reached or not. Then the optimal solution can be obtained by iterations. Particles update themselves by tracing two extreme values in each iteration.

The number of particles is called the population size m (generally the number of initialization particles is 20~40). The position in d dimension space for particle i is expressed as $x_i=(x_{i1}, x_{i2}, \dots, x_{id})$ ($i=1, 2, \dots, m$), and its velocity is expressed as $v_i=(v_{i1}, v_{i2}, \dots, v_{id})$ which decides the displacement of the iteration units for particles in searching space. The fitness of the particle is decided by the fitness function of practical problems. According to the fitness of each particle, the personal best value of each particle is updated as $p_{best}=(p_1, p_2, \dots, p_d)$ and the global best value is updated as $g_{best}=(g_1, g_2, \dots, g_d)$. A particle updates the velocity and position by dynamically tracing the individual optimal and the global optimal. A particles update the velocity and position by the following formula:

$$v_{ij}(t+1)=w*v_{ij}(t)+c1*rand()*(p_j(t)-x_{ij}(t))+ c2*rand()*(g_j(t)-x_{ij}(t)). \tag{2}$$

$$x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t+1). \tag{3}$$

In the formula, $j=1, 2, \dots, d$; t is the number of iterations; $c1$ and $c2$ are learning factors; w is an inertia weight which can make PSO adjust the ability of searching local and global optimization, and a linear inertia weight is usually adopted; $rand()$ is the random function which generates the values between 0 and 1.

In each dimension, the velocity of each particle cannot exceed the maximum velocity v_{max} during the updating

procedure. The larger v_{max} can guarantee the global searching ability of particle swarms, and the smaller v_{max} can strengthen the local searching ability of swarms. At the same time, the position of particles in each dimension is limited in the scope of x_{max} . The flow chart of this algorithm is shown in Fig. 3: The specific steps are shown as follows:

- (1) Initialize all the particles: initialize randomly the initial position and velocity in the permitted scope;
- (2) Compute the fitness value of each particle, that is to say that calculates the objective function values of each particle;
- (3) Search the P_{best} of each particle: for each particle, compare the fitness value to the best value of the particle. If it is better, the fitness value of the particle is as the best value;
- (4) Search the g_{best} of current iteration: for each particle, compare the fitness value to the global best value. If it is better, the fitness value of the particle is as the global best value;
- (5) According to formula (2) and (3), update the position and velocity;
- (6) Check the termination conditions, if it meets the conditions, stop iteration; else return to (2).

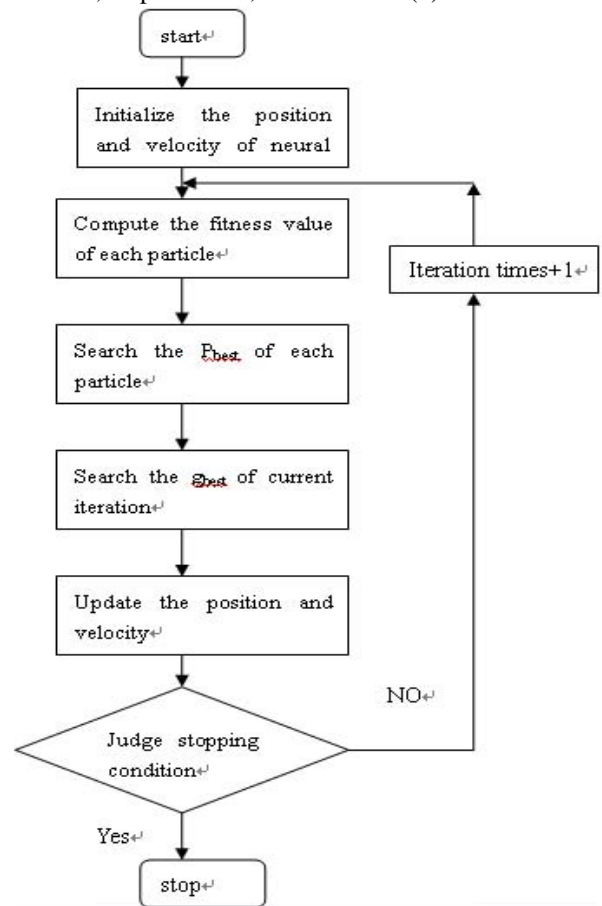


Figure 3. The flow chart of the PSO.

B. Optimized BP network based on PSO

1. The algorithm introduction

BP network adopts the gradient descent method, which possesses low convergence velocity and poor generalization capability, and easily makes the algorithm get into the local minimum. Because PSO avoids meeting the condition of differentiable function and the process of functional derivation in the gradient descent, and also avoids the selection, crossover and mutation operator of genetic algorithm, it can shorten the time of training neural network, Therefore PSO can be adopted to train the parameters of BP instead of the gradient descent of BP network, which can improve BP algorithm performance, make it uneasily into local minimum and enhance the generalization performance^{[18]-[21]}.

The method of training the BP network based on PSO is as: use the position of each particle in the PSO to represent the set of weights in the current iteration of BP network, and the dimension of particle is decided by the number of weights; use the output error of BP network as the fitness function of the PSO. In each iteration, all particles update the position according to the computed new velocity.

In optimized BP network based on PSO, $x_i=(x_{i1}, x_{i2}, \dots, x_{id})$ is a group of parameters, each dimension represents the weight or threshold, d is the number of all weights and thresholds in BP network. The fitness function of particles is shown as follows:

$$I_i = \sum_j (Y_{i,j} - y_{i,j})^2 \tag{4}$$

$$I_{popIndex} = \frac{1}{n} \sum_{i=1}^n I_i \tag{5}$$

In the formula, n is the number of samples, $Y_{i,j}$ is the ideal output for sample i , $popIndex=1, \dots, popSize$, $popSize$ is particle swarm size. The flow chart of this algorithm is shown in Fig. 4:

2. Algorithm design and realization.

The algorithm combines PSO with BP network by using PSO to optimize the weight and threshold of neural network which can improve the performance of BP. The fitness function of particles is obtained by MSE of BP network. Specific steps are shown as follows:

Step 1: Determine the structure of neural network, including the neuron number of input layer, output layer, and hidden layers.

Step 2: Initialize particle swarms, including the position and velocity of each particle, particle swarm size, learning factor $c1, c2$ and inertia weight w and so on, determine the fitness function. MSE of neural network can be used as fitness to guide population for searching.

Step 3: Calculate the fitness, use all the training samples to calculate in forward propagation for each particle, generate the training error of particles with training samples, calculate the fitness with formula (5).

Step 4: Update the individual extremum and global optimization.

Step 5: Update position and velocity of each particle with formula (2) and (3), update inertia weight w .

Step 6: Judge the stopping condition of the algorithm, judge the algorithm whether meeting the maximum iteration or meeting the prescribed standard error. If it meets the condition, the optimum solution will be generated. Otherwise return step 3 after increasing iterative times.

In the optimized BP network based on PSO, each element in particle vector is composed of weights and thresholds of BP network, and the fitness function of particles is obtained by MSE of BP network. All these can lead to achieve the fusion of PSO fully.

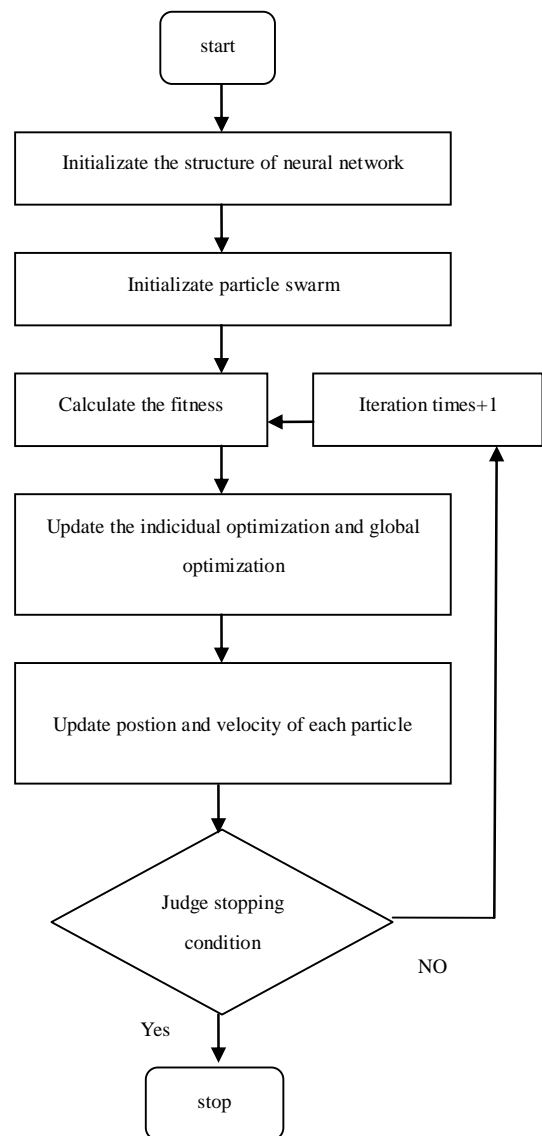


Figure 4. The flow chart of the optimized BP network based on PSO.

V. CASE STUDY

A. Parameter setting

In order to prove that the proposed method in this paper is better than the BP network, here this paper predicts the object-oriented software. The object-oriented software quality has six facets, including functionality, reliability, usability, efficiency, maintainability and portability. But because the neural network of multiple input -multiple output can be divided into many neural networks of multiple input -one output, so the reliability, functionality, usability, efficiency, maintainability and portability of software can be predicted respectively. This experiment is to predict the object-oriented software reliability. Data is from kc1 database in the metric data plan of NASA, and the samples are shown in table I. The model used CK metric index which shows the internal

properties of object-oriented software: CBO、DIT、NOC、RFC、WMC、LCOM for input and the reliability for the output of the prediction model^{[22]-[26]}. Thus we constitute the object-oriented software reliability prediction model which possesses six inputs and one output.

The experiment adopts 28 samples to train the optimized BP network based on PSO; the number of hidden nodes is 12, so the structure of the BP network based on PSO is 6-12-1. The parameters' setting of PSO: the number of particles is 30; c1=2, c2=1.8; w linear decreases from 0.9 to 0.3 with the iteration.

We make data normalization processing in (0,1) before training, which can accelerate the training speed and improve precision. From the 28 samples, 20 groups of samples are selected as training samples and the rest of samples are as validation data.

TABLE I.
SAMPLE DATA

The number of sample	CBO	DIT	LCOM	NOC	RFC	WMC	reliability
1	24	4	100	0	110	73	0.3401
2	19	4	100	0	78	30	0.0381
3	13	1	88	0	99	99	0.8729
4	21	4	100	0	68	30	0.0099
5	17	2	90	0	69	36	0.6873
6	10	1	88	0	22	22	0.856
7	19	3	94	0	53	17	0.4122
8	14	2	90	0	37	25	0.6501
9	18	2	100	1	38	13	0.7334
10	6	2	59	0	33	25	0.3352
11	4	1	75	2	8	8	0.5657
12	1	4	100	0	53	22	0.0324
13	3	2	100	0	38	11	0.9028
14	10	2	46	0	23	15	0.0129
15	0	1	100	1	98	98	0.2882
16	14	3	96	0	48	18	0.1703
17	14	4	87	5	61	26	0.1564
18	17	5	93	0	54	6	0.5314
19	12	2	83	0	52	39	0.5055
20	18	2	100	1	38	13	0.7334
21	14	2	86	0	37	10	0.781
22	1	4	100	0	53	22	0.0324
23	20	4	95	0	85	28	0.1685
24	9	2	93	0	4	14	0.8221
25	9	1	83	0	14	14	0.5244
26	11	2	80	1	28	15	0.6822
27	10	1	88	0	25	25	0.8537
28	0	1	100	1	35	35	0.5341

B. Results Analysis

The figure of error curves with BP and optimized BP based on PSO are shown in Fig. 5 and Fig. 6. From Fig. 5 we know that the BP achieve the optimal error in the iteration of 204, and the error is 0.0364. From Fig. 6 we know that the optimized BP based on PSO achieve the

optimal error in the iteration of 98, and the error is .0102. The optimized BP network based on PSO shortens the training time and improves the convergence speed, and the MSE is smaller than BP network.

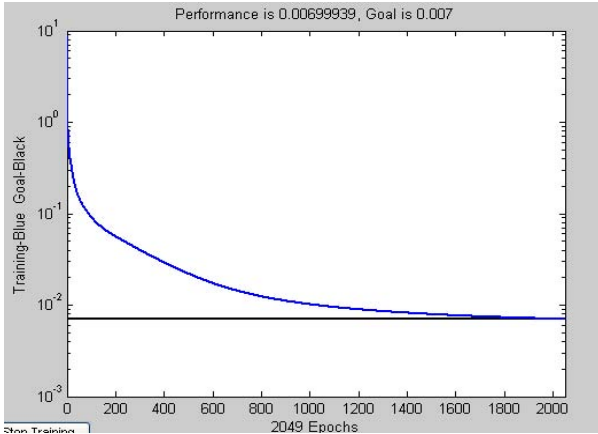


Figure 5. The figure of error curve with BP.

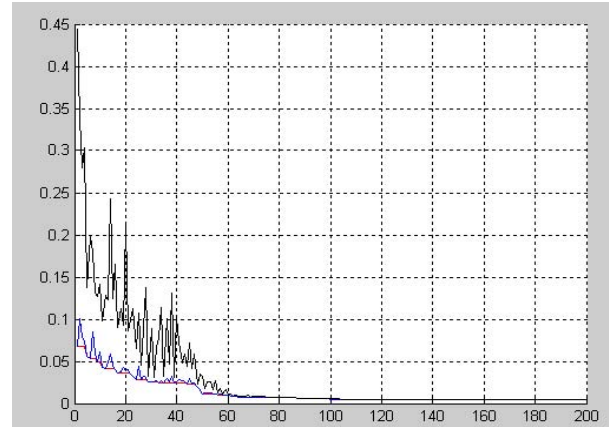


Figure 6. The figure of error curve with optimized BP based on PSO.

TABLE II.
COMPARISON WITH LEARNING METHODS

Name	Epoch	MSE
BP	2049	0.0364
PSO-BP	98	0.0102

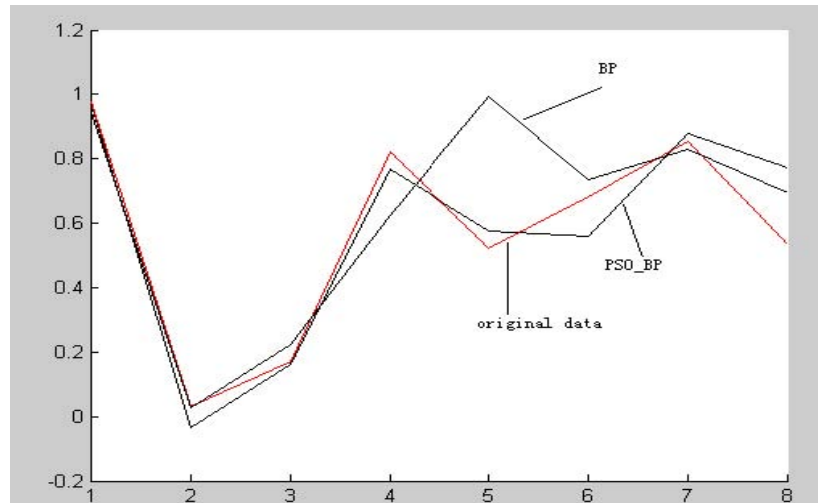


Figure 7. Curve fitting of validation data.

Table II. shows that achieving the desired error only needs 98 iterations by optimized BP network based on PSO, while it needs 2049 iterations by the tradition BP network. The optimized BP network based on PSO shortens the training time and improves the convergence speed, and the MSE is smaller than BP network. From Fig. 7, we can see that the optimized BP network based on PSO can better validate the data. The experiment expresses the optimized BP network based on PSO is better than the traditional BP network, which can improve BP algorithm performance, make it uneasily into local minimum and enhance the generalization performance. Also it can better response the nonlinear relation better between metric data and quality property, and it establishes the objective and quantitative contact between software internal properties and external properties.

VI. CONCLUSION

Software quality is the natural attribute of software, which is influenced by all kinds of uncertainty factors in

software development process, so we should choose the right model to reflect the relationship between metric and software quality. Software quality prediction model is the key technology in software quality evaluation system, by which the objective and quantitative relationship is established between the software internal properties (metrics) and external properties (quality indicators). It can avoid the interference of subjective factors, and make the evaluation result be in keeping with the actual quality status of the evaluated software, and is helpful for management decisions [2]. The paper introducing the PSO and BP network into the software quality prediction model with using the reasonable data analysis technique can obtain the software external properties which cannot be directly measured.

The paper combines PSO with BP by which the software reliability is predicted, and an optimized BP network prediction model with 6 metrics for inputs is established. Results show that the optimized BP network based on PSO can quickly and accurately predicts software quality, and it overcomes the defects that

parameters are made by experts' experience. The model can rightly determine the relationship between internal properties and external properties. However the numbers of the hidden nodes and hidden layers in the model are set at the beginning by experience. How to optimize the BP network structure by PSO in order to get the optimum network structure is the next research work.

REFERENCES

- [1] Zheng Renjie, Yin Renkun, Tao YongLei. Practical software engineering[M]. BeiJing:Tsinghua university press,2002
- [2] L IHu, SH Ixiao-Hua, YANG Hai-Yan, and GAO Zhong-Yi. Software quality evaluating technique[J]. Computer research and development, 2002; 39(1): 61-67
- [3] Hatton, L.C. Safety Related Software Development: Standards, Subsets, Testing, Metrics, Legal Issues, McGraw-Hill, 1994
- [4] Fenton, N.E, Lawrence S., Glass, R. Science and substance: a challenge to software engineers, IEEE Software, 1994, pp.86-95
- [5] Khoshgoftaar, T.M. Seliya, N. Tree-based software quality estimation models for fault prediction, Proceeding of the 8th IEEE Symposium on Software Metrics, 4-7 June 2002, pp.203-214
- [6] Khoshgoftaar, T.M., Liu, Y., et al. Multiobjective module-order model for software quality enhancement, IEEE Transactions on Evolutionary Computation, 2004. Vol8, No.6,pp.593-608
- [7] Cai KaiYuan, The software reliability engineering foundation, Tsinghua university press, 1995
- [8] Cai KaiYuan, A fuzzy software reliability confirmation model, aeronautical journal, 1993,11:95-98
- [9] N.Karunanithi, D.Whitley, Y.K. Malaiya. Using neural networks in reliability prediction. IEEE Software, 1992, 9(4): 53-59
- [10] Nidhi Gupta, Manu Pratap Singh. Estimation of software reliability with execution time model using the pattern mapping technique of artificial neural network. Computer and Operations research, 2005, 32(1):187-199
- [11] Yu Lijun, Ben Kerong. Realization and Analysis of Software Reliability Based on Neural Networks. Computing Technology and Automation, 2002, 21(3):40-42
- [12] Liang Tian, Afzel .Noore. Evolutionary neural modeling for software cumulative failure time prediction. Reliability Engineering and System Safety, 2005, 87(1):45-51
- [13] Liang Tian, Afzel Noore. On-line prediction of software reliability using an evolutionary connectionist model. Journal of Systems and Software, 2005, 77(2):173-180
- [14] K.K.Aggarwal, Yogesh Singh, Pravin Chandra, Manimala Puri. Evaluation of various training algorithms in a neural network model for software engineering applications. ACM sigsoft software engineering notes, 2005, 30(4):1-4
- [15] S.L.Ho, M.xie, T.N.Goh. A study of connectionist models for software reliability prediction. Computers & Mathematics with Applications, 2003, 46(7):1037-1045
- [16] Cheng Duping, Qian Hongbing. Software Quality and Metrics[J]. Computer Engineering and Applications, 2002; 38(7): 80-83
- [17] Ma Huimin, Song Yu, Xu Zhengwei. An Application of General Regression Neural Network to Software Quality Prediction[J], Computer Engineering and Applications, 2004, 29:217-219
- [18] Chen Shu Xu Baoguo Wang Haixia Wu Xiaopeng. Study of fermentation process based on PSO-BPNN[J]. Computers and Applied Chemistry, 2007. 24(7) :885 – 887
- [19] ZENG Wan-li; WEI Ren-yong; CHEN Hong-ling. Research and Application of BP Neural Network Based on Improved PSO Algorithm[J]. Computer Technology and Development, 2008. 18(4) :49 – 51
- [20] Shi Y, Eberhart R C. A modified particle swarm optimizer C .Proc. IEEE Int'l Conf. on Evolutionary Computation, NJ , 1998:69 73.
- [21] LIU Hong-bo, WANG Xiu-kun, MENG Jun. Neural Network Training Algorithm Based on Particle Swarm Optimization[J]. Mini-micro Systems, 2005, 26(4) :638 - 640
- [22] Mie Mie Thet Thwin, Tong-Seng Quah. Application of neural network for predicting software development faults using objectoriented design metrics[C]//Proceeding of 9th International Conference on Neural Information Processing(ICONIP' 02), 2002, 5: 2312-2316.
- [23] Quah Jon T S, Mie Mie Thet Thwin. Prediction of software readiness using neural network [C]//Proceedings of 1st International Conference on Information Technology & Applications(ICITA 2002). Australia. 25-28 Nov 2002.
- [24] Mie Mie Thet Thwin, Tong-Seng Quah. Application of neural networks for software quality prediction using object oriented mettics[J]. Journal of Systems and Software, 2005, 76(2): 147-156.
- [25] S. Kanmani, V. Rhymend Uthariaraj, V.Sankaranarayanan, P. Thambidurai. Object oriented software quality prediction using general regression neural networks[C]//ACM SIGSOFT Software Engineering Notes, 2004, 29(5): 1-6.
- [26] ZHANG Jin, HU Ping. Analyse and Design on C&K Object-Oriented Software Metrics Suit. Microprocessors, 2008,4

Kewen Li, associate professor, College of Computer and Communication Engineering, China University of Petroleum, Dongying, Shandong, 257061, China; School of Management, Tianjin University, Tianjin,300072, China.

Jisong Kou, School of Management, Tianjin University, Tianjin, 300072, China

Lina Gong, College of Computer and Communication Engineering, China University of Petroleum, Dongying, Shandong, 257061, China.