RESEARCH ARTICLE OPEN ACCESS

# Evaluation Technology of Software Reliability by using the BP Neural Network

D.Jyothirmai[1], M.Suresh Kumar[2], Dr.Katta Subba Rao[3]
[1](CSE, BVRIT, Narsapur),
[2](Scientist-E,DRDO,Hyderabad,Telangana),
[3] (CSE, BVRIT, Narsapur)

**Abstract:**

In the Software Industry, the Software Reliability linked with Neural Networks are very important. Here, In this paper We briefly introduces the BP(Back Propagation) neural network and also discuss about the application of BP neural network algorithm in evaluating the reliability of software, it constructs the evaluation index system of software reliability based on the neural network, and finally design the software reliability evaluation system. This system takes the scores of influence factors(main factors) of software reliability as input which are marked by user, and then output the evaluation result of software reliability computed by the trained neural network.

*Keywords*—**BP neural network; influence factors, software reliability, evaluation system**

## I. INTRODUCTION:

In the network side, the artificial neural network contains a large number of simple processing units, these are connected with each other into the network according to a certain way. It has successfully solved many problems that modern computer could not solved and showed good performance of intelligence and potential application prospect at many areas, such as pattern recognition, signal processing, prediction and evaluation, automatic control, fault diagnosis, system identification, combinatorial optimization, medicine and economics.

BP(Back Propagation) neural network is one of the most widely used neural networks. This paper discusses the application of BP neural network to predict software reliability, and designs a software reliability evaluation system.

## II. BP NEURAL NETWORK

The BP neural network is a multilayer feed forward neural network .It contains one or more hidden layers, which are trained by the error back propagation algorithm . The learning process of BP algorithm consists of two processes, which are the positive propagation of signal and back propagation of error. The two processes are to adjust the weight matrix of each layer. The processes will be operating until the error of the network output is reduced to an acceptable degree, or the learning time is to a predetermined number.

BP neural network is the most widely used at present, which is mainly because that,.

it has the following important capabilities

### *A. Nonlinear Mapping*

Multilayer feed forward neural network can learn and store a large number of input-output mappings, so it is not necessary to understand the mathematical relationship of the mappings. As long as enough samples can be provided for the BP network to train and learn, it is able to complete the nonlinear conversion from input space to the output space.

### *B. Generalization*

In many cases, we require that neural network has good fitting ability of the existing training sets, and we also require that it can make better prediction to the same distribution data beyond the training set. This is related to the generalization ability of neural network which refers to the ability of the neural network model already trained can make correct prediction to the sample beyond the training set. Training is not a simply remembering the sample data, but to grasp the inherent regularity implied in the sample.

### *C. Fault Tolerant*

Another advantage of the multilayer feed forward neural network is that the input samples are allowed to have some errors. This is because the adjustment of weight matrix is the process of extracting statistical characteristics from a large number of samples. The knowledge which can reflect the correct rules comes from all the samples, and the error of the individual sample cannot affect the adjustment result of weight matrix.

The BP Neural Network Contains the Inputs and Output. The Inputs taken as Voltage, Current, Temparature and Cycle Time and the output as Sec is shown in the below figure:1

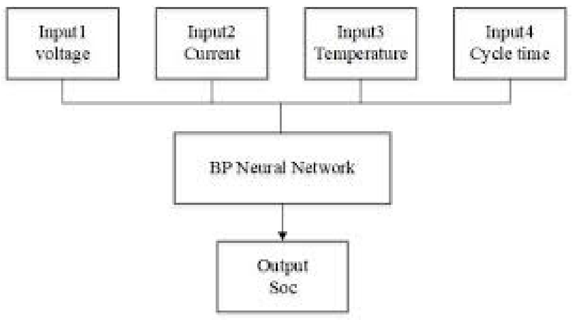

Figure:1.Input and Output of the BP Neural Network

## III. Design of Software Reliability Evaluation System:

Software reliability has become the focus of people's attention and research as an important measure of software quality The most direct method of software reliability measurement is to test the failure of software in the actual operating environment. However, this approach is flawed. Firstly, it requires the software to put into operation and the operation time will be relatively long.

Secondly, people want to estimate the software reliability in development stage to test whether it achieves the desired goal. So people put forward the method of software reliability modelling But at present, the accuracy of many software reliability models is limited, that is, the model can achieve a higher level of expected just for only one or a few software engineering projects or a certain stage of data. Such models are often not robust, and a single model is not adapted to the minor changes in software engineering projects, which makes the error increase dramatically

Due to the lack of existing reliability models, this paper establishes a relational model between software reliability assessment value and influence factors by making the advantages of the neural network -- self-learning, self-organizing, adaptive ability. Flow chart of software reliability evaluation based on BP neural network is shown in Fig. 2

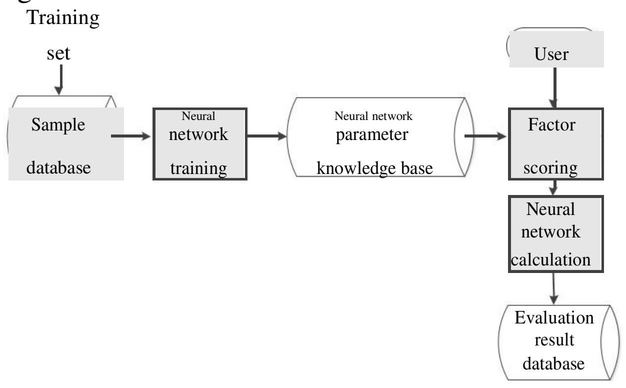
Training set
User
Sample database
Neural network training
Neural network parameter knowledge base
Factor scoring
Neural network calculation
Evaluation result database

Fig. 2. Flow chart of software reliability evaluation based on BP neural network

This method trained the BP network by the samples of influence factors scores marked by experts. The system selects ten relatively important and representative software reliability influence factors, so that the generalization ability of the trained neural network will be better. When evaluating the software, the system reads the weights of BP neural network stored in the knowledge base and the values of ten influence factors marked by user, and then finally get the reliability of the software after operation.

### *A. The Main Factors of Software Reliability*

Software reliability influence factors can be divided into the following categories: software itself factors, human factors, development factors, environmental factors, software reliability engineering factors and other factors

If the number of factors which is the input layer in the evaluation index system is too large, it will reduce the generalization ability of the network and ultimately affect the accuracy of the system. So we select ten factors as evaluation index. They are software architecture, software requirements, coupling degree with hardware, software process maturity, human factors, software development process, environmental factors, software reuse, code quality, and other factors.

### *B. Design of BP Neural Network*

#### *1) Structure and parameters of BP network:.*

The software reliability evaluation problem can be regarded as the nonlinear mapping relationship between software reliability assessment value and scores of influence factors. A three-layer BP network can approach the mapping relation with any precision [8], so the system adopts three layers of BP network structure.

#### *2) Number of input neurons:*

This system selects ten factors which are relatively important and representative as the evaluation indexes. So the number of input layer neurons is identified as 10 and the input parameters are the score of the influencing factors.

#### *3) Number of output neuron:.*

The software finally gets the evaluation value of the software reliability, so the output layer has only one neuron. It is a numerical value which ranges between 0 and 1. The higher the output value means the software has the better reliability

#### *4) Number of hidden layer neuron:.*

The function of the hidden layer neurons is to extract and store the inherent rules from the learning samples. Each neuron has a number of weights and each weight is a parameter to enhance the ability of network mapping.

If the number of hidden layer neurons is too small, the ability of extracting knowledge from a sample will decline, and it's not sufficient to reflect the rules of learning samples. If the number of hidden layer neurons is too large, it will learn the irregular knowledge of the samples, which can reduce the generalization ability. The system calculates the appropriate number of hidden layer neurons by defining a function

#### 5)*Learning samples for training:*

The learning samples set is generally an authoritative evaluation result with high reliability, which can be evaluated by the experts.

## IV.CONCLUSION

In this paper, the prediction of software reliability is realized by constructing BP artificial neural network. As the trained BP neural network has strong ability of pattern recognition, it can no longer depend on the condition of the software in the actual operating environment. BP neural network model can also continue to optimize by choosing more reasonable influencing factors and training with a representative the sample set.

## V.REFERENCES

[1] L. Q. Han, editor. Theory, Design and Application of Artificial Neural Network, Chemical Industry Press, Beijing, 2002.

[2] J. H. Sui, L. C. Fan, H. S. Zhang, "Research on Software Reliability Prediction Technology," Computer Engineering, Vol. 32, No. 1, pp. 67-70, 2006.

[3] M. Cong, M. Y. Lu, Y. F. Bai, "Study and Implementation of Software Reliability Prediction Method," Journal of Beijing University of Aeronautics and Astronautics, Vol. 28, No. 1, pp. 34-38, 2002.

[4] P. Tian, W. T.Tian, Z. Liu, K. Qiang, "The Research of Embedded Software Reliability Evaluation System,"2008 International Conference on MultiMedia and Information Technology.

[5] J. H. Liu, N. Li, "Influence Factor Analysis of Software Reliability Based on The Comparative Analysis of Literature," Electronic Design Engineering, Vol. 19, No. 21, pp. 21-30, 2011.

[6] W. Shuang, J. G. Qiu, "Evaluation Method of Campus Website Based on BP Neural Network," Equipment Manufacturing Technology, No. 11, pp. 65-68, 2009.

[7] W. H. Lu, "The Application of Artificial Neural Network BP Algorithm in Evaluating the Website," Sci-Tech Information Development & Economy, Vol. 17, No. 6, pp. 170-172, 2007.

[8] J. D. Musa, "Software Reliability Data," Data & Analysis Centre for Software, January 1980.

[9] R. Iyer and I. Lee, "Measurement-based analysis of software reliability," Handbook of Software Reliability Engineering, McGraw-Hill, pp. 303 – 358, 1996.

[10] J. D. Musa and K. Okumoto, "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement," in ICSE, EEE Press Piscataway. NJ, USA: Proceedings of the 7th International Conference on software Engineering, pp. 230–238, 1984.

[11] T. R. Benala, S. Dehuri, and R. Mall, "Computational Intelligence in Software Cost Estimation: An Emerging Paradigm," ACM SIGSOFT Software Engineering, vol. 37, no. 3, pp. 1–7, May 2012.

[12] IEEE, "Standard glossary of software engineering terminology," Standards Coordinating Committee of the IEEE Computer Society, 1991.

[13] Simon Haykin, "Neural Networks A Comprehensive Foundation", Pearson Prentice Hall, 2nd Edition, 2001.

[14] L. Tian and A. Noore, "Software Reliability Prediction Using Recurrent Neural Network with Bayesian Regularization," International Journal of Neural Systems, vol. 14, no. 3, pp. 165–174, June 2004.

[15] A. L. Goel, "Software reliability models: Assumptions, limitations, and applicability," IEEE Transaction on Software Engineering, vol. 11, no. 12, pp. 1411–1423, December 1985

[16] P. Werbos, "Generalization of Back propagation with Application to Recurrent Ga Market Model," Neural Network, vol. 1, pp. 339–356, 1988.

[17] R. Shadmehr and D. Z. DSArgenio, "A Comparison of a Neural Network Based Estimator

and Two Statistical Estimators in a Sparse and Noisy Data Environment," in IJCNN, vol. 1, Washington D.C, pp. 289–292, June 1990. 36

[18] N. Karunanithi, Y. Malaiya, and D. Whitley, "Prediction of Software Reliability Using Neural Networks," in Proceedings IEEE International Symposium on Software Reliability Engineering. Austin, TX: IEEE, pp. 124– 130, May 1991.

[19] T. M. Khoshgoftaar, A. S. Pandya and H. More, "A Neural Network Approach For Predicting Software Development Faults." Research Triangle Park, NC: Proceedings of Third International Symposium on Software Reliability Engineering, pp. 83–89, October 1992.

[20] Y. Singh and P. Kumar, "Prediction of Software Reliability Using Feed Forward Neural Networks," in Computational Intelligence and Software Engineering (CiSE), I. Conference, Ed. IEEE, pp. 1–5, 2010.

[21] N. Karunanithi and D. Whitley, "Prediction of Software Reliability Using Feed forward and Recurrent Neural Nets," in Neural Networks, 1992. IJCNN, vol. 1. Baltimore, MD: IEEE, pp. 800–805, June 1992.

[22] R. Sitte, "Comparison of software-reliability-growth predictions: neural networks vs. parametric-recalibration," Reliability, IEEE Transactions, vol. 48, no. 3, pp. 285–291, September 1999.

[23] N. RajKiran and V. Ravi, "Software Reliability Prediction using Wavelet Neural Networks," in International Conference on Computational Intelligence and Multimedia Applications, vol. 1. Sivakasi, Tamil Nadu: IEEE, pp. 195 – 199, December 2007.

[24] J. H. Lo, "The Implementation of Artificial Neural Networks Applying to Software Reliability Modeling," Control and Decision Conference, 2009. CCDC '09, Chinese, pp. 4349 – 4354, June 2009

[25] L. Zhao, J. pei Zhang, J. Yang, and Y. Chu, "Software reliability growth model based on fuzzy wavelet neural network," in 2nd International Conference on Future Computer and Communication (ICFCC), vol. 1. Wuhan: IEEE, pp. 664– 668, May 2010. 37

[26] R. Mohanty, V. Ravi, and M. R. Patra, "Hybrid Intelligent Systems for Predicting Software Reliability," Applied Soft Computing, vol. 13, no. 1, pp. 189–200, August 2013.