Research Journal of Applied Sciences, Engineering and Technology 6(7): 1246-1251, 2013
DOI:10.19026/rjaset.6.3939
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: October 30, 2012 Accepted: December 21, 2012 Published: July 05, 2013

Research Article

A BP Neural Network Based on Improved Particle Swarm Optimization and its Application in Reliability Forecasting

[1]Heqing Li and [2]Qing Tan
[1]School of Mechanical and Electrical Engineering, Central South University, Changsha 410083, China
[2]School of Automobile and Mechanical Engineering, Changsha University of Science of Technology, Changsha 410076, China

Abstract: The basic Particle Swarm Optimization (PSO) algorithm and its principle have been introduced, the Particle Swarm Optimization has low accelerate speed and can be easy to fall into local extreme value, so the Particle Swarm Optimization based on the improved inertia weight is presented. This method means using nonlinear decreasing weight factor to change the fundamental ways of PSO. To allow full play to the approximation capability of the function of BP neural network and overcome the main shortcomings of its liability to fall into local extreme value and the study proposed a concept of applying improved PSO algorithm and BP network jointly to optimize the original weight and threshold value of network and incorporating the improved PSO algorithm into BP network to establish a improved PSO-BP network system. This method improves convergence speed and the ability to search optimal value. We apply the improved particle swarm algorithm to reliability prediction. Compared with the traditional BP method, this kind of algorithm can minimize errors and improve convergence speed at the same time.

Keywords: BP improvement, neural network, reliability prediction, the improved PSO

INTRODUCTION

The Particle Swarm Optimization (PSO), proposed by Kennedy and Eberhart (1995), Eberhart and Kennedy (1995) and Eberhart and Shi (2001), was based on the optimal algorithm of swarm intelligence and It guides optimal search through swarm intelligence producing by the corporation and competition among particles. Because the algorithms has the virtues of the simple principle, the easy realization, prompt self-optimized paces and robustness, now it has been widely used in function optimization, neural works, fuzzy system control and so on. However, in early stage, the searching speed is fast, but later it becomes slower. Particle swarm shows intensive similarity, thus falling into local optimum.

The artificial Neural Network possesses the ability of processing database in parallel, organizing and learning by itself. It has made great achievements in many fields. Especially neural network technique has been used in reliability and few researchers apply the method to predict failures (Ying-Jie and Chang-Hua, 2005; Hai-Dong *et al*., 2003; Jin-Yong *et al*., 2004). Theoretically, BP Neural Network may approach any continual nonlinear function. However, affected greatly by the sample, it may fall into local minimum value, thus it can't ensure that it converge the minimum value in overall situations.

Therefore, current research emphasis has been on optimal improvement of BP Neural Network. Recently, PSO algorithm is introduced into BP network optimization by using iterate algorithm of particle swarm instead of gradient correction algorithm of BP network. This method can shorten the time of training network and improve the convergence speed of BP algorithm. However, according to different system object, if basic particle swarm algorithm is modified, the performance of BP network is improved and its practicality is better. For balancing between the global search and the local search and improving the precision of the result, this study proposed the nonlinear strategies for decreasing inertia weight based on the idea of the existing linear decreasing inertia weight. The result of simulation shows that the optimizing methods speed the rapidity of convergence obviously and the precision of simulation increase greatly.

THE OPTIMAL ALGORITHM OF PARTICLE SWARM

The optimal algorithm of improved particle swarm is not dependent of fields of problems. However, it uses the code of decision variable as operation object and

Corresponding Author: Heqing Li, School of Mechanical and Electrical Engineering, Central South University, Changsha 410083, China

adaption function as searching objects. Furthermore, it can use the information from various searching points. It applies to solve the problem about nonlinearity and non-differentiable function and multiple objectives. It has been applied to many scientific fields (Jin-Yong *et al*., 2004; Xiang *et al*., 2008; Jie-Ping *et al*., 2009), but it still has many problems.

The optimal algorithm of basic particle swarm: Supposing in the D-dimensional objects searching space, there is a community composed of N particle. The "I" particle represent a D-dimensional vector, $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$. It means that the "i" particle represents its position in this space. Every position of particle "X" is a potential solution. If we put "x" into objective function, we can know the adaptive value. We can know whether the "x" is the optimal answer based on the adaptive value. The speed of particle is also a D-dimensional, it also recorded as $v_i = (v_{i1}, v_{i2}, \ldots, v_{id})$. We record the particle I to the h times, the optimal position was $p_i = (p_{i1}, p_{i2}, \ldots, p_{id})$. All the particles to the h times, the optimal position was $p_{gd} = (p_{i1}, p_{i2}, \ldots, p_{id})$. The basic formulas are as follows:

$$v_{id}^{t+1} = wv_{id}^{t} + c_1 r_1^t (p_{id}^t - x_{id}^t) + c_2 r_2^t (p_{gd}^t - x_{id}^t) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2)$$

where,

C_1 & C_2 : Speeding coefficient, adjusting the maxim step length that flying the best particle in whole situation and the individual best particle respectively. Appropriate C_1 and C_2 can speed up the convergence and avoid falling into partial optimality

r_1 & r_2 : Random number between 0 and 1, for controlling the weight of speed

W : Inertia factor. It was oriented toward overall searching

We usually take the original value as 0.9 and make it to 0.1 with the addition and reduction of the times of iteration. It mainly used to total searching, making the searching space converge to a certain space. Then we can get the solution in high degree of accuracy by partial refined researching (Yi-Shan *et al*., 2009).

The optimal algorithm of improved particle swarm: With the increasing number of dimension of problems, basic PSO algorithm is easily falling into partial extreme value, thus influence the optimal function of algorithm. Someone brought up with improved algorithm. Many scholars' research shows that "w has a great influence on the algorithm of particle swarm (Xiao-Rong and Shu-Xian, 2008; Hao and Xiao-Lei, 2008). When the "w" is bigger, the algorithm has a strong ability in total searching and when the "W" is smaller, it is good for partial searching. "Therefore, in recent years, some scholars brought up many schemes (Jiang-Hong *et al*., 2006; Xuan *et al*., 2009). According to formula (2), literature (Jiang-Hong *et al*., 2006) came up with LDW (Linearly Decreasing Inertia Weight) that is:

$$w = w_{\max} - \frac{t \times (w_{\max} - w_{\min})}{t_{\max}} \quad (3)$$

where,

w_{max} & w_{min} : The maximum and minimum value of W

t : The step of iteration

t_{max} : The maximum iteration step

However, there are still problems in formula (3). In the primary period of operation, if it detects the optimal point, it wants to converge to the optimal point promptly. However, the linear reduction slows down the speed of convergence of algorithm. In the later period of function, with the reduction of "w", it may make the ability of total searching decline and the variety awaken. Finally it may easily falling into partial optimum (Xiu-Ye and Hong-Xia, 2010). In this text, we use the PSO method of nonlinear variation weight with momentum to improve this method. That is:

$$w = w_{\max} - \frac{t \times (w_{\max} - w_{\min})}{2^{\theta} t_{\max}} \quad (4)$$

2^{θ} is momentum, when in $\theta = t/t_{\max}$, t is smaller, 2^{θ} is near to 1 and w is near to $w_{\max}$., it ensure the ability of total searching.. With the increasing of t, w reduces in non linearity, ensuring the searching ability in partial areas. In the later period ($t = t_{\max}$), avoiding the problems caused by the decrease of w. That is, the reduction ability of total searching and the decline of variety.

THE IMPROVED PSO-BP ARTIFICIAL NEURAL NETWORK ALGORITHM

BP neural network: The standard BP Neural Network consists of input layer, one or several hidden layers and an input layer, as the Fig. 1.

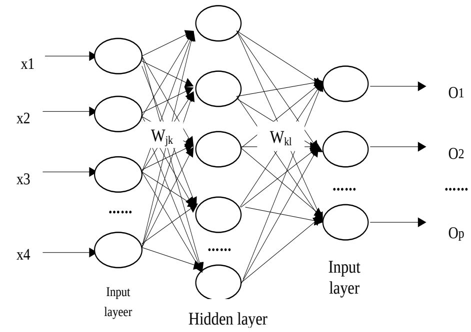


Fig. 1: The structure design of BP neural network

The node action function of BP neural network is generally "S" function. Common activation function f (x) is derivable Sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{5}$$

Error function R is:

$$R = \frac{\sum (Y_{mj} - Y_j)^2}{2} \quad (j = 1, 2, \ldots, n) \tag{6}$$

In this formula, Y_j is expected out Y_{mj} is actual output n is sample length.

The uniform expression of weight modified formula of BP algorithm is:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_{pj} o_{pj} \tag{7}$$

where,

W_{ij} = The connecting weight of neurons
η = Networks learning rate
O_{pj} = The output of sample p
δ_{pj} = Error correction value

BP algorithm: The specific process of BP algorithm can be generalized as follows:

Step 1: Select n samples as a training set.

Step 2: Initialize weight and biases value in neural network. The initialized values are always random numbers between (-1, 1).
Every sample in the training set needs the following processing:

Step 3: According to the size of every connection weight, the data of input layer are weighted and input into the activation function of hidden layer and then new values are obtained. According to the size of every connection weight, the new values are weighted and input into the activation function of output layer and the output results of output layer are calculated.

Step 4: If there exists error between output result and desired result, the calculation training is wrong.

Step 5: Adjust weight and biases value.

Step 6: According to new weight and biases values, the output layer is calculated. The calculation doesn't stop until the training set meets the stopping condition.

The improved particle swarm algorithm optimizes neural network: The method that we apply Improved Particle Swarm Optimization to train BP Network is: improved PSO-BP algorithm is to use improved PSO algorithm to optimize the original weight and the threshold value. When the algorithm ends, we can find the point near the overall situation optimal point. On the base of improved PSO algorithm, we can use BP algorithm to search overall situation by starting from here and then achieve the network training goal. In the particle swarm, every particle's position represents weights set among the BP network during the resent iteration. The dimension of every particle is decided by the number of the weight and the threshold value serving as connecting bridge.

The concrete process can be narrated as follows:

Step 1: Initialization: n_i is the number of neurons in the hidden layer no represent the number of neurons in input layer. So, the dimension of particle swarm D is:

$$D = n_i \times n_h + n_h \times n_o + n_h + + n_o \tag{8}$$

Step 2: Setting fitness function of particle swarm' in this text, we choose mean square error in BP Neural Network as fitness function of particle swarm:

$$E = \frac{1}{M} \sum_{K}^{m} \sum_{j=1}^{n_0} (y_{kj} - \bar{y}_{kj}) \tag{9}$$

Y_{kj} : The output in theory based on sample K
$\bar{y}_{kj}$: The virtual output based on sample K
M : The number of Neural Network

Step 3: Using the improved particle swarm algorithm to optimize the weight and the threshold value of BP network.

Step 4: Coming to the optimal weight and the threshold value based on formula 10:

$$g_{best} = [h_1, h_2, \ldots, h_{n_h}, o_1, o_2, \ldots, o_{n_o}, ih_1, ih_2, \ldots, ih_{n_i \times n_h}, ho_1, ho_2, \ldots, h_{o_h \times n_o}] \tag{10}$$

where,

h_i (i = 1, 2,…, n_k) : The threshold value in the hidden layer
o_i (i = 1, 2,…, n_o) : The threshold value in the output layer
ih_i (i = 1, 2,…, $n_i \times n_h$) : The weight between the hidden layer and the input layer
ho_i (i = 1, 2,…, $n_h \times n_o$): The weight between the hidden layer and the output layer

Step 5: Letting the optimal weight and the threshold value as the original weight and the threshold value of BP network and then put them into Neural Network for training. Adjusting the weight and the threshold value based on BP algorithm until the function index of the network's Mean Square Error (MSE) <e. e is the preset expected index.

THE APPLICATION OF THE IMPROVED PSO-BP NETWORK ALGORITHM IN RELIABILITY FORECASTING

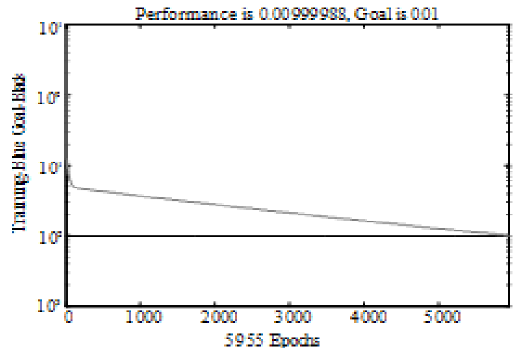
The fundamental information of reliability forecasting: Various failures inevitably occur during using product, and mean time between failures for product is one of important index in product reliability. However, it is very difficult that traditional reliability forecasting and test evaluation give satisfied results. Relative errors between predicted value and observed value reach from several tens percent to several thousand percent according to statistics (Michael *et al*., 1993). At present, neural network technique has been used in reliability. Few researchers apply the method to predict failures, and first failure time of product (Shan-Yu and Hong-De, 2005). On the basis, this paper put forward to forecasting methods for mean time between failures based on improved PSO neural networks.

The application of algorithm: Neural Network is to train the sample that have already results and then spread the dealing model to research area. Therefore, when we use the model of the sample; we train the sample from training samples from field survey sample points and mean time between failures of product is used as input and output of Neural Network. Prediction model is adopted time series forecasting method and date is from a series of time between failures. Time series is represented by x (t), where t = 0, 1, 2,…. The time series forecasting method forecasts a variable only according to historical data, namely, relations between a variable value and its front series values are the nonlinear functional relationship, that is described by:

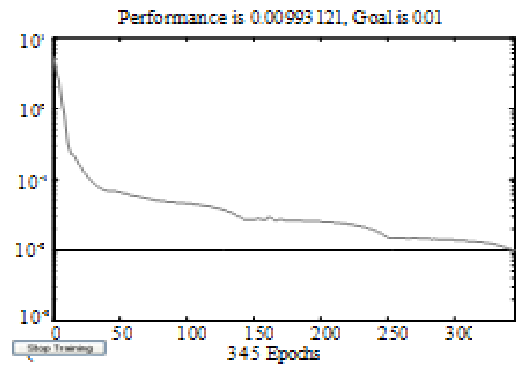
$$x(n+k) = F(x(n), x(n-1), \cdots x(n-m+1))$$

where, $x(n + k)$ is defined by failures time from time n failure to time n+k failure and k = 1, $x(n)$, $x(n\text{-}1)$,… $x(n - m + 1)$ is m historical data before time n. The rolling forecasting method is taken in order to forecast better accurate. According to network complex degree and prediction accuracy, the number of the input node is 8, thus the input node is determined. Three layer feed forward neural networks is used in the model. Front eight mean times between failures are used as input of network and the network forecasts the next mean time between failures. According to determining the principle of hidden mode, the number of hidden nodes is 7 by trial-and-error method and the number of output node is 1 by actual need. After testing by trial-and-error method, we adopt the structure containing the 8 input nodes, 7 hidden layer nodes and 1 output node.

The model is applied to forecasting mean time between failures of vibratory roller of certain Road & bridge company. Forty two sample data is obtained during the period 2000-2004.

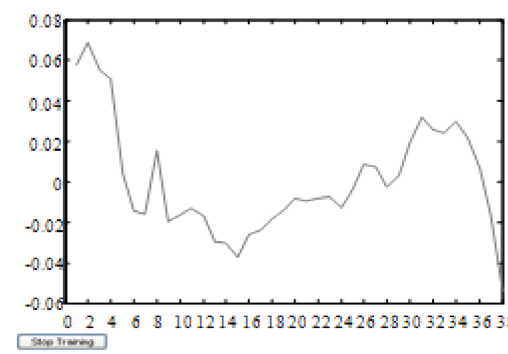


(a) BP algorithm learning curve

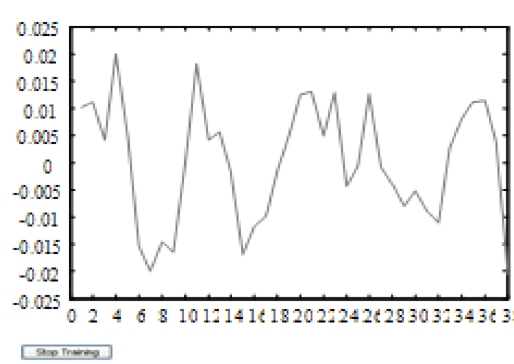


(b) Improved PSO-BP algorithm learning curve

Fig. 2: Simulation result



(a) Training relative error curve of BP neural network



(b) Training relative error curve of IPSO-BP neural network

Fig. 3: Training relative error curve

Table 1: Comparisons of BP neural network and improved PSO-BP neural network forecasting results

Detecting sample	Expected output	BP algorithm output	Improved PSO-BP algorithm output	BP algorithm relative error	Improved PSO-BP algorithm relative error
1	1320.00	1425.60	1326.40	8%	2%
2	1377.00	1448.61	1398.09	5.20%	1.53%
3	1442.00	1476.90	1407.25	2.42%	-2.41%
4	1517.00	1468.46	1544.00	-3.20%	1.78%

RESULT ANALYSIS

We take 38 samples from 42 as training sample, 4 for detecting samples and then we use 2 groups of sample to train and detect the standard BP Neural Network and the BP Neural Network optimized by improved particle swarm, the results are as follows.

The process of concrete network training is as Fig. 2, from Fig. 2a, we can conclude that BP network needs iteration for 5995 times until the error is converged to 0.01 as the Fig. 2b shows, the improved PSO-BP Neural Network only need iteration for 345 times until the error is converged to 0.01. Therefore, compared with standard BP network, the improved PSO-BP neural network increased the convergence speed obviously.

After being trained for 1000 times, the relative error of 38 training samples are showed by Fig. 3 we can conclude from Fig. 3 that, the relative error of samples trained by improved PSO-BP Neural Network is in the range of less 2%. However, the relative error of samples trained by BP Neural Network is in the range of 7%.

Detecting 2 kinds of Neural Network by 4 samples, we show the imitating results are in Table 1, we can conclude that the relative error of imitating results trained by improved PSO-BP Neural Network can be controlled in the range of 2%. The imitating result is the same with real results. However, the relative error in the imitating results of Neural Network is greater than the improve PSO-BP Neural Network.

CONCLUSION

- The improved particle swarm algorithm in this study helps solve the defect of the traditional particle swarm algorithm, which easily falls into local extreme value and has low convergence accuracy. This kind of algorithm optimizes the weight and the threshold value of BP Neural Network and achieves the parameter combination that the optimal particle corresponds in BP Neural Network. Therefore, it optimizes parameter of BP Neural Network and improves the generalization ability and self-study ability of BP Neural Network.
- The method for forecasting reliability of product based on improved PSO-BP Neural Network has many advantages. For instance, it has high prediction precision, good stability, high convergence speed. The relative error of forecasting precision based on improved PSO-BP network is better than that based on BP network. This method has many advantages over traditional BP Neural Network in following aspects. For a start, it can reduce iteration times. Then it has great output stability and high convergence speed and precision of forecasting. It proves that adopting improved PSO-BP Neural Network model for forecasting reliability of product is attainable and practical.
- This study brings up an algorithm that the improved particle swarm optimized Neural Network. It provides a scientific and an effective way for forecasting reliability of product.

REFERENCES

Eberhart, R.C. and J. Kennedy, 1995. A new optimizer using particle swarm theory. Proceeding of the 6th International Symposium on Micro Machine and Human Science, pp: 39-43.

Eberhart, R.C. and Y. Shi, 2001. Particle swarm optimization: Development, applications and resources. Proceeding of Congress on Evolutionary Computation, Piscataway, NJ IEEE Press, pp: 81-86.

Hai-Dong, Z., M. XU-Dong and L. Shi-Pin, 2003. Military plane fault prediction system based on neural network model [J]. Syst. Eng. Electron., 25(7): 894-896.

Hao, P. and H. Xiao-Lei, 2008. BP network training algorithm based on particle swarm optimization. Comp. Eng. Appl., 44(9): 67-69.

Jiang-Hong, H., L. Zheng-Rong and W. Zhen-Chun, 2006. Adaptive particle optimization algorithm and simulation. J. Syst. Simul., 18(10): 2969-2971.

Jie-Ping, C., C. Dao-Cai and L. Li, 2009. Application of BP network model based on PSO for the forecast of drought and flood. J. Shengyang Agric. Univ., 40(1): 118-121.

Jin-Yong, M., H. Chang-Qing, G. Hua and L. Kejie, 2004. Study on fault forecasting of air control equipment based on neural network. Syst. Eng. Electron., 26(7): 992-993.

Kennedy, J. and R.C. Eberhart, 1995. Particle swarm optimization. Proceeding of IEEE International Conference on Neural Networks, Perth, Australia, pp: 1942-1948.

Michael, J.C., E.M. David, J.S. Thomas and A. Malhotra, 1993. Comparison of electronics-reliability assessment approaches. IEEE Trans. Reliabil., 42(4): 542-546.

Shan-Yu, Y. and W. Hong-De, 2005. Analysis on main fan first failure time based on neural network technique. J. China Coal Soc., 30(6): 741-745.

Xiang, L., C. Ji-Feng and X. Jun, 2008. Power demand forecasting based on BP neural network optimized by clonal selection particle swarm. J. Hunan Univ., 35(6): 41-52.

Xiao-Rong, W. and L. Shu-Xian, 2008. Study on BP neural network optimization based on improved PSO algorithm. J. Bohai Univ., 29(2): 189-192.

Xiu-Ye, W. and P. Hong-Xia, 2010. The Improvement of Particle Swarm and Intelligent Trouble Diagnosis. China Defense Press, Beijing, pp: 101-102.

Xuan, H., Z. Jun and Z. Zhi-Hui, 2009. Faster particle swarm optimization with random inertia weight. Comp. Eng. Design, 30(3): 647-650.

Yi-Shan, X., Z. Bi and Y. Xiu-Wen, 2009. BP neural network and its applications based on improved PSO. Comp. Eng. Appl., 45(35): 233-235.

Ying-Jie, L. and H. Chang-Hua, 2005. The performance analysis of missile inertia device fault prediction based on neural networks. Tact. Miss. Technol., 27(2): 52-55.