BP Neural Network Algorithm Optimized by Genetic Algorithm and Its Simulation

Junsheng Jiang

School of Mechatronics and Vehicle Engineering, Weifang University
Weifang, 261061, China

Abstract

Aiming at the drawbacks of slowly converging and easily getting into the local minimum appearing in the BP neural network, this paper combines the general optimization of genetic algorithms together with the local optimization of BP neural networks to improve the performance of BP neural networks. The numerical experiment shows that, compared with the original BP neural network, the optimized BP neural network can effectively reduce the average error of the model calculation and prediction, greatly cut the times of iteration, and raise the calculation accuracy and convergence speed. This paper also demonstrates the ability of genetic algorithms to improve the performance of BP neural networks.

Keywords: *BP neural network, genetic algorithm, optimization*

1. Introduction

In recent years, artificial neural networks have been successfully applied to system identification, pattern identify and control fields. As the neural network theory is still in progress, there is not a set of ways to guide the design process. Now the design of neural network mainly focuses on two sides: one is the optimum structural design of neural networks, the other the algorithm of neural network's weight. At homes and abroad, many researches have been conducted on these two sides and proposed a lot of optimization methods.

The gradient descent method is usually used for training BP networks, whose advantages are fast in local optimization because of only one direction. However, it easily converges to local optimization and then has poor performance on global optimization. The genetic algorithm was put forward by professor Holland in 1975, which is based on biological evolution and genetic variation. The genetic algorithm makes population evolution ceaseless by means of choosing copy and genetic operator and can obtain optimal solutions at last. It has good performance of global optimization because of its searching the optimal solution in the whole solution space, however, at high cost. Some researches indicate that the genetic algorithms can reach the 90 percent of the global optimization quickly, but the convergent speed becomes slow clearly in the last period.

The paper introduces a BP neural network optimized by genetic algorithms and the BP neural network takes advantages of the gradient descent method and genetic algorithms. The simulations demonstrate that the optimized algorithm has faster convergent speed than the original algorithm.

2. Principle of Genetic Algorithm

The genetic algorithm is a random search algorithm that simulates natural selection and evolution. It searches the total solution space and can find the optimal solution globally over a domain. The basic operation of a genetic algorithm is as follows:

(1) Choosing Operation

The choosing operation means to choose the individuals from the old group and add them to the new group in a certain probability. The probability of the individual selected is relevant to the fitness value: the better fitness value of the individual, the greater probability of being chosen.

(2) Crossover Operation

The crossover operation means to select two individuals from the existing individuals and generate a new better individual through the exchange and combination of two chromosomes. The crossover process is to choose two chromosomes from the groups and select randomly one point or more points to be exchanged. The crossover operation is shown in Figure 1.

$$\begin{matrix} A:1100\vdots 01011111 \\ B:1111\vdots 01010000 \end{matrix} \rightarrow \begin{matrix} A:1100\vdots 01010000 \\ B:1111\vdots 01011111 \end{matrix}$$

Fig.1. Crossing Operation

(3) Mutation Operation

The mutation operation is to choose one individual from the group and then select a point of chromosomes for variation operation to generate better individuals. The mutation operation is shown in Figure 2.

$$\text{A}:1100\ \ 0101\ \ 1111 \rightarrow \text{A}:1100\ \ 0101\ \ 1101$$

Fig.2. Variation Operation

3. BP Neural Network Optimized by Genetic Algorithm

3.1 Flowchart of BP Neural Network Optimized by Genetic Algorithm

The BP neural network optimized by the genetic algorithm consists of three parts: network structure determination, genetic algorithm optimization and network prediction. The structure of a BP neural network is determined by the input and output parameters of the fitting function, by which the length of individuals of the genetic algorithm can be determined. The genetic algorithm is exploited to optimize the weights and thresholds of the BP neural network. Each individual of the population consists of all weights and thresholds of a network and the individual fitness value is calculated by the individual fitness function. Subsequently, the genetic algorithm finds the best fitness value corresponding to the individual by means of the selection, crossover, and mutation. Next, the BP neural network obtains the optimal initial weights and threshold provided by the genetic algorithm and predicts the function output after the network has been trained.

3.2 The Procedure of the Genetic Algorithm

The BP Neural network optimized by the genetic algorithm uses the genetic algorithm to optimize the initial weights and thresholds of the BP neural network and makes the optimized BP neural network better predict the output. The BP neural network optimized by genetic algorithm is as follows:

(1) Population Initialization

The individual encoding is real-coded and each individual is a real string which consists of connection weights of the input layer and the hidden layer, the threshold of hidden layer, connection weights of hidden layer and output layer, and the threshold of output layer. The individual contains all weights and thresholds of the neural network. We can form a neural network with the determined structure, weights and thresholds under the conditions of the network structure being known.

(2) Fitness Function Calculation

To begin with, the initial weights and thresholds of the BP neural network are obtained by the individual; then, the output is predicted after the BP neural network has been trained with the training data. Use the sum of the absolute value of error between the prediction output and desired output as the fitness value of individual and the formula is shown as

$$F = k(\sum_{i=1}^{n} abs(y_i - o_i)) \quad (1)$$

In the formula, n is the number of output nodes of the network, y_i is the desired output of the ith node of the neural network, o_i is the prediction output of the ith node and k is the coefficients.

(3) Selection Operation

The genetic algorithm operation has the roulette selection method, the tournament method and so on. This paper adopts the roulette method, which is based on the proportional fitness. The choice probability p_i of each individual i is

$$f_i = k / F_i \quad (2)$$

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \quad (3)$$

where, F_i is the fitness value of the individual i, k is the coefficients and N is the number of individual species.

(4) Crossover Operation

The crossover method uses the real crossover method because the individuals use real-coded. The crossover between the kth chromosome a_k and the lth chromosome a_i is as follows:

$$\left.\begin{aligned} a_{kj} &= a_{kj}(1-b) + a_{lj}b \\ a_{lj} &= a_{lj}(1-b) + a_{kj}b \end{aligned}\right\} \quad (4)$$

where, b is a random number between zero and one.

(5) Mutation Operation

The jth gene a_{ij} of the ith individual is selected to mutate. The mutation operation is as follows:

$$a_{ij} = \begin{cases} a_{ij} + (a_{ij} - a_{\max}) * f(g) & r \geq 0.5 \\ a_{ij} + (a_{\min} - a_{ij}) * f(g) & r < 0.5 \end{cases} \quad (5)$$

$$f(g) = r_2(1 - g / G_{\max})$$

where, a_{max} and a_{min} are the upper and the lower bounds of the gene a_{ij}, respectively, r_2 is a random number, g is the current iteration number, G_{max} is the maximum number of evolution and r is a random number between zero and one.

4. Simulation example

The approximation function which has been used widely in the neural network was here used to verify the feasibility and effectiveness of the optimized algorithm. The function is as follows:

$$f(x) = \frac{\sin(8x)}{(x-0.25)^2 + 0.1} \quad (6)$$

Using each of the gradient descent algorithm and the generic method, only low approximation accuracy can be derived. If we further raise the approximation accuracy, then the gradient descent method is easy to fall into local minima. In addition, the genetic algorithms can quickly converge to the field of minimal errors, but it is difficult to achieve the final convergence. In this paper, the BP neural network and the genetic algorithm were together used to approximate the function. The best individual fitness value obtained by the optimized BP neural network is shown in Figure 3. Comparisons between the network prediction errors of the original and the optimized networks are shown in Figure 4. As seen in Figure 4, the BP network optimized by the genetic algorithm has more prediction accurate than the original network, since the root-mean-square of the optimized network is 5.2284×10^{-5} and that of the original BP network is 2.2356×10^{-4}. Thus, compared with the original network, the optimized network delivers better performance.

5. Conclusions

The original BP neural network is easy to fall into local minimum. To overcome the shortcoming, the paper combined the genetic algorithms with the gradient descent. The proposed algorithm in this paper can fully take the advantages of two algorithms, effectively avoid the local minimum and be able to achieve the desired weights of the BP network quickly. The simulation shows that the proposed algorithm can effectively solve the local minimum problem in the neural network training, and the mean square error of the prediction can greatly reduced.

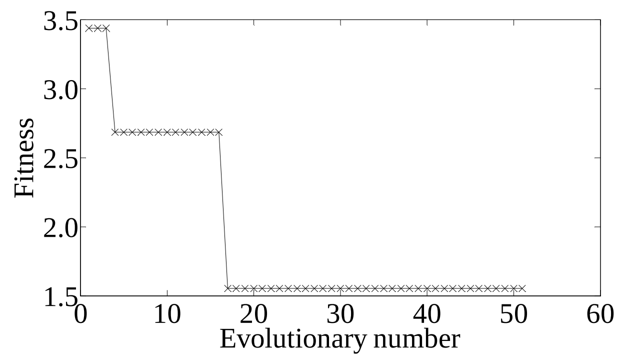

Fig. 3 The best individual fitness value

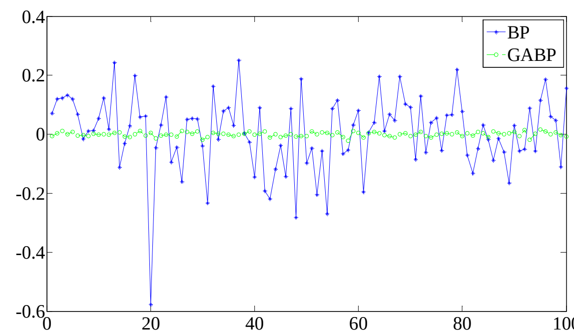

Fig. 4 Comparisons between forecasting errors of the un-optimized and optimized networks

References

[1] Wang Xiaoping, Cao Liming, Genetic Algorithm-Theory, Application and Realization. XI'AN: Xi'An JiaoTong University Press, 2002.

[2] LI Mu, HE Yi-gang, ZHOU Shao-wu, TAN Wen, "Hybrid Genetic Neural Network Method for Predicting Chaotic Time Series", Journal of System Simulation, vol.20, no.21, 2008,pp.5824-5828.

[3] WU Ling, JIANG Zhou-hua, GONG Wei, LI Yang, "GA-NN-Based Predicting Model of Activity of Multiple Slag System", Journal of Northeastern Universty (Natural Science), vol.29, no.12, 2008, pp.1725-1728.

[4] Xie Hongmei, Liao Xiaoping, Lu Yuhai, "GA-BP Algorithm and Its Application in the Forecasting of Part Quality" , China Mechanical Engineering, vol. 19, no. 22, 2008, pp.2711-2714.

[5] Seyedyaser Bozorgirad, Mohammad Ishak Desa and Antoni Wibowo ,"Genetic Algorithm Enhancement to Solve Multi Source Multi Product Flexible MultistageLogistics Network", International Journal of Computer Science Issues, Vol. 9, Issue 3, No 2, May 2012 pp.157-164.

[6] Sayedmohammadreza Vaghefinezhad and Kuan Yew Wong", A Genetic Algorithm Approach for Solving a Flexible Job ShopScheduling Problem" International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1, May 2012 pp.85-90.

[7] V. Sai Shanmuga Raja and Dr. S.P. Rajagopalan ,"A Neuro-Genetic System for Face Recognition", International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1, May 2012 pp.263-267.