TELKOMNIKA, Vol. 11, No. 10, October 2013, pp. 5588 ~ 5593
ISSN: 2302-4046

A Novel Method to Optimize the Structure of BP Neural Networks

Changming Qiao, Shuli Sun*
Institute of Electronic Engineering, Hei Longjiang University (XU)
Harbin, 150080, China, Ph.: +86-13796204098
Corresponding author, e-mail: hlju501@126.com

Abstract

It has been a long time that there is not a so good method to determine the number of neurons in hidden layer for BP neural network. For this problem, a novel algorithm based on Akaike Information Criterion (AIC) to optimize the structure of the BP neuron networks is proposed in this paper. At the same time, this paper gives the upper and lower bounds for classical AIC to overcome its shortcomings. The simulation experiment shows that this method can select a more suitable network structure, and can ensure the minimal output error with the optimal structure of the network.

Keywords: *BP neuron networks, AIC, network structure*

1. Introduction

The algorithm of neural networks has been proposed for many years and has achieved quite good results in many fields, such as, artificial intelligence, information fusion, pattern recognition, fault diagnosis, intelligent control and so on [1-3]. Among all the algorithms of neural networks until now, BP neural networks (BPNN) is applied more frequently and widespread. It is a kind of typical multi-layer feed forward neural networks, and can solve many difficult problems with very complex nonlinear. However, some shortcomings for classical BPNN algorithm have been found in the long-term use of it. One of them is can not use an effective method to determine the number of neurons in the hidden layer, in many practical applications, the method of “trial and error” or “empirical formula” is still used. For this problem, many experts and scholars have studied some different solutions.

Reference [4] proposed that to determine a range for the number of neurons in the hidden layer with the empirical formula first, and then expand the range and to find the optimal value within it. But the essence of this method is still the trial and error, so its practical value is not widespread. Reference [5] proposed an adaptive merging and growing algorithm (AMGA), it is the combination of the genetic algorithm and growth algorithm. Although this algorithm has a certain value, the problems of does not know when to terminate the algorithm and high computational complexity are very obvious, especially when dealing with large scale classification problems. Reference [6-8] proposed an algorithm based on Agent. This algorithm is relatively good and can be used widely, but the computational complexity of this algorithm is so high, not suitable for the hardware implementation of the neural networks. Reference [9] proposed a method to determine the scale of hidden layer for the single hidden layer binary neural networks. The theory of this method is so rigorous, and has high theoretical significance for practical applications. But in the end, the paper also pointed out that the upper bound determined by this method whether is the certainly upper bound needs to be discussed and proved.

For this problem, this paper proposed to use AIC which is used for determine model order in system identification theory to optimize the number of neurons in hidden layer for BPNN. Moreover, this paper detailed analyses the shortcomings of AIC and gives the solution. The simulation experiment shows that this method can select the most suitable number of neurons in hidden layer quickly, and the mean square error (MSE) of the entire network output is minimal. At the same time, it is easy to implement and has low computational complexity.

Received April 18, 2013; Revised June 23, 2013; Accepted July 10, 2013

2. BPNN and Its Defects

In 1974, P. Werbos proposed a learning algorithm suitable for multi-layer networks in his doctoral thesis [15]; Later in 1986, the U.S. PDP team studied the algorithm deeply and proposed BP algorithm, so the neural networks which trained by this algorithm is named BPNN. The typical BPNN has a network topology with 3 layers [10], including input layer, hidden layer and output layer, as shown in Figure 1. Where, $x_1, x_2, \cdots, x_n$ are network inputs, $y_1, y_2, \cdots, y_l$ are network outputs, W_{ij} and V_{ki} are connection weights. Generally, the BP algorithm includes four steps: the forward spread of the samples, the calculation of the output error, the back spread of the error and the adjustment of the weights and thresholds.

The hidden layer of BPNN can be considered as an internal interpretation for the input layer. It is mainly used to extract the characteristics between a kind of input mode and other input modes, and pass them to the output layer. This process can be seen as the process of weights adaptive adjustment. According to the Kolmogorov theorem, if a neural network with 3 layers is consist of the Sigmoid-type neurons and the number of neurons in the hidden layer is enough, this neural network can complete an arbitrary nonlinear mapping with arbitrary accuracy. But it does not mean that the more the better, the performance of a neural network is usually evaluated by calculating the output error of test samples. Barroll considers that the error comes from two aspects [11]: the approximation error (bias) and estimation error (variance). When the number of neurons in hidden layer increases, the approximation error will decrease gradually; but the estimation error will gradually increase simultaneously, so it needs a balance between them. If the number of neurons in the hidden layer is excessive, it is possible for the network to train noise and other redundant information included in the data. It will result in over-training and the network will fall into local minimum points with a high probability. On the contrary, if the number of neurons in the hidden layer is too little, the accuracy of the network output is low, cannot reflect the nonlinear relationship between the input and output data. Therefore, how to determine an appropriate number of neurons in the hidden layer is the key to construct effective BPNN.

3. AIC and Its Defects

The overview of AIC: In 1973, Japanese scholar Akaike proposed a selection criterion for statistical model called Akaike Information Criterion (AIC) [14-15], as shown in formula (1).

$$AIC(k) = -2\ln^{(L)} + 2k \quad (1)$$

Where, *k* is the number of parameters that can be adjusted independently in the model, reflecting the complexity of the model; *L* is the maximum of model likelihood function, reflecting the fitting accuracy. The process of this algorithm is: first estimates the parameters with the method of maximum likelihood, and then calculates the value of the likelihood function and AIC(k), the model will be the best fitted model when the value of AIC(k) is minimal. AIC will give a suitable model through make a balance between fitting accuracy and model complexity. Specifically, assume the real number of adjustable parameters in the system is n_0, calculates the value of AIC (k) from $k = 1$. In the beginning, *k* is far less than n_0, the fitting accuracy must be poor, so the value of the former part $-2\ln^{(L)}$ in the formula (1) is greater and playing a leading role; with the increasing of *k*, the value of $-2\ln^{(L)}$ gradually decreases when *k* close to the n_0, while the latter part 2*k* gradually increases and playing the leading role. Therefore, a minimal point appears at n_0.

The analysis of defects: Firstly, the classical AIC focuses on the effects caused by residuals and model order just from the mathematical aspect, lack the physical understanding of the model. Therefore, it may lead to cannot identify all modes during the modes analysis. Moreover, the AIC has no lower bound of model order, so it is so possible that the model order given by the AIC is less than the real value in the application.

Secondly, although there is an upper bound of the model order in practical application generally, it may be ∞ in some extreme cases. If so, the calculate process of the AIC will endlessly, that means the AIC will not converge.

4. Derivation and Improvement

Theoretical derivation: It can be seen from the formula (1) that the maximum of model likelihood function *L* is the only value needs to be calculated when the AIC is applied to the BPNN. Generally, assume the parameter needs to be adjusted in a neural network model is $\theta \in R^m$, including all connection weights and thresholds, input variables $X \in R^k$ and output variables $Y \in R^l$. So a neural network model can be expressed as:

$$\left\{ Y = f(X,\theta); \theta \in R^m, X \in R^k, Y \in R^l \right\} \tag{2}$$

Assume the training sample set of the network is $\left\{ (x_i, y_i), i = 1,2,\cdots,N \right\}$, where *N* is the number of training samples; $x_i = \left\{ x_i^{(1)}, x_i^{(2)} \cdots x_i^{(P)} \right\}$ is the input vector, *P* is the number of neurons in the input layer; $y_i = \left\{ y_i^{(1)}, y_i^{(2)} \cdots y_i^{(K)} \right\}$ is the output vector, *K* is the number of neurons in the output layer. Also, assume the response function of output neurons is Sigmoid. After fully trained, when the ith sample is input, if the actual input and the ideal input of the kth output neuron are $c_i^{(k)}$ and $d_i^{(k)}$ respectively, the input error of this output neuron is:

$$\varepsilon_i^{(k)} = c_i^{(k)} - d_i^{(k)} \tag{3}$$

It can be known from the central limit theorem that error $\varepsilon_i^{(k)}$ can be seen as independent random variables, and its probability distribution obey normal distribution, that is $\varepsilon_i^{(k)} \sim N(\mu_k, \sigma_k^2)$, where,

$$\mu_k = \frac{1}{N}\sum_{i=1}^{N} \varepsilon_i^{(k)}, \sigma_k^2 = \frac{1}{N}\sum_{i=1}^{N} (\varepsilon_i^{(k)} - \mu_k)^2 \tag{4}$$

The conditional probability density function of $d_i^{(k)}$ is [14]:

$$f(d_i^{(k)} \mid x_i^{(k)}, \theta) = f(\varepsilon_i^{(k)} \mid \theta) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\left(\frac{\varepsilon_i^{(k)} - \mu_k}{\sqrt{2\pi}\sigma_k} \right)^2 \right) \tag{5}$$

According to the Sigmoid transfer function, we can get the conditional probability density of output neurons:

$$f(a_i^{(k)} \mid x_i^{(k)}, \theta) = \left[\frac{1 - a_i^{(k)}}{a_i^{(k)}} a_i^{(k)^2} \sqrt{2\pi}\sigma_k \right]^{-1} \exp\left(-\left(\frac{\ln\left(\frac{1 - a_i^{(k)}}{a_i^{(k)}} \right) - \mu_k}{\sqrt{2\pi}\sigma_k} \right)^2 \right) \tag{6}$$

where, $a_i^{(k)}$ is the actual output of the kth output neuron of the ith sample. Assume the network output error is:

$$r_i^{(k)} = y_i^{(k)} - a_i^{(k)} \tag{7}$$

then:

$$f(r_i^{(k)} \mid x_i^{(k)}, \theta) = f(a_i^{(k)} \mid x_i^{(k)}, \theta) \tag{8}$$

the maximal likelihood function is:

$$L = \prod_{i=1}^{N} \prod_{k=1}^{K} f(r_i^{(k)} \mid x_i^{(k)}, \theta) \tag{9}$$

Take the formula (9) into the formula (1):

$$AIC\_BP = -2\ln(\prod_{i=1}^{N} \prod_{k=1}^{K} f(r_i^{(k)} \mid x_i^{(k)}, \theta)) + (P + K + 1)M \tag{10}$$

where, *P*, *M* and *K* are the number of neurons in the input, hidden and output layer respectively.

The improved method: Consider the shortcomings of AIC said in section 3.2, that is, the parameter *M* in the formula (10) lack of lower bound and its upper bound is not clear. Using the following formula to determine the lower bound of *M*:

$$M_{\min} = \log_2^P \tag{11}$$

Generally, the parameter *M* has an upper bound, but in some extreme cases, especially when the determinate lower bound is not so good, it is easy to lead to calculate endlessly. In fact, it does not need to execute to the real upper bound of the model in the calculation process, only needs a stage upper bound of *M* and given by the following formula:

$$M_{\max} = \sqrt{P + K} + a \tag{12}$$

where: *a* is constant between 0~10.

The implementation steps: According to the method described above, the implementation steps as follows:

Step 1: Determine the *P* and *K* according to the practical application of the system;

Step 2: Calculate the initial lower bound $M_{\min}$ and upper bound $M_{\max}$ of the *M*;

Step 3: Circulating train the network and calculate the AIC_BP values in the range of $M_{\min}$~$M_{\max}$;

Step 4: If the minimal point of AIC_BP does not appear, set $M_{\min}=M_{\max}$, $M_{\max}=2M_{\max}$, then go to Step 2; Otherwise, training stopped.

5. Experiment Analysis

The experiment is to complete a complex nonlinear function regression. The function is:

$$y = 20 + x_1^2 - 10\cos(2\pi x_1) + x_2^2 - 10\cos(2\pi x_2) \tag{13}$$

1000 groups of input data randomly generated firstly, where 800 groups as training samples, 200 groups as testing samples. Then, tests the algorithm 3 times with 10%, 20% and 30% noise respectively, the noise is added randomly. According to formula (13), it is sure that there are 2 nodes in the input layer, and 1 node in the output layer. We can also calculate the initial lower bound ($M_{\min}$) and upper bound ($M_{\max}$) of neurons in the hidden layer are 1 and 5 respectively, when the value of a is 3. Moreover, trains the network following the algorithm in section 4.3, calculates the value of AIC_BP and output MSE, the results as shown in Figure 1 and 2.

It can be seen from Figure 1 that the minimal value of AIC_BP appears when the number of neurons are 36, 44 and 54 respectively. Moreover, with the increasing of neurons in the hidden layer, the AIC_BP decreases first and then increases. When the percentage of noise increases, the required neurons also increase for the best structure of BPNN. The results clearly show that the algorithm proposed in this paper is correct. At the same time, the output MSE shows the same trend with AIC_BP, as shown in Figure 2. Also, the minimal value of MSE appears at the same moment with AIC_BP. At last, tests the BPNN with the structure of 2x36x1.

Figure 3 shows that the fitting result is very good, almost coincides with the real image of the function, there are a little errors on the edge of the image. Thus, the nonlinear mapping ability of the network is very good with this structure.

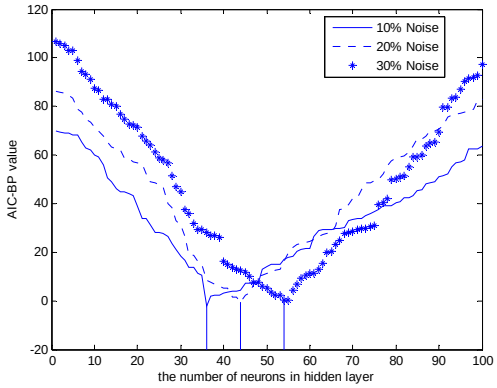


Figure 1. The change trend of AIC_BP

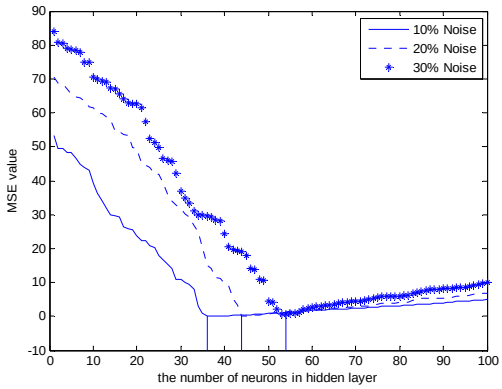
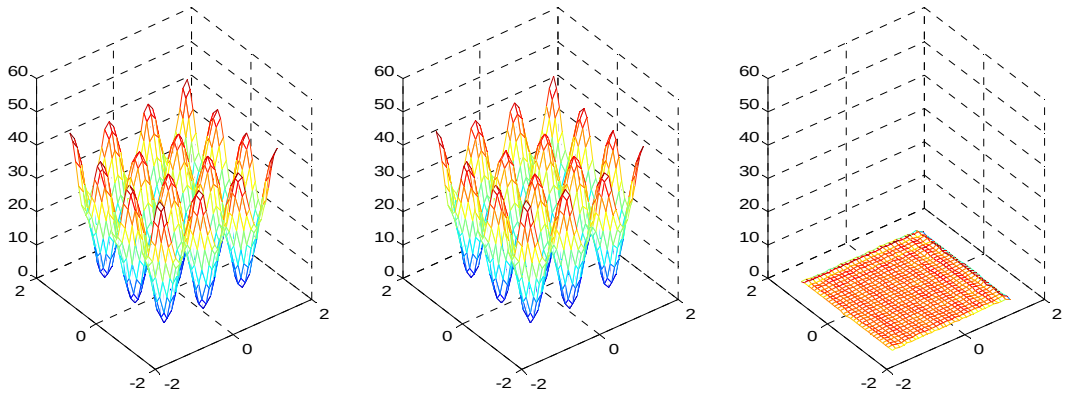


Figure 2. The change trend of output MSE



(a) Real function image (b) BP neural network fitting image (c) Error image

Figure 3. Fitting results and error

6. Conclusion

There is a problem that the number of neurons in the hidden layer can only be determined empirically when using the BP neural network, this paper by means of the AIC criterion for model order in the information theory, its upper and lower bounds are also given, then the optimal selection method of BP neural network structure based on improving AIC criterion is proposed. The simulation result shows that we can select the optimal model structure suitable for the practical problems with this method, and get very satisfied output results with this structure. But it is very important to determine the initial lower bound of AIC criterion, otherwise the calculation may not converge, if so, it should be properly reduce the initial lower bound calculated from the formula (11) and then start to select network structure.

Acknowledgements

This work was supported in part by Natural Science Foundation of China under Grant NSFC-60874062, Program for High-qualified Talents under Grant Hdtd2010-03, and Electronic Engineering Province Key Laboratory.

References

[1] H Zhao, J Zhang. Pipelined Chebyshev Functional Link Artificial Recurrent Neural Network for Nonlinear Adaptive Filter. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*. 2010; 40: 162-172.

[2] Omer Deperlioglu, Utku Kose. An educational tool for artificial neural networks. *Computers and Electrical Engineering*. 2011; 37: 392-402.

[3] Abe T, Saito T. *An approach to prediction of spatio-temporal patterns based on binary neural networks and cellular automata*. IEEE International Joint Conference on Neural Networks. 2008; 2494-2499.

[4] Yan hong, Guan Yan-ping. Method to Determine the Quantity of Internal Nodes of Back Propagation Neural Networks and Its Demonstration. *Control Engineering of China*. 2009; 16(S1): 100-102.

[5] Islam MM, Sattar MA, Amin F, et al. A New Adaptive Merging and Growing Algorithm for Designing Artificial Neural Networks. *IEEE Trans on Systems, Man, and Cybernetics—Part B: Cybernetics*. 2009; 39(3): 705-718.

[6] Gao Peng-yi, Chen Chuan-bo, Qin sheng. A Novel Algorithm to Optimize the Hidden Layer of Neural Networks. *Computer Engineering & Science, China*. 2010; 32(5): 30-33.

[7] YU Zhijun. RBF Neural Networks Optimization Algorithm and Application on Tax Forecasting. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013. 11(7).

[8] Patricia Melin, Victor Herrera, Danniela Romero, Fevrier Valdez, Oscar Castillo. Genetic Optimization of Neural Networks for Person Recognition based on the Iris. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(2): 309-320.

[9] Lu yang, Yang juan, Wang qiang. The Upper Bound of The Minimal Number of Hidden Neurons for The Parity Problem in Binary Neural Networks. *Science China*. 2012; 42(3): 352-361.

[10] Li ying Wang zheng, Ao Zhi-guang. Optimization for Breakout Prediction System of BP Neural Network. *Control and Decision, China*. 2010; 25(3): 453-456.

[11] Barron AR. Approxmation and estimation bounds for artificial neural networks. *Machine Learning*. 1994; 14: 115-133.

[12] Hannane J. The Estimation of the Order of an ARMA Process. *The Annals of Statistics*. 1980; 8(5): 1071-1081.

[13] Stoica P, Selen Y. Model-order selection: a review of information criterion rules. *IEEE Signal Processing Magazine*. 2004; 21: 36-47.

[14] Xie Xiao-heng, He You-hua. On the Error of Kernel Estimation for Conditional PDF and Its Optimal Bandwidth Selection. *Or Transactions, China*. 2008; 12(3): 13-22.

[15] Werbos PJ. Beyond regression: New tools for prediction and analysis in the behavioral sciences. [PhD thesis]. Cambridge (MA): Harvard University, 1974.