Electric Power Load Forecasting of Babol City Based on BP Neural Network

Seyed Ahmad Sheibat Alhamdy, Amir Pourghassem, Morteza Gholam Ahmadi, Milad Padidarfard

Department of industrial management, Firoozkooh branch , Islamic Azad University, Firoozkooh, Iran
sheibat@yahoo.com

Abstract: Modeling and predicting electricity consumption play a vital role both in developed and developing countries for policy makers and related organizations. Improve load forecasting technology level is not only beneficial to plan power management and make reasonable construction plan, but also good for saving energy and reducing power cost, and then, it can improve the economic benefits and social benefit for power system. BP neural network is one of the most widely used neural networks and it has many advantages in the power load forecasting. Matlab has become the best technology application software which has been internationally recognized, the software has many characteristics, such as data visualization function and neural network toolbox, for these, it is the essential software when we do some research on neural network.
[Seyed Ahmad Sheibat Alhamdy, Amir Pourghassem, Morteza Gholam Ahmadi, Milad Padidarfard. **Electric Power Load Forecasting of Babol City Based on BP Neural Network.** *Life Sci J* 2013;10(2):950-953] (ISSN:1097-8135). http://www.lifesciencesite.com. 133

Keywords: Electric power load, Matlab, BP neural network, forecast.

1. Introduction

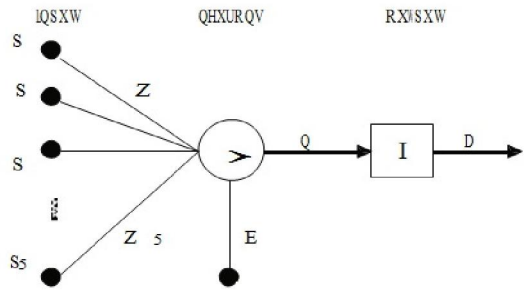
Along with the rapid development of modern science and technology various load forecasting methods has been put out. It is a well-known fact that Artificial Neural Networks (ANN) can model any nonlinear relationship to an arbitrary degree of accuracy by adjusting the network parameters. It is also better to use models that can handle nonlinearities among variables as the expected nature of the consumption data of electric power. Using BP neural network technology for power load forecasting is a new research method, it can imitate human intelligence, adapt itself for large non- structure and the law, besides, it can get the information by autonomous learning, memory, reasoning and optimization calculation, and Matlab can realize the BP neural network for forecasting.

2. Material and Methods

The BP neural network is Back Propagation neural network, the signal is forward but the error is spread propagation, it is the most sophisticated neural network and is the most widely used. BP network is a multilayer feed forward neural network, when the transfer function of neurons is 'S' function, the output of entire network is limited in a small range, but if the transfer function of neurons is 'purelin' function, the output can take any value[1]. Fig. 1 gives a basic BP neurons model, the number of input is R, all inputs are connected by proper weights called w, and the output of network can be expressed as:

$$a= f(w\times p+b) \quad (1)$$

'f' is the transfer function which connect input and output, the BP neurons model is as follows:



$$a= f(w\times p+b)$$

Figure 1. BP neurons model

2.1. Number of Network Layer

In fact, increase the number of hidden layers can improve the nonlinear mapping capacity of BP network, but if the number of hidden layers above a certain value, it will increase training time of network. Therefore, in applying the BP neural network for predicting, select a network which has one hidden layer is enough.

2.2. Number of Neurons in Each Network Layer

The node number of input and output is closely related with the sample, according to the historical data, the number of neurons in the input is 4 and the number of neurons in the output is 1. In reality, we can get the number of neurons in hidden layer according to the empirical formula, the common

empirical formula is:

$$i = \sqrt{(n+m)} + a \qquad (2)$$

'i' is the number of neurons in hidden layer, 'n' is neurons number of input layer, 'm' is neurons number of output layer, 'a' is constant and 1<a<10. Thus, we can set the number of neurons of hidden layer in this neural network for 11.

3. Results

The learning of BP neural network is to be supervised, it need to provide input vector 'p' and expectations 't', the weights and deviation in training process are adjusted according to corresponding network performance, and finally we can achieve the expected function. The default performance function of network is average variance in forward neural network, and the goal of learning process is how to make average error minimize [2,3,4].

3.1. The Weights Adjust between Hidden Layer J and Output Layer P

In BP algorithm, the adjustment and output of weights are relative to the partial differential, the partial differential is[5]:

$$\frac{\partial E(n)}{\partial W_{jp}(n)} = -e_{kp}(n) - f(u_p^p(n)) - V_j^l(n) \qquad (3)$$

Define the local gradient as:

$$\delta_p^p(n) = -\frac{\partial E(n)}{\partial u_p^p(n)} = e_{kp}(n) - f(u_p^p(n)) \qquad (4)$$

The iteration value of wjp(n) between hidden layer J and output layer P is:

$$W_{ip}(n+1) = W_{ip}(n) + \Delta w_{ip}(n) \qquad (5)$$

3.2. The Weights Adjust between Hidden Layer I and Hidden Layer J

It is similar with above, the weights adjust between hidden layer I and hidden layer J
is also along with the descent direction of gradient, and the adjust value is:

$$\Delta W_{ij}(n) = \eta \frac{\partial E(n)}{\partial W_{ij}(n)} = \eta \delta_j^l(n) - V_j^I \qquad (6)$$

The local gradient is:

$$\delta_j^l(n) = \frac{\partial E(n)}{\partial u_j^l(n)} = -\frac{\partial E(n)}{\partial V_j^l(n)} \cdot \frac{\partial V_j^l(n)}{\partial u_j^l(n)} = -\frac{\partial E(n)}{\partial V_j^l(n)} \cdot f(u_j^i(n)) \qquad (7)$$

The iteration value between hidden layer I and hidden layer J is:

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) \qquad (8)$$

3.3. The Weights Adjust between Hidden Layer I and Hidden Layer J

It is similar with above, the adjust value between any two nodes from two different layers is:

$$\Delta w_{mi} = \eta \delta_i^l(n) - x_{\mathrm{Im}}(n) \qquad (9)$$

The local gradient is:

$$\delta_i^I = f(u_i^I(n)) - \sum_{j=1}^{l} \delta_j^l(n) w_{in}(n) \qquad (10)$$

4. Discussions

4.1. Sample Data Processing

The number of node in input layer, output layer and hidden layer depend on the complexity of the object. The sample data is the quantity of electric power load in Babol city between 2008 until 2012, that each year has been divided into six time interval that because of difficulty in data gathering we could get 4 time interval of each year but for year 2012 we have 5 time period that they are shown in below figure:

Table 1. The domestic electric power consumption between 2009 until 2012 in BABOL city

Time priode	Quantity(MWH)	Month	Quantity(MWH)
2009,1	3.6825	2011,2	2.8068
2009,2	2.6774	2011,3	4.8969
2009,3	5.0120	2011,4	3.2887
2009,4	2.7364	2012,1	3.6173
2010,1	3.4834	2012,2	3.4518
2010,2	3.8609	2012,3	5.1535
2010,3	6.3699	2012,4	2.3356
2010,4	4.4994	2012,5	3.4581
2011,1	3.0120		

We input the quantity of four months, with these four data, we can get the quantity of fifth month, we also take the quantity of 2010.04 and 2010.05 as the test samples, it is as follows:

Table 2. Sample data

training sample	Input	Output
1	2009.01-2009.04	2009.05
2	2009.02-2009.05	2009.06
3	2009.03-2009.06	2009.07
4	2009.04-2009.07	2009.08
5	2009.05-2009.08	2009.09
6	2009.06-2009.09	2009.10
7	2009.07-2009.10	2009.11
8	2009.08-2009.11	2009.12
9	2009.09-2009.12	2010.01
10	2009.10-2010.01	2010.02
11	2009.11-2010.02	2010.03
test sample		
1	2009.12-2010.03	2010.04
2	2010.01-2010.04	2010.05

4.2. Model Building

Input the command in command window of Matlab:

```
p0(:,1:44)=0
p0(1,1:17)=[3.6825 2.6774 5.0120 2.7364 3.4834
3.8609 6.3699 4.4994 3.0120 2.8068 4.8969 3.2887
3.6173 3.4518 5.1535 2.3356 3.5 ]
for j=1:3
for i=(j-1)*11+1:j*11
p(:,i)=[p0(i) p0(i+1) p0(i+2) p0(i+3)]';
t(i)=p0(i+4);
end
net=newff(minmax(p),[11,1],{'logsig','purelin'},'trainl
m'),
net.trainparam.show=1000 ,
net.trainparam.epoch=2000,
net.trainparam.goal=1e-4,
[net,tr]=train(net,p,t);
```

We can get the training diagram of neural network as follows (Figure 2):

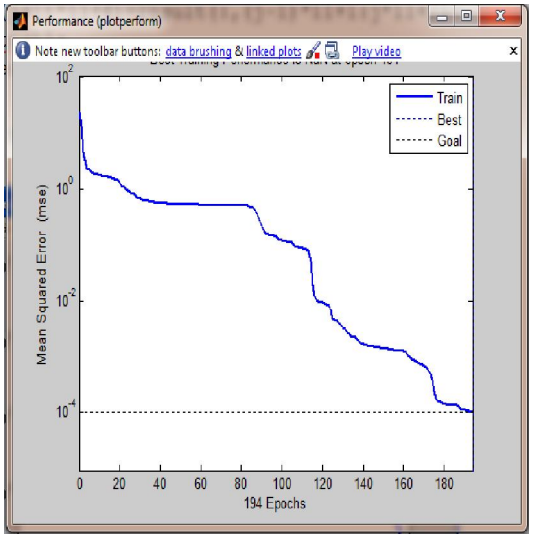


Figure 2. Training diagram of neural network

5.3. Simulate the Network

Input the command:

```
ptest(:,1)=[p0(j*11+1) p0(j*11+2) p0(j*11+3)
p0(j*11+4)];
 ptest(:,2)=[p0(j*11+2) p0(j*11+3) p0(j*11+4)
p0(j*11+5)];
ttest(1)=p0(j*11+5);
 ttest(2)=p0(j*11+6);
 result_test=sim(net,p)
 result_test1=sim(net,ptest);
 result=[result_test result_test1];
p0(:,j*11+3:(j+1)*11+4)=result(:,(j-
1)*11+1:j*11+2);
 ttest2=[t ttest];
 result2=[ttest2' result' (result-ttest2)']
end
```

Among results there exist 3 columns, in the first column we have two types of data,the First one is expected output and in the second one new data is generated by software from previous one for new prediction , In the second column we have two types of data the first one is predicted output from actual data and the second one is predicted output from software simulated data. The third column is absolute errors, use these data, we can calculate the relative error, they are 0.0007%, 0.0003%, -0.0025%, -0.0002%, 0.0013%, 0.0012%, 0.0218%, 0.0071%, -0.0025%, 0.0082%, -0.0255%, -4.1444%, -8.3207%. It can be seen that the error of training sample is very small, but the error of test sample is 10% above, obviously if we want to use this network extrapolation to get more accurate prediction, we must get a large number of samples.

4.4 Drawing Shows

The expected output and actual output are shown below (Figure 3):

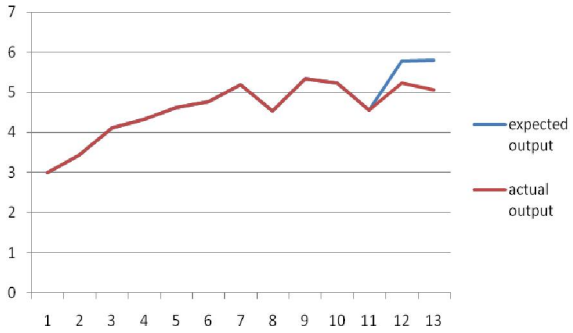


Figure 3. Contrast diagram between expected output and the actual output

Table 3. The expected output, forecasting results and forecasting error.

Real Data	Predicted Data	Prediction Tolerance
3.4834	3.4841	0.0007
3.8609	3.8612	0.0003
6.3699	6.3674	-0.0025
4.4994	4.4992	-0.0002
3.0120	3.0133	0.0013
2.8068	2.8080	0.0012
4.8969	4.9187	0.0218
3.2887	3.2958	0.0071
3.6173	3.6148	-0.0025
3.4518	3.4600	0.0082
5.1535	5.1280	-0.0255
6.3676	6.3699	0.0023
4.4915	4.4931	0.0015
3.0156	3.0141	-0.0015
2.8068	2.8066	-0.0002
4.8960	4.8824	-0.0137
3.2966	3.3015	0.0049
Predicted Data(assumed)	New Predicted Data	Prediction Tolerance
3.6122	3,6110	-0.0012
3.4492	3,4703	0.0212
5.1525	5,1615	0.0090
3.7403	3,7470	0.0067
3.0020	3,0008	-0.0012
3.0208	3,0211	0.0003
2.8163	2,8173	0.0010
4.9161	4,9075	-0.0086
3.2870	3,2861	-0.0009
3.6201	3.6237	0.0036
3.4671	3.4384	-0.0287
5.1608	5.1782	0.0174
3.7365	3.7340	-0.0025
2.9953	2.9964	0.0011
6.5385	6.5374	-0.0011

1/15/2013

In the practical application of artificial neural network, BP neural network is the most popular. The neural network toolbox of Matlab can set learners free from the complex programming work.

Corresponding Author:
Dr. seyed ahmad sheibat alhamdy
Department of industrial management
Firoozkooh branch ,Islamic Azad University, Firoozkooh, Iran E-mail: sheibat@yahoo.com

References

1. Deng, J.L.: Control problems of grey systems. Syst. Control Lett. 1, 288–294 (1982).
2. Yao, A.W.L., Chi, S.C., Chen, C.: Development of an integrated grey fuzzy-based electricity management system for enterprises. Energy 30, 2759–2771(2005).
3. Ranjan, M., Jain, V.K.: Modeling of electrical energy consumption in Delhi. Energy 24, 351–361.
4. Hsu, C.C., Chen, C.: Applications of improved grey prediction model power demand forecasting. Energy Convers Manage 44, 2241–2249 (2003).
5. Chang, N.B., Tseng, C.: Optimal evaluation of expansion alternatives for existing air quality monitoring network by grey compromise programming. Environ Manage 56, 61–67 (1999).