



Research on the Prediction of Logistics Demand for Emergencies Based on BP Neural Network

Kang Ming, Zhao Ying, and Zhang Jing^(✉)

Beijing University of Posts and Telecommunications, Beijing 100876, China
zhangjing@bupt.edu.cn

Abstract. Logistics demand forecasting is a prerequisite and an important part of logistics system planning and optimization, especially in emergencies, where short-term, massive and multi-discipline material demands put forward extremely high requirements on the guarantee capacity of logistics systems. In this paper, a logistics demand prediction model based on time series is constructed for the logistics demand characteristics of emergency events. Since the BP neural network method has the advantages of non-linear mapping capability, self-learning and self-adaptive capability, the BP neural network method is used to solve the model, and finally the model is verified and improved by practical cases. The results show that the model and method used in this study can better predict the logistics demand under unexpected events, which meets the need for rapid prediction of logistics demand in the early stage of unexpected events and is of great significance to improve the efficiency of logistics under unexpected events.

Keywords: Logistics · Demand forecasting · Time series · BP neural network

1 Introduction

There are four broad categories of emergencies in China: natural disasters, accidents, public health and social security incidents. The impact on society and families of economic and casualty losses caused by major emergencies is difficult to estimate, such as SARS in 2002 and COVID-19 that continues to this day last year. Logistics demand forecast is a premise and an important part of logistics system planning and optimization, especially in emergencies, short-term, a large number of real-time rescue needs, logistics demand forecast accuracy and timeliness put forward higher requirements. Traditional logistics demand forecasting methods require high data integrity and quality, complex model structure and poor timeliness, which make it difficult to meet the needs of rapid logistics demand forecasting in the early stage of emergencies. Therefore, this paper constructs a time series-based logistics demand forecasting model for the logistics demand characteristics of emergencies, solves it using BP neural network method and conducts case validation, which is important for improving the response efficiency of logistics system under emergencies and reducing disaster losses.

2 Literature Review

Liu Hui [1] Science and Technology Management Research used BP neural networks to price the IPO in order to overcome the disadvantages of existing valuation methods such as insufficient information and subjective pricing process. This is because compared with the traditional IPO pricing methods, BP neural networks still have the superior ability to deal with nonlinear relationships in the presence of insufficient information; Zhao Minglan [2] advantages of BP neural networks to build an IPO pricing model for the GEM; Wen Ke [3] used dynamic parameters to optimize the traditional BP neural network in building a risk warning model for securities companies, further improving its accuracy in this application. It can not only keep the network training error small enough, but also make the network weights and thresholds smaller; Yang Limin [5] et al. selected some risk indicators by studying the risk early warning of securities companies, and used the L-M algorithm to optimize the BP neural network to some extent; Zhang Guozheng [6] used BP neural network to build a model for risk early warning, and venture investors or venture capital institutions can predict investment risks well when selecting investment projects; Hu Yanjing [7] et al. improved the traditional BP neural network and predicted risk indicators; Guo Peng [8] et al. established a risk prediction model by BP neural network. After verification, compared with other models, the model prediction used in this paper is more comprehensive and objective, and has a good development prospect; Li Haitang [9] first performed principal component analysis, and then optimized it by combining the optimization algorithm of particle swarm with BP neural network to achieve the short-term accurate prediction of grain pile temperature; Ye Fei [10] combined three algorithms of genetic, BP network and particle swarm to predict the Si content in blast furnace iron; Song Bo [11] used BP neural network method to optimize the clinical path modeling and conducted simulation experiments with actual data; Huang Xiaolong [12] introduced the genetic algorithm and used it to optimize the BP neural network to form a non-holiday intercity passenger flow prediction model; Zhao Fanghui [13] used the collected sample data to construct a PSO-BP neural network and used the model to predict the residential demand in Hefei city in the next three years.

3 BP Neural Network Algorithm

3.1 Basic Concepts

BP (Back Propagation) neural networks are divided into two processes.

- 1) Work signal forward-transmission sub-process
- 2) Error signal back-transmission sub-process

In a BP neural network, a single sample has m inputs and n outputs, and there are several hide layers between the input and output layers. A three-layer BP network can complete an arbitrary m -dimensional to n -dimensional mapping. These three layers are the input layer, the hide layer and the output layer, as shown in Fig. 1.

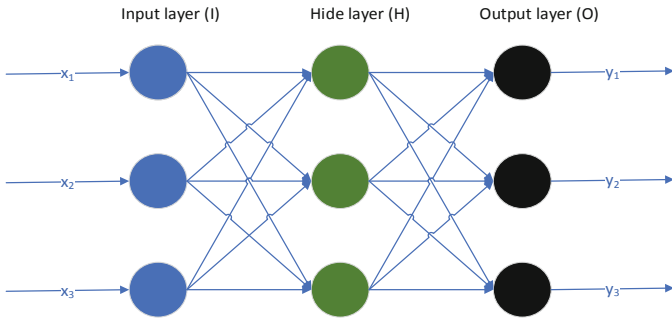


Fig. 1. BP neural network layers

The input variables are first passed as nodes in the input layer to the nodes in the intermediate layer, the implicit layer. The number of layers of the hide layer should be determined according to the problem under study. It can be designed as multiple hide layers or a single hide layer. Different levels of hide layers will form networks of different complexity and accuracy. The hide layers can transform and process the information from the input layer and finally get the result. The error can be gradually reduced by adjusting the weights and thresholds of the input layer-hide layer and hide layer-output layer nodes. The above steps are repeated several times to train the network. The training can be ended when the number of training times reaches a set value or the output error is within the set range.

3.2 Calculation Steps

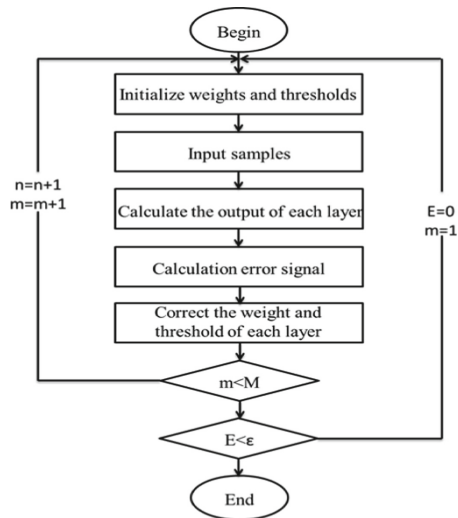


Fig. 2. Flowchart of BP neural network algorithm

The training process of BP neural network is as follows: the first stage is forward-propagation. The data is first transferred from the input layer to the hide layer, and then to the output layer after relevant calculations and processing are performed at that layer. At this point, the output value is obtained. By this time, the error between the calculated value and the expected value is calculated. If the value is within a reasonable range, training will be stopped, otherwise error retransmission occurs. The sample data is replaced by the main median error of back-propagation. Errors go from the output layer to the hide layer, then propagate to the input layer. In this propagation process, the network can adjust the weight and threshold value of each layer of nodes until the error gradually reduces to an acceptable range or reaches the training times. The following is the detailed description process of the algorithm [14] (Fig. 2).

- 1) Input sample data
- 2) First start the calculation of forward-propagation
The output of the i -th node in the input layer is as follows:

$$Y_i = f(x_i) \quad (1)$$

The input of the h -th node in the hide layer is as follows:

$$I_h = \sum_n \omega_{hi} I_i + \theta_i \quad (2)$$

The output of the h -th node in the hide layer is as follows:

$$Y_h = f(I_h) \quad (3)$$

- 3) Calculate the input of the j -th node of the output layer as follows:

$$I_j = \sum_n \omega_{jh} Y_h + \theta_j \quad (4)$$

The actual output value of the j -th node in the output layer is as follows:

$$Y_j = f(I_j) \quad (5)$$

- 4) Calculate the output error as follows:

$$E_k = \frac{\sum_{(j=1)}^M (d_j - Y_j)^2}{2} \quad (6)$$

- 5) Modify all weights and thresholds in the network:
- 6) Judge whether all samples have been trained, if not, the new sample data is then provided to the network for training, repeat step 2); if all the training is completed, start step 7).
- 7) The total error value is calculated as follows:

$$E = \sum_{k=1}^K E_k = \frac{1}{2 \sum_{k=1}^K \sum_{j=1}^M (T_j^k - Y_j^k)^2} \tag{7}$$

- Judge whether $E < \varepsilon$ is satisfied, if so, stop training, otherwise go to step 8).
- 8) Judge whether the training times have reached the set value. If the training has been completed, stop the training; if not, go back to step 2) and start training again.

4 Model Establishment

The model constructed in this paper predicts the material demand based on time series, and deduces the unknown quantity of material demand in the next few days by the known material demand in the previous days [15].

4.1 Determine the Network Layers

Determine the Number of Network Layers and the Nodes of Input Layer and Output Layer. This model is a three-layer model. Using the materials of the first 7 days of the earthquake disaster as input data, the material demand of the 8th day is predicted by running the BP model b of the MATLAB program. Then the data of the 8th day is put into the database, and the material demand of the 9th day is predicted by the material demand of the 2nd–8th days, and so on. So the number of nodes in the input layer is 7 and the number of nodes in the output layer is 1.

Determine the Number of Hide Layer Nodes. Same as BP model a, this model has only one hide layer. Based on the above empirical formula, this model conducts experiments on BP neural network and determines that the interval of the number of hide layer nodes is [5], [17]. And then the number of nodes is increased sequentially to train the network starting from the minimum number of nodes in this range.

By using trial-and-error method in this paper, we determine the number of nodes in the hide layer corresponding to 16 when the minimum network error is 0.0049.

Determine the Samples. According to the random function of RANDPERM, the training samples are [1–3, 5–8, 11] and the test samples are [4, 9, 10].

4.2 Set the Network Parameters

When setting network parameters for this model, the following parameters need to be considered:

Function Select. Function selection of this model is the same as BP model a.

Learning Rate. After several training sessions with this model, the learning rate was finally set to 0.03.

Expected Error. After several training sessions of this model network, the expected error of this model was determined to be 0.001, where the number of training sessions was set to 20 and the error metrics were RMSE and MEAP. Once the parameters are set, the samples can be trained, and the execution code is shown in the attached page.

5 Case Analysis

The vegetable demand at the time of the Jiuzhaigou earthquake is selected here as the data source needed for the material demand forecasting model, as shown in Table 1. The empirical case data are mainly obtained through the statistical data from the statistical bureau of the region.

Table 1. Total demand for vegetable materials (unit: g) in a region from 8 to 25 days

Date	Aug. 8	Aug. 9	Aug. 10	Aug. 11	Aug. 12	Aug. 13
Demand	1251514.5	1419551	1613274.5	1380102.5	1076933	1248046.5
Date	Aug. 14	Aug. 15	Aug. 16	Aug. 17	Aug. 18	Aug. 19
Demand	990989.5	1331856.5	1377008.5	1851725	1069538	1918305.5
Date	Aug. 20	Aug. 21	Aug.22	Aug.23	Aug. 24	Aug. 25
Demand	1649501.5	1540871.5	1257320	1632867	2210255	1292586.5

First observe the size [sample size, number of indicators] by Size function, here set to [8, 11], that is, the sample size is 11, the number of indicators is 8, of which 7 are input indicators, 1 is output indicators, meaning that inputting the first 7 days of the demand for supplies in the disaster area, and then predicting the number of supplies needed on the 8th day by running the BP model of MATLAB program (Tables 2 and 3).

Table 2. Input data

Dataset	Serial number	Input data					
		X_i	$x_i + 1$	$x_i + 2$	$x_i + 3$	$x_i + 4$	$x_i + 5$
Training set	1	1251514.5	1419551	1613274.5	1380102.5	1076933	1248046.5
	2	1419551	1613274.5	1380102.5	1076933	1248046.5	990989.5
	3	1613274.5	1380102.5	1076933	1248046.5	990989.5	1331856.5
	4	1380102.5	1076933	1248046.5	990989.5	1331856.5	1377008.5
	5	1076933	1248046.5	990989.5	1331856.5	1377008.5	1851725
	6	1248046.5	990989.5	1331856.5	1377008.5	1851725	1069538
	7	990989.5	1331856.5	1377008.5	1851725	1069538	1918305.5
	8	1331856.5	1377008.5	1851725	1069538	1918305.5	1649501.5
Test set	1	1377008.5	1851725	1069538	1918305.5	1649501.5	1540871.5
	2	1851725	1069538	1918305.5	1649501.5	1540871.5	1257320
	3	1069538	1918305.5	1649501.5	1540871.5	1257320	1632867

Table 3. Output data

Dataset	Serial number	Output value	
		xi + 6	xi + 7
Training set	1	990989.5	1331856.5
	2	1331856.5	1377008.5
	3	1377008.5	1851725
	4	1851725	1069538
	5	1069538	1918305.5
	6	1918305.5	1649501.5
	7	1649501.5	1540871.5
	8	1540871.5	1257320
Test set	1	1257320	1632867
	2	1632867	2210255
	3	2210255	1292586.5

The input and output data are normalized between [0,1], as Tables 4 and 5.

Table 4. Input data normalization

Dataset	Serial number	Input Data					
		Xi	xi + 1	xi + 2	xi + 3	xi + 4	xi + 5
Training set	1	0.3027	0.4622	0.6711	0.4196	0.0927	0.2772
	2	0.4979	0.6711	0.4196	0.0927	0.2772	0.0000
	3	0.7230	0.4196	0.0927	0.2772	0.0000	0.3676
	4	0.4521	0.0927	0.2772	0.0000	0.3676	0.4163
	5	0.0998	0.2772	0.0000	0.3676	0.4163	0.9282
	6	0.2986	0.0000	0.3676	0.4163	0.9282	0.0847
	7	0.0000	0.3676	0.4163	0.9282	0.0847	1.0000
	8	0.3960	0.4163	0.9282	0.0847	1.0000	0.7101
Test set	1	0.4485	0.9282	0.0847	1.0000	0.7101	0.5930
	2	1.0000	0.0847	1.0000	0.7101	0.5930	0.2872
	3	0.0913	1.0000	0.7101	0.5930	0.2872	0.6922

Table 5. Output data normalization

Dataset	Serial number	Output value	
		xi + 6	xi + 7
Training set	1	0.0000	0.2300
	2	0.2796	0.2695
	3	0.3166	0.6857
	4	0.7059	0.0000
	5	0.0644	0.7441
	6	0.7606	0.5084
	7	0.5401	0.4132
	8	0.4510	0.1646
Test set	1	0.2184	0.4938
	2	0.5264	1.0000
	3	1.0000	0.1955

Training and testing the input and output, running MATLAB yields the following results (Fig. 3):

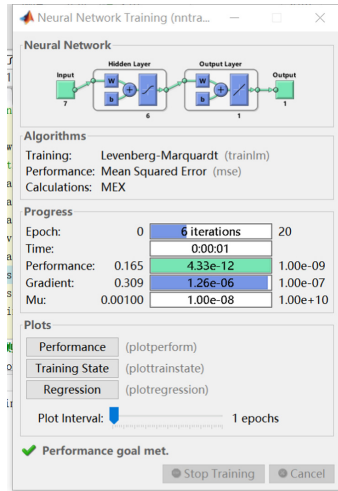


Fig. 3. Training results

Model D was trained 6 times and stopped, with regression evaluation metric $mse 1.00e-08$ and performance indicators = $4.33e-12$. Figure 4 is the training process diagram, the image shows that the error decreases gradually with the number of training sessions (Fig. 5).

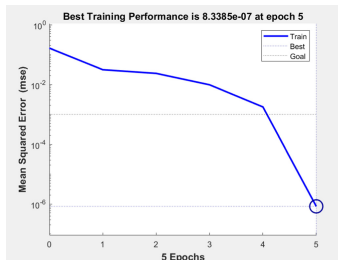


Fig. 4. Training process diagram

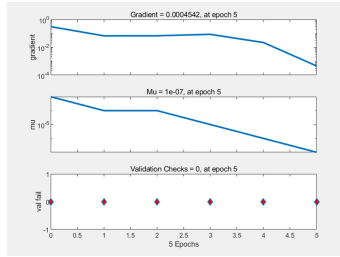


Fig. 5. Training status diagram

Figure 6 shows that the horizontal and vertical axes are fitted to a linear image when they are very similar (Table 6).

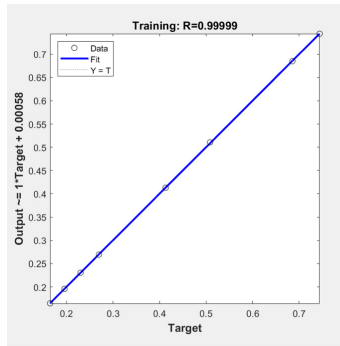


Fig. 6. Exporting object values

Table 6. Network output value

Training sample predicted values	0.5084	0.6857	0.1955	0.4132
	0.1646	0.7441	0.23	0.2695
Test sample predicted values	0.5202	0.655	0.0094	
Final overall predicted values	0.23	0.2695	0.6857	0
	0.7441	0.5084	0.4132	0.1646
	0.4938	1	0.1955	

The results were back-normalized as in Table 7.

Table 7. Inverse normalization results

Training samples	1649501.5	1851725	1292586.5	1540871.5
	1257320	1918305.5	1331856.5	1377008.5
Test samples	1069538	1632867	2210255	

The results of the root mean square and relative errors are calculated in Table 8.

Table 8. Validation results

Root mean square of the training data	Root mean square of the test data	Root mean square of all samples
2.37436028	744475.8894	388789.8533
Relative error of the training samples	Relative error of the test sample	Relative error of all samples
7.7624E-05	39.28789194	10.71493607

Finally, a graph is drawn to make the result analysis more intuitive: Fig. 7 shows the scatter plot of the training samples, in which the red cores are the actual values and the blue circles are the predicted values, which overlap when they are very close to each other.

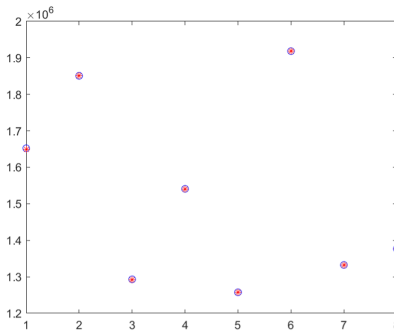


Fig. 7. Scatterplot of actual and predicted values of training samples

In order to see it more intuitively, we subtract the actual value from the predicted value, and the following is the result of the specific value of the subtraction (Fig. 8).

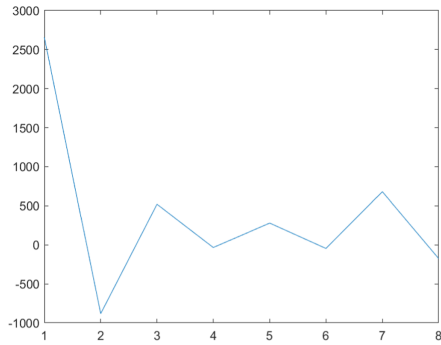


Fig. 8. Difference between actual and predicted values of training samples

The graphs of the results generated from the test samples are as follows (Figs. 9 and 10):

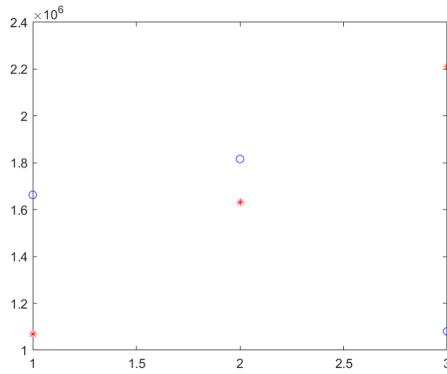


Fig. 9. Scatter plot of actual value predicted value of the test sample

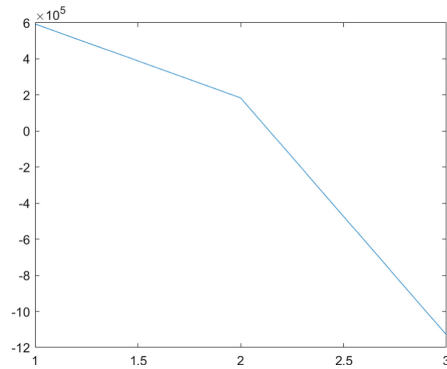


Fig. 10. The difference between the actual and predicted values of the test samples

The results of the analysis of the actual and predicted values of the overall sample are as follows (Figs. 11 and 12):

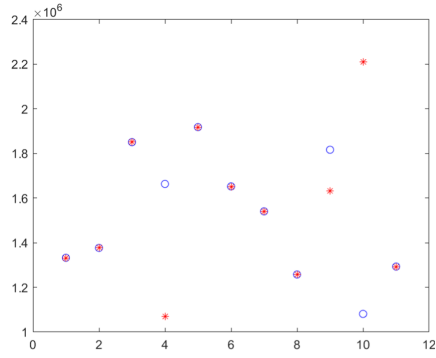


Fig. 11. Overall sample scatter plot

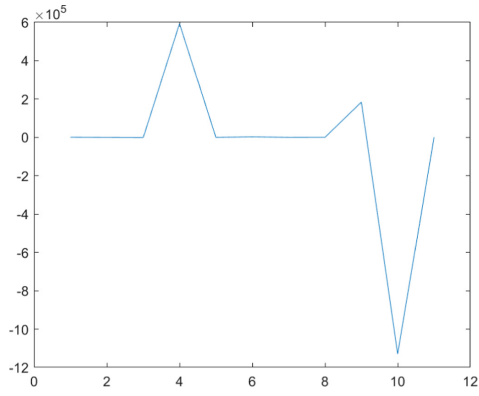


Fig. 12. Difference between the actual and predicted values of the overall sample

Percentage error (Fig. 13):

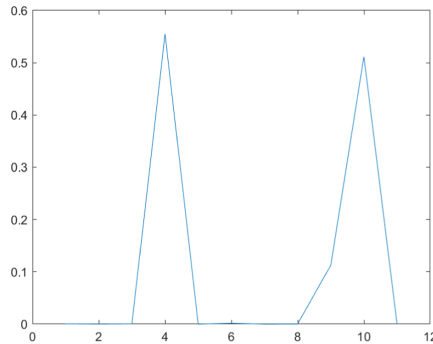


Fig. 13. Error percentage graph

6 Conclusion

The BP neural network model constructed in this paper is a direct prediction model, by inputting the actual amount of materials in the days before the disaster and then making analysis and prediction of the amount of materials in the next few days, so that it can further modify and improve the accuracy of emergency materials on the basis of the initial demand prediction and improve the prediction accuracy. The model is validated by actual cases, and it is concluded that the predicted material requirements under the contingency scenario are basically the same as the actual occurrence of the requirements, so the application of this model can reasonably forecast the material requirements under the contingency.

Acknowledgement. Supported by the National Key Research and Development Program of China (Grant No. 2018YFB1403101) and the Fundamental Research Funds for the Central Universities (Grant No. 2020RC22).

References

1. Liu Hui, F.: The Research on IPO with BP Neural Network. Wuhan University of Technology, China (2008)
2. Zhao Minglan, F.: A study on IPO pricing of GEM listed companies based on BP neural network. *Lanzhou J.* 77–80 (2010)
3. Wen Ke, F.: Investment bank risk prediction model of dynamic parameters of the neural network. *Sci. Technol. Bull.* **31**(09), 192–195 (2015)
4. Yang Limin, F.: Investment banks risk early warning based on BP neural network. *J. Anhui Univ. Technol. (Nat. Sci. Edn.)* 96–100 (2006)
5. Zhang Guozheng, F.: Research on early warning of venture capital risk based on neural network. *Sci. Technol. Manag. Res.* 182–184 (2006)

7. Hu Yanjing, F.: BP artificial neural network model: a new visual angle of the financial risk early-warning. *J. Chongqing Technol. Bus. Univ. (West. Econ. Forum)*, 68–71 (2003)
8. Guo Peng, F.: Research of BOT project risk assessment based on BP neural network. *Sci. Technol. Manag. Res.* 210–214 (2015)
9. Li, Haitang, F.: Research on grain pile temperature prediction model based on improved BP neural network. Henan University of Technology, China (2019)
10. Ye Fei, F.: Prediction Method of Silicon Content in Blast Furnace Hot Metal Based on Improved BP Neural Network. Anhui University of Technology, China (2019)
11. Song Bo, F.: Clinical path optimization based on BP neural network. *Comput. Technol. Dev.* **30**(04), 156–160 (2020)
12. Huang Xiaolong, F.: Study on passenger flow prediction of intercity passenger line based on improved BP neural network. Harbin Institute of Technology, China (2019)
13. Zhao Fanghui, F.: Research on Housing Demand Forecast Based on PSO-BP Neural Network in Hefei City. Hebei University of Engineering, China (2020)
14. Fan Rui, F.: Research on Demand Prediction of Large Earthquake Emergency Materials Based on PSO-BP Neural Network. Beijing Jiaotong University, China (2020)
15. Kang Lijun, F.: Application of Particle Swarm Optimization BP Neural Network in Emergency Material Demand Forecasting. Lanzhou Jiaotong University, China (2013)