TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.6, June 2014, pp. 4451 ~ 4456
DOI: 10.11591/telkomnika.v12i6.5481

The Prediction of Granulating Effect Based on BP Neural Network

Fang Li*[1], Kaigui Wu[2], Guanyin Zhao[3]
College of Computer Science, Chongqing University, Chongqing, China
*Corresponding author, e-mail:hnzy1988@163.com[1], kaiguiwu@cqu.edu.cn[2], 563799972@qq.com[3]

Abstract

During the granulation process of Iron ore sinter mixture, there are many factors affect the granulating effect, such as chemical composition, size distribution, surface feature of particle, and so on. Some researchers use traditional fitting calculation methods like least square method and regression analysis method to predict granulation effects, which exists big error. In order to predict it better, we build improved BP (Back propagation) neural network model to carry out data analysis and processing, and then obtain better effect than traditional fitting calculation methods.

***Keywords**: iron ore sinter mixture, size distribution, granulation effects, BP, neural network, fitting calculation*

1. Introduction

Based on the granulating mechanism of sinter mixture, there are many factors influence granulating effect, In certain conditions of granulation equipment and operating conditions, the main factor is the sinter mixture's own nature, including the chemical composition of the material, size distribution, moisture capacity ,microscopic structure and other factors. The main chemical composition of sinter mixture is TFe, FeO, SiO_2, CaO, Al_2O_3, MgO, MnO, TiO_2, K_2O, Na_2O, S, P, In which, CaO, Al_2O_3, MgO are conducive to granulation, but SiO_2 has an adverse effect to granulation. The content of these chemical constituents should be used as the input parameters of the model. While the rest of the chemical ingredients, such as MnO, TiO_2, K_2O, Na_2O, S, P, which has low content, will not be considered for the purpose of reducing the complexity of the model. Similarly, we select <0.2mm,0.2-0.7mm and 0.7-3mm as size distribution input, and other two parameter like moisture capacity, moisture content are considered.

So there are nine parameters as granulating effect prediction model's input, that is CaO, Al_2O_3, MgO, SiO_2, <0.2mm, 0.2-0.7mm and 0.7-3mm, moisture capacity, moisture content.

The performance quality of iron ore sinter mixture granulation is determined by permeability; however it is not measured in actual production but in experimental conditions. We use content of 3-8mm in the granulation to evaluate the permeability in actual production. There are two output parameters in granulating effect prediction model, permeability and 3-8mm granularity content.

The BPNN is a forward multi-layer network，which bases on BP algorithm, and the topological structure as a layered feed-forward network, is composed of the input layer, hidden layer and output layer. In essence, the BPNN algorithm makes the input and output of a set of samples into a nonlinear optimization problem with using the gradient descent algorithm optimization technique, which uses the iterative solution to get the right value [1].

2. Granulating Neural Model

In this paper, we build a three layers BP neural network model as Figure 1. In it the three layers are denoted as Input layer, Hide layer, and Output layer.

Received December 23, 2013; Revised February 16, 2014; Accepted March 3, 2014

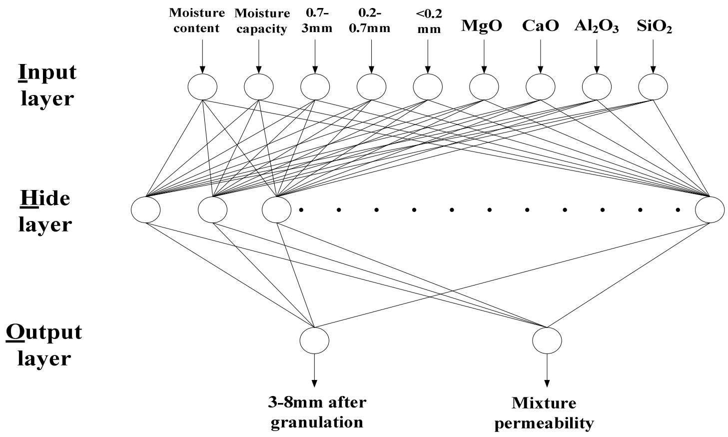

Figure 1. Granulating Neural Model

From Figure 1, we can see there are nine input nodes in the network, which are moisture capacity, moisture content, CaO, Al_2O_3, MgO, SiO_2, <0.2mm, 0.2-0.7mm and 0.7-3mm. We make those parameters as granulating effect prediction input.

There are also two output nodes in the network, which are permeability and 3-8mm granularity content. We make those two parameters as granulating effect prediction output.

3. BP Algorithm

Back Propagation neural network is one kind of neural networks with most wide application. It is based on gradient descent method which minimizes the sum of the squared errors between the actual and the desired output values [2, 3].

Suppose p is the input of network, a is the output of neurons in hidden layer, o is the output of neurons in output layer, r is the number of input nodes, s is the number of neurons in hidden layer, t is the number of neurons in output layer, w1 is the connection weight of hidden layer, w2 is the connection weight of output layer [4, 5].

The output of i neurons in hidden layer:

$$a_i = f_1\left(\sum_{j=1}^{r} w1_{ij} p_j + b1_i\right) \quad (1)$$

The output of i neurons in output layer:

$$o_k = f_2\left(\sum_{i=1}^{s} w2_{ki} a_i + b2_k\right) \quad (2)$$

In (1) and (2), f1, f2 are the excitation function in hidden layer and output layer respectively, b1,b2 are the threshold value in hidden layer and output layer, respectively. In which, i=1,2,…,s; k==1,2,…,t.

The training of BP network is realized by updating the connection weight according to error between real data and respect value [6].

Now we define the error function:

$$E_p(k) = \frac{1}{2}\sum_{k=1}^{t}(t_k - o_k)^2 \quad (3)$$

In (3), tk and ok is the real output and respect value respectively [7]. The total error function:

$$E(k)=\sum_{p=1}^{r}E_p(k) \tag{4}$$

Then compute the fluctuating value of connection weight [8]:

$$\begin{aligned}\Delta w2(k+1)&=-\mu\frac{\partial E(k)}{\partial w2(k)}\\&=u\sum_{p=1}^{r}[\delta_2(k)o_k]\end{aligned} \tag{5}$$

$$\begin{aligned}\Delta w1(k+1)&=-\mu\frac{\partial E(k)}{\partial w1(k)}\\&=u\sum_{p=1}^{r}[\delta 1(k)a_k]\end{aligned} \tag{6}$$

We can adjust the connection weight [9]:

$$w1(k+1)=w1(k)+\Delta w1(k+1) \tag{7}$$

$$w2(k+1)=w2(k)+\Delta w2(k+1) \tag{8}$$

4. Improved Model

The BP algorithm is simple, easy, small amount of calculation, and has the parallel advantages, so it is one of the largest and most mature training algorithms for network training at present. The essence of the algorithm is to solve the minimum value of the error function 6. Because it uses the method of steepest descent in nonlinear programming, there exists following problems [10, 11].

(1) Slow convergence, low learning efficiency;

(2) Easily falling into local minima.

In order to make the model more accurate, we use momentum adaptive learning rate adjustment algorithm. The weights and threshold adjustment formula with additional momentum factor [12]:

$$\Delta w_{ij}(k+1)=(1\text{-}mc)u\delta_i p_j+mc\Delta w_{jj}(k) \tag{9}$$

$$\Delta b_i(k+1)=(1\text{-}mc)u\delta_i+mc\Delta b_i(k) \tag{10}$$

In which, k is the training times, we take 10000, mc is the momentum factor, we take 0.9 is the weight between i node in hidden layer and j node in input layer; is the adjustment weight for hidden layer and is the adjustment threshold for hidden layer. At the same time it is not an easy thing to select appropriate learning rate for a particular problem. To solve the problem, it is natural to adjust the learning rate automatically in training process. The adaptive learning rate adjustment formula [13, 14]:

$$u(k+1)=\begin{cases}1.05u(k) & E(k+1)<E(k)\\0.7u(k) & E(k+1)>1.04E(k)\\u(k) & other\end{cases} \tag{11}$$

E(k) is sum of squared errors for the k step. The selection of the initial learning rate can be optional, we take 1.0 [15].

This method can ensure that the network train the samples by a learning rate which is always acceptable to the network. The system setting is showed in Table 1 including accuracy, rate, time, momentum factor.

Table 1. System Settings

System settings	Data settings
System accuracy	0.001
Learning rate	1.0
Training time	10000
Momentum factor	0.9

5. Model Solution

In order to get the relation between moisture capacity and moisture content, we collect and measure different ores from different factory.

In this paper, we use 40 groups of data as samples; take 32 groups as training samples which is selected randomly from the samples as showed in Table 2 then use 8 groups as forecast samples as showed in Table 3.

Table 2. Training Samples

No	moisture content	3-8mm/%	Permeability/mmH2O
1	5.1	29.85	288.00
2	6.86	53.80	216.00
3	8.12	61.33	230.00
4	5.73	63.48	220.00
5	6.68	61.20	228.00
6	6.58	59.90	228.00
7	6.27	31.96	286.00
8	5.21	27.37	652.00
9	5.00	33.69	674.00
10	6.24	35.04	570.00
11	8.38	51.68	314.00
12	7.99	59.16	196.00
13	7.04	40.96	256.00
14	9.32	60..27	208.00
15	7.61	43.21	286.00
16	6.33	31.35	588.00
17	5.585	25.92	550.00
18	7.85	55.41	404.00
19	6.65	66.02	196.00
20	5.42	54.79	288.00
21	5.68	40.24	596.00
22	6.34	58.23	413.00
23	6.27	47.94	296.00
24	7.10	42.24	248.00
25	7.71	42.86	232.00
26	8.99	60.67	200.00
27	8.53	54.56	196.00
28	8.50	61.08	206.00
29	7.88	66.11	224.00
30	7.24	49.35	246.00
31	5.52	32.16	566.00
32	7.80	49.08	566.00

Table 3. Forecast Samples

No	Moisture content	3-8mm/%	Permeability/mmH2O
1	7.2	49.22	260.00
2	5.78	32.21	442.00
3	6.82	36.87	286.00
4	4.98	63.45	250.00
5	6.90	25.16	820.00
6	5.42	70.23	210.00
7	6.58	42.08	236.00
8	8.61	63.33	248.00

Then we use c++ to write BP Algorithm, and make a small software showed in Figure 3 to train samples and get the forecast value.

The BP software should firstly input training data as training sample, the large training time for the number of neurons and the number of input layer, hidden layer, and output layer. When the training is stopped, the forecast result will be stored.

From Figure 2, we can see that before training system accuracy is set to 0.001, training time is set to 10000 times, learning rate is set to 0.8. The input num represents the number of node in input layer, the hidden num represents the number of node in hidden layer, the output num represents the number of node in output layer. When the training time is up to 8000, the total error is 0.0009992(0.0009992<0.001), the training is stopped.

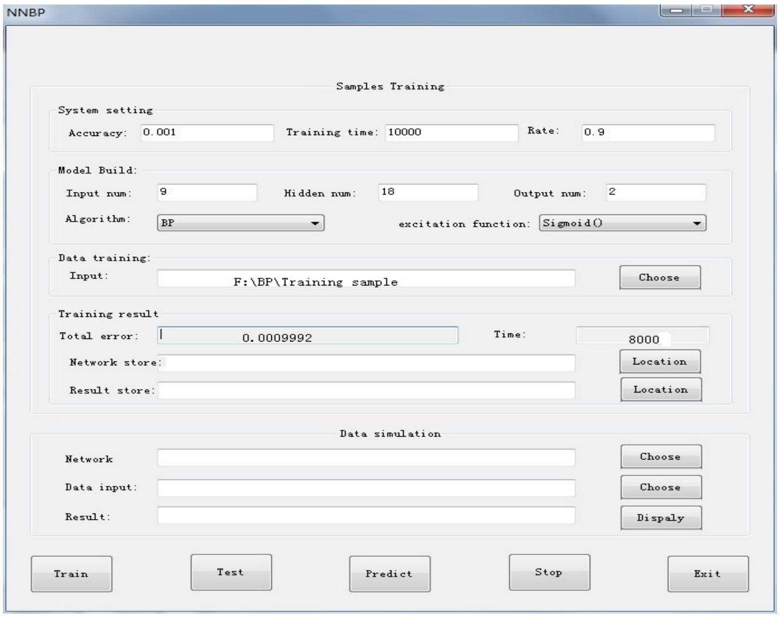

Figure 2. Main Interface of the Software

Train them to solve the weights from input layer to the hidden layer and from the hidden layer to the output layer, and then take the other samples as forecast samples, analyzing the difference between the forecast value (forecast incidence) and the actual data as showed in Figure 3 and Figure 4:

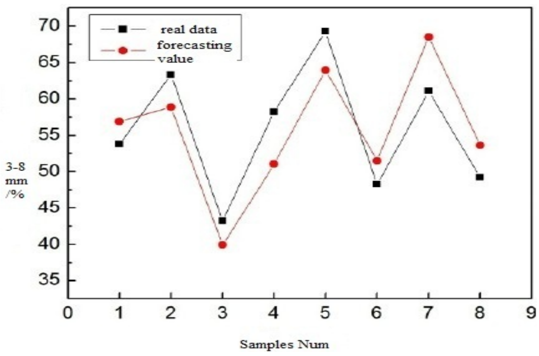

Figure 3. Prediction Results for 3-8mm Granules Percentage

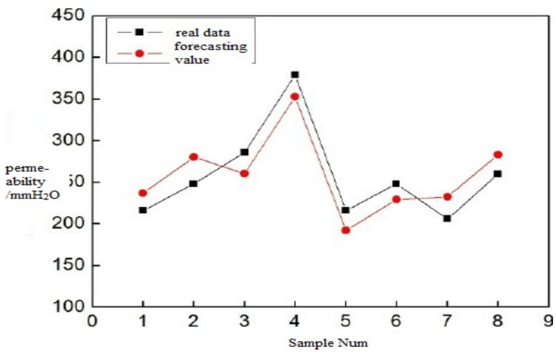

Figure 4. Prediction Results for Permeability

Sequentially comparing relative errors stay between 6%~8%, and the accuracy of the model reach 92%. So we can get this conclusion following:

(1) It is feasibility to predict granulating effect using BPNN model; and the model obtained very good effect.

(2) The BP network has the strong misalignment to approach ability; the fitting precision is good between the output and the samples.

In this paper, neural network is applied to the modeling process of granulation which is complex, nonlinear, dynamic, multivariable, difficulty in modeling. We obtain better effect than traditional fitting calculation methods.

In future, the model will play a certain role in granulating production.

Acknowledgements

This work is supported by the Major Research Project of the National Natural Science Foundation of China under Grant No.90818028, Natural Science Foundation Project of CQ CSTC: 2011BB2064.

References

[1] Jiang Hui Yu, Dong Min, Yang Feng. *Application of BP neural network into prediction of nitrobenzene compound in toxicity*, International Conference on Genetic and Evolutionary Computing, 2nded. 2008; 170-173.

[2] BHM Sadeghi. A BP-neural network predictor model for plastic injection molding process. *Journal of Materials Processing Technology*. 2000; 103(3): 411–416.

[3] Tie Wang, Chao Wang. The Fault Diagnosis of Bora Engine CH Emissions based on Neural network. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(8): 2343-2350.

[4] Yang Shanxiao, Yang Guangying. Emotion Recognition of EMG Based on Improved L-M BP Neural Network and SVM. *Journal of Software*. 2011; 6(8): 1529-1536.

[5] Luo Yaozhi, Tong Ruofei. Study of BP network for a cylinder shell's support identification. *Advanced Materials Research, Advances in Civil Engineering and Architecture Innovation*. 2012; 2050-2055.

[6] Li Jing-Rui, Wang Gang, Zhou Yun-Jin, Gong Yun-Fei. *Workpiece pattern recognition based on ART2 nerual network*. Harbin Gongye Daxue Xuebao/Journal of Harbin Institute of Technology. 2009; 41(3): 117-120.

[7] Shuiping Zeng, Lin Cui, Jinhong Li. Diagnosis System for Alumina Reduction Based on BP Neural Network. *Journal of Computers*. 2012; 7(4): 929-933.

[8] Shah, Manthan, Gaikwad, Vijay, Lokhande Shashikant, Borhade Sanket. *Fault identification for I.C. engines using artificial nerual network*. International Conference on Process Automation, Control and Computing, PACC. 2011.

[9] Y Alginahi, MA Sid-Ahmed, MAhmadi. *Local thresholding of composite documents using multi-layer perceptron nerual network*. Midwest Symposium on Circuits and Systems. 2004; I209-I212.

[10] Nabil EL Kadhi, Karim Hadjar, Nahla EL Zant, *A Mobile Agents and Artificial Neural Networks for Intrusion Detection*, Journal of Software. 2012; 7(1): 156-160.

[11] Saravanan P, Nagarajan S. *An adaptive learning approach for tracking data using visual and textual features*, International Conference on Trendz in Information Sciences and Computing, 2nded. TISC-2010. 2010; 192-196.

[12] Dongchao Ma, Zhibo Zhang, Yuanyuan Jia. A Software Components Automated Testing System on Reconfigurable Routing Platform. *Journal of Convergence Information Technology*. 2013; 8(1): 69 -76.

[13] Li Xinwu, Guan Pengcheng. A Novel Algorithm of Network Trade Customer Classification Based on Fourier Basis Functions. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(11): 6434-6440.

[14] Zhu Lei, Yang Dan, Wu Ying-bo. Selection of software reliability model based on BP neural network. *Computer Engineering and Design*. 2007; 28(17).