

Application of Improved BP Neural Network in Controlling the Constant-Force Grinding Feed

Zhaoxia Chen, Bailin He, and Xianfeng Xu

Key Laboratory of Ministry of Education for Conveyance and Equipment,
East China Jiaotong University, Nanchang, P.R. China
czx79@126.com

Abstract. BP neural network is applied to control the amount of feed, which is the key problem during the constant-force grinding. Firstly, BP neural network is constructed. Because its convergence is slow and local minimums often occur, the adaptive learning rate is used and certain momentums are added to improve BP neural network. Then the feature parameters in time and frequency domain are picked up in grinding vibration signals. With these feature parameters BP neural network is trained. As the result it makes the amount of grinding feed recognized precisely. Comparing the practical amount of feed with the set one, the system sends commands to increase or decrease the feed. So the amount of feed regains the set one. This method realizes the auto-control of the grinding feed, and puts forward a new method for constant-force grinding. It combines the features in time domain with those in frequency domain, and overcomes the limitation of the method which picks up feature parameters only in time domain or in frequency domain. At the same time this method provides a new clue of integrating other feature parameters in the grinding. Practice proves its good effect.

Keywords: BP neural network, Feature parameter, Constant-force grinding, Feed.

1 Introduction

The grinding is a method of machining with high precision. As a kind of grinding, the constant-force grinding is advanced to form curve shapes, which controls the force in the grinding and makes the wheel contact the workpiece uniformly. The premise of realizing the constant-force grinding is to recognize the feed of the wheel. Great attention has been paid by the researchers to the ways of monitoring the grinding on line [1] [2]. Generally, several feature parameters which can reflect the state pattern of grinding are the main basis for this work. Thus whether the on-line monitoring can be done depends on the effectiveness of the feature parameters. In some references the feature parameters are picked up only in time domain or in frequency domain [3] [4], and by this method the grinding feed cannot be recognized precisely inevitably. As BP neural network has the distinctive advantage in merging various feature parameters, this paper presents the application of improved BP neural network on the basis of picking up the feature parameters of the vibration signals in the grinding. The

aim is to recognize the grinding feed, control the amount of feed automatically and realize the constant-force grinding.

2 Principle of BP Neural Network

BP neural network defines the learning process consists of the forward propagation of the signal and backward propagation of error. In the forward propagation, the input sample is transmitted from the input layer, after processed by every hidden layer in sequence, then to the output layer. If the actual output cannot match the expected one (i.e. teaching signal), the backward propagation of error begins. This propagation means passing the output error through the hidden layers to the input layer reversely in some way. At the same time, the error is allocated to all units of each layer to obtain the error signal of each layer, which is the basis of correcting every unit weight. This adjustment of the weight for each layer is cycled, which is just the training process of the network. The process ends off when the output error is diminished to what is accepted or when the learning times is up to what is set in advance.

The main structure of BP neural network is that, the network is composed of nodes at different levels and the outputs from the nodes at a certain level are passed to those at next level. These outputs are amplified, attenuated or inhibited for different connection weight. Except the input layer, the input to each node is the weighted sum of the output from the former node. The stimulation output of each node is determined by the input, the activation function and the threshold. Figure 1 shows the main structure of BP neural network.

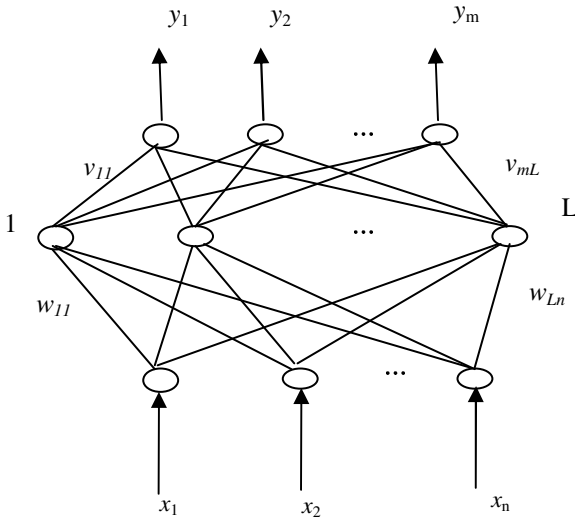


Fig. 1. The main structure of BP neural network

The learning algorithm of BP neural network is the gradient search rule to make the mean square deviation between the actual and expected outputs of the network least. As the error is propagated backward, it is corrected simultaneously. This is the learning process of the network, which contains two steps: forward calculation and backward adjustment. The activation function is S-typed:

$$f(x) = \frac{1}{1 + e^{-x}} . \quad (1)$$

The input to node l in the hidden layer is expressed by

$$net_l(x) = \sum_{j=1}^n w_{lj}x_j + \theta_l . \quad (2)$$

The output from node l in the hidden layer is determined according to

$$h_l = s(net_l(x)) . \quad (3)$$

The input to node k in the output layer is expressed by

$$net_k(h) = \sum_{l=1}^L v_{kl}h_l + \theta_k . \quad (4)$$

The output from node k in the output layer is determined according to

$$y_k = s(net_k(h)) . \quad (5)$$

$$y_k = s\left(\sum_{l=1}^L v_{kl}s\left(\sum_{j=1}^n w_{lj}x_j + \theta_l\right) + \theta_k\right) . \quad (6)$$

$$k = 1, 2, \dots, m$$

For a continuous mapping $F: R^n \rightarrow R^m$, a sample is the data representing the input and output, which is $(x_1, x_2, \dots, x_n; o_1, o_2, \dots, o_m)$ and is fit for the relationship:

$$F(x_1, x_2, \dots, x_n) = (o_1, o_2, \dots, o_m) . \quad (7)$$

The error of network can be calculated by following equation:

$$E = \frac{1}{2} \sum_{k=1}^m (y_k - o_k)^2 = \frac{1}{2} \sum_{k=1}^m e_k^2 . \quad (8)$$

(here, $e_k = y_k - o_k$ that is the difference between network output and mapping output.)

To make the network master this sample, the error must be as little as possible. This can be realized by adjusting the weights v_{kl} and w_{ij} , the simplest method of which is gradient descent algorithm.

$$\begin{aligned} \frac{\partial E}{\partial v_{st}} &= \sum_{k=1}^m e_k \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial net_k} \frac{\partial net_k}{\partial v_{st}} . \\ &= e_s s'(net_s(h)) h_t = \delta_s h_t \end{aligned} \quad (9)$$

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}} &= \sum_{k=1}^m e_k \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial net_k} \frac{\partial net_k}{\partial w_{ij}} \\ &= \sum_{k=1}^m e_k s'(net_k(h)) v_{ki} s'(net_i(x)) x_j . \\ &= \sum_{k=1}^m \delta_k v_{ki} s'(net_i(x)) x_j = \bar{\delta}_i x_j \end{aligned} \quad (10)$$

$$\bar{\delta}_i = \left(\sum_{k=1}^m \delta_k v_{ki} \right) s'(net_i(x)) . \quad (11)$$

The correction value Δv_{kl} and Δw_{ij} is proportional to error function and the relationship can be expressed

$$\Delta v_{kl} = \eta \delta_s h_t . \quad (12)$$

$$\Delta w_{ij} = \eta \bar{\delta}_i x_j . \quad (13)$$

The threshold θ is variable and also need correcting, the principle of which is the same as the correction of weights.

Because local minimums often occur in BP neural network, certain momentums should be added to avoid it. Meanwhile, the adaptive learning rate must be used to speed the training [5] [6].

3 Analysis of Grinding Experiment

The grinding experiment is made on the CNC external cylindrical grinding to obtain constant-force grinding. In the experiment, the acceleration sensor is fixed on the side of grinding carriage and the samples are picked up with the high-speed A/D sampler, the frequency (fs) of which is 1MHz. The workpiece is cylindrical and made from GCY15. The wheel is made from red corundum the granularity of which is 80. First, the vibration signals are recorded when the feed is $1\mu\text{m}$ 、 $2\mu\text{m}$ 、 $3\mu\text{m}$ respectively.

Then the feature parameters about different feed are picked up to train the network. Finally, the trained network is used to recognize the feed, ensuring it is $2\mu\text{m}$.

3.1 Features Pick-Up of Signals

Some work must be done to fulfill real-time monitoring. That is to analyze appropriate data, judge the current feed, and give commands the grinding tool should go ahead or back. For instance, if the current feed is calculated to be $3\mu\text{m}$ and the expected is $2\mu\text{m}$, the wheel is controlled to retreat for $1\mu\text{m}$. If the quantity of data analyzed each time is too big, the processing time is long and the operation is not live. If the quantity of data analyzed each time is too little, the features picked up are representative. After a lot of tests, the quantity of data analyzed each time is 500.

From the 500 data analyzed each time, some features should be picked up in time and frequency domains, to study the features of signals comprehensively. In time domain, they are mean and standard deviation. In frequency domain, they are maximum value of amplitude spectrum and frequency there. A part of feature parameters are showed in Table 1, 2 and 3. Tests show any one in the four kinds of features cannot recognize the feed individually, because they are overlapped in every feed. So it is necessary to integrate the four kinds of features to recognize the feed precisely.

Table 1. Feature parameters for the $1\mu\text{m}$ feed

mean	standard deviation	maximum value of amplitude spectrum	frequency (KHz)
1.423	55.269	16.591	87.890
0.383	50.913	16.278	83.984
1.170	38.368	10.379	76.171
0.570	47.805	13.018	87.890

Table 2. Feature parameters for the $2\mu\text{m}$ feed

mean	standard deviation	maximum value of amplitude spectrum	frequency (KHz)
1.092	38.570	10.649	58.593
0.566	39.267	10.842	72.265
0.403	50.029	13.463	91.145
0.266	49.210	20.331	82.465

Table 3. Feature parameters for the $3\mu\text{m}$ feed

mean	standard deviation	maximum value of amplitude spectrum	frequency (KHz)
0.944	60.669	21.766	58.593
0.465	36.084	8.764	42.968
0.929	58.294	21.263	68.359
1.209	41.587	17.381	58.593

3.2 Construction and Training of Improved BP Neural Network

The first thing is to determine the structure of the network. According to Kolmogov theorem [7], an $N \times (2N+1) \times M$ three-layered network is designed as recognizer, whose structure is shown in Figure 2. Here, N is the number of components of input vectors and M is the number of output categories. In this paper, N is 4 and M is 3. So in the network there are 4 input nodes, 9 hidden-layer nodes and 3 output nodes. When the output node is expressed as $(1,0,0)$, the feed is $1\mu\text{m}$. Similarly, $(0,1,0)$ means $2\mu\text{m}$ and $(0,0,1)$ means $3\mu\text{m}$. In BP network model, the activation function of hidden layers is tansig and the one of output layers is logsig. Because in the three elements of output vectors one is 1 and the other are 0, the outputs must be processed by competitive transferring function [8] [9].

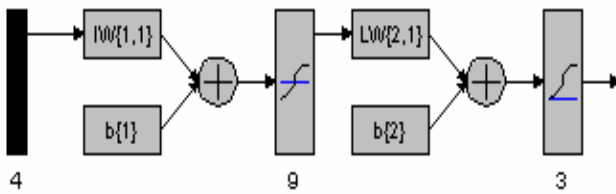


Fig. 2. The structure of the actual BP Neural network

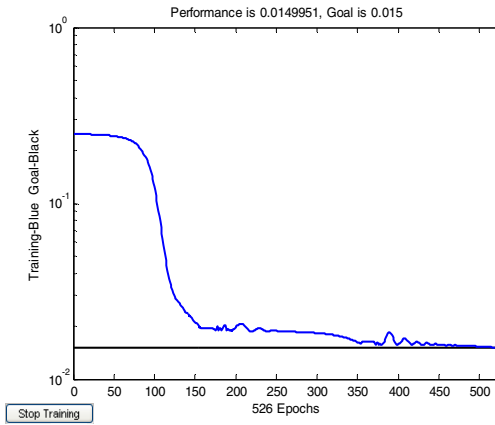


Fig. 3. The training error curve of the network

In order to train the network more fully, 40,000 sampling data are chosen, and from every 500 sampling data the features are picked up. The adaptive learning rate is used and certain momentums are added to improve BP neural network, because the velocity of its converge is very slow, and the network falls into the situation of local extrema easily. Figure 3 shows the training error curve of the network. From Figure 3, it can be seen that the network converges is very well.

The program is compiled in MATLAB. The main code including interpretation is as following.

```
%Read the training data from file which is made from
sampling data.
p=load('tezheng.txt');
%According to the feature of network, p must be
transposed.
p=p';
%Construct the network
net=newff(minmax(p),[9 3], {'tansig','logsig'},
'traingdx');
%Set the starting value of network
net.LW{2,1}=net.LW{2,1}*0.01;
net.b{2}=net.b{2}*0.01;
%Set the parameter of the network
net.performFcn='sse';
net.trainParam.goal=0.015;
net.trainParam.show=100;
net.trainParam.epochs=5000;
net.trainParam.mc=0.97;
%Read the target value of the network.
t=load('target.txt');
%According to the feature of network, t must be
transposed.
t=t';
%Training the network.
[net, tr]=train(net, p, t);
```

3.3 Verification of the Network

To verify the recognition ability of the improved BP network, the author collects corresponding data respectively when the feed is $1\mu\text{m}$, $2\mu\text{m}$ and $3\mu\text{m}$. After picked up, the feature parameters are inputted into the network. The network can recognize the feed quickly and precisely. The verification code in MATLAB is as following.

```
%Read the data used to test the network from file which
is made from sampling data.
pp=load('test.txt');
%According to the feature of network, pp must be
transposed.
pp=pp';
%Test the network
tt=sim(net, pp)•
%The output should handled by competitive transferring
function, so that the outcome can be 0 or 1
tt=compet(tt);
```

4 Ending

The feature parameters in time and frequency domain of grinding vibration signals are fused into the improved BP neural network successfully. The feed of grinding tool can be recognized precisely. This method provides a new clue of integrating feature parameters from different domains in the grinding. Also, this method makes it possible that constant-force grinding transforms from experience-control to auto-control. Practice proves its good effect.

References

1. Zhang, X., Kuhlenkotter, B., Knueper, K.: An Efficient Method for Solving the Signorini Problem in the Simulation of Free-form Surfaces Produced by Belt Grinding. *International Journal of Machine Tools & Manufacture*, 641–648 (2005)
2. Li, H.: Determination of Frequency Calculation Range of Feature Parameters in Grinding State Monitoring. *Tool Engineering*, 75–78 (2006)
3. Leea, D.E., Hwanga, I., Valenteb, C.M.O.: Precision Manufacturing Process Monitoring with Acoustic Emission. *International Journal of Machine Tools & Manufacture*, 176–188 (2006)
4. Karpuschewski, B., Wehmeier, M., Inasaki, I.: Grinding Monitoring System Based on Power and Acoustic Emission Sensors. *Annals CIRP*, 235–240 (2000)
5. Ferrari, S., Stengel, R.F.: Smooth function: Approximation Using Neural Networks. *IEEE Trans. on Neural Networks*, 24–38 (2005)
6. Hornik, K., Stinchcombe, M., White, H.: Universal Approximation of An Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks. *Neural Networks*, 551–560 (1990)
7. Yang, Y., Xie, G.-S.: Handwritten Digit Recognition Based on BP Neural Network. *Journal of East China Geological Institute*, 383–386 (2003)
8. Tang, D., Sun, H., Wang, H.: Milling Deformation Forecast with BP Neural Network. *Manufacturing Technology & Machine Tool*, 48–50 (2007)
9. Wu, J., Li, Y.: Intrusion detection model based on BP neural network and feature selection. *Computer Engineering and Applications*, 114–118 (2008)