

## Bayesian Neural Network of Rolling Force Prediction for Hot-Strip Mill

Xiaodan Zhang<sup>1</sup>, Rui LI<sup>2</sup>, Yanliang YE<sup>\*3</sup>

<sup>1</sup>Electrical and Information Engineering Department, Beihua University, Jilin,132013, P. R. China

<sup>2</sup>School of Automation & Electrical Engineering, USTB, Beijing 100083, P. R. China

<sup>3</sup>Scientific Research Office, Beihua University, Jilin,132013, P. R. China

\*Corresponding author, e-mail: [z1314oy@163.com](mailto:z1314oy@163.com)<sup>1</sup>, [lirui012@163.com](mailto:lirui012@163.com)<sup>2</sup>, [yylxcy@126.com](mailto:yylxcy@126.com)<sup>\*3</sup>

### Abstract

For obtaining relative accurate rolling-mill model is difficulty by the simple mathematical method, due to the complexity of the actual production scene and the non-linear relationship between variables, this paper firstly proposes an improved Bayesian regularization neural network model according to these measured data of 1580 production line. In this model, the paper constructs the improved Bayesian neural networks by the introduction of bound terms that represents the network complexity in the objective function. At last, the simulation result proves the effectiveness and validity of the model and the prediction accuracy of the model algorithm is superior to the traditional model.

**Keywords:** hot continuous rolling, rolling force prediction, neural network, Bayesian regularization

**Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.**

### 1. Introduction

The characteristics of the rolling process are non-linear, large-delay, strong coupling and parameter variation. Since state parameters of the control system are constant change, the traditional control model cannot well adapted due to its shortcomings. Taking into account of the learning neural network, many studies showed that predictive control effect using traditional BP algorithm or LM algorithm, which use neural networks to create rolling force model and use on-site measured data for training and learning, is remarkable. Since neural network training time is usually too long, and the sample data contains noise, there are problems of training times too much or the network scale too large and other issues, so it tends to make the network to remember unnecessary details when neural network training. If the noise included in the training process of network data are recorded, the new data may result in incorrect output, that is to say the traditional neural network algorithm does not have good generalization function, and there is a few problems such as the difficulty to control complexity degree of the model and the difficulty to overcome over-fitting data and so on [1]. According to the project specific issues, comparing with the BP algorithm, LM algorithm and Bias algorithm, this paper proposes an improved Bias method of neural network prediction for the rolling force of the hot rolling mill, so as to obtain the better mathematical model.

### 2. The Principle of Bayesian Regularization Neural Network

The method of Bayesian regularization mainly through modifying the training performance functions of neural network to improve their marketing capabilities. Input variables are two categories in actual system, one can be observed, and the other is also uncontrollable and unobserved. But the two variables have an impact on the output system. Let  $X$  be an observable variable,  $X=[x_1, x_2...x_n]$ . Then the following relationship between the system output  $d$  and the input  $x$ :

$$d = f(x) + \varepsilon \quad (1)$$

Where  $f$  represents the effect of the unobservable inputs to the outputs in the system;  $\varepsilon$  represents the random variable with a distribution.

The training performance function of neural network is generally using the mean square error function. Assuming the error function  $E_0$  is:

$$E_0 = \frac{1}{2} \sum_{k=1}^K \sum_{n=1}^N (y'_{nk} - y_{nk})^2 \quad (2)$$

Where  $N$  represents the number of samples,  $K$  represents the output number of neural network,  $y_{nk}$  represents the expected output,  $y'_{nk}$  represents the actual output of network.

Although the function that make the above-mentioned objective function reach to minimize has infinite, the neural network has local minimum. Aiming at the above-mentioned problems, it can be solved by regularization theory which adds a constraint term to obtain stable and useful solutions. In general, if  $F(x)$  is smooth, it will have the interpolation ability. When the network weight is small, the network output is smoother. So using smoothness constraint as a constraint term, it can effectively reduce the network weight. Then the objective function is:

$$F = \alpha \cdot \frac{1}{2} \sum_{i=1}^W w_i^2 + \beta \cdot \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^N (y'_{nk} - y_{nk})^2 = \alpha J_w(F) + \beta J_D(F) \quad (3)$$

### 3. Improved Bayesian Regularization Neural Network Algorithm

Bayesian neural networks put the probability distribution of weight value (threshold value) in the whole space as the starting point, considers the parameter as a random variable, considers the objective function as the likelihood function of training data, and the right decay term corresponds to the priori probability distribution of the network parameters, and integration the prior probability distribution assumption of the parameters, and the parameters of the posterior distribution can be constantly adjusted after the observing data are given. The prediction results of Bayesian neural network are based on an average of the posterior distribution of the parameters, a single model is mapped to a point in parameter space, and all models are mapped to the entire parameter space, in order to guarantee strong generalization ability of the network in theory [2].

Assuming the network structure  $H$  is given (primarily the number of hidden layer neurons) and the network model  $d_i=f(x_i, W, H)$  is given. In the absence of the sample data, the prior distribution of the weights (threshold) is  $p(w|\alpha, H)$ ; the posteriori distribution is  $p(w|D, \alpha, \beta, H)$  with the sample data set  $D = \{x^N, d^N\}$ . According to the Bayesian rule [3] is:

$$p(w|D, \alpha, \beta, H) = \frac{p(D|w, \beta, H)p(w|\alpha, H)}{p(D|\alpha, \beta, H)} \quad (4)$$

Where  $p(D|w, \beta, H)$  represents the likelihood function,  $p(D|\alpha, \beta, H)$  represents normalization factor,  $w$  represents the weight value (threshold) vector. The knowledge on the weights distribution is little when there is no data; therefore the prior distribution is a very wide distribution. It can be converted to a compact posterior distribution when the data are obtained, the weight value only in a very small range will produce consistent with the performance of the network map [4].

#### 3.1. Prior Probability

In the absence of the prior knowledge of weights,  $p(w|\alpha, H)$  follows the Gauss distribution that the mean is 0 and the variance is  $1/\alpha$  [5]:

$$p(w|\alpha, H) = \frac{1}{z_w(\alpha)} \exp(-\alpha \sum_{i=1}^W w_i) \quad (5)$$

Thus, the value of the normalization factor  $ZW(\alpha)$  is:

$$z_w(\alpha) = \int_{-\infty}^{+\infty} \exp(-\alpha \sum_{i=1}^W w_i) dw = \left(\frac{2\pi}{\alpha}\right)^{\frac{W}{2}} \quad (6)$$

### 3.2. Approximate Probability

Assuming the noise smoothing function with a Gauss distribution that the mean is 0 and the variance is  $1/\beta$  produces the desired output  $d$ , for a given input  $x$ , the observed probability of the output  $d$ :

$$p(d_n | x_n, w, \beta, H) \propto \prod_{k=1}^K \exp\left(-\frac{\beta}{2} [d_{nk} - y_{nk}(x_n, w, \beta, H)]^2\right) \quad (7)$$

If each sample independently selected data,

$$p(D | w, \beta, H) = \prod_{n=1}^N p(d_n | x_n, w, \beta, H) = \frac{1}{z_D(\beta)} \exp(-\beta J_D) \quad (8)$$

The normalization factor  $z_D(\beta) = \int_{-\infty}^{+\infty} \exp(-\beta J_D) dD$ ,  $J_D = \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^N (y_{nk} - c_{nk})^2$ , therefore,  
 $z_D(\beta) = \left(\frac{2\pi}{\beta}\right)^{\frac{N}{2}}$ . Where  $N$  is the input vector dimension.

### 3.3. Optimized and Solved

The Prior Probability function and approximate probability function into equation (8), we can obtain:

$$p(w | D, \alpha, \beta, H) = \frac{1}{z_M(\alpha, \beta)} \exp(-\beta J_D - \alpha J_w) = \frac{1}{z_M(\alpha, \beta)} \exp[-M(w)] \quad (9)$$

Where,  $z_M(\alpha, \beta) = \int_{-\infty}^{+\infty} \exp(-\beta J_D - \alpha J_w) dw$ .

If the sample data reaches a certain number, the posterior distribution tends to Gaussian distribution. If the posterior distribution curve simultaneously satisfies the sufficiently narrow and the sharply enough peak, you can further simplify the problem, namely using the Taylor expansion obtain  $z_M(\alpha, \beta)$ . Assume that  $w_{MP}$  is the weight value (threshold value) to which  $B$  is the minimum value corresponding. The Taylor expansion of  $M(w)$  in the vicinity of  $w_{MP}$  is:

$$M(w) \approx M(w_{MP}) + \frac{1}{2} (w - w_{MP})^T \nabla \nabla M(w_{MP}) (w - w_{MP}) \quad (10)$$

$$\nabla \nabla M(w_{MP}) = \beta \nabla \nabla J_D(w_{MP}) + \alpha \nabla \nabla J_w(w_{MP}) = \beta \nabla \nabla J_D(w_{MP}) + \alpha I \quad (11)$$

Where,  $\nabla \nabla$  represents the second derivative, therefore:

$$z_M(\alpha, \beta) = (2\pi)^{\frac{W}{2}} \{\det[\nabla \nabla M(w_{MP})]\}^{-\frac{1}{2}} \exp[-M(w_{MP})] \quad (12)$$

### 3.4. Approximate Calculation of Hessian Matrix

If you want to optimize the solution, the first is to calculate *Hessian* matrix when  $M(w)$  in the minimum point of  $w_{MP}$ . The formula (11) shows that the calculation amount of  $\nabla\nabla J_D(w_{MP})$  is large. Therefore, *Hessian* matrix can be further simplified to improve computing speed.

Make  $\varepsilon_{nk} = (y_{nk} - c_{nk})$ , then:

$$\frac{\partial J_D}{\partial w} = \left\{ \sum_{k=1}^K \sum_{n=1}^N \frac{\partial \varepsilon_{nk}}{\partial w_i} \right\} \quad (13)$$

$$\nabla\nabla J_D(w_{MP})_{ij} \approx \sum_{k=1}^K \sum_{n=1}^N \left[ \frac{\partial \varepsilon_{nk}}{\partial w_i} \cdot \frac{\partial w_{nk}}{\partial w_j} + \varepsilon_{nk} \cdot \frac{\partial^2 \varepsilon_{nk}}{\partial w_i \partial w_j} \right] \quad (14)$$

### 3.5. Determination of Hyper-parameters $\alpha$ and $\beta$

Hyper-parameters  $\alpha$  and  $\beta$  can be obtained by calculating the posterior distribution:

$$p(\alpha, \beta | D, H) = \frac{p(D | \alpha, \beta, H) p(\alpha, \beta | H)}{p(D | H)} \quad (15)$$

Assume that the prior distribution  $p(\alpha, \beta | H)$  meet a very wide distribution function. Because the normalization factor  $p(D | H)$  has nothing to do with  $\alpha, \beta$  in the above formula, so the problem of obtaining the maximum a posteriori distribution could be transformed into the problem of solving maximum likelihood function. Because the approximate function  $p(D | \alpha, \beta, H)$  is the normalization factor of the formula (15), then:

$$p(D | \alpha, \beta, H) = \frac{p(D | w, \beta, H) p(w | \alpha, H)}{p(w | D, \alpha, \beta, H)} \quad (16)$$

Uniting the (8) and (9), we can obtain:

$$p(D | \alpha, \beta, H) = \frac{z_M(\alpha, \beta)}{z_D(\beta) z_W(\alpha)} \quad (17)$$

For formula (7) taking the logarithm:

$$\begin{aligned} \ln(p(D | \alpha, \beta, H)) &= -\alpha J_W(w_{MP}) - \beta J_D(w_{MP}) + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) \\ &\quad - \frac{1}{2} \ln \{ \det[\nabla\nabla M(w_{MP})] \} + \frac{W}{2} \ln \alpha \end{aligned} \quad (18)$$

If the characteristic value of  $\beta \nabla\nabla J_D(w_{MP})$  is  $\{\lambda_i\}, i=1, 2, \dots, W$ , we can obtain that the characteristic value of  $\nabla\nabla J_D(w_{MP})$  is  $\{\lambda_i + \alpha\}$  by the formula (11). Also, because  $J_D$  is a normal error term, then:

$$\frac{d}{d\alpha} \ln \{ \det[\nabla\nabla M(w_{MP})] \} = \frac{d}{d\alpha} \ln \left[ \prod_{i=1}^W (\lambda_i + \alpha) \right]$$

$$= \sum_{i=1}^W \frac{1}{\lambda_i + \alpha} = \text{tr}[\nabla \nabla M(w_{MP})]^{-1} \quad (19)$$

Since  $\lambda_i$  and  $\beta$  are proportional, therefore:

$$\begin{aligned} \frac{d\lambda_i}{d\beta} = \frac{\lambda_i}{\beta} &\Rightarrow \frac{d}{d\beta} \ln\{\det[\nabla \nabla M(w_{MP})]\} = \frac{d}{d\beta} \ln\left[\prod_{i=1}^W (\lambda_i + \alpha)\right] = \\ \frac{d}{d\beta} \sum_{i=1}^W [\ln(\lambda_i + \alpha)] &= \sum_{i=1}^W \frac{1}{\lambda_i + \alpha} \cdot \frac{d\lambda_i}{d\beta} = \frac{1}{\beta} \sum_{i=1}^W \frac{\lambda_i}{\lambda_i + \alpha} \end{aligned} \quad (20)$$

Respectively make the partial derivative of A and B in the formula (20) equal to 0, you can get:

$$\begin{aligned} 2\alpha J_W(w_{MP}) &= W - \sum_{i=1}^W \frac{\alpha}{\lambda_i + \alpha} = \sum_{i=1}^W \frac{\lambda_i}{\lambda_i + \alpha} = \gamma \\ 2\beta J_D(w_{MP}) &= N - \sum_{i=1}^W \frac{\lambda_i}{\lambda_i + \alpha} = N - \gamma \end{aligned} \quad (21)$$

Therefore, the maximum prominence of  $\alpha_{MP}$  and  $\beta_{MP}$  can be obtained:

$$\alpha_{MP} = \frac{\gamma}{2J_W(w_{MP})}, \beta_{MP} = \frac{N - \gamma}{2J_D(w_{MP})} \quad (22)$$

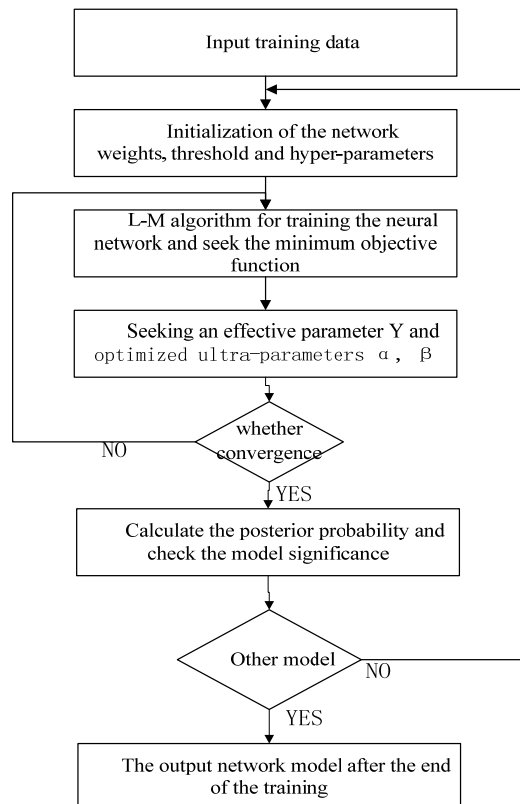


Figure 1. Bayesian neural network training flow chart

Where  $\gamma$  represents the number of parameters works in reducing the network performance index function,  $\gamma \in (0, W)$ .

In summary, the Bayesian neural network is an iterative process, each iteration involves three inferences [6]: the first layer inference is to maximize  $P(w|\alpha, \beta, H)$  under the conditions of hyper-parameters; the second layer inference is to optimize hyper-parameters  $\alpha, \beta$ , and to infer the most possible hyper parameters; the third layer inference is the significant degree of calculation model, and select the best model [7].

#### 4. Simulation

In the theoretical analysis, the paper made a series of simulation experiments based on the measured data of a 1580 hot strip mill production line which consists of slab yard, furnace, roughing mill group, finishing mill, coiling machines and other equipment. The strip design thickness of 1580 hot-rolling mill is 1.2mm~12.7mm and width is 700mm~1750mm. The main varieties include low-carbon steel, silicon steel, carbon-structural steel, micro-alloy steel, low-alloy steel. The steel strength class are  $\sigma_b \leq 65 \text{kg/mm}^2$ ,  $\sigma_s \leq 50 \text{kg/mm}^2$ .

The seven rack four-roll mill of 1580 finishing mill was arranged in tandem, the seven mills referred to as  $F_1 \sim F_7$ , the distance between each rack is 5800mm. four roller pairs (PC) mill, roll crossing with unilateral transmission form. Four roller pair-cross (PC) mill is used by  $F_2 \sim F_7$ , the form of unilateral cross-use drive is adopted by roll.  $F_1$  has the negative roll bending,  $F_2 \sim F_7$  has the positive bending [8].

To calculate the rolling speed and rolling time frame: According to the rule of volume flow rate, taking into account the distance of the roughing mill exports to the finishing mill entrance is 18m, and the distance between each rack is 5.8m, The two frame transmission time of the rolled piece in each can be calculated, and the total rolling time also can be obtained. The results are shown in Table 1. The pre-set roll gap values of each frame are shown in Table 2.

Table 1. The Rolling Speed and Time of Rolling Mill

Framework	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$
Rolling speed(m/s)	0.867	1.4743	2.1825	3.0225	4.21	4.8672	5.176
Exit speed(m/s)	6.1834	7.8959	8.8373	9.5083	11.8712	11.2095	8.2363
Over time mill(s)	0.2222	0.1064	0.0651	0.0422	0.0301	0.0239	0.0199
Total time rolling(s)	0.2793	0.1161	0.0727	0.049	0.0364	0.0289	0.0253

Table 2. The Pre-set Roll Gap Value of each Frame

Framework	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$
gap value(mm)	16.584	9.296	5.95	5.165	3.649	3.711	3.596

The selection principle of the input and output variables: there is a clear and definite relationship that produces a greater impact between the selected variables and the studied output variables when the neural network variables are selected; the selected variables can be detected or calculated in the actual production process [9].

We can know that there are many factors affecting the rolling pressure changes by the mechanism analysis of the rolling process and prior experience, such as the entrance thickness of rolled plate, exit thickness, reduction rate, rolling temperature, rolling speed, roll diameter, chemical composition content and so on. According to the scene measured data, the determine parameters of each layer is as follows:

The input layer: the C content, the Si content, the Mn content, the Cu content, the entry thickness (H), outlet thickness (h1), rolling width (B), the rolling temperature (T), the rolling time (t1), reduction ratio (e1). The output layer: the rolling force (P).

The number of neurons in the input layer is 10, the number of neurons in the output layer is 1.

#### 4.1. The Prediction Rolling Force of BP Neural Network

In the paper, 610 groups' normalized data are used as the training samples; a variable learning rate BP algorithm is used to train the network. The transfer function of between the input layer and the hidden layer adopts the hyperbolic S-shaped transfer function; the neuron of the output layer uses the neuron linear transfer function. The frequency of the learning process is 50, the training time is 5000 and the expected error is set to  $1e-004$ .

Figure 2 shows the error of the BP algorithm forecasting rolling force and the actual value, and the error of the calculated value of rolling force and the actual value. The abscissa is the sample-point; the vertical axis is the error value. As can be seen, the use of neural networks, the forecasting error values of using BP neural networks is between  $[-200, 300]$ , but the traditional calculation errors is between  $[-600, 400]$ . Namely the forecasting rolling force fitting curve of BP neural network is good. Therefore, the prediction accuracy of rolling force can be improved by the use of neural network model.

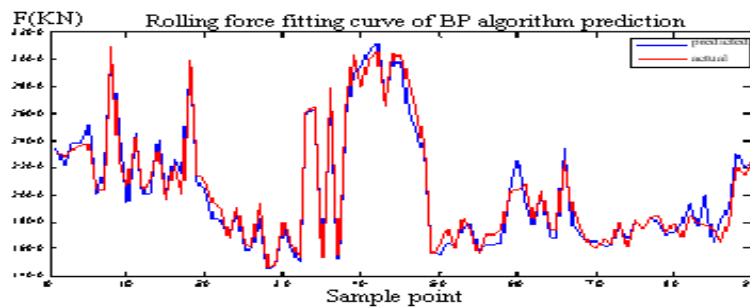


Figure 2. The Rolling Force Fitting Curve of BP Algorithm Prediction

#### 4.2. The Forecasting Model of Levenberg- Marquardt Algorithm

In the same parameter settings, 610 groups' normalized data set 610 as the training samples, Levenberg-Marquardt algorithm is used to train the network.

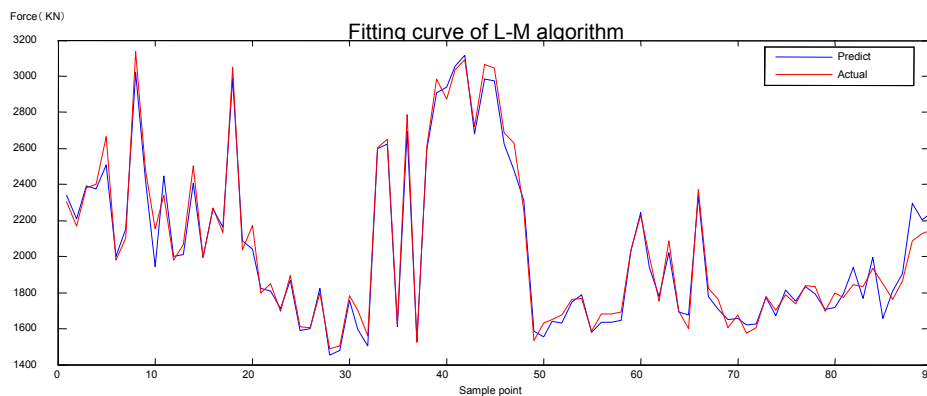


Figure 3. Fitting Curve of L-M Algorithm

As can be seen, L-M algorithm forecasting by the rolling force and the actual value of the error, the prediction error values of the model using the LM algorithm is between  $[-200, 200]$ , which error is smaller than the error of the traditional BP algorithm. However, we found that the network prediction accuracy is not guaranteed when the number of hidden nodes changes in experiments. For example, taking the 30 hidden nodes, the training of the network is completed to achieve the accuracy of  $9.9356e-005$  at step 150. But we found the prediction error is quite

large and the rolling force prediction curve deviates significantly from the actual rolling force values when the rest data of 90 groups were used to detect. This is due to excessive emphasis on training precision, leaving the design of the network is large, resulting in more powerful network function mapping, resulting in excessive adaptation phenomenon. This is the limitations of LM algorithm: The accuracy of the network training objectives cannot be determined, the error is set too small in order to the lack of trained network generalization and the low precision; Error that be set too large will easily lead to excessive generalization and non-versatility.

**4.3. Bayesian Regularization Neural Network Mode**

In the same parameter settings, Bayesian neural network adds the constraints of the squared weights and the number of valid parameter in network training objective function. The training Results: MSE = 0.0058, SSE = 1.3961.

Figure 4 shows the comparing error curve of the forecast rolling force of Bayesian network and the actual rolling force. As can be seen, the prediction error basically distributed in [-200, 200], but the traditional calculation error distribution in [-600, 400]. Figure 5 shows the curve of the predictive rolling force and the measured value. We can see the fitting degree is good from the graph. Therefore, the accuracy of Bayesian neural networks can establish a good Rolling force prediction model.

Table 3. The Training Result Contrast of Three Network Model

		N=4	N=5	N=6	N=7	...	N=26
Variable learning rate BP	MSE	0.0145	0.0097	0.0094	0.0092		0.0059
	SSE	3.4744	2.3212	2.2665	2.1998		1.4190
Levenberg- Marquardt	MSE	0.0084	0.0076	0.0068	0.0052		9.9363e-007
	SSE	2.0068	1.8203	1.6262	1.2371		2.3847e-004
Bayesian regularization	MSE	0.0109	0.0078	0.0071	0.0058		0.0026
	SSE	2.6058	1.8818	1.7015	1.3961		0.6196

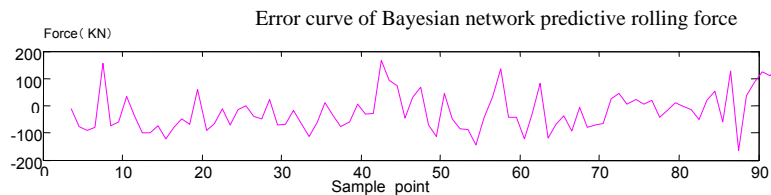


Figure 4. Error Curve of Bayesian Network Predictive Rolling Force

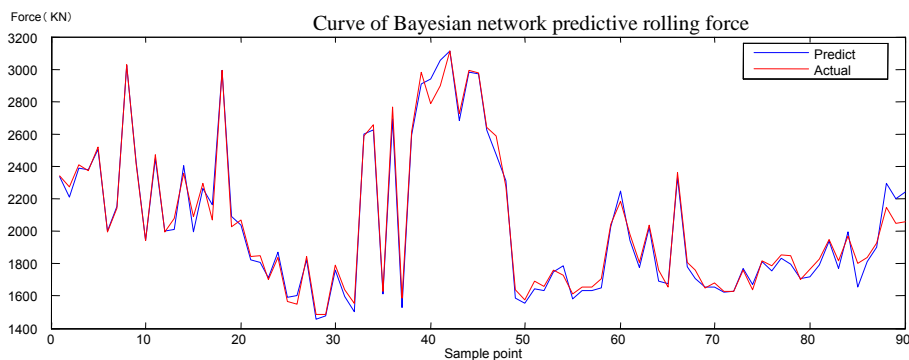


Figure 5. Curve of Bayesian Network Predictive Rolling Force

Using Bayesian neural network and introducing the posterior probability of statistics allows the network to avoid falling into local extreme and to reduce the training time. In the experiment, each training time, the error indicator and the network performance values of the Bayesian neural network training model are relatively stable. When the size of the network



increases, it did not appear the phenomenon of "over-fit", which avoids trying to choose the best approach to determine network size. In the same parameter settings, Table 3 shows the training results of taking the different type of hidden nodes by contrast variable-learning rate BP algorithm, L-M optimization algorithm and Bayesian regularization algorithm.

In the same parameter settings, Table 3 shows the training results of taking the different type of hidden nodes by contrast variable-learning rate BP algorithm, L-M optimization algorithm and Bayesian regularization algorithm.

The comparing the results show that L-M algorithm need to calculate the Jacobin matrix and the Hessian matrix, and larger storage space. When the number of parameters is very large, L-M algorithm may not be practical. In addition, the approximation precision of neural network trained by train function for learning samples is very high, it is easy to realize "over-match" for the sample data points. But for non-learning samples (such as validation learning-effect sample), the approximation error will appear a singular phenomenon that decreased and then risen along with the increases in the number of neural network training, which cannot guarantee the generalization ability of the network. However, trainer adds the weights of the network to the performance function, select the optimum weights and thresholds groups so can reduce the weight range in order to make the network output smoother, ease the lack of generalization capability, and ensure that the Bayesian training network is stability and robustness.

## 5. Conclusion

Based on the background of a 1580mm hot strip mill production line, This paper describes the structure and function of modern hot rolling control system, put forward the multi-level control strategy and research on the finishing mill model based on LEVEL 2. Considering the network stability using the BP algorithm is poor and the generalization ability is low, Bayesian neural networks is introduced into the constraint in the traditional neural function according to the complexity of the actual production site. Combined with the non-linear characteristics of the variables, higher precision neural network prediction model has been obtained based on the measured data. At last, Study and the experimental result found that the prediction accuracy of the optimized model has been significantly improved, and the stability of its network, the training speed and the generalization capabilities are superior to the traditional network neural network.

## References

- [1] Liyan Dong *Research of Application Foundation on Bayesian Networks*. Changchun: Jilin University. 2007.
- [2] Wei Dong, Zhang Minglian, Jiang Zhijian, Sun Ming. *Neural Network Non-linear Modelling Based on Bayesian Methods*. Computer Engineering and Applications. 2005; 11(1): 5.
- [3] MN Haimeer. Bayesian-neural network approach for probabilistic modeling of bacterial growth/no-growth interface. *International Journal of Food Microbiology*. 2003; (82): 233-243.
- [4] Zhang Jixian, Yi Xia, *Neural Network and Its Application in Engineering*. Beijing: China Mechine Press. 1996.
- [5] Man Leung Wong, Shing Yan Lee, Kwong Sak Leung. Data Mining of Bayesian Network Using Cooperative Coevolution. *Decision Support Systems*. 2004; (8): 451-472.
- [6] Qiang S, Ai-min W, Yun-su Z. The combination Prediction of BTP in Sintering Process based on Bayesian Framework and LS-SVM. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(8): 4616-4626.
- [7] Honglian L, Hong F, Ju T, et al. A Novel Prediction Algorithm of DR Position Error Based on Bayesian Regularization Back-propagation Neural Network. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 11(7): 4004-4011.
- [8] Hu Xianlei, Wang Jun, Wang Zhaodong. Process Model Setup System of 3340mm Plate of Shougang Group Co. Steel Rolling. 2003; 20(1): 42-44.
- [9] Lv Cheng, Wang Guodong, Liu Xianghua, Jiang Zhengyi, Yuan Jianguang, Xie Qi. *High-precision Prediction of Rolling Load of Finishing Stands with Neural Network*. Iron & Steel. 1998; 33(3): 33-35.