Int J Adv Manuf Technol (2017) 93:3325–3338
DOI 10.1007/s00170-017-0711-5

ORIGINAL ARTICLE

Prediction of bending force in the hot strip rolling process using artificial neural network and genetic algorithm (ANN-GA)

Zhen-Hua Wang[1] · **Dian-Yao Gong**[1] · **Xu Li**[1] · **Guang-Tao Li**[1] · **Dian-Hua Zhang**[1]

Received: 8 May 2017 / Accepted: 26 June 2017 / Published online: 15 July 2017
© Springer-Verlag London Ltd. 2017

Abstract An artificial neural network (ANN) optimized by genetic algorithm (GA) is an established prediction model of bending force in hot strip rolling. The data are collected from factory of steel manufacture. Entrance temperature and thickness, exit thickness, strip width, rolling force, rolling speed, roll shifting, target profile, and yield strength of strip are selected to be independent variables as network inputs. MATLAB software is utilized for establishing GA-ANN model and achieving the purpose of obtaining the bending force as results of setup model, as well as the GA method is used to optimize the initial weights and biases of the backpropagation neural network. Mean absolute error (MAE), mean absolute percentage error (MAPE), root mean squared error (RMSE), and correlation coefficient are adapted to evaluate the performance of the model. The predictive results are compared with the measured results to verify the accuracy of the GA-ANN prediction model. It is found that the optimization effect is the best with the population size 40 crossover probability of 0.7 and the mutation probability of 0.05 at the same time, the fitness function value can reach 80.7. In addition, the ANN architecture 9-11-1 trained with Bayesian regulation "trainbr" function has the best performance with mean absolute error of 0.01 and correlation coefficient of 0.983. With a deeper understanding of neural networks through the analysis of the GA-ANN model, the proposed model can be flexibly used for on-line controlling and rolling schedule optimizing.

Keywords Hot strip rolling · Artificial neural network (ANN) · Genetic algorithm (GA) · Bending force

1 Introduction

The hydraulic bending roll control is one of the main methods for hot strip shape control. The control principle is that the external bending moment is used to change the contact pressure distribution between work roll and back-up roll, and the deflection of the work roll is controlled by a hydraulic cylinder mounted between the bearing seats to improve the flatness and profile. The control technology is characterized in that the crown of the roll can be adjusted rapidly, and it has the advantages of no hysteresis control. Combining with other flatness control methods, the flatness and profile adjustment ability can be further enhanced. The schematic diagram of hydraulic roll bending technique is shown in Fig. 1. The prediction accuracy of the roll bending force has an impact on the flatness and profile control accuracy, especially the head-end of the strip. The high prediction accuracy is beneficial to the closed loop feedback control of the bending force. In the actual production process, the bending force is calculated according to the requirements of the temperature, thickness, width, rolling force, material, the thermal expansion of the roll, and the roll wear as well as the target flatness and crown. The configuration model of the hot rolling bending force is complex because some rolling factors related to the model parameters are nonlinear, strong coupling, large detection error, and so on. The response time of the mathematical

✉ Zhen-Hua Wang
wangzhenhuaneu@yeah.net

1 The State Key Laboratory of Rolling and Automation, Northeastern University, Shenyang 110819, Liaoning, People's Republic of China

model which is established by the traditional theory is slow in the production practice, and the control precision is low. Appropriate bending force can be only obtained after rolling through a few pieces of steel. In recent years, artificial intelligence algorithm model has been gradually applied to many engineering fields and has a pretty good performance. Artificial neural network (ANN) is one of the methods that is suitable to deal with the internal relations of complex model because of its highly nonlinear, large amounts of data parallel processing, high robustness, and fault tolerance. So, a kind of ANN is explored in this paper to predict the bending force in the hot strip rolling process.

ANNs have been widely used in many fields due to their unique properties, and the application in the field of rolling is very representative. Many scholars in the world have been studying. Jeon and Kim [1] designed a neural network model to calculate appropriate rolling force and rolling torque quickly and accurately before strip reached the rolls and executed in the rolling system optimization strip manufacturing process. Portmann et al. [2] applied the neural network to the rolling mill control system based on the traditional mathematical model. Larkiola et al. [3] introduced the neural network into the strip shape and thickness integrated control system in cold rolling process. Similarly, Yao [4] also used neural network in the process of shape control in hot strip rolling. Pican et al. [5] utilized neural networks to calculate the rolling force in a temper mill, and they attempted to use multiple networks to solve the problem of local point degradation in the single network. Lee and Lee [6] proposed a neural network model based on long-term learning to improve the prediction accuracy of the rolling force in hot strip rolling, the main idea was that in the pre calculation stage, neural network was used to correct the traditional model and improve the hit ratio of the strip head thickness. SIEMENS used more than 20 neural networks in its factory [7], and typical applications include the rolling force calculation model and the hot strip temperature calculation model. Chun et al. [8] studied the error propagation network model for the flow stress and rolling force prediction of aluminum alloy during hot rolling process. Lee and Choi [9] discussed the factors that affect the performance of the neural network and presented an on-line adaptive rolling force setting calculation network model. Son et al. [10] proposed an on-line prediction model for the rolling force of hot rolling. Moussaoui et al. [11] found that the prediction accuracy of rolling force in hot strip rolling could be greatly improved by combining neural network with analytical model. Yang et al. [12] proposed a neural network model that did not rely on the traditional mathematical model and empirical model but it improved the prediction accuracy of rolling force and rolling torque. Yang et al. [13] used finite element model to produce rolling data and developed a neural network model of roll load prediction with the obtained data. The rolling force and rolling torque of hot strip rolling neural network prediction model were also proposed by Bagheripoor and Bisadi [14], the accuracy of the model was verified by comparing with the results of the finite element model.

Although artificial neural networks have been widely used in the prediction of rolling force, little research has been done on the prediction of roll bending force. The main purpose of this research is to develop a neural network model that can able to predict the roll bending force accurately in hot strip rolling. Firstly, a model was built using data sets acquired from a stainless steel hot rolling mill production line. Secondly, the network model

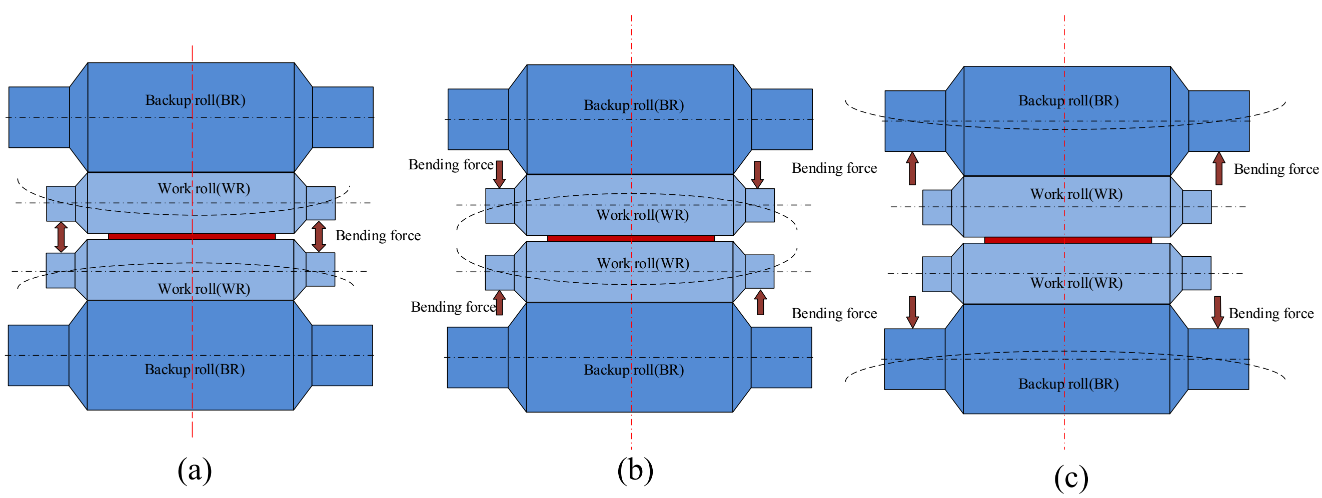

Fig. 1 Schematic diagram of hydraulic roll bending technique. **a** Positive bending of work roll. **b** Negative bending of work roll. **c** Bending of back-up roll

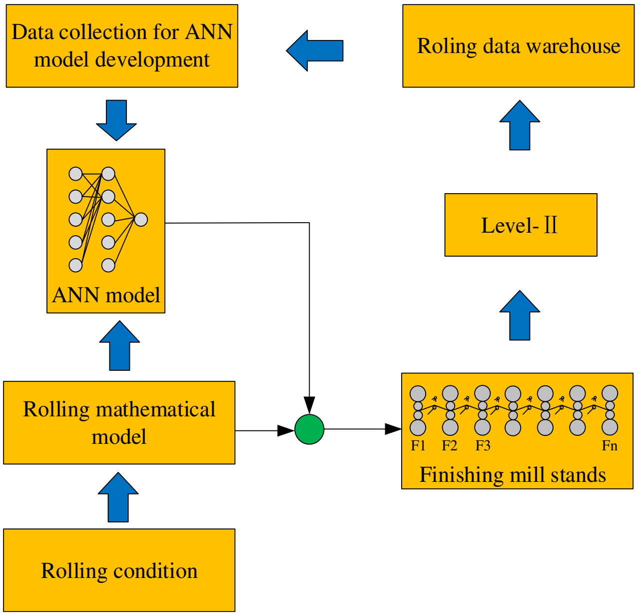

Fig. 2 Neural network data collection and application

was optimized by genetic algorithm (GA). Finally, the accuracy of the model was analyzed. The analysis results show that the proposed model is feasible to apply in the prediction of bending force in hot strip rolling and provides a method for rolling schedule optimizing.

2 Artificial neural network and database

2.1 Neural network theory

The theory of ANN is inspired by the animal brain neuron structure and its ability to deal with huge information [15], it is a kind of neural network to imitate animal behavior characteristics of the distributed parallel processing algorithm of the mathematical model. This network achieves the purpose of processing information by adjusting the relationship between a large number of nodes connected to each other, and it has the ability of self-learning and is adaptive.

The most common neural network is called “multilayer feed forward neural networks” [16], including input layer, one or more hidden layer, and output layer. Backpropagation (BP) network is an outstanding representative of multilayer feed forward neural network using the error backpropagation algorithm. Error backpropagation algorithm contains two stages, namely, forward propagation of signal and backpropagation of error. The input layer neurons receive input variables, hidden and output layer neurons process signal, and output layer neurons export the final results. In other words, the input layer neurons only accept the input variables, hidden layer and output layer contain functional neurons. The neural network learning process is based on the training set to adjust the connection weights between neurons and biases of each functional neuron.

In the first stage of the algorithm, the hidden layer neurons are received signals from the input layer neurons, these signals are transmitted by connecting links with adjustable weights. In this layer, the total values of input layer neurons are received and they are compared with the biases of the current neuron, and then, the output of the neuron is processed by the activation function. The neurons of the hidden layer weighted sum inputs variables as shown by Eq.(1), and the neurons of the output layer weighted sum inputs variables as shown by Eq.(2) [17].

$$O_i = f\left(\sum_{j=1}^{M} w_{ij}x_j + \theta_i\right) \tag{1}$$

$$O_k = \phi\left[\sum_{i=1}^{q} w_{ki} f\left(\sum_{j=1}^{M} w_{ij}x_j + \theta_i\right) + a_k\right] \tag{2}$$

where f and ϕ are the activation function of hidden layer and output layer, respectively. M and q are the input

Table 1 Descriptive statistics of the input parameters

Number	Parameter	Unit	Mean	Standard deviation	Minimum	Maximum
1	Entrance temperature	°C	1035	13.38	988	1082
2	Entrance thickness	mm	2.47	0.03	2.38	2.57
3	Exit thickness	mm	2.24	0.02	2.24	2.35
4	Strip width	mm	1252	2.13	1248	1258
5	Rolling force	kN	8940	711.86	7430	10,590
6	Rolling speed	m/s	8.70	0.01	8.65	8.74
7	Roll shifting	mm	96	2.02	93	103
8	Target profile	μm	66	1.15	62	69
9	Yield strength	MPa	457	7.10	433	482

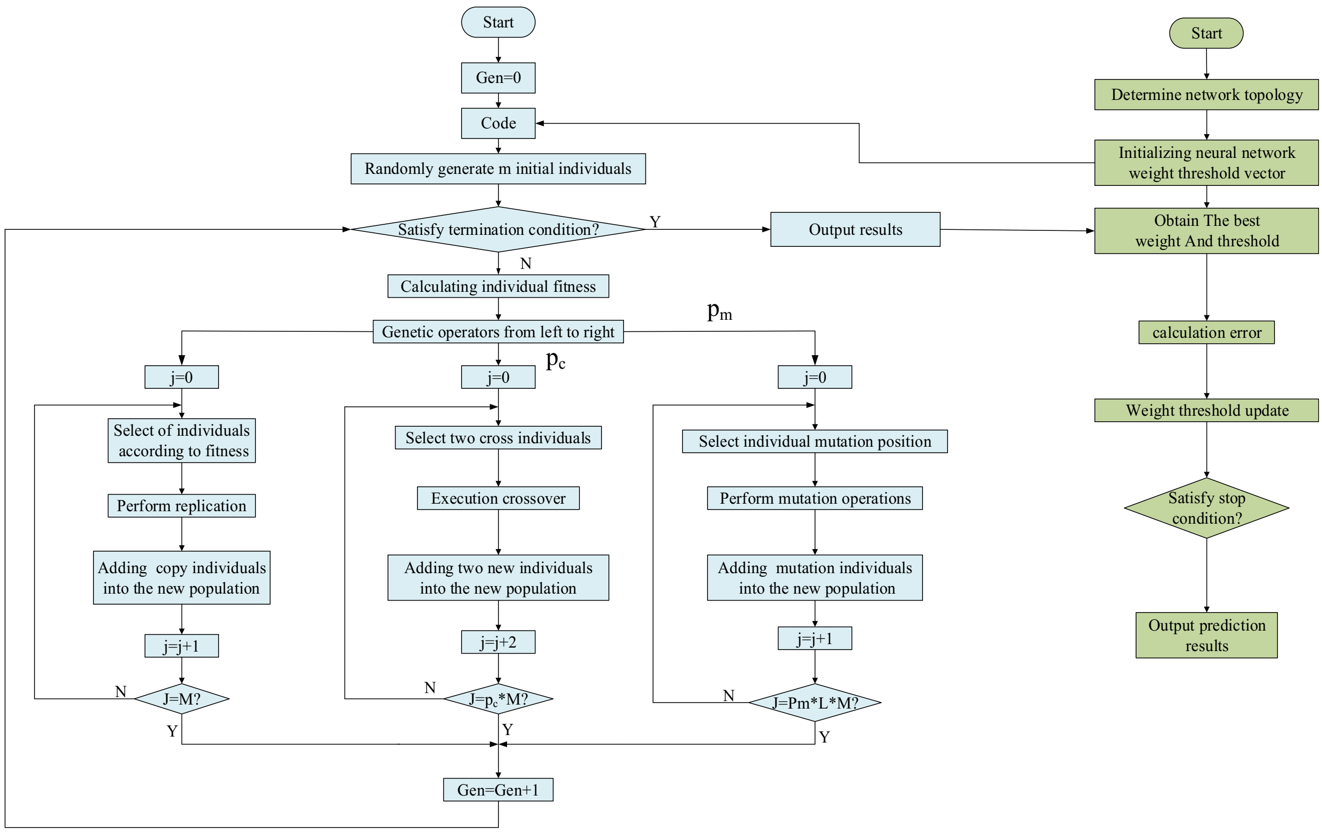

Fig. 3 The flow chart of the ANN optimization process based on GA

vector dimension for input layer and output layer, respectively, w_{ij} and w_{kj} are the weights of input layer to hidden layer and the weights of hidden layer to output layer, respectively. θ and a_k are the biases of hidden layer and output layer, respectively.

Tangent sigmoid ("tansig"), log-sigmoid ("logsig"), and linear ("Purling") are the most common used transfer functions, that is activate function, to solve regression problems [18]. In this study, the performance of the model with different transfer function combinations will be discussed. "Logsig" and "tansig" formula are as follows:

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{3}$$

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{4}$$

The second stage is the error backpropagation. The output error of the whole network is calculated according

Table 2 Some ANN learning algorithm for the test

No	Abbreviation	Learning algorithm
1	trainbfg	BFGS Quasi-Newton
2	*trainbr*	*Bayesian regulation*
3	traincgf	Conjugage gradient backpropagation with Fletcher-Reeves restarts
4	traincgp	Conjugage gradient backpropagation with Polak-Ribiere restarts
5	traingd	Gradient descent
6	traingdx	Gradient descent with momentum and adaptive learning rate (LR)
7	trainlm	Levevberg-Marquardt
8	trainoss	One step secant
9	trainrp	Rprop
10	trainscg	Scaled conjugate gradient
11	traincgb	Gradient descent with Beale-Powell restarts

Table 3 Comparison of different training algorithms

Training algorithm	Target reached	Number of epochs	R		
			Training set	Testing set	All
trainbfg	0.0617	4	0.8957	0.9025	0.8968
trainbr	*0.0608*	*2*	*0.9645*	*0.9653*	*0.9647*
traincgf	0.0890	5	0.8681	0.8375	0.8608
traincgp	0.0768	2	0.8624	0.8539	0.8631
traingd	0.0999	37	0.8213	0.8555	0.8200
traingdx	0.0971	35	0.8324	0.8276	0.8302
trainlm	0.0652	1	0.9473	0.9410	0.9460
trainoss	0.0757	7	0.8983	0.9122	0.8990
trainrp	0.0798	5	0.8563	0.8496	0.8571
trainscg	0.0675	4	0.8830	0.8710	0.8742
traincgb	0.0660	4	0.8841	0.8658	0.8844

to the error gradient descent method to adjust the weights and biases of each layer until output of the modified network is close to the expected value. For sample p, the error criterion function is as follows [19]:

$$E_p = \frac{1}{2}\sum_{k=1}^{L}\left(T_k^p - O_k^p\right)^2 \tag{5}$$

The total MSE criterion function for training samples P is as follows [20]:

$$E_P = \frac{1}{2P}\sum_{p=1}^{P}\sum_{k=1}^{L}\left(T_k^p - O_k^p\right)^2 \tag{6}$$

where T_k^p and O_k^p are target value and predicted value, respectively.

The learning process ends when the total error function reaches the desired value or meets the desired requirements [21].

2.2 Database of neural network

The performance of neural networks is largely dependence of the input samples. In order to build a network with strong generalization ability, the sample data must choose appropriate and a representative. If the representative of the sample set is poor, such as many contradictory and redundant samples, network performance cannot achieve the desired effect. The final stand rolling data of a 1580-mm hot rolling process of stainless steel mill are collected to use for experiments. According to the three sigma criterion, the error data and noise data are removed to get the on-line data of 1144 pieces of steel. Figure 2 shows neural network data collection and application. Table 1 shows the data distribution statistics of minimum, maximum, average, and standard deviations for each input variables. Seventy seven percent (1110) of them are used as the training set, which is used to adjust the weights and biases of the network, and the remainder is used as the testing set to test network generalization performance.

The data need to be normalized when the training samples are input into the network. The purpose is to cancel the difference between the numbers of different dimensional data, and to avoid prediction error increase because of the great difference between input and output data, using the following formula to data normalizing:

$$x_i = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{7}$$

where $x_{\max}$ and $x_{\min}$ are the maximum and minimum number of data sequences; x_i is the normalization value of the data i.

The number of neurons in the input layer is determined by the dimension of the input vector. According

Table 4 Comparison of different numbers of hidden layers

Number of hidden layers	Target reached	Number of epochs	R		
			Training set	Testing set	All
1	*0.00992*	*43*	*0.9830*	*0.9781*	*0.9822*
2	0.01	78	0.9832	0.9702	0.9817
3	0.0987	79	0.9826	0.9768	0.9817

Table 5 The influence of different transfer function combinations on network performance

Function type in hidden and output layer	Target reached	Number of epochs	*R*		
			Training set	Testing set	All
[Logsig-logsig]	0.1280	1000	0.8859	0.8881	0.8862
[Logsig-tansig]	0.0099	28	0.9813	0.9767	0.9823
[*Logsig-purelin*]	*0.0098*	*49*	*0.9829*	*0.9792*	*0.9824*
[Tansig-logsig]	0.1280	165	0.8840	0.8774	0.8830
[Tansig-tansig]	0.0097	28	0.9814	0.9607	0.9830
[Tansig-purelin]	0.00986	43	0.9825	0.9742	0.9822
[Purelin-logsig]	0.1180	1000	0.8800	0.8779	0.8797
[Purelin-tansig]	0.0360	1000	0.9690	0.9623	0.9679
[Purelin-purelin]	0.0184	1000	0.9677	0.9631	0.9670

to the characteristics of network structure, the input data are selected to have a great influence on the bending force and the physical quantity can be determined by measuring or calculating. The input units of the model are entrance temperature, entrance thickness, exit thickness, strip width, the rolling force, rolling speed, roll shifting, target profile, and yield strength. The output unit is selected as the work roll bending force.

3 Genetic algorithm optimization scheme

Although BP algorithm is considered as an excellent algorithm for its good fault tolerance and adaptive capability, it also has obvious shortcomings, including that the convergence speed is slow and BP algorithm falls into local minimum point easily. In order to ameliorate these disadvantages, different heuristic algorithms are proposed to optimize the BP network [22]. GA is a kind of parallel stochastic search optimization method which is based on the genetic mechanism and biological evolution theory [23]. It is the nature of "evolution, the theory goes, guarantees survival to the fittest" which was introduced to optimize the parameters of forming the encoding series group. GA uses the alternation of selection, crossover, and mutation operator to obtain the global optimum by fast searching in the whole solution space. Therefore, the neural network based on GA can fully combine the advantages of the two algorithm to improve the performance of the bending force prediction model [24]. GA optimization BP neural network is divided into three parts: determining BP neural network structure, GA optimization, and BP neural network prediction. GA optimization network flow chart is shown in Fig. 3.

3.1 Population initialization

Each individual encodes a real string by adopting real coding method in this paper. String is composed of four parts: the weights of input layer to the hidden layer, the biases of hidden layer, the weights of hidden layer to output layer, and the bias output layer. The individual

Table 6 Performance details of different ANN architectures

ANN architecture	Network performance				
	Performance goal achieved	Number of epochs	*R*		
			Training	Testing	All
9-3-1	0.01130	255	0.9805	0.9822	0.9807
9-5-1	0.00998	73	0.9831	0.9748	0.9820
9-7-1	0.01000	258	0.9825	0.9812	0.9825
9-9-1	0.00980	49	0.9832	0.9827	0.9831
9-11-1	*0.00958*	*56*	*0.9839*	*0.9829*	*0.9837*
9-13-1	0.00981	37	0.9831	0.9774	0.9823
9-15-1	0.00995	40	0.9832	0.9740	0.9819
9-20-1	0.00975	47	0.9832	0.9776	0.9824

Table 7 The best configuration for ANN

Network	Feed forward backpropagation network
Learning algorithm	Bayesian regulation
Transfer function for hidden layer	"Logsig" function
Transfer function for output layer	"Purelin" function
Performance function	MSE
Number of input layer neurons	9
Number of hidden layer neurons	11
Number of output layer neurons	1

includes all weights and bias value of neural network, when the network structure is determined, a structure with certain weights and bias value will be formed.

3.2 Fitness function

Fitness is a measure of the degree of each individual in the group that can help to find the optimal solution in optimization process. Individuals with high fitness are more likely to inherit to the next generation. A measure of individual fitness is called fitness function. The sum of absolute prediction error between the output and the expected value is set as fitness F. The calculation formula is as follows:

$$F = \frac{1}{\sum_{i=1}^{n} \left| y_i^* - \hat{y}_i \right|} \tag{8}$$

Where n is the output node number of network; y_i^* and $\hat{y}_i$ are the expected output and predicted output for the ith node of network, respectively.

3.3 Select operation

The selection process means selecting individuals with a strong vitality in the group. Select operator is used to choose individuals in the population. The roulette wheel selection method is adopted which formula is as follows:

$$p_i = \frac{F_i}{\sum_{j=1}^{N} F_i} \tag{9}$$

Where F_i is the fitness value of ith individual, N is the number of population.

3.4 Crossover operation

Crossover is also called recombination, and the crossover operation in GA is that the two pairs of chromosomes are exchanged with each other in some way. The real number crossover operation method is adopted. Crossover operation method of the kth chromosome A_k and the lth chromosome A_l in the j position are as follows:

$$A_{kj} = A_{kj}(1-\eta) + A_{lj}\eta \tag{10}$$

$$A_{lj} = A_{lj}(1-\eta) + A_{kj}\eta \tag{11}$$

where η is a random number between 0 and 1.

3.5 Mutation operation

The mutation operation in GA is to replace the gene value of some loci in the chromosome coding sequence

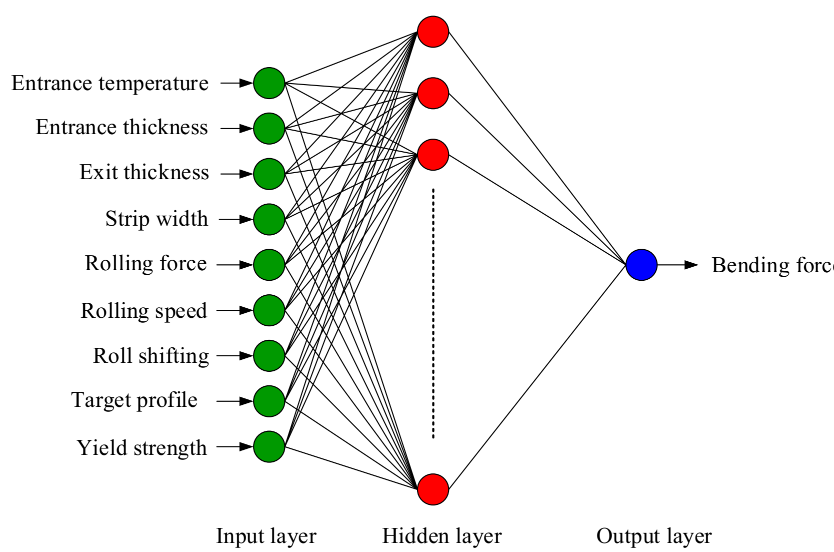

Fig. 4 Schematic illustration of the neural network structure

Table 8 Parameter design of the L9 Taguchi orthogonal analysis for GA

Experiment no.	Design set			Design level		
	Population size	Crossover probability	Mutation probability	Population size	Crossover probability	Mutation probability
1	20	0.7	0.025	1	1	1
2	20	0.8	0.05	1	2	2
3	20	0.9	0.1	1	3	3
4	30	0.7	0.025	2	1	1
5	30	0.8	0.05	2	2	2
6	30	0.9	0.1	2	3	3
7	40	0.7	0.025	3	1	1
8	40	0.8	0.05	3	2	2
9	40	0.9	0.1	3	3	3

with other alleles of the locus. The gene A_{ij} is selected to mutation, mutation operation are as follows:

$$A_{ij} = \begin{cases} A_{ij} + \left(A_{ij} - A_{\max}\right) f(g) & r_1 > 0.5 \\ A_{ij} + \left(A_{\min} - A_{ij}\right) f(g) & r_1 \leq 0.5 \end{cases} \tag{12}$$

$$f(g) = r_2 \left(1 - \frac{G}{G_{\max}}\right)^2 \tag{13}$$

where $A_{\max}$ is the upper bound of A_{ij}; $A_{\min}$ is lower bound of A_{ij}; G is the current iteration times; $G_{\max}$ is maximum times of evolution; r_1 and r_2 are random number between 0 and 1.

4 Model development and comparative analysis

After establishing the GA-ANN model structure, the next stage is to find the parameter configuration that enables the network model to have the best generalization capability. These parameters include the network learning algorithm, the number of hidden layers, the optimal combination of transfer function for hidden layer and the output layer, and the number of neurons in the hidden layer. The generalization ability of network under different parameters were checked by correlation coefficient (R). In general, the R value is larger, the generalization ability of the network is higher. In this paper, a large number of experiments were carried out in order to obtain the best network parameters with good generalization ability [14]. The experiments were completed on a computing platform equipped with MATLAB software.

4.1 Determining the best configuration for artificial neural network

Experiments were first conducted to determine the optimal network learning algorithm, the influence of the common learning algorithms in Table 2 on the generalization ability of model was tested. In addition to the learning algorithm, other parameters of the network remain the same, specifically, single hidden layer network. The number of neurons in hidden layer was 11, the transfer function of hidden layer and output layer were “logsig” and “purelin,” respectively, the target error MSE and learning rate were set to 0.01 and 0.1, respectively. The test results are shown in Table 3.

According to Table 3, the network based on “trainbr” learning algorithm has fast iterative speed in the training process, and the correlation coefficients both in the

Table 9 Taguchi analysis results

Design level	Factors		
	Population size	Crossover probability	Mutation probability
1	−38.89	−38.69	−38.85
2	−38.72	−38.78	−38.41
3	−38.71	−38.86	−39.07
Delta values	*0.18*	*0.17*	*0.66*
Rank	2	3	1

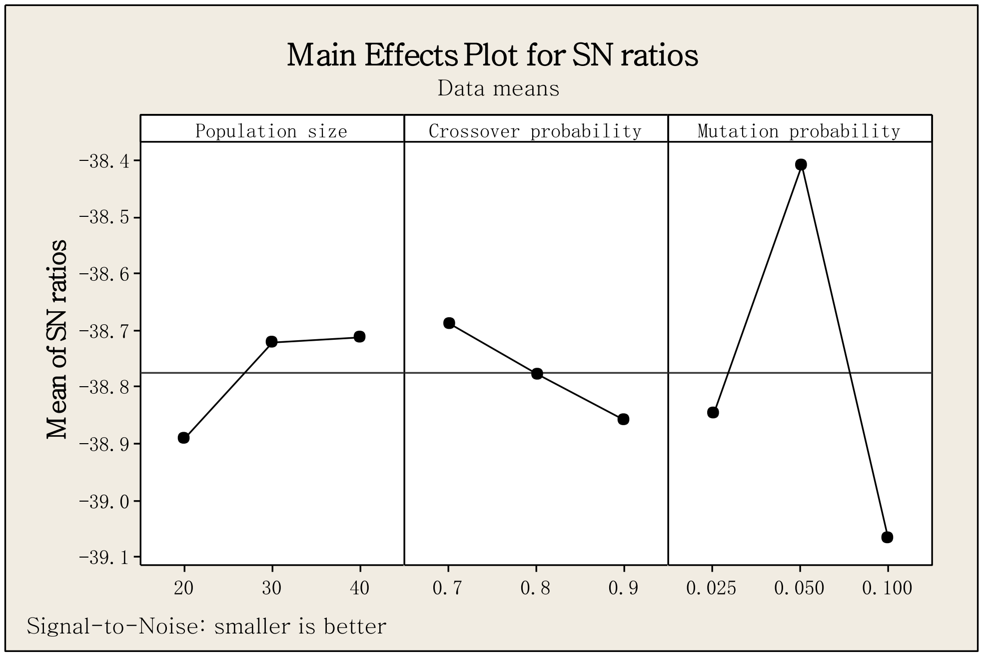

Fig. 5 The mean of SN rations for each factor and its levels

training set and testing set are higher than those of other learning algorithms. Therefore, the optimal learning algorithm is chosen as "trainbr."

The second stage experiments were designed to find the best hidden layer number of the network. In this part, the network performance was tested when the network layer number was 1, 2, and 3. The model used the "trainbr" learning algorithm, and the other parameters remain the same as the previous experiments. The experimental results are shown in Table 4.

It is obvious from Table 4 that increasing the hidden layer number of network not only cannot improve the network performance but also increase the number of network iterations. In view of this, the choice of the single hidden layer network is appropriate.

Then, the experiments were carried out to determine the optimal combination of the activation functions of the network hidden layer and the output layer. The different combinations of "logsig," "tansig," and "Purling" should be used for network testing. The other network parameters in the test were the same with the previous experiments. The details of the test results are listed in Table 5.

According to Table 5, when the transfer functions of the hidden layer and the output layer are "logsig" and "Purling," the correlation coefficient of the network is

Table 10 Prediction of Taguchi results

Population size	Crossover probability	Mutation probability	SN ratios	Data means
20	0.7	0.025	−38.8760	87.9063
20	0.8	0.050	−38.5261	84.5307
20	0.9	0.100	−39.2646	92.0081
30	0.7	0.050	−38.2715	81.7931
30	0.8	0.100	−39.0172	89.3521
30	0.9	0.025	−38.8753	87.8827
40	*0.7*	*0.05*	*−38.2604*	*81.6843*
40	0.7	0.100	−38.9190	88.3114
40	0.8	0.025	−38.7844	86.9235
40	0.9	0.050	−38.4273	83.4664

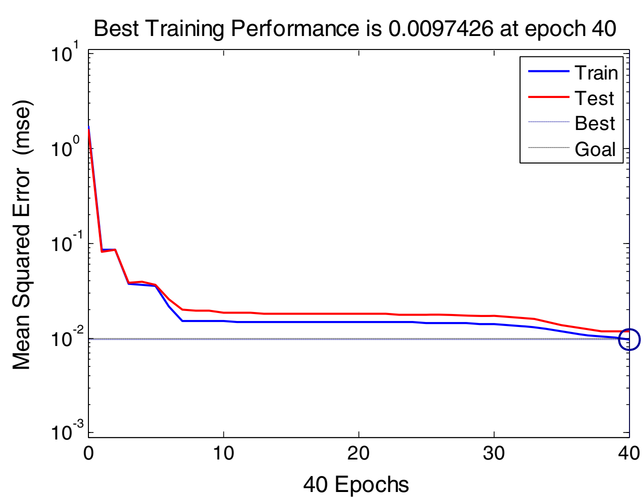

Fig. 6 The training performance of the best configuration of GA-ANN

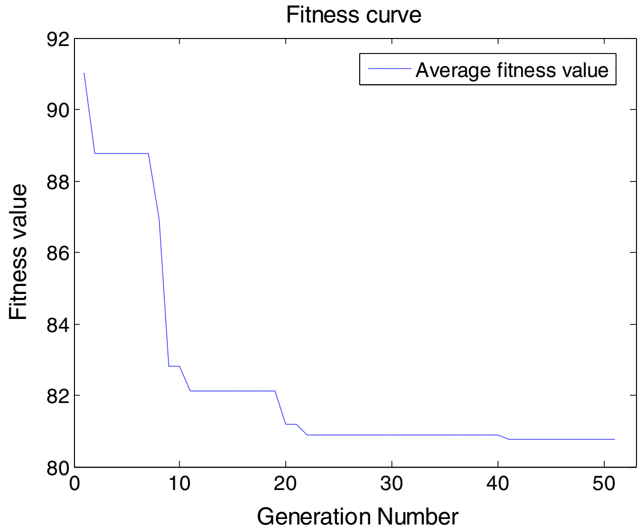

Fig. 7 The curve of average fitness change

higher and the number of iterations is less, which is the ideal choice under the conditions of this paper.

The last experiments were conducted to find the required number of neurons in the hidden layer. The number of nodes in hidden layer has a great influence on the performance of neural network. Generally, more hidden layer nodes can bring better network performance, but also may lead network training time to become longer. Less hidden layer nodes may make the network generalization ability insufficient. At present, there is not an ideal analytic formula which can be used to determine the number of nodes in the hidden layer neurons, which is usually based on the empirical formula. In this paper, the performance of neural networks with 3–20 neurons were investigated considering the presence of a single layer structure. The results for each combination are given in Table 6.

As can be seen from Table 6, when the number of neurons in the network hidden layer is 11, the performance is the best. So far, the parameters of the neural network all have been determined, listed in Table 7 specifically, and the topology of the network is shown in Fig. 4.

4.2 Determining the parameter of genetic algorithm

The crossover probability controls the crossover operator in GA, in other words, it controls the frequency at which the crossover operation is used. The mutation algorithm is a kind of repair and supplement of some genetic genes

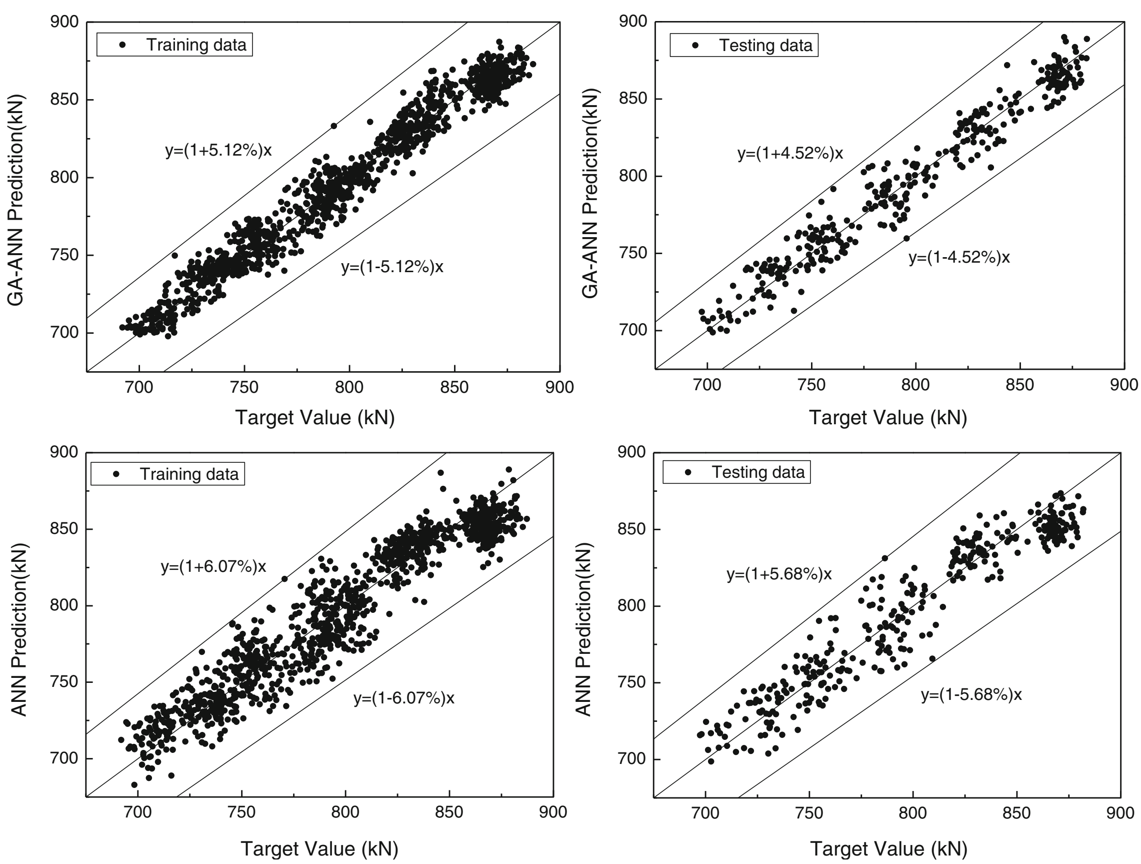

Fig. 8 Scatter plot of GA-ANN and ANN model in prediction of training set and testing set

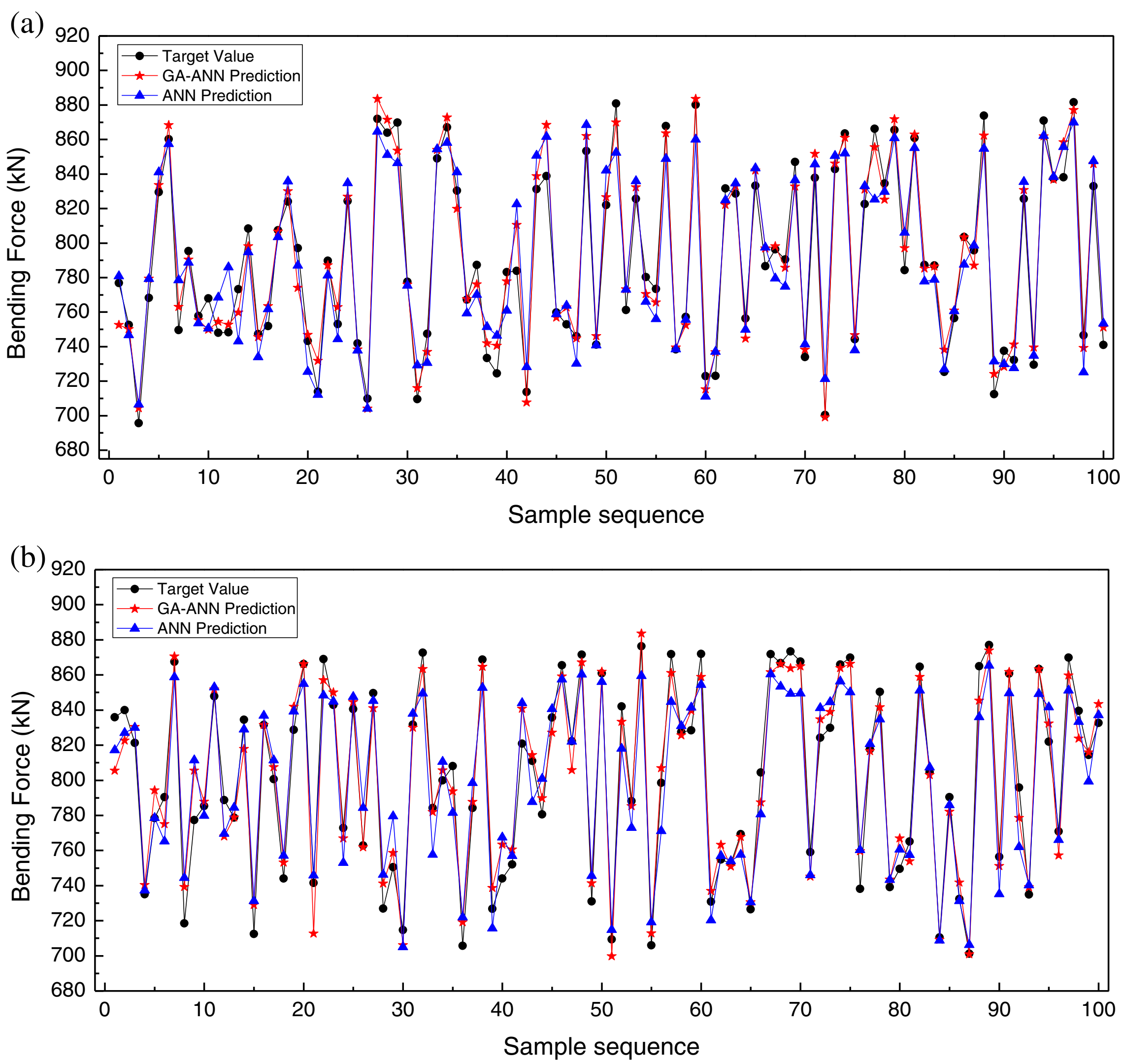

Fig. 9 Model capability of predicting accurate bending force for **a** training set and **b** testing set

which may be lost in the process of crossover. It can prevent the GA from converging to local optimal solution as soon as possible. Mutation probability controls the frequency of mutation operation. Only the crossover operator

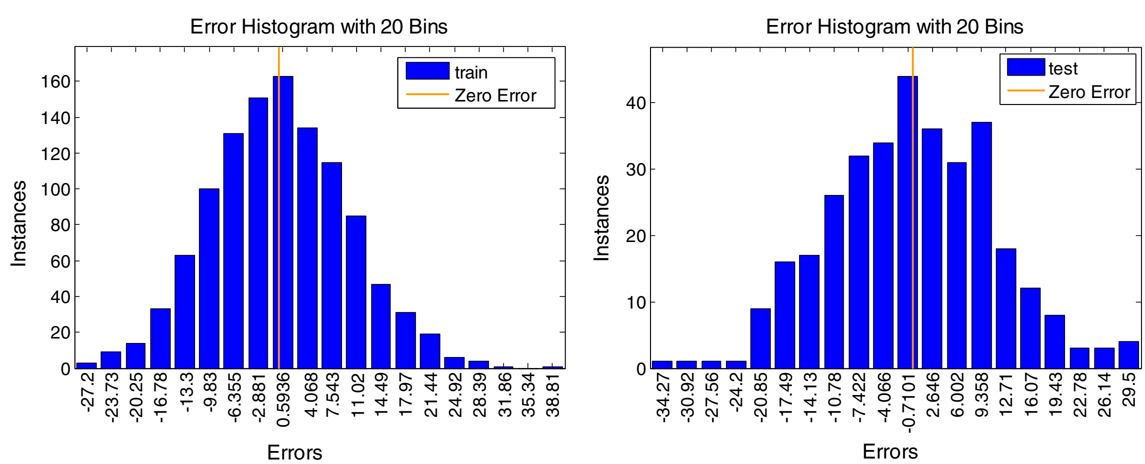

Fig. 10 Error histogram of predicted bending force from actual target results for training set and testing set

Table 11 The error indexes of GA-ANN model for training and testing predictions

Error type	Training set		Testing set	
	GA-ANN	ANN	GA-ANN	ANN
R	0.9825	0.9550	0.9776	0.9580
MAE	7.7672	12.6900	8.8906	12.9836
MAPE (%)	0.9770	1.5914	1.1199	1.6241
RMSE	9.7734	26.6608	11.126	29.6538
Maximum percentage error (%)	5.1200	6.0661	4.5200	5.6843

and mutation operator cooperate with each other to complete the global search and local search of the search space, so that the GA can search the optimal solution with good search performance. Taguchi method can economically solve industrial problem. In this paper, in order to obtain the optimum combination parameters of GA, the Taguchi method of orthogonal design was adopted to elicit the interaction between parameters [25]. An important index in Taguchi analysis is the signal-to-noise ratio, it is used to analyze experimental data and find the optimum parameter combinations [26]. In this section, three main parameters of the GA were designed for three levels to be investigated, including population size, mutation rate, and crossover rate. The fitness function value of GA was used as Taguchi analysis response. L9 orthogonal design was selected to carry out experiments and calculate the effects of factors, specific design is shown in Table 8. Minitab software was used to analyze and calculate the signal-to-noise ratio of each factor. According to the situation in this paper, the signal-to-noise ratio selected the criteria "Smaller is better," formula for calculating the signal-to-noise ratio is shown as follows: [27].

$$\frac{S}{N}ratio = \eta_j = -10\cdot\log\left(\frac{1}{n}\sum_{i=1}^{n} y_{ij}^2\right) \tag{14}$$

where $y_{ij} = i$th replicate of jth response; n = number of repetitions = 1,2,…,n; and j = 1,2,…,k.

According to Taguchi analysis, delta value have great influence to the response, so the mutation probability had the biggest impact on the response, which is 0.66 (as shown in Table 9). Population size and crossover rate are found as second and third important factors. Figure 5 shows the main effect diagram of signal-to-noise ratio and the Table 10 is used to show predicted Taguchi results. Optimal combination of GA parameters are found as 40, 0.7, and 0.05 for the population size, crossover rate, and mutation rate, respectively.

5 Model analysis and discussion

The neural network optimized by GA is proposed to predict the bending force in hot strip rolling. Parameters of model are determined by experiments. The model performance curve of the neural network in the training process is shown in Fig. 6, as can be seen from the graph, the best performance of the network is up to 0.009742 when the number of iterations is 40. The fitness function value with the evolution of the number of curves is shown in Fig. 7, best average fitness value is 80.7 in the process of GA optimize the weights and biases of neural network.

The regression performance of the GA-ANN and ANN model are drawn in scatter diagram as Fig. 8 which shows the training set and the testing set data. It is

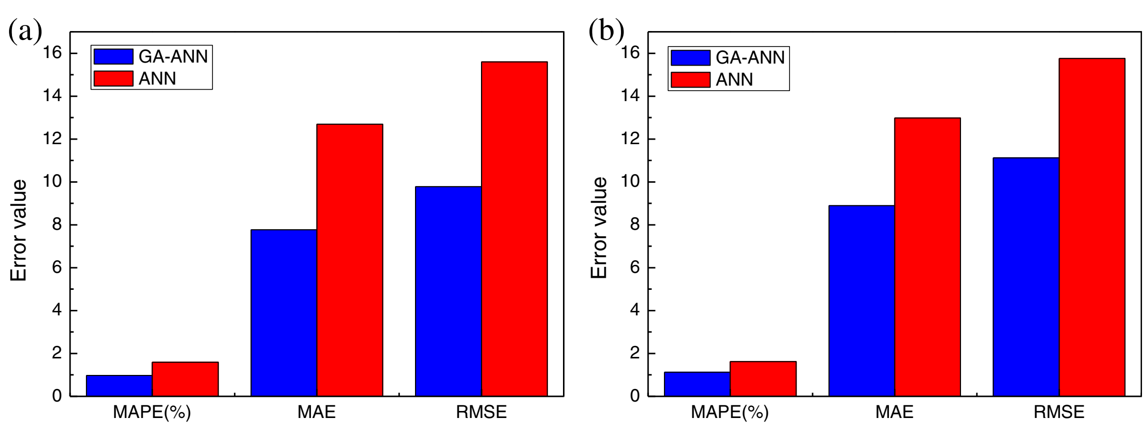

Fig. 11 Error histogram comparison between GA-ANN model and ANN model

obvious from the diagram that data points of GA-ANN model are evenly distributed on both sides of the $y = x$ line whether the training data used or the testing data that does not participate in the training when the model is established. The regression effect of ANN model is bad by comparison with GA-ANN because of its wide error band. Figure 8 also shows the percentage error deviations between the predicted and actual values of GA-ANN on the two data sets. The percentage error of the training set is less than 5.12%, and the testing set is less than 4.52%. Correspondingly, this value is 6.07 and 5.68% on the ANN model, respectively. This fully shows that the neural network model based on GA optimization is more accurate in prediction accuracy. Comparison of the predicted results with actual target ones is given in Fig. 9. As can be seen from Fig. 9, random sample prediction effect is consistent both in the training set and the testing set, it means higher prediction accuracy is obtained. Besides, whether the error distribution is reasonable, it also has a great significance to the feasibility of the proposed model. The training set and testing set error distribution histogram is shown in Fig. 10, the diagram shows that both in the training and testing set, the error distribution histogram is stable, its shape is high in the middle and low on both sides, the overall error is approximately symmetrical normal distribution.

In order to evaluate the effect of bending force prediction model more synthetically and quantitatively, the correlation coefficient, mean absolute error (MAE), and root mean square error (RMSE) are chosen as the absolute error indexes of the model. Mean absolute percentage error (MAPE) and maximum percentage error are chosen as the relative error indexes. As seen in Table 11, a high coefficient correlation and a low MAPE are obtained for the training and testing data sets. The proposed GA-ANN model for bending force prediction has correlation coefficients of 0.9825 and 0.9776 for training and testing set, respectively, the MAPE is about 0.9770 and 1.1199, respectively. Furthermore, the model also provides a reasonable MAE and RMSE both in the training set and testing set, the values of MAE in the two sets are 7.7672 and 8.8906, respectively. Similarly, the values of RMSE are 9.7734 and 11.1255 in the training set and testing set, respectively. Figure 11 shows a comparison of the correlation error indicators on the GA-ANN and ANN models. The error of GA-ANN is obviously smaller than that of ANN model. Through a comprehensive analysis of Figs 8, 9, 10, 11 and Table 11, conclusions can be drawn that the GA-ANN model proposed in this paper realized the accurate prediction of roll bending force in hot strip mill and the model shows very high generalization ability for the new samples.

6 Conclusions

An intelligence optimization algorithm is proposed using ANN and GA to prediction of the bending force accurately in hot strip rolling process, which aims to improve the quality of the final products in steel manufacturing. The neural network is trained and tested by the data collected from hot strip rolling plant. The optimal network topology is determined by several experiments. A Bayesian regulation learning algorithm with a single hidden layer and 11 neurons is found as the best performing network. Taguchi method based on sensitivity analysis is carried out to find the best parameters for GA to improve the neural network performance successfully. GA-ANN model prediction results have a high degree of consistency with the actual bending force value, and it is proved that the neural network is an effective tool for bending force prediction in hot strip rolling. The model proposed in this paper provides a new method for the bending force setting optimization and has wide industrial application potential because of the advantages of simple structure, high precision, and strong promotion.

Acknowledgements This study was financially supported by the National Natural Science Foundation of China (No.: 51634002), National Intelligent Manufacturing Equipment Development Project of China (No.: 2014-2558), and the Fundamental Research Funds for the Central Universities (No.: N140704001).

References

1. Jeon E, Kim S (2000) A study on the texturing of work roll for temper rolling. J Korean Soc Mach Tool Eng 9(4):7–16
2. Portmann NF, Lindhoff D, Sorgel G, Gramckow O (1995) Application of neural networks in rolling mill automation. Iron Steel Eng 72(2):33–36
3. Larkiola J, Myllykoski P, Korhonen A, Cser L (1998) The role of neural networks in the optimisation of rolling processes. J Mater Process Technol 80:16–23
4. Yao X (1996) Applications of artificial intelligence for quality control at hot strip mills
5. Pican N, Alexandre F, Bresson P (1996) Artificial neural networks for the presetting of a steel temper mill. IEEE Expert 11(1):22–27
6. Lee D, Lee Y (2002) Application of neural-network for improving accuracy of roll-force model in hot-rolling mill. Control Eng Pract 10(4):473–478
7. Poppe T, Obradovic D, Schlang M (1995) Neural networks-reducing energy and raw-materials requirements. Siemens Rev:24–27
8. Chun M, Biglou J, Lenard J, Kim J (1999) Using neural networks to predict parameters in the hot working of aluminum alloys. J Mater Process Technol 86(1):245–251
9. Lee D, Choi S (2004) Application of on-line adaptable neural network for the rolling force set-up of a plate mill. Eng Appl Artif Intell 17(5):557–565

10. Son J, Lee D, Kim I, Choi S (2005) A study on on-line learning neural network for prediction for rolling force in hot-rolling mill. J Mater Process Technol 164:1612–1617
11. Moussaoui A, Selaimia Y, Abbassi HA Hybrid hot strip rolling force prediction using a Bayesian trained artificial neural network and analytical models. In: American Journal of Applied Sciences, Citeseer
12. Yang Y, Linkens D, Talamantes-Silva J, Howard I (2003) Roll force and torque prediction using neural network and finite element modelling. ISIJ Int 43(12):1957–1966
13. Yang Y, Linkens D, Talamantes-Silva J (2004) Roll load prediction—data collection, analysis and neural network modelling. J Mater Process Technol 152(3):304–315
14. Bagheripoor M, Bisadi H (2013) Application of artificial neural networks for the prediction of roll force and roll torque in hot strip rolling process. Appl Math Model 37(7):4593–4607
15. Haykin S, Lippmann R (1994) Neural networks, a comprehensive foundation. Int J Neural Syst 5(4):363–364
16. Bishop CM (2006) Pattern recognition, vol 128. Machine learning
17. Deng J, Gu D, Li X, Yue Z (2005) Structural reliability analysis for implicit performance functions using artificial neural network. Struct Saf 27(1):25–48
18. Khayet M, Cojocaru C (2012) Artificial neural network modeling and optimization of desalination by air gap membrane distillation. Sep Purif Technol 86:171–182
19. Mukhopadhyay A, Iqbal A (2005) Prediction of mechanical properties of hot rolled, low-carbon steel strips using artificial neural network. Mater Manuf Process 20(5):793–812
20. Mukhopadhyay A, Iqbal A (2006) Comparison of ANN and MARS in prediction of property of steel strips. Applied Soft Computing Technologies: The Challenge of Complexity: 329–341
21. Shahani A, Setayeshi S, Nodamaie S, Asadi M, Rezaie S (2009) Prediction of influence parameters on the hot rolling process using finite element method and neural network. J Mater Process Technol 209(4):1920–1935
22. Liu C, Ding W, Li Z, Yang C (2016) Prediction of high-speed grinding temperature of titanium matrix composites using BP neural network based on PSO algorithm. Int J Adv Manuf Technol 89(5):1–9
23. Holland John H (1975) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan, Ann Arbor
24. Ma C, Zhao L, Mei X, Shi H, Yang J (2016) Thermal error compensation of high-speed spindle system based on a modified BP neural network. Int J Adv Manuf Technol 89(9): 3071–3085
25. Yuce B, Rezgui Y, Mourshed M (2016) ANN–GA smart appliance scheduling for optimised energy management in the domestic sector. Energ Buildings 111:311–325
26. Hou T, Su C, Liu W (2007) Parameters optimization of a nano-particle wet milling process using the Taguchi method, response surface method and genetic algorithm. Powder Technol 173(3):153–162
27. Khoualdia T, Hadjadj AE, Bouacha K, Abdeslam DO (2016) Multiobjective optimization of ANN fault diagnosis model for rotating machinery using grey rational analysis in Taguchi method. Int J Adv Manuf Technol 89(9):3009–3020