# IMPROVED ARTIFICIAL NEURAL NETWORK BASED ON INTELLIGENT OPTIMIZATION ALGORITHM

*Y. Xu*[*†]; *M. He*[†]

**Abstract:** Neural network based on back-propagation (BP) algorithm is a widely used prediction model. However, the nodes number of the first hidden layer, the learning rate and momentum factor are usually determined manually, which affects the forecast accuracy of network. Therefore, in this paper, to improve the forecast accuracy, firstly, the nodes number of the first hidden layer is selected adaptively based on minimizing mean square error (MSE). Secondly, improved genetic algorithm (GA) is proposed to train the learning rate and momentum factor dynamically, which includes multi-point crossover and single point mutation. Thirdly, we construct a new neural network model based on the adaptively selected nodes number of the first hidden layer, the dynamically selected learning rate and momentum factor, which is called HN-GA-BP neural network model. Finally, the proposed neural network model is used to forecast the carbon dioxide contents in China for fifty years. Experimental results demonstrate the effectiveness of the proposed HN-GA-BP neural network model.

Key words: *neural network, genetic algorithm, hidden layer node, learning rate, momentum factor*

*Received: January 12, 2018* **DOI:** 10.14311/NNW.2018.28.020
*Revised and accepted: August 14, 2018*

## 1. Introduction

Carbon dioxide is the primary anthropogenic greenhouse gases in the earth's atmosphere. With rapid progress of the industrialization and urbanization, carbon dioxide emissions produced by human activities are considered as the main reason of increase in atmospheric concentrations [1]. According to the annual survey of China Meteorological Administration, the concentration of carbon dioxide had reached the highest level in history [2]. Therefore, forecasting the carbon dioxide contents accurately is important to human society and may help in making better environmental strategies, even better for humans' daily life. Generally, most of

*Yi Xu – Corresponding author; Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui University, Hefei 230039, China, E-mail: xuyi1023@126.com

†Minghui He; School of Computer Science and Technology, Anhui University, Hefei 230601, China.

 

the traditional methods for forecasting the carbon dioxide contents are mathematical or manual calculation, which cannot forecast intelligently. Apparently, with the gradual increasing data of carbon dioxide, these traditional methods will be inadequate, even lead to inaccurate forecast results. Compared with traditional methods, neural network is an effective mathematical calculation model, which can accurately calculate like human brain's neural system. Nowadays, neural networks have been widely applied in many fields, such as energy forecasting, intelligent computing and classification problems etc [3]. Especially, neural network models have also been applied for predicting carbon dioxide contents. Moreover, neural networks have capability of fast training and accurate forecasting, which can handle many complex forecast problems effectively [4].

Generally, neural network contains three layers: the input layer, the hidden layer and the output layer, which makes the input and output of a set of samples into a nonlinear problem, using gradient descent algorithm optimization technique to get the final forecast values [5]. JP Skon provided a Multilayer Perception (MLP) neural network with back-propagation (BP) algorithm for forecasting indoor air carbon dioxide [6]. Based on neural network model with single hidden layer, Yan [7] summarized and provided the single hidden layer sigmoid feedforward neural network model (SLFNN), which has been widely used in many forecasting fields. V. Bevilacqua [8] proposed a neural network model in order to forecast carbon dioxide contents based on some input attributes. C. Gallo [9] proposed a short term method for forecasting carbon dioxide contents with back-propagation algorithm. The above mentioned neural network models were constructed based on BP algorithm (BPNN). Usually, the topology structure and training parameters are determined by manual setting or experience, which affects the forecasting accuracy. So how to design the network structures and set parameters reasonably, which is crucial for the prediction accuracy.

However, the network structures and parameters are difficult to determine, which restricts the network performance [10]. For the network structures, the hidden layers are the most important, and the number of hidden layers and the number of hidden layer nodes will determine the overall structures of a network and the way of information transmission. Generally, neural network with single hidden layer has fast training speed [11, 12], but it cannot deal with nonlinear problems effectively. However, with the increase of layer number, the computational complexity of the network will increase at an exponential rate, which affects the efficiency of network training [13]. Therefore, in practical applications, double hidden layer is a good choice, which can deal with nonlinear problems more effectively. Besides, the nodes number of hidden layer is crucial for network training accuracy. However, existing neural network models often set the nodes number of hidden layer depend on artificial or mathematical formulas [14–16], which is too subjective and cannot adjust dynamically. In addition, learning rate and momentum factor are two important parameters for training accuracy of neural network. Usually, learning rate is small to maintain the convergence speed and stability of neural network training [17]. On the contrary, if the learning rate is too large, this may lead to instability of the network training. Similarly, setting momentum factor appropriately can make the network weight update quickly, and avoid the network falling in local minima [18–20]. However, in above models, the learning

rate and momentum factor are set manually, which leads to imprecise training results. To improve the prediction performance of neural network, many researchers use intelligence optimization algorithms to construct network structures and set parameters. A. Jamali [21] combined the neural network and genetic algorithm to forecast the emission of carbon dioxide with multi-object optimization method. S. Yu [22] applied genetic algorithm neural network (GANN) to forecast carbon dioxide contents with recoded chromosome of training parameters. Moreover, based on the swarm intelligence method and BP neural network, W. Sun [23] combined particle swarm optimization with BP neural network to construct (PSO-BP) model, and applied the proposed model to forecast carbon dioxide contents.

Therefore, in this paper, to improve the forecast accuracy, we optimize the neural network from two aspects. Firstly, according to the mean square error of each iteration of network training, the nodes number of the first hidden layer is selected adaptively, which can minimize the mean square error. Secondly, using mean square error function to define the fitness function, improved genetic algorithm (GA) is proposed to train the learning rate and momentum factor dynamically, which includes multi-point crossover and single point mutation. On this basis, the HN-GA-BP neural network model is proposed based on the adaptively selected nodes number of the first hidden layer, the dynamically selected learning rate and momentum factor. The proposed network model is applied to forecast the carbon dioxide contents in China. Experimental results demonstrate the effectiveness of the proposed HN-GA-BP neural network model.

The remainder of this paper is organized as follows. In Section 2, we give the method to select the first hidden layer nodes number adaptively. In Section 3, we give the method of training learning rate and momentum factor dynamically using the improved genetic algorithm. Based on the adaptively selected nodes number of the first hidden layer, the dynamically selected learning rate and momentum factor, we construct HN-GA-BP neural network model. Some experimental results are discussed in Section 4. In Section 5, we conclude this paper.

## 2. The method of selecting the first hidden layer nodes number adaptively

Neural network is an effective mathematical model designed as the structure of the nervous system. The model was presented for the first time by McCulloch and Pitts [24] and involves a set of nodes, a set of weights and activation functions. Neural networks simulate the human brain and consist of an interconnected network of neurons and synapses. Neurons accept inputs from other neurons and produce an output by firing their synapse. Neurons perform a weighted sum on all of their inputs and then the result goes through an activation function to produce an output. Neural network is organized into layers. There is an input layer, an output layer and one or more hidden layers. The hidden layers are the fundament and important component of the neural network that perform the actual computations.

Generally, the network performance is reflected by the mean square error of the training results. If the mean square error is small, then the network topology structure is reasonable. Hence, in this paper, we select the nodes number of the

first hidden layer adaptively according to the mean square error of each iteration of network training, which can avoid the blindness of manually setting the nodes number. However, if we use the same method to set the nodes number of the second hidden layer, the training time will increase significantly. In addition, the nodes number of the second hidden layer depends on the number of input nodes. So in this paper, we set the nodes number of the second hidden layer as $(2m + 1)/2$ ($m$ is the nodes number of the input layer) according to Hecht-Nielsen's method [25], which is simple and effective.

The topology structure of back-propagation neural network with double hidden layers is shown in Fig. 1.

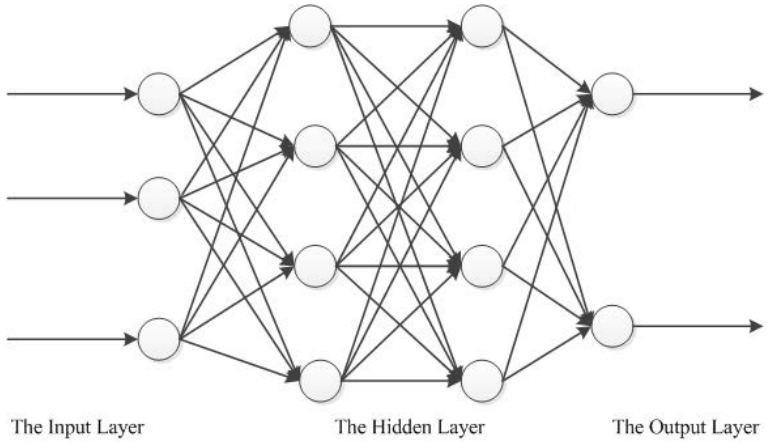


**Fig. 1** *The topology structure of back-propagation neural network with double hidden layers.*

The back-propagation neural network is a supervised learning method that uses a gradient descent method to minimize the error between the predicted output and the target output. Repeat above calculation and modification of the weights and thresholds until the mean square error function reaches a termination condition. According to the way of information transmission in back-propagation neural network with single hidden layer [5], the information transmission in back-propagation neural network with double hidden layers is defined as follows.

$m$ : The input layer nodes number
$n$ : The first hidden layer nodes number
$s$ : The second hidden layer nodes number
$N$ : The total number of sample
$x_j^p$ : The $p$-th sample input of $j$-th node in input layer ($1 \leq p \leq N$)
$w_{ij}$ : The weight between $j$-th node in input layer and $i$-th node in the first hidden layer
$w_{ki}$ : The weight between $i$-th node in the first hidden layer and $k$-th node in the second hidden layer
$w_{qk}$ : The weight between $k$-th node in the second hidden layer and $q$-th node in output layer

$b_i$ : The threshold of $i$-th node in the first hidden layer
$a_k$ : The threshold of $k$-th node in the second hidden layer
$b_q$ : The threshold of $q$-th node in output layer
$\varphi(x)$ : The excitation function
$Q$ : The output layer nodes number
$o_q^p$ : The $p$-th sample target output of $q$-th node in output layer $(1 \leq q \leq Q)$

The input of $i$-th node in the first hidden layer is defined as [26]

$$\text{net}_i^p = \sum_{j=1}^{m} w_{ij} x_j^p + b_i. \tag{1}$$

The output of $i$-th node in the first hidden layer is defined as

$$y_i^p = \varphi(\text{net}_i^p) = \varphi\left(\sum_{j=1}^{m} w_{ij} x_j^p + b_i\right). \tag{2}$$

The input of $k$-th node in the second hidden layer is defined as

$$\text{net}_i^p = \sum_{i=1}^{n} w_{ki} y_i^p + a_k. \tag{3}$$

The output of $k$-th node in the second hidden layer is defined as

$$y_k^p = \varphi(\text{net}_k^p) = \varphi\left(\sum_{i=1}^{n} w_{ki} y_i^p + a_k\right). \tag{4}$$

The output of $q$-th node in output layer is defined as

$$y_q^p = \varphi(\text{net}_q^p) = \varphi\left(\sum_{k=1}^{s} w_{qk} y_k^p + b_q\right). \tag{5}$$

Combined with Eqs. (1), (2), (3), (4), (5), the mean square error in output layer is defined as

$$\text{MSE} = \frac{1}{N}\frac{1}{Q}\sum_{p=1}^{N}\sum_{q=1}^{Q}(y_q^p - o_q^p)^2. \tag{6}$$

The mean absolute error is defined as

$$\text{MAE} = \frac{1}{N}\frac{1}{Q}\sum_{p=1}^{N}\sum_{q=1}^{Q}\left|y_q^p - o_q^p\right|. \tag{7}$$

The mean absolute percentage error is defined as

$$\text{MAPE} = \frac{1}{N}\frac{1}{Q}\sum_{p=1}^{N}\sum_{q=1}^{Q}\left|\frac{y_q^p - o_q^p}{o_q^p}\right|. \tag{8}$$

**Algorithm 1** The method of selecting the nodes number of the first hidden layer adaptively.

**Input:** Initial parameters and functions of back-propagation neural network.
**Output:** The nodes number of the first hidden layer.
**Step1:** Set $S$ as the maximum nodes number of the first hidden layer.
**Step2:** Initialize iteration number $i = 1$, mean square error vector $res\,(i) = 0$.
**Step3:** Initialize the back-propagation neural network, set the nodes number of the first hidden layer $number_1 = i$, set the nodes number of the second hidden layer $number_2 = (2m + 1)/2$($m$ is the nodes number of the input layer), set the activation function in the first hidden layer and the second hidden layer is Sigmoid, set the activation function in output layer is Purelin, set the learning function is Trainlm.
**Step4:** Train the back-propagation network, then according to formula (6), calculates mean square error (MSE) of $i$-th training results, $res\,(i) = \text{MSE}$.
**Step5:** Iteration number $i = i+1$, when $i = S$, jump to Step6; Otherwise repeat Step3 ˜Step5.
**Step6:** For $1 \leq i \leq S$, calculate the minimum value of $res\,(i)$, then record $i$.
**Step7:** If $i < (m + Q)/2$, $i = (m + Q)/2$; if $i > (2m + 1)$, $i = (2m + 1)$, where $Q$ is the nodes number of output layer.
**Step8:** Output $i$, which is the selected nodes number of the first hidden layer.

According to the value of mean square error, the method of selecting the nodes number of the first hidden layer adaptively is described as Algorithm 1.

Note that, in this paper, the nodes number of input layer ($m$) are more than the nodes number of output layer ($Q$). Hence, we set $[(m + Q)/2, (2m + 1)]$ as the range of the nodes number of first hidden layer in Algorithm 1, which can maintain the convergence of neural network [27, 28].

## 3. Improved genetic algorithm to train learning rate and momentum factor

Genetic algorithm was proposed by J.H. Holland in 1975 inspired by species evolution [29]. Genetic algorithm simulates the evolution process of biology in nature, which means that survival of the fittest in natural selection. First, the genetic algorithm initializes the population based on the chromosome coding and constructs the fitness function according to the objective of problem solving. Second, each individual in the population exchanges partial genes with other individual with a specified probability, which is crossover operation. Then, some genes in this individual can mutation with a specified probability, which is mutation operation. Finally, after crossover operation and mutation operation, each individual has been evolved and has a new chromosome structure [30]. After many iterations of evolution, the genetic algorithm will get the best individual.

In this paper, we use the improved genetic algorithm to train the learning rate and momentum factor dynamically, which can avoid the blindness of manually setting. We define the learning rate as $Lr$, and the momentum factor as $Mc$. In

general, $Lr$ is set small to maintain the convergence speed and stability of network training [17]. However, if $Mc$ is too small, the network weights will be updated slowly [18]. Usually the value of $Lr$ should be smaller than $Mc$. For example, Wang [18] set $Mc$ in the rage of 0.0 to 1.0 and set $Lr$ to be 0.01. Masood [19] set the value of $Lr$ to be 0.1 to maintain the convergence speed and stability of the network training. Narayanan [20] indicated that the value of $Lr$ should be limited and cannot be larger. In improved genetic algorithm, we use binary representation to denote the chromosome. We suppose the accuracy of solution (training results) are three decimal. Binary representation of chromosome should satisfy Len $= L/A$, where Len is the length of binary representation of solution; $L$ is the interval length of solution domain; $A$ is the accuracy of solution [29]. In order to ensure accuracy and make the binary representation as short as possible, we set the solution domain of $Lr$ in $[0.05, 0.1]$ and the solution domain of $Mc$ in [0.6, 1.0] by experience and many times experiments. Using binary representation and to ensure accuracy, the interval [0.05, 0.1] should be divided into 50 equal parts at least. As $32 = 2^5 < 50 < 2^6 = 64$, the chromosome of leaning rate $Lr$ requires 6 binary bits. Similarly, the interval $[0.6, 1.0]$ should be divided into 400 equal parts at least. As $256 = 2^8 < 400 < 2^9 = 512$, the chromosome of momentum factor $Mc$ requires 9 binary bits. Apparently, the genetic representation of one individual has 15 binary bits. The genetic representation of learning rate and momentum factor is shown in Fig. 2.

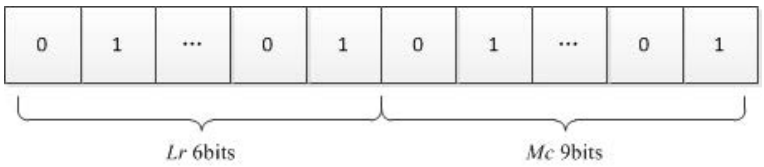


**Fig. 2** *The genetic representation of learning rate and momentum factor.*

In genetic algorithm, the individual fitness determines the probability that an individual is selected [17]. In this paper, the fitness function is defined as:

$$f = \text{MSE} = \frac{1}{N}\frac{1}{Q}\sum_{p=1}^{N}\sum_{q=1}^{Q}(y_q^p - o_q^p)^2. \tag{9}$$

In the process of dynamically selecting learning rate and momentum factor, we use multi-point crossover as the crossover operation, which crosses arbitrary gene fragments of two chromosomes randomly with a probability between $[0, 1]$. Note that, the length of gene fragment is no more than the maximum length of chromosome. We use single point mutation as the mutation operation, which selects arbitrary point in individual chromosome randomly to mutation with a probability between $[0, 1]$. Then, we use fitness function to calculate individual fitness of new chromosome and select the individual with the minimum fitness to next generation. Repeat above process until we obtain the best individual with minimum fitness, which includes learning rate and momentum factor. The method of training learning rate and momentum factor by improved genetic algorithm is described in Algorithm 2.

**Algorithm 2** The method of training learning rate and momentum factor by improved genetic algorithm.

**Input:** Initial parameters and fitness function of improved genetic algorithm.
**Output:** The learning rate and momentum factor.

**Step1:** Set $N_{it}$ as the maximum evolution number, set $N_{size}$ as the maximum population number, set current evolution number $i = 1$, set the genetic representation of learning rate and momentum factor as Fig. 2, set the fitness function is MSE as formula (6).

**Step2:** Calculate initial fitness of all individuals, and record the minimum fitness.

**Step3:** If current evolution number $i \leq N_{it}$ and individuals meet crossover rate or mutation rate, individuals perform multi-point crossover or single point mutation respectively.

**Step4:** Calculate individual fitness of new chromosome and select the individual with the minimum fitness to next generation.

**Step5:** $i = i + 1$. If $i > N_{it}$, jump to Step6; otherwise return to Step3.

**Step6:** Output the learning rate and momentum factor.

In the following, based on the adaptively selected nodes number of the first hidden layer, the dynamically selected learning rate and momentum factor, we construct a novel neural network model named HN-GA-BP neural network. We suppose the HN-GA-BP neural network model has double hidden layers and apply the proposed neural network model to forecast the carbon dioxide contents. When the proposed model training process is completed, we output the typical results MSE, MAE, MAPE and the predicted carbon dioxide contents. The training process of HN-GA-BP neural network model is described as Algorithm 3.

**Algorithm 3** The training process of HN-GA-BP neural network model.

**Input:** Carbon dioxide dataset.
**Output:** MSE, MAE, MAPE and predicted carbon dioxide contents.

**Step1:** Load dataset and normalize the dataset into $[-1, 1]$.

**Step2:** Use Algorithm 1 to determine the nodes number of first hidden layer.

**Step3:** Use Algorithm 2 to train learning rate and momentum factor dynamically.

**Step4:** Based on the adaptively selected nodes number of the first hidden layer, learning rate and momentum factor, we construct the HN-GA-BP neural network model.

**Step5:** Use the HN-GA-BP neural network model to forecast the carbon dioxide contents.

**Step6:** Renormalize the forecasting results from $[-1, 1]$.

**Step7:** Output MSE, MAE, MAPE and the predicted carbon dioxide contents.

## 4. Experiments and analyses

In this section, to demonstrate the effectiveness of the proposed neural network model, we use the HN-GA-BP neural network model, the single hidden layer sigmoid feedforward neural network model (SLFNN) [7], the back-propagation neural network model (BPNN) [6, 9], the PSO-BP neural network model [23] and the genetic algorithm neural network model (GANN) [21, 22] to predict the Chinese carbon dioxide contents of 50 years (1961 to 2010) [2] and compare the average mean square error, average mean absolute error and average mean absolute percentage error of the four neural network models. The experiments are performed on MATLAB 2010b. The data set is divided into two subsets, the training set (1961 to 1990) and the test set (1991 to 2010) respectively. The input features include gaseous fuel consumption, liquid fuel consumption, solid fuel consumption, $CO_2$ emissions (Kg per GDP) and $CO_2$ emissions (metric tons per capita). The outputs are the prediction values of carbon dioxide contents. The learning function is *LM* and the goal is 1e-4.

Therefore, the HN-GA-BP neural network model has 5 input nodes and one output node. To calculate conveniently, we normalize the datasets into $[-1, 1]$. Firstly, we initialize the HN-GA-BP neural network weights of the first hidden layer, the second hidden layer and the output layer is $[-1, 1]$ randomly. The thresholds of the first hidden layer, the second hidden layer and the output layer is $[-1, 1]$ randomly. The maximum nodes number of the first hidden layer is 20. The nodes number of the second hidden layer is 6 according to the formula $(2m + 1)/2$ ($m$is the nodes number of the input layer). The activation function in the first hidden layer and the second hidden layer is Sigmoid. The activation function in output layer is Purelin. Secondly, we use Algorithm 1 to get the nodes number of the first hidden layer adaptively and determine the topology of the whole network. Then we use Algorithm 2 to train the learning rate and momentum factor dynamically. The size of population is 20, the maximum evolution number is 100, the crossover factor is 0.8, and the mutation factor is 0.2. Finally, based on the adaptively selected nodes number of the first hidden layer, the dynamically selected learning rate and the momentum factor, we construct the HN-GA-BP neural network and apply this model to forecast the carbon dioxide contents.

For comparing, we set the learning rate is 0.05 and momentum factor is 0.8 for SLFNN, BPNN, PSO-BP neural network models. In SLFNN neural network model with single hidden layer, we set the nodes number of hidden layer is 5. In BPNN neural network model with double hidden layers, we set the nodes number of each hidden layer as 5, and set the activation function in first hidden layer and second hidden layer is Sigmoid, the activation function in output layer is Purelin, set the learning function is *LM*. In PSO-BP neural network model, we set the nodes number of hidden layer is 7, the size of particles is 20, the maximum iteration number is 100, the space range is $[-3, 3]$, the velocity range is $[-1, 1]$, the random parameters is in $[0, 1]$, the cognitive factor and the social factor are 1.5 and 2.5 respectively. In GANN neural network model, we set the nodes number of hidden layer is 7, the size of population is 20, the maximum evolution number is 100, the crossover factor is 0.8, and the mutation factor is 0.2.

We run each neural network model 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 times and calculate the average MSE, average MAE and average MAPE, which are used as the final results. The training results of HN-GA-BP neural network are shown in Tab. I. The training results of SLFNN neural network are shown in Tab. II. The training results of BPNN neural network are shown in Tab. III. The training results of PSO-BP neural network are shown in Tab. IV. The training results of GANN neural network are shown in Tab. V.

| Number | Training times | MSE (AVG) | MAE (AVG) | MAPE (AVG) |
|---|---|---|---|---|
| 1 | 5 | 0.00000120 | 0.00078362 | 0.0018 |
| 2 | 10 | 0.00000116 | 0.00069538 | 0.0018 |
| 3 | 15 | 0.00000113 | 0.00070941 | 0.0018 |
| 4 | 20 | 0.00000118 | 0.00073379 | 0.0018 |
| 5 | 25 | 0.00000117 | 0.00073697 | 0.0018 |
| 6 | 30 | 0.00000115 | 0.00073776 | 0.0017 |
| 7 | 35 | 0.00000116 | 0.00073589 | 0.0017 |
| 8 | 40 | 0.00000116 | 0.00073793 | 0.0017 |
| 9 | 45 | 0.00000116 | 0.00074130 | 0.0017 |
| 10 | 50 | 0.00000115 | 0.00073699 | 0.0017 |

**Tab. I** *The training results of HN-GA-BP neural network model.*

| Number | Training times | MSE (AVG) | MAE (AVG) | MAPE (AVG) |
|---|---|---|---|---|
| 1 | 5 | 4.3061 | 2.0626 | 6.4544 |
| 2 | 10 | 4.2800 | 2.0327 | 6.3828 |
| 3 | 15 | 5.4293 | 2.2833 | 7.1464 |
| 4 | 20 | 3.9401 | 1.9131 | 6.0417 |
| 5 | 25 | 4.4293 | 2.0632 | 6.4666 |
| 6 | 30 | 4.7477 | 2.1212 | 6.6227 |
| 7 | 35 | 4.7940 | 2.1422 | 6.6865 |
| 8 | 40 | 4.7408 | 2.1432 | 6.7077 |
| 9 | 45 | 4.7607 | 2.1256 | 6.6754 |
| 10 | 50 | 4.7656 | 2.1284 | 6.6530 |

**Tab. II** *The training results of SLFNN neural network model.*

From Tab. I to Tab. V, we can conclude that the performance of SLFNN neural network model and BPNN neural network model are not very good and cannot reach the preset goal. Obviously, the performance of HN-GA-BP neural network model, PSO-BP neural network model and GANN neural network model are significantly better than SLFNN and BPNN, which can reach the preset goal in many cases.

Fig. 3 to Fig. 5 plots the logarithmic values of average MSE, MAE and MAPE of the five neural network models.

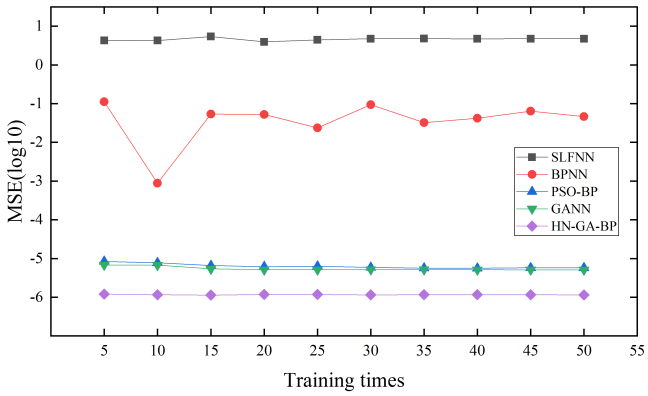


**Fig. 3** *The logarithmic values of average MSE of the five neural network models.*

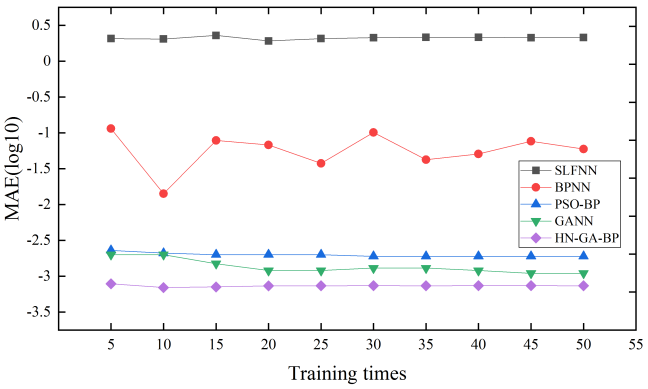


**Fig. 4** *The logarithmic values of average MAE of the five neural network models.*

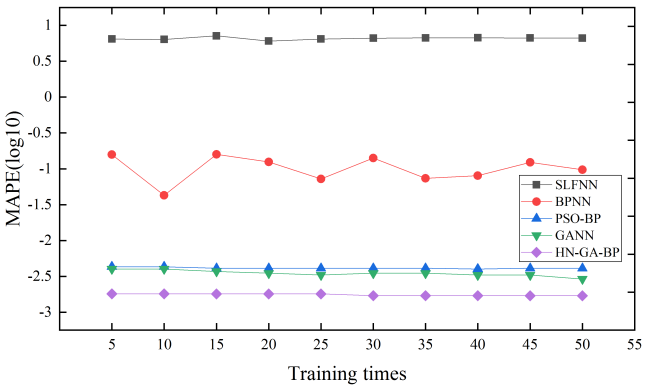


**Fig. 5** *The logarithmic values of average MAPE of the five neural network models.*

| Number | Training times | MSE (AVG) | MAE (AVG) | MAPE (AVG) |
|---|---|---|---|---|
| 1 | 5 | 0.1117 | 0.1146 | 0.1579 |
| 2 | 10 | 0.0877 | 0.0142 | 0.0426 |
| 3 | 15 | 0.0538 | 0.0784 | 0.1587 |
| 4 | 20 | 0.0523 | 0.0679 | 0.1246 |
| 5 | 25 | 0.0237 | 0.0374 | 0.0721 |
| 6 | 30 | 0.0946 | 0.1010 | 0.1411 |
| 7 | 35 | 0.0324 | 0.0422 | 0.0735 |
| 8 | 40 | 0.0422 | 0.0507 | 0.0804 |
| 9 | 45 | 0.0639 | 0.0763 | 0.1229 |
| 10 | 50 | 0.0466 | 0.0593 | 0.0969 |

**Tab. III** *The training results of BPNN neural network model.*

| Number | Training times | MSE (AVG) | MAE (AVG) | MAPE (AVG) |
|---|---|---|---|---|
| 1 | 5 | 0.00000838 | 0.0023 | 0.0043 |
| 2 | 10 | 0.00000773 | 0.0021 | 0.0043 |
| 3 | 15 | 0.00000658 | 0.0020 | 0.0041 |
| 4 | 20 | 0.00000614 | 0.0020 | 0.0041 |
| 5 | 25 | 0.00000632 | 0.0020 | 0.0041 |
| 6 | 30 | 0.00000588 | 0.0019 | 0.0041 |
| 7 | 35 | 0.00000562 | 0.0019 | 0.0041 |
| 8 | 40 | 0.00000559 | 0.0019 | 0.0040 |
| 9 | 45 | 0.00000574 | 0.0019 | 0.0041 |
| 10 | 50 | 0.00000573 | 0.0019 | 0.0041 |

**Tab. IV** *The training results of PSO-BP neural network model.*

| Number | Training times | MSE (AVG) | MAE (AVG) | MAPE (AVG) |
|---|---|---|---|---|
| 1 | 5 | 0.00000677 | 0.0020 | 0.0040 |
| 2 | 10 | 0.00000673 | 0.0020 | 0.0040 |
| 3 | 15 | 0.00000538 | 0.0015 | 0.0037 |
| 4 | 20 | 0.00000514 | 0.0012 | 0.0035 |
| 5 | 25 | 0.00000512 | 0.0012 | 0.0033 |
| 6 | 30 | 0.00000518 | 0.0013 | 0.0035 |
| 7 | 35 | 0.00000517 | 0.0013 | 0.0035 |
| 8 | 40 | 0.00000512 | 0.0012 | 0.0033 |
| 9 | 45 | 0.00000512 | 0.0011 | 0.0033 |
| 10 | 50 | 0.00000510 | 0.0011 | 0.0029 |

**Tab. V** *The training results of GANN neural network model.*

From Fig. 3 to Fig. 5, it is clear that the performance of HN-GA-BP neural network model is better than that of SLFNN, BPNN, PSO-BP and GANN neural network models. For instance, the HN-GA-BP neural network model exhibits 79.69% decrease in minimum average MSE compared with PSO-BP neural network model. The HN-GA-BP neural network model exhibits 95.10 % and 63.40 % decrease in minimum average MAE compared with BPNN neural network model and PSO-BP neural network model respectively. The HN-GA-BP neural network model exhibits 96.00 % and 57.50 % decrease in minimum average MAPE compared with BPNN neural network model and PSO-BP neural network model respectively. Moreover, compared with GANN neural network model, the HN-GA-BP neural network model exhibits 77.42 % , 36.78 % and 41.37 % decrease in minimum average MSE, MAE and MAPE respectively. With the increase of training times, the average MSE, average MAE and average MAPE of HN-GA-BP neural network model are tend to be more stable than that of other four neural network models.

In order to further compare the performance of these five neural network models, we calculate the average predicted carbon dioxide contents. The average predicted carbon dioxide contents (Kt) of these five network models are shown in Tab. VI.

| Year | Target | SLFNN | BPNN | PSO-BP | GANN | HN-GA-BP |
|---|---|---|---|---|---|---|
| 1991 | 2584538.270 | 10570488.470 | 2559196.197 | 2584291.196 | 2583290.096 | 2580440.026 |
| 1992 | 2695982.067 | 10526241.630 | 2685261.873 | 2692181.631 | 2690187.617 | 2687515.110 |
| 1993 | 2878694.009 | 10691628.820 | 2871678.897 | 2874013.611 | 2874413.801 | 2877579.974 |
| 1994 | 3058241.330 | 10799000.070 | 3117647.199 | 3059074.593 | 3059271.513 | 3059476.027 |
| 1995 | 3320285.150 | 11072107.180 | 3368471.127 | 3311534.082 | 3314534.482 | 3316431.238 |
| 1996 | 3463089.131 | 11222792.650 | 3525228.314 | 3465358.357 | 3465158.367 | 3465573.547 |
| 1997 | 3469510.048 | 11192271.130 | 3503893.695 | 3481577.633 | 3481177.683 | 3470983.662 |
| 1998 | 3324344.519 | 11002788.440 | 3380094.241 | 3337319.241 | 3341319.291 | 3328390.913 |
| 1999 | 3318055.614 | 11031249.060 | 3343026.427 | 3320163.977 | 3310063.877 | 3315222.547 |
| 2000 | 3405179.867 | 11213820.210 | 3405723.435 | 3408408.583 | 3409418.681 | 3407767.187 |
| 2001 | 3487566.356 | 11331202.780 | 3502944.039 | 3482949.692 | 3482049.682 | 3480648.472 |
| 2002 | 3694242.143 | 11618271.820 | 3711284.474 | 3689988.348 | 3699918.308 | 3697861.810 |
| 2003 | 4525177.009 | 12397660.600 | 4571354.160 | 4521839.201 | 4525839.271 | 4527851.845 |
| 2004 | 5288166.032 | 13492707.130 | 5333459.729 | 5286132.356 | 5286932.656 | 5289761.489 |
| 2005 | 5790016.984 | 14052627.820 | 5893055.317 | 5777714.901 | 5770714.901 | 5801152.070 |
| 2006 | 6414463.080 | 14799219.800 | 6399139.935 | 6411296.737 | 6411096.719 | 6414575.224 |
| 2007 | 6791804.714 | 15272112.470 | 6735613.951 | 6790257.101 | 6807257.101 | 6792529.298 |
| 2008 | 7035443.861 | 15590620.390 | 7058239.351 | 7033249.757 | 7034249.757 | 7034992.465 |
| 2009 | 7692210.895 | 16025488.340 | 7857719.091 | 7691157.492 | 7691257.491 | 7691526.129 |
| 2010 | 8286891.952 | 16520456.800 | 8189845.538 | 8286174.172 | 8276470.682 | 8285953.085 |

**Tab. VI** *The average predicted carbon dioxide contents (Kt) of four network models.*

Apparently, with the increase of training times, the predicted carbon dioxide contents of HN-GA-BP neural network model are tend to be more accuracy than that of other four neural network models. In summary, the HN-GA-BP neural network model has better performance in forecasting carbon dioxide contents, which demonstrates the effectiveness of the proposed neural network model.

# 5. Conclusions

Neural networks have been applied in many applications. The network structures and parameters are crucial for forecast accuracy and difficult to determine. Therefore, in this paper, to improve the forecast accuracy, we improve the neural network from two aspects. First, according to mean square error of each iteration of network training, the nodes number of the first hidden layer are selected adaptively, which can minimize the mean square error. Then, improved genetic algorithm is proposed to train the learning rate and momentum factor dynamically. On this basis, the HN-GA-BP neural network model is proposed based on the adaptively selected nodes number of the first hidden layer, the dynamically selected learning rate and momentum factor. We apply the proposed model to forecast the carbon dioxide contents in China. Experimental results demonstrate the effectiveness of the proposed neural network model. In the future, we will investigate other intelligent optimization algorithms to learn other parameters of neural network model, which may further optimize neural network performance.

## Acknowledgement

The author would like to thank the editors and the anonymous reviewers for their valuable comments and suggestions to improve the paper. This work was supported by the National Natural Science Foundation of China (No. 61402005), the Natural Science Foundation of Anhui Province, China (No. 1308085QF114), the Higher Education Natural Science Foundation of Anhui Province, China (No. KJ2013A015), and the Scholarship of China Scholarship Council.

# References

[1] SHETE A.F., GHATASHEH N., FARIS H. Forecasting global carbon dioxide emission using auto-regressive with eXogenous input and evolutionary product unit neural network models. The *6th International Conference on Information and Communication Systems (ICICS).* 2015, pp. 182–187, doi: 10.1109/IACS.2015.7103224.

[2] National Bureau of Statistics of China, http://data.stats.gov.cn/easyquery.htm?cn=C01, (accessed 08.20.2016).

[3] PRIETO A., PPRIETO B., ORTIGOSA E.M. Neural networks: An overview of early research, current frameworks and new challenges. *Neurocomputing.* 2016, 214, pp. 242–268, doi: 10.1016/j.neucom.2016.06.014.

[4] ASKARZADEH A., REZAZADEH A. Artificial neural network training using a new efficient optimization algorithm. *Applied Soft Computing.* 2013, 13(2), pp. 1206–1213, doi: 10.1016/j.asoc.2012.10.023.

[5] LI F., WU C., Wu K. An improved back propagation neural network model and its application. *Journal of Computers.* 2014, 9(8), pp. 1858–1862, doi: 10.4304/jcp.9.8.1858-1862.

[6] SKÖN J.P., JOHANSSON M., RAATIKAINEN M. Modelling indoor air carbon dioxide concentration using neural network. *World Academy of Science Engineering & Technology.* 2012, 14(15), pp. 16, doi: 1307-6892/11723.

[7] PINGFAN Y. Some Views on the research of multilayer feedforward neural network. *Acta Electronica Sinica.* 1999, 27(5), pp. 82–85, doi: 10.3321/j.issn:0372-2112.1999.05.024.

[8] BEVILACQUA V., INTINI F., KÜHTZ S. A model of artificial neural network for the analysis of climate change. *The 29th Annual International Sympsium on Forecasting.* 2008, pp. 22–25, http://hdl.handle.net/11563/9566.

[9] GALLO C., CONTò F., FIORE M. A neural network model for forecasting $co_2$ emission. *AGRIS on-line Papers in Economics and Informatics.* 2014, 6(2), pp. 31, http://ageconsearch.umn.edu/record/182488.

[10] HAGAN M.T., DEMUTH H.B. BEALE M.H. Neural network design. *Boston: PWS.* 1996, doi: 10.1007/1-84628-303-5.

[11] PERALTA J., GUTIERREZ G., SANCHIS A. ADANN: Automatic design of artificial neural networks. *Proceedings of the 10th annual conference companion on Genetic and evolutionary computation.* 2008, pp. 1863–1870, doi: 10.1145/1388969.1388991.

[12] SHI F., WANG X., YU L. Analysis 30 neural network cases with matlab. *Beijing, Beihang University Press.* 2010, ISBN: 978-7-5124-0034-4.

[13] NEGNEVITSKY M. Artificial intelligence: a guide to intelligent systems. *New York : Addison Wesley.* 2002, pp. 111–143, doi: 10.1016/j.artmed.2011.04.006.

[14] WANG L., ZENG Y., CHEN T. Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. *Expert Systems with Application.* 2015, 42(2), pp. 855–863, doi: 10.1016/j.eswa.2014.08.018.

[15] MI C., YANG J., LI S. In Prediction of accumulated temperature in vegetation period using artificial neural network. *Mathematical & Computer Modelling.* 2010, 51(11-12), pp. 1453–1460, doi: 10.1016/j.mcm.2009.10.005.

[16] WAGARACHCHI N.M., KARUNANANDA A.S. Optimization of multi-layer artificial neural networks using delta values of hidden layers. *2013 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB).* 2013, pp. 80–86, doi: 10.1109/CCMB.2013.6609169.

[17] YI X. Selection of initial weights and thresholds based on the genetic algorithm with the optimized back-propagation neural network. *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD).* 2015, pp. 173–177, doi: 10.1109/FSKD.2015.7381935.

[18] LILI W., HONGBO L., FENG C. Identification of flow regimes based on adaptive learning and additional momentum bp neural network. *2016 6thInternational Conference on Instrumentation & Measurement, computer, Communication and Control.* 2016, pp. 574–578, doi: 10.1109/IMCCC.2016.29.

[19] MASOOD S., DOJA M.N., CHANDRA P. Analysis of weight initialization methods for gradient descent with momentum. *2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI).* 2015, pp. 131–136, doi: 10.1109/ICSCTI.2015.7489618.

[20] NARAYANAN S.J., BHATT R.B., PERUMAL B. Improving the accuracy of fuzzy decision tree by direct back propagation with adaptive learning rate and momentum factor for user location. *Procedia Computer Science.* 2016, pp. 506–513, doi: 10.1016/j.procs.2016.06.104.

[21] JAMALI A., AHMADI P., JAAFAR M.N.M. Optimization of a novel carbon dioxide cogeneration system using artificial neural network and multi-objective genetic algorithm. *Applied Thermal Engineering.* 2014, 64(1–2), pp. 293–306, doi: 10.1016/j.applthermaleng.2013.11.071.

[22] YU S., WEI Y.M., WANG K. A PSO-GA optimal model to estimate primary energy demand of China. *Energy Policy.* 2012, 42(2), pp. 329–340, doi: 10.1016/j.enpol.2011.11.090.

[23] SUN W., XU Y. Using a back propagation neural network based on improved particle swarm optimization to study the influential factors of carbon dioxide emissions in Hebei Province China. *Journal of Cleaner Production.* 2016, 112, pp. 1282–1291, doi: 10.1016/j.jclepro.2015.04.097.

[24] MCCULLOCH W.S., PITTS W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* 1943, 5(4), pp. 115–133, doi: 10.1007/BF02459570.

[25] HECHT-NIELSEN R. Kolmogorov's mapping neural network existence theorem. *IEEE International Conference on Neural Networks.* 1987, doi: 10.1109/CCMB.2013.6609169.

[26] ZHOU Y., ZHU Q., HUANG H. Prediction of acute hypotensive episode in ICU using chebyshev neural network. *Journal of Software.* 2013, 8(8), pp. 1923–1931, doi: 10.4304/jsw.8.8.1923-1931.

[27] WANG H.S., WANG Y.N., WANG Y.C. Cost estimation of plastic injection molding parts through integration of pso and bp neural network. *Expert Systems with Applications.* 2013, 40, pp. 418–428, doi: 10.1016/j.eswa.2012.01.166.

[28] LEEMA N., NEHEMIAH H.K, KANNAN A. Neural network classi?er optimization using differential evolution with global information and back propagation algorithm for clinical datasets. *Applied Soft Computing.* 2016, 49, pp. 834–844, doi: 10.1016/j.asoc.2016.08.001.

[29] HOLLAND J.H. Adaptation in natural and artificial systems. *MIT Press.* 1992, 6(2), pp. 126–137, doi: 10.1109/CCMB.2013.6609169.

[30] AHMAD F., ISA N.A.M., HUSSAIN Z. A genetic algorithm based multi-objective optimization of an artificial neural network classifier for breast cancer diagnosis. *Neural Computing & Applications.* 2013, 23(5), pp. 1427–1435, doi: 10.1007/s00521-012-1092-1.