International Journal of Automation and Computing

15(3), June 2018, 267-276
DOI: 10.1007/s11633-017-1107-6

Gesture Recognition Based on BP Neural Network Improved by Chaotic Genetic Algorithm

Dong-Jie Li Yang-Yang Li Jun-Xiang Li Yu Fu

School of Automation, Harbin University of Science and Technology, Harbin 150080, China

Abstract: Aim at the defects of easy to fall into the local minimum point and the low convergence speed of back propagation (BP) neural network in the gesture recognition, a new method that combines the chaos algorithm with the genetic algorithm (CGA) is proposed. According to the ergodicity of chaos algorithm and global convergence of genetic algorithm, the basic idea of this paper is to encode the weights and thresholds of BP neural network and obtain a general optimal solution with genetic algorithm, and then the general optimal solution is optimized to the accurate optimal solution by adding chaotic disturbance. The optimal results of the chaotic genetic algorithm are used as the initial weights and thresholds of the BP neural network to recognize the gesture. Simulation and experimental results show that the real-time performance and accuracy of the gesture recognition are greatly improved with CGA.

Keywords: Gesture recognition, back propagation (BP) neural network, chaos algorithm, genetic algorithm, data glove.

1 Introduction

Recently, gesture recognition has been the hot research field of human-computer interaction[1−4], and the key parameters to measure the effective of gesture recognition algorithm are instantaneity and accuracy. The deep learning of neural network is applied to every field of technology[5], for the back propagation (BP) neural network has strong advantages of good adaptability, simple principle, easy implementation, etc., and it is often used for gesture recognition. Jiang and Ruan[6] used CAS_Glove data glove to collect the flexion angle of hand joints, and then used the BP neural network to learn and recognize the gesture, which finally obtained the good recognition effect. But the BP neural network is easily affected by the initial condition of parameters and falls into local minimum during the learning process[7]. Zeng[8] talked about how to improve the learning rate and momentum factor of BP neural network, and the learning speed was increased to some degree, but its ascending space is limited, and different systems need different test constantly. You et al.[9] used simulated annealing (SA) to optimize the BP neural network, the random disturbance was added to the weight and threshold during the iteration of the traditional BP neural network, which not only makes the network out of local optimum, but also improves the learning ability of the network. Salim[10] adopted particle swarm optimization (PSO) to improve the BP neural network. The weights and thresholds of the neural network are used as the initial population, and then the global optimization is realized with different search speed according to the different locations of the individuals in the search space. At the same time, the particles of the PSO falling into the local minimum are optimized with the self-adaptive mutation operator to particle algorithm in PSO.

Genetic algorithm (GA)[11,12] is an intelligent algorithm which is first proposed by Professor John in 1975. The algorithm is a heuristic global search algorithm, which is widely used in the fields of function optimization, algorithm combination, robotics and image processing, etc.[13] Its main principle is to encode the research object to the initial individual, and then the best individual is obtained after the individual selection, crossover and mutation operation according to the fitness function. In[14], the weights and thresholds of the BP neural network are encoded by the GA algorithm for the reverse correction process. At the same time, the combination of binary coding and real number coding are used to increase the searching range. Simulation results show that the combination of GA-BP can improve the accuracy of the system greatly.

The main features of the genetic algorithm are directly object operating and without relevant to the function derivability and continuity[15]. Therefore, the defect of BP neural network, which is easy to fall into local minimum can be solved easily by using genetic algorithm to optimize the BP neural network. But in view of the condition of multiple variables and constraints in the process of gesture recognition[16], optimization of the process

Research Article
Manuscript received September 21, 2016; accepted October 9, 2017; published online January 4, 2018
This work was supported by Natural Science Foundation of Heilongjiang Province Youth Fund (No. QC2014C054), Foundation for University Young Key Scholar by Heilongjiang Province (No. 1254G023) and the Science Funds for the Young Innovative Talents of HUST (No. 201304).
Recommended by Associate Editor Victor Becerra

needs a long time and is prone to cause the phenomenon of gene loss.

In recent years, the chaos theory[17, 18], a kind of nonlinear dynamic algorithm, is mainly used in two ways. On one hand, it is used as a separate algorithm for engineering application, including secure transmission, cryptography, telecommunications, etc. On the other hand, it is used as an optimization algorithm combining with other algorithms to improve the performance of the algorithm. Its main principle is to make the solution space of the research object mapped to the chaotic space by a specific way[19, 20]. Using the randomness, ergodicity and regularity of chaotic variables, the global optimal solution can be obtained through repeatedly traversing of all the motion states, and finally the chaotic variables are reflected to the solution space from chaotic space. Yin et al.[21] introduced the nonlinear self-feedback chaos learning algorithm to the weights of BP neural network, it has faster learning speed and higher search accuracy, but the separate chaos algorithm is not suitable for wide range search space for the low search speed.

From the above, this paper combines the chaos algorithm and genetic algorithm (CGA) to recognize the hand gesture. The basic idea is to achieve global optimization through continuous iteration. Firstly, the general optimal solution is obtained after the selection, crossover and mutation operators of each iteration with genetic algorithm, then the general optimal solution is optimized to the accurate optimal solution by adding chaotic disturbance. The optimal results of the chaotic genetic algorithm are used as the initial weights and thresholds of the BP neural network to recognize the gesture.

2 CGA-BP gesture recognition overall design

The data of gesture is captured with the 5DT data gloves[22] in this paper. The data glove has 8-bit open bending characteristics, and it is comfortable and without data drift. The data glove has 14 sensors to test the bending angle of five fingers, and the value of each sensor is [0, 1]. Fig. 1 shows the sensor distribution of 5DT data glove.

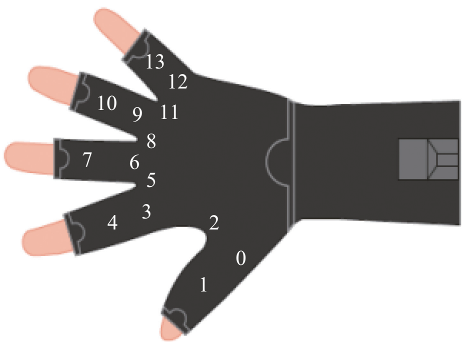


Fig. 1 Sensor distribution of 5DT data glove

The absolute value of the sensor is different when different operators dress it, and even the same operator makes the same gesture. But the absolute value of the sensor and the relative change in the range of finger motion are small, so it is possible to establish a common gesture template which is for multi-user collaborative operation. We collected 300 sets gesture data of persons from different ages and gender. 270 sets of data are used to establish gesture template for the learning process and the left 30 sets of data are used to neural network testing. Ten gesture templates icon and the corresponding digital definition are shown in Table 1.

Table 1 Definition of gesture template

Gesture to figure	Digital definition	Gesture to figure	Definition of digital
	10000		11111
	11000		10111
	11100		10011
	10010		10110
	11001		10101

Three layers BP neural network is adopted for gesture recognition in this paper, and the neurons number of the three layers are 14–13–5. The 14 neurons of input layer correspond to the data glove's 14 sensor value after uniformization. The 5 neurons of output layer are the composition of bent(1) and extension(0) for each finger. The neurons number of the hidden layer can be calculated with the following empirical formula[23]:

$$n_1 = \sqrt{n_0 + n_2} + a \tag{1}$$

where n_0, n_1 and n_2 are the numbers of neurons in the input layer, hidden layer and output layer, respectively; a is an arbitrary constant between 1 and 10.

3 Problems of BP neural network

This paper studies the three layers BP neural network established in Section 2. Assuming the sample number that need to learn in the gesture template library is N, and the neurons numbers for each layer are n_0, n_1 and n_2, respectively. The activation function in the layer m and the node i is $Z_{i,P}^{m}$ in the sample P. The weight of the connection is $\omega_{i,j}^m$ between the layer m with the node j and the layer m-1 with the node i, and the threshold of this node is expressed as $\eta_{0,j}^m$. The activation function of each layer is defined as $f(x)$, $O_{j,P}^m$ is the output value of the layer m and the node i in the sample P, and $T_{j,P}$ is the expected output of the output layer in nodes j.

The positive learning process of the BP neural network is

$$Z_{i,P}^m = \eta_{0,i}^m + \sum_{j=1}^{n_{m-1}} \omega_{j,i}^m O_{j,P}^{m-1} \tag{2}$$

$$O_{i,P}^m = f(Z_{i,P}^m). \tag{3}$$

Usually, the squared error between the actual output and the expected output of the network is used as the evaluation criterion, its specific formula is

$$E = \frac{1}{2}\sum_{P=1}^{N}\sum_{j=1}^{n_2}\left(T_{j,P} - O_{j,P}^2\right)^2. \tag{4}$$

The learning process of reverse BP neural network is constantly adjusting the weights and thresholds.

Assuming the learning rate is α, and the learning process is

$$\begin{cases} \Delta\omega_{i,j}^2 = -\alpha\dfrac{\partial E}{\partial\omega_{i,j}^2} \\ \Delta\eta_{0,j}^2 = -\alpha\dfrac{\partial E}{\partial\eta_{0,j}^2} \\ \Delta\omega_{i,j}^1 = -\alpha\dfrac{\partial E}{\partial\omega_{i,j}^1} \\ \Delta\eta_{0,j}^1 = -\alpha\dfrac{\partial E}{\partial\eta_{0,j}^1}. \end{cases} \tag{5}$$

Assuming k is the current learning times, the revision process of the weights and thresholds is

$$\omega_{j,i}^m(k) = \omega_{j,i}^m(k-1) + \Delta\omega_{j,i}^m(k) \tag{6}$$

$$\eta_{0,j}^m(k) = \eta_{0,j}^m(k-1) + \Delta\eta_{0,j}^m(k). \tag{7}$$

In order to make the values of (6) and (7) stable, which means obtaining the extreme value, (5) should equal to zero, i.e.,

$$\frac{\partial E}{\partial\omega_{i,j}^2} = \frac{\partial E}{\partial\eta_{0,j}^2} = \frac{\partial E}{\partial\omega_{i,j}^1} = \frac{\partial E}{\partial\eta_{0,j}^1} = 0. \tag{8}$$

However, if the error is still large when (8) holds, the weight and threshold value can not be revised, which means it falls into a local minimum. So we use the strategy that combined genetic algorithm and chaos algorithm to adjust the weights and thresholds to improve the speed and accuracy of the algorithm.

4 Genetic algorithm

For a general optimization problem:

$$\min f(x_1, \cdots, x_r) \tag{9}$$

$$\text{s.t. } a_i \le x_i \le b_i, i = 1, \cdots, r \tag{10}$$

where x_i is the study variable, the variable range of x_i is $[a_i, b_i]$, and r is the number of variables.

The basic steps of genetic algorithm are:

1) Individual encoding: Encoding is usually realized through the binary coding and the actual coding, however we adopt the real number coding method in this paper to encode x_i, which represents the weight value and threshold of BP neural network. The coding length is $14 \times 13 + 13 + 13 \times 5 + 5 = 256$, the population size is $sizepop = 15$, the maximum hereditary algebra is $gen = 20$.

2) Determining of the fitness function: The actual output of the BP neural network is T_i, the expected output is O_i, so the error of BP neural network is defined as

$$err = \sum_{i=1}^{n_2}\frac{1}{2} \times (T_i - O_i)^2. \tag{11}$$

Because the optimal solution in this paper is the minimization problem, the reciprocal of BP neural network is selected as the fitness value, which is

$$\text{fitness}(k) = \frac{1.0}{err(k)}(k = 1, 2, \cdots, sizepop). \tag{12}$$

3) Selection process: The selection process is an important part in the population evolution, and the common selection procedures including roulette selection method[24], tournament selection method[25], $(\mu + \lambda)$ selection[26], sorting selection, elitist selection[27] truncation selection, etc. We combine the roulette selection method with elitist selection method to produce the next generation individuals. The individuals with the largest fitness value are not operated for crossover and mutation, but directly transfered to the next generation through heredity. The left individuals are chosen using the roulette wheel selection method, and the probability that each individual is chosen as the next generation is

$$p_s(k) = \frac{\text{fitness}(k)}{\sum\limits_{k=1}^{sizepop-1}\text{fitness}(k)}. \tag{13}$$

4) Crossover operation: Crossover operation[28] plays an important role in the genetic algorithm, and it is closely related to the search ability of genetic algorithm[29]. The simulation binary crossover is adopted to realize the crossover operation, and the crossover coefficient r_i can be calculated firstly with the following equa-

tion:

$$r_i = \begin{cases} (2u_i)^{\frac{1}{(\eta_c+1)}}, & u_i < 0.5 \\ \dfrac{1}{2(1-u_i)}^{\frac{1}{(\eta_c+1)}}, & u_i \geq 0.5 \end{cases} \tag{14}$$

where u_i is a random number which belongs to [0, 1], $i \in [0, length-1]$, $\eta_c = 10$, x_i^1 and x_i^2 are progenies of the individuals determined by the parent individuals p_i^1 and p_i^2 through the following formula:

$$\begin{cases} x_i^1 = 0.5[(1+r_i)p_i^1 + (1-r_i)p_i^2] \\ x_i^2 = 0.5[(1+r_i)p_i^1 + (1+r_i)p_i^2]. \end{cases} \tag{15}$$

5) Variation operation: The main effect of variation is the introduction of variation operators to generate new individuals[30, 31], and enrich the pattern diversity. In this paper, the polynomial form of variation rule is adopted, and the variation coefficient δ can be obtained with formula (16).

$$\delta = \begin{cases} (2u)^{\frac{1}{(\eta_m+1)}} - 1, & u < 0.5 \\ 1 - [2(1-u)]^{\frac{1}{(\eta_m+1)}}, & u \geq 0.5 \end{cases} \tag{16}$$

where u is a random number which belongs to [0, 1], η_m is a random number, and we defined $\eta_m = 10$. After mutation, each individual gene can be expressed as:

$$x_i = p_i + \delta\Delta_{\max} \tag{17}$$

where $\Delta_{\max}$ is the maximum range of variation.

6) Evolution stopping criteria: When the number of iterations $k = gen$ or after five consecutive iteration, the best individual remains the same, then the genetic algorithm ends, and the optimal solution is output.

5 Chaos algorithm

The ergodicity of the chaos algorithm ensures that the chaotic variable can traverse each point of the space and jump out of the local optimum. The sensitivity to the initial value makes the two individuals completely different even their fitness values are very near. It can not only reserve the best individual, but also maintain the diversity of the population. In this paper, we use the combination of Tent mapping and Logistic mapping to generate chaotic sequence. Tent mapping[32] is used to generate the initial value of chaotic variables, and the Logistic mapping[33] of genetic algorithm is used for chaotic disturbance.

5.1 Tent mapping

Tent mapping is a piecewise linear mapping and it has the performance of simple structure, good sensitivity to the initial value and good correlation[34]. The classic Tent mapping is

$$x_{n+1} = \begin{cases} \dfrac{1}{\mu}x_n, & 0 \leq x_n < \mu \\ \dfrac{1}{\mu}(1-x_n), & \mu \leq x_n \leq 1. \end{cases} \tag{18}$$

Taking constant $\mu = 0.5$, in this case the sequence obtained for the different parameters has the uniform density distribution. The defects of the classic Tent mapping are that the segmentation of the adjacent point has stronger related feature. Because the digital accuracy is limited in the calculation process, the obtaining of some inherent fixed point from the initial value after a finite iteration becomes meaningless. In this paper, we use a linear combination to avoid the shortcomings mentioned above, and the basic expression is

$$x_{n+1} = \begin{cases} 2x_n + p\sin(qx_n), & x_n < 0.5 \\ 2(1-x_n) - p\sin(qx_n), & x_n \geq 0.5. \end{cases} \tag{19}$$

where p and q need to satisfy the following constraint conditions respectively:

$$p = 2\pi n, n \in Z \cap n \neq 0$$
$$0 \leq \frac{2}{q}\arccos(-\frac{2}{pq}) + p\sin\left[\arccos(-\frac{2}{pq})\right] \leq 1.$$

Setting the initial value $x_0 = 0.36$, and the chaotic sequence is produced according to (19). Fig. 2 shows the distribution histogram of the initial values of the random sequence mapped with the Tent mapping. Fig. 3 is the distribution histogram of the initial values mapped with Logistic mapping. According to Figs. 2 and 3, it can be concluded that the initial values generated with Tent mapping can avoid the peak states on the terminals of 0 and 1 generated with the Logistic mapping, that means the initial values are more smooth.

5.2 Learning algorithm

The crossover and mutation operators of genetic algorithm increase the diversity of the population, but the

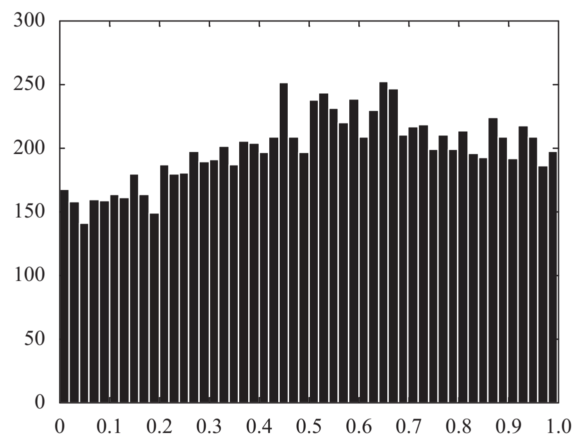


Fig. 2 Distribution histogram of Tent mapping

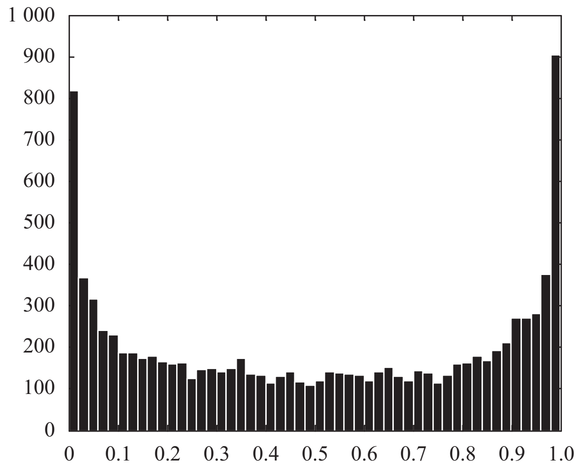


Fig. 3 Distribution histogram of Logistic mapping

optimal speed is slow. So we add the chaotic disturbance to the variables of genetic population to speed up the optimization. Chaos optimization is realized by chaotic variables, and we adopt the improved Tent mapping to complete the space map, (19) is the specific calculation equation.

Assume the individual is $X^* = \{x_1^*, x_2^*, \cdots, x_i^*, \cdots, x_n^*\}$ after the selection, crossover and mutation of genetic algorithm, and the chaos algorithm is adopted to disturb this individual, the perturbation formula is

$$\delta'_k = (1-\alpha)\delta^* + \alpha\delta_k \tag{20}$$

where δ_k is the chaos disturbance vector of the k-th iteration, δ^* is the vector of optimal solution $X^* = \{x_1^*, x_2^*, \cdots, x_i^*, \cdots, x_n^*\}$ after chaotic change, and δ_k^* is the random chaotic perturbation vector. The variable scale chaos method is used to get the value of α, and the specific equation is

$$\alpha = 1 - \left(\frac{k-1}{k}\right)^n. \tag{21}$$

At the beginning of iteration, the value of α is larger, and the space of chaos traversing is larger, so a rough search is carried out in this case. With the increase of the number of iterations, the value of α is gradually reduced and the detailed search works.

The fitness value of the new individual was calculated after each iteration. The new individual disturbed with chaos will be chosen as the current optimal individual when its fitness value is larger than the original individual, otherwise, iterating continues. If the optimal individual values have no change after 5 consecutive iterations, iteration will be stopped. The optimal solution will be output and mapped to the solution space from chaotic space. The mapping formula is as follows:

$$x_{mi} = a_i + (b_i - a_i)u_{mi}. \tag{22}$$

6 Algorithm learning process

The principle of CGA optimizing the BP neural network is encoding the weights and threshold values of BP neural network, and then the optimal solution is obtained using the ergodicity of chaotic algorithm and global convergence of genetic algorithm. Finally, this optimal solution is used as the initial weights and thresholds of BP neural network to predict the gesture recognition template. The algorithm flow chart of the overall design is shown in Fig. 4.

7 Simulation

7.1 Simulation of algorithm function

In order to show the effectiveness of CGA algorithm modifying the BP neural network, three commonly used standard test functions are selected: Rosenbrock function, Ackley function and Griewank function. The main characteristics of the three test functions and the function curves are shown in Table 2.

The BP neural network, GA modified BP neural network and CGA modified BP neural network are implemented using Matlab. Each algorithm runs 50 times, and the mean value, maximum value, and minimum value of the error are calculated, and the number of iterations to find the global optimal solution is obtained. The final errors of the three algorithms after 1 000 iterations are shown in Table 3, and the average iteration steps to find the global optimal solution of each algorithm are shown in Table 4.

The data in Tables 3 and 4 show that the BP neural network may fall into the local optimization in the process of global optimization with the three kinds of standard test functions. GA-BP algorithm can solve this problem of BP neural network but the recognition rate is still limited. CGA-BP optimization algorithm can be able to find the global optimal solution only after a few iterations. It can overcome the defects of the pure BP algorithm, GA-BP algorithm, and the algorithm convergence speed is improved greatly.

7.2 Matlab simulation

In order to verify the effectiveness of the CGA algorithm modifying BP neural network in gesture recognition, we use Matlab to complete the simulation. The setting of simulation parameters are as follows:

1) Sample: Number of total samples is 500, learning samples is 400, the number of test sample: 100.

2) The BP neural network: Learning factor $\eta = 0.2$, expected error $error = 10^{-4}$, maximum number of learning maxstep = 5 000, and the hidden layer and output layer function adopt the Logsig function and Purelin function respectively.

3) Chaos genetic algorithm: Crossover probability

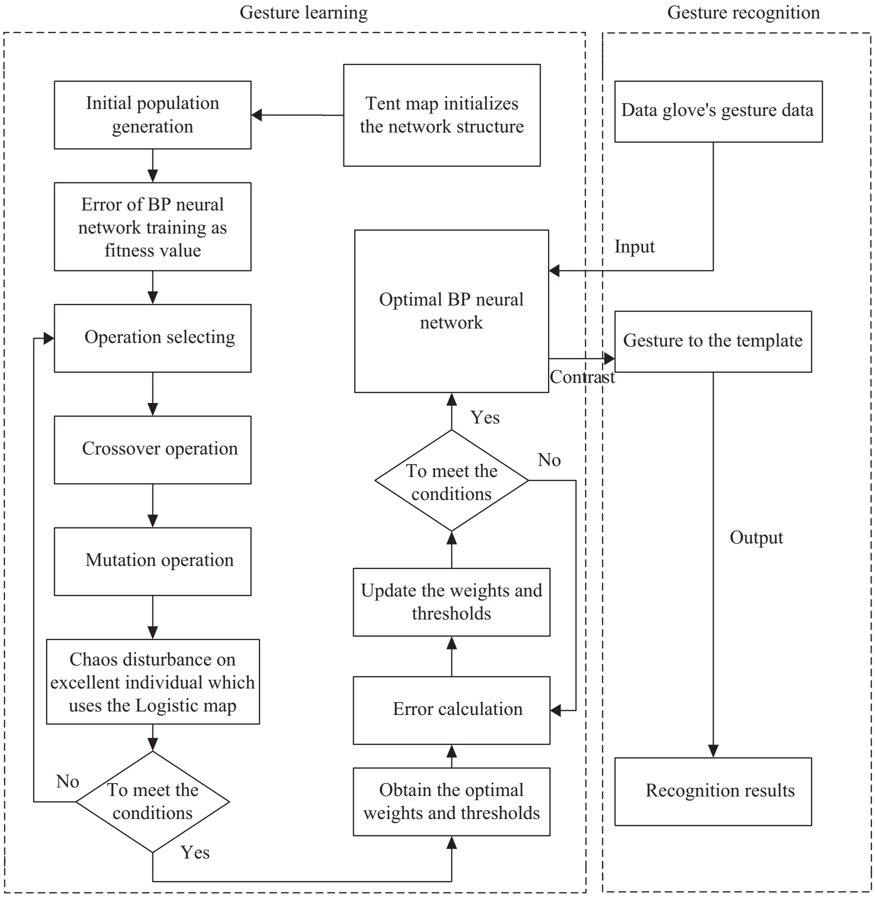
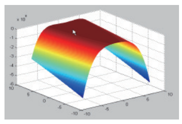
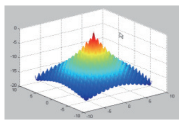
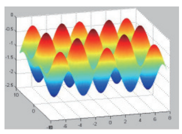


Fig. 4 Flow chart of CGA modifying BP network

Table 2 Main features of the test function

Function	Expression	Variable interval	Function curve
Rosenbrock	$f(x)=\sum_{i=1}^{n-1}(100(x_i^2-x_{i+1})^2+(x_i-1)^2)$	[–8, 8]	
Ackley	$f(x)=20+e-20\exp\left(-0.2\sqrt{\frac{\sum_{i=1}^{n-1}x_i^2}{n}}-\exp\frac{\sum_{i=1}^{n}\cos 2\pi x_i}{n}\right)$	[–8, 8]	
Griewank	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[–8, 8]	

$p_c=0.6$, mutation probability $p_m=0.1$.

Learning with BP neural network, genetic algorithm improved BP neural network, and CGA improved BP neural network, the learning error's falling curves of the three networks are shown in Figs. 5–7.

In Fig. 5, the final system error is 0.005 655 8 after 5 000 times learning with BP neural network, and the error change rate is near zero, but it falls into the local optimum. In Fig. 6, GA modifies the BP neural network and learns 5 000 times. The error is 0.000 418. The results are very near to the desired error, and the error rate is not zero, that means it will not fall into the local minimum, but its learning time is much longer. In Fig. 7, the BP neural network modified by CGA learns only 608 times, the final error has reached the ideal error and the learning time is greatly reduced. The final error is smaller than

Table 3 Error statistics after 1000 iterations

Function		BP	GA-BP	CGA-BP
Rosenbrock	Maximum	0.072	0.180	0.000
	Minimum	0.005	0.001	0.000
	Average	0.054	0.009	0.000
Ackley	Maximum	0.124	0.054	0.000
	Minimum	0.006	0.007	0.000
	Average	0.035	0.012	0.000
Griewank	Maximum	0.118	0.038	0.000
	Minimum	0.007	0.000	0.000
	Average	0.021	0.002	0.000

Table 4 Number of average iterations (Global optimal solution is not found)

Function	BP	GA-BP	CGA-BP
Rosenbrock	1823	982	524
Ackley	–	901	508
Griewank	–	912	521

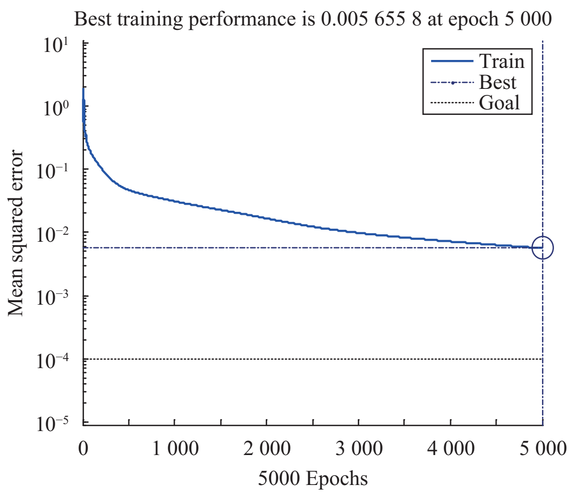


Fig. 5 Learning error curve of BP

the results of GA modified BP neural network. The analysis curves of linear regression of different algorithms modified BP neural network are shown in Figs. 8–10. The dotted line represents the regression curve of the expected output, and the solid line shows the linear regression curve after the sample learning.

In Fig. 8, the slope of the linear regression curve is 0.991 94. The final study data is scattered around 1, and the dispersion is larger, which is [0.83, 1.125]. The slope of the linear regression curve is 0.999 74, and the final data is [0.9, 1.05] in Fig. 9. In Fig. 10, the slope of the linear regression curve is 0.999 94, and the final data is [0.96, 1.02]. From the above data analysis, it can be obtained that the CGA can optimize the BP neural network, avoid the defects of BP neural network falling into the local minimum, and improve the recognition speed and accuracy.

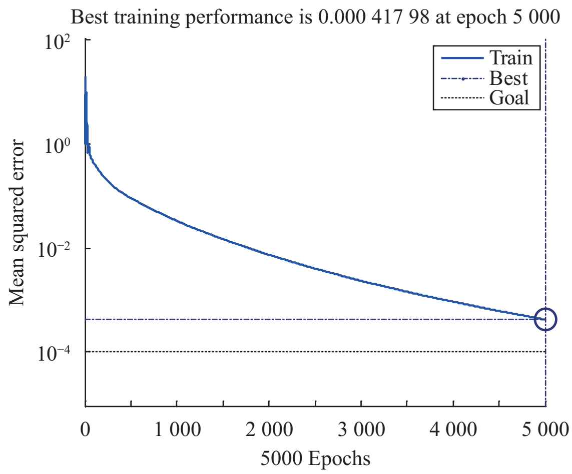


Fig. 6 Learning error curve of BP modified by GA

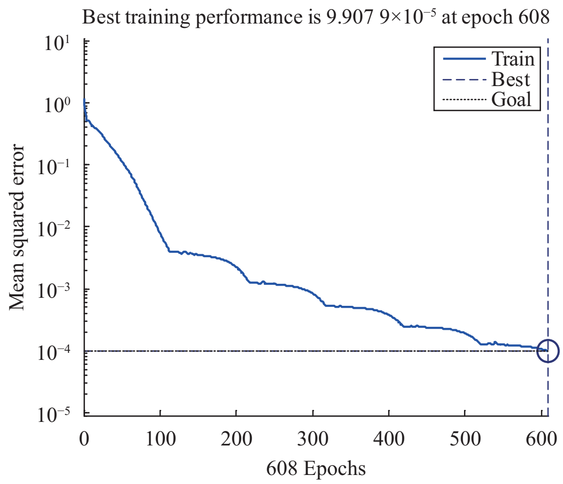


Fig. 7 Learning error curve of BP modified by CGA

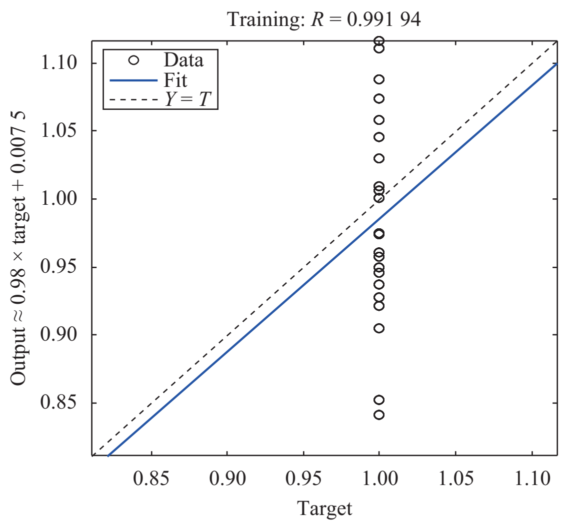


Fig. 8 Linear regression curve of BP

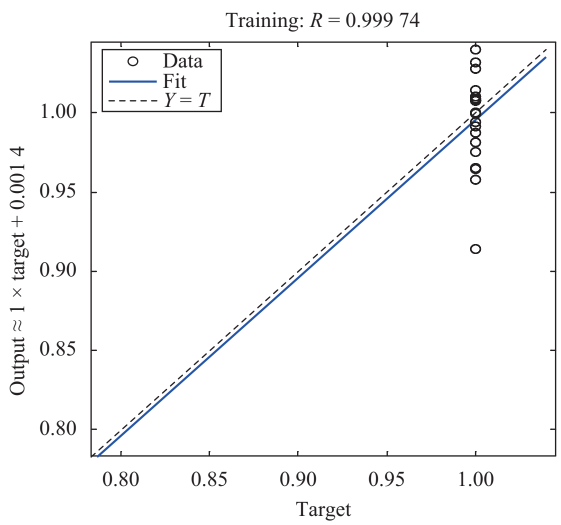


Fig. 9 Linear regression curve of BP modified by GA

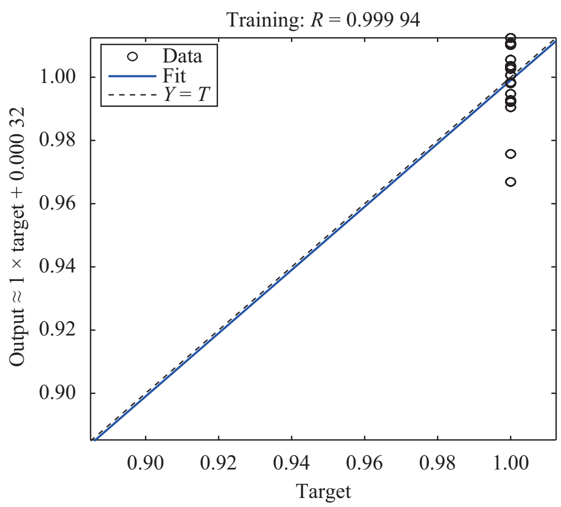


Fig. 10 Linear regression curve of BP modified by CGA

7.3 Application program development

The signal data obtained by the operator wearing the data glove is a continuous data flow. In the application development process, the timer is used to sample the continuous data flow, and the discrete static gesture is obtained. The measured gesture can only be compared with the existing data template in the database to output the optimal solution. In order to avoid external interference, an error threshold is set up. When the error is less than the threshold, it begins to recognize, conversely, the invalid gesture is considered. A gesture of 16 s matching data flow curve is shown in Fig. 11.

Simulation results of Matlab: Through simulation, it can be found that when the error is less than 0.1, two consecutive segments of gesture may have the interference phenomena; but when the error is greater than 0.1, all gesture can be recognized normally. Therefore, we set 0.1 as the error threshold.

The hand gesture recognition platform of data glove is developed based on VS2008 MFC, two similar gesture recognition results are shown in Figs. 12 and 13.

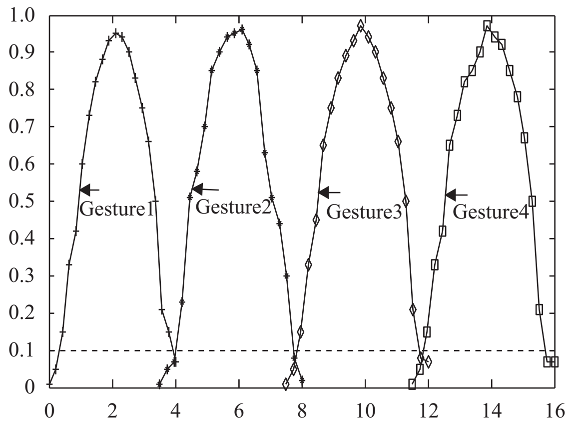


Fig. 11 Matching curve of gesture recognition

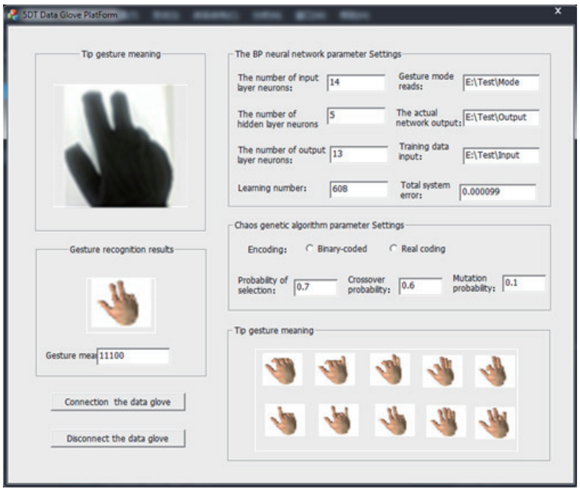


Fig. 12 First gesture recognition process

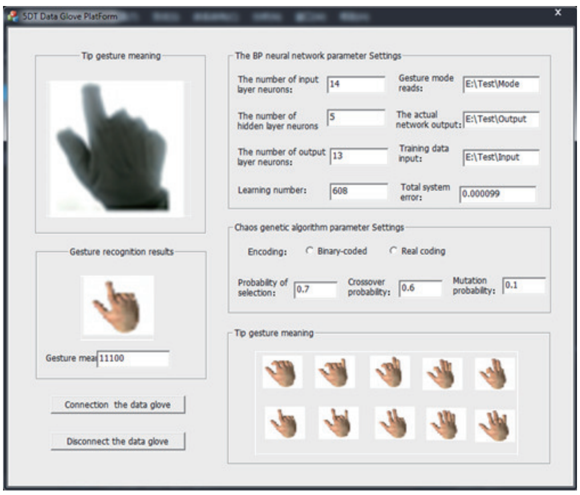


Fig. 13 Second gesture recognition process

The BP neural network is used in the actual gesture recognition after learning. Ten different operators are selected to wear the data glove for gesture recognition, and each person repeats the same gesture for 30 times. We record the times of correct recognition and the average recognition rate. Gesture recognition results are shown in Table 5, and the average recognition rate can reach 98.67%.

Table 5 Recognition results of gesture

Gesture number	1	2	3	4	5	Recognition rate 98.67%
Correct identification number	297	296	299	296	295	
Gesture number	6	7	8	9	10	
Correct identification number	295	294	296	299	293	

8 Conclusions

The paper uses the combination of chaos algorithm and genetic algorithm to optimize the BP neural network and to recognize the gesture based on BP neural network. First of all, the topology of BP neural network is established, the weights and threshold value of neural network are encoded to form the individual species, and the initial optimal solution is obtained through the genetic algorithm selection, crossover and mutation operations. Then, the chaotic disturbance is added to the initial optimal solution to make the iteration working to find the optimal solution. Finally, the optimal solution is used as the initial weights and thresholds of BP neural network to learn. Experimental results show that the BP neural network is easy to fall into local minimum, and the convergence is slower. The modified BP neural network based on genetic algorithm solves the problem of BP neural easy to fall into the local minimum, but the learning speed is not increased obviously. The CGA modified BP neural network can not only solve the problem that BP neural network is easy to fall into local minimum, but also accelerate the convergence speed, and the precision of the optimized BP neural network is improved obviously.

References

[1] Z. X. Huang, B. Peng, J. Wu. Research and application of human-computer interaction system based on gesture recognition technology. In *Proceedings of International Conference on Computer and Computing Technologies in Agriculture*, Beijing, China, pp. 210–215, 2012.

[2] C. N. Song. Android-based remote-control with real-time video surveillance for Wi-Fi robot. In *Proceedings of International Conference on Trustworthy Computing and Services*, Beijing, China, pp. 382–388, 2013.

[3] C. Lee, M. Kim, J. Park, J. Oh, K. Eom. Development of wireless RFID glove for various applications. *Security-Enriched Urban Computing and Smart Grid*, T. H. Kim, A. Stoica, R. S. Chang, Eds. Berlin, Heidelberg, Germany: Springer, pp. 292–298, 2010.

[4] S. S. Rautaray, A. Agrawal. Vision based hand gesture recognition for human computer interaction: A survey. *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015. DOI: 10.1007/s10462-012-9356-9.

[5] N. Zhang, S. Ding, J. Zhang, Y. Xue. Research on pointwise gated deep networks. *Applied Soft Computing*, vol. 52, pp. 1210–1221, 2017. DOI: 10.1016/j.asoc.2016.08.056.

[6] L. Jiang, Q. Q. Ruan. Research of gesture recognition based on neuron networks. *Journal of Beijing Jiaotong University*, vol. 30, no. 5, pp. 32–36, 2006. (In Chinese)

[7] L. S. Xiang, F. Qi, X. Y. Liu. A new optimization method for neural tree network model. *Control and Decision*, vol. 28, no. 1, pp. 73–77, 2013. (In Chinese)

[8] X. Q. Zeng. Study on the parameter optimization problem of BP neural network in the modeling. *Meteorological Monthly*, vol. 39, no. 3, pp. 333–339, 2013. (In Chinese)

[9] L. H. You, J. J. Wu, Y. Wang, S. J. Song. Optimized BP neural network based on simulated annealing algorithm for pH value prediction. *Chinese Journal of Sensors and Actuators*, vol. 27, no. 12, pp. 1643–1648, 2014. (In Chinese)

[10] Z. Salim. New time-varying fuzzy sets based on a PSO midpoint of the universe of discourse. *International Journal of Automation and Computing*, vol. 13, no. 4, pp. 392–400, 2016. DOI: 10.1007/s11633-016-0988-0.

[11] S. N. Pawar, R. S. Bichkar. Genetic algorithm with variable length chromosomes for network intrusion detection. *International Journal of Automation and Computing*, vol. 12, no. 3, pp. 337–342, 2015. DOI: 10.1007/s11633-014-0870-x.

[12] E. A. Dil, M. Ghaedi, A. Asfaram, F. Mehrabi, A. A. Bazrafshan, A. M. Ghaedi.. Trace determination of safranin O dye using ultrasound assisted dispersive solid-phase micro extraction: Artificial neural network-genetic algorithm and response surface methodology. *Ultrasonics Sonochemistry*, vol. 33, pp. 129–140, 2016. DOI: 10.1016/j.ultsonch.2016.04.031.

[13] C. A. C. Coello, E. M. Montes. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, vol. 16, no. 3, pp. 193–203, 2002. DOI: 10.1016/S1474-0346(02)00011-3.

[14] J. W. Peng, W. H. Lv, H. Y. Xing, X. J. Wu. Temperature compensation for humidity sensor based on improved GA-BP neural network. *Chinese Journal of Scientific Instrument*, vol. 34, no. 1, pp. 153–160, 2013. (In Chinese)

[15] L. Q. Xiao, H. X. Wang, H. L. Cheng, X. J. Xu. Topology optimization of ERT finite element model based on improved GA. *Chinese Journal of Scientific Instrument*, vol. 33, no. 7, pp. 1490–1496, 2012. (In Chinese)

[16] Q. Li, J. Gong, J. F. Tang. Multi-objective particle swarm optimization algorithm for cross-training programming. *Control Theory & Applications*, vol. 30, no. 1, pp. 18–22, 2013. (In Chinese)

[17] M. Islam, M. R. Rana, T. Rahman, M. Shahjahan. A biologically plausible neural network training algorithm with composite chaos. In *Proceedings of the 15th International Conference on Computer and Information Technology*, IEEE, Chittagong, Bengal, pp. 15–20, 2012.

[18] L. G. Chen, H. D. Chiang, N. Dong, R. P. Liu. Group-based chaos genetic algorithm and non-linear ensemble of neural networks for short-term load forecasting. *IET Generation, Transmission & Distribution*, vol. 10, no. 6, pp. 1440–1447, 2016. DOI: 10.1049/iet-gtd.2015.1068.

[19] A. Rasoolzadeh, M. S. Tavazoei. Prediction of chaos in non-salient permanent-magnet synchronous machines. *Physics Letters A*, vol. 377, no. 1–2, pp. 73–79, 2012. DOI: 10.1016/j.physleta.2012.10.053.

[20] S. Farzin, P. Ifaei, N. Farzin, Y. Hassanzadeh, M. T. Aalami. An investigation on changes and prediction of Urmia Lake water surface evaporation by chaos theory. *International Journal of Environmental Research*, vol. 6, no. 3, pp. 815–824, 2012.

[21] L. S. Yin, Q. Jiang, Q. Z. Hu. Research on wavelet neural network traffic flow model and rediction based on chaos algorithm. *Chinese Journal of Scientific Instrument*, vol. 30, no. 6, pp. 405–409, 2009. (In Chinese)

[22] K. R. Nirmal, N. Mishra. 3D graphical user interface on personal computer using p5 Data Glove. *International Journal of Computer Science Issues*, vol. 8, no. 5, pp. 155–160, 2011.

[23] Y. J. Wang, X. D. Wang, Y. Q. Zhou, Y. X. Yan. GA-BP network based battery SOC prediction for quasi anti-damage power supply. *Electric Machines and Control*, vol. 14, no. 6, pp. 61–65, 2010. (In Chinese)

[24] J. F. Yao, C. Mei, X. Q. Peng. The application research of the chaos genetic algorithm (CGA) and its evaluation of optimization efficiency. *Acta Automatica Sinica*, vol. 28, no. 6, pp. 935–942, 2002. (In Chinese)

[25] N. Tosun, L. Özler. A study of tool life in hot machining using artificial neural networks and regression analysis method. *Journal of Materials Processing Technology*, vol. 124, no. 1-2, pp. 99–104, 2002. DOI: 10.1016/S0924-0136(02)00086-9.

[26] I. Benacer, Z. Dibi. Extracting parameters of OFET before and after threshold voltage using genetic algorithms. *International Journal of Automation and Computing*, vol. 13, no. 4, pp. 382–391, 2016. DOI: 10.1007/s11633-015-0918-6.

[27] L. Mourelle, R. E. Ferreira, N. Nedjah. Migration selection of strategies for parallel genetic algorithms: Implementation on networks on chips. *International Journal of Electronics*, vol. 97, no. 10, pp. 1227–1240, 2010. DOI: 10.1080/00207217.2010.512021.

[28] X. P. Chen, S. L. Yu. Improvement on crossover strategy of real-valued genetic algorithm. *Acta Electronica Sinica*, vol. 31, no. 1, pp. 71–74, 2003. (In Chinese)

[29] A. Avancini, M. Ceccato. Comparison and integration of genetic algorithms and dynamic symbolic execution for security testing of cross-site scripting vulnerabilities. *Information and Software Technology*, vol. 55, no. 12, pp. 2209–2222, 2013. DOI: 10.1016/j.infsof.2013.08.001.

[30] Y. Q. Huang, C. Y. Liang, S. L. Yang, Q. Lu. Interactive genetic algorithm based on accelerating convergent mutation strategy. *Journal of System Simulation*, vol. 19, no. 9, pp. 1913–1916, 2007. (In Chinese)

[31] C. M. Fernandes, J. L. J. Laredo, A. M. Mora, A. C. Rosa, J. J. Merelo. The sandpile mutation operator for genetic algorithms. In *Proceedings of International Conference on Learning and Intelligent Optimization*, Rome, Italy, pp. 552–556, 2011.

[32] D. P. Tian, T. X. Zhao. Particle swarm optimization based on Tent map and Logistic map. *Journal of Shaanxi University of Science & Technology*, vol. 28, no. 2, pp. 17–23, 2010. (In Chinese)

[33] M. W. Fagerland, D. W. Hosmer. Tests for goodness of fit in ordinal logistic regression models. *Journal of Statistical Computation and Simulation*, vol. 86, no. 17, pp. 3398–3418, 2016. DOI: 10.1080/00949655.2016.1156682.

[34] M. Javidi, R. Hosseinpourfard. Chaos genetic algorithm instead genetic algorithm. *The International Arab Journal of Information Technology*, vol. 12, no. 2, pp. 163–168, 2015.

Dong-Jie Li received the B. Sc. degree in mechanical design manufacturing and automation, Harbin University of Science and Technology, China in 2004, the M. Sc. degree in mechatronic engineering from Harbin University of Science and Technology, China in 2007, and the Ph. D. degree in mechanical manufacture and automation from Harbin University of Science and Technology, China in 2009. From 2010 to 2013, she was a post-doctoral researcher at Harbin Institute of Technology. She visited the Georgia Institute of Technology, USA from March 2016 to March 2017. She is currently a professor of Harbin University of Science and Technology. She is the author and co-author of more than 50 publications.

Her research interests include micro/nano manipulation, application of intelligent control, and intelligent mechatronic system.

E-mail: dongjieli2013@163.com (Corresponding author)
ORCID iD: 0000-0002-1005-7890

Yang-Yang Li received the B. Sc. degree in automation, Harbin University of Science and Technology, China in 2015. His now a postgraduate in control theory and control engineering, Harbin University of Science and Technology, China.

His research interest is robot intelligent control.

E-mail: 18246058061@163.com

Jun-Xiang Li received the B. Sc. degree in automation, Harbin University of Science and Technology, China in 2013, the M. Sc. degree in pattern recognition and intelligent system from Harbin University of Science and Technology, China in 2016. He is now working for Delta Electronics Co. Ltd.

His research interests include pattern recognition and intelligent control system.

E-mail: 1007844868@qq.com

Yu Fu received the B. Sc. degree in automation, Harbin University of Science and Technology, China in 2015. She is now a postgraduate in control engineering, Harbin University of Science and Technology, China.

Her research interest is application of intelligent control.

E-mail: 173325586@qq.com