



---

## MATLAB network model simulation based on BP neural network

Chenchen Li, Yuan Fu, Xiaolin Wang, Shulong Dong, Ziwen Song

School of Transportation and Vehicle Engineering, Shandong University of Technology, Zibo 255000, China

**Abstract** With the development of china's economy this year, the demand for automobiles is increasing, so the number of car accidents in the country each year is also more and more concerned, this paper first collects the number of car accidents in the national calendar year, and then uses these data to make predictions with MATLAB using BP neural network model and improved radial base function network, and finally analyzes the advantages and disadvantages of the model according to the prediction results, so as to achieve a deep understanding of the principle and application of BP neural network and radial base function network.

**Keywords** artificial neural network; BP neural network; network model simulation; MATLAB data processing

---

### Introduction

Matlab [1-2] is a high-performance numerical computing and visualization software introduced by MathWorks in 1982, which combines numerical analysis, matrix operations, signal processing, and graphics display in one body to create a convenient and user-friendly user environment. MatLAB's launch has attracted wide attention from experts and scholars in various fields, and its powerful extension capabilities provide the basis for applications in various fields. These experts and scholars have introduced MATLAB toolboxes, which mainly include signal processing, control systems, neural networks, image processing, robust control, nonlinear system control design, system identification, optimization, fuzzy logic, wavelets, splines, communications and statistics, and the toolbox is constantly increasing, these toolboxes for research and engineering applications in various fields provide a powerful tool. In addition, MATLAB has the following advantages:

- (1) the language is simple, library functions are rich, compressing all unnecessary programming work.
- (2) Operators are rich, grammar restrictions are not strict, program design freedom is large, and the program can be ported well, basically do not make modifications can be run on various models of computers and operating systems.
- (3) Graphics are powerful and the visualization of data is very simple.
- (4) The openness of the original procedure. With the exception of the in-house functions, all MATLAB core and toolbox files are readable and modifiable source files that users can modify and add to their own files to form a new toolbox.

With the development of analytical chemistry, chemical econometrics has been produced. The subject is a marginal subject formed by the intersection of chemistry, mathematics, statistics and computer science, which uses mathematical and statistical methods to design or select optimal measurement programs and experimental methods, and obtains the maximum information by analyzing chemical measurement data. The computational problems in chemical metrology are generally more complex, especially the processing of high-dimensional data involves a large number of matrix operations, and the MATLAB language provides powerful tools for chemical test design and data processing because of its unique advantages and powerful computing functions.



## 1. Neural network introduction

### 1.1 Artificial neural network

Artificial neural networks are mathematical models that process information using structures similar to those of the brain's synapses. In engineering and academia is also often referred to directly as neural networks or neural networks. Neural networks are computational models consisting of a large number of nodes (or neurons) and interconnects. Each node represents a specific out-of-the-way function called an incentive function. The connection between each of the two nodes represents a weighted value for the signal passing through the connection, called a weight, which is equivalent to the memory of an artificial neural network. The output of the network varies depending on how the network is connected, the weight value and the excitation function. The network itself is usually an approximation of some algorithm or function in nature, or it may be an expression of a logical strategy. Artificial neural networks have the following characteristics:

(1) nonlinear nonlinear relationships are a common feature of nature. The wisdom of the brain is a nonlinear phenomenon. Artificial neurons are activated or inhibited in two different states, which mathematically manifest themselves as a nonlinear network relationship. Networks of thresholded neurons provide better performance and improve fault tolerance and storage capacity.

(2) Non-limited a neural network is usually made up of multiple neurons connected widely. The overall behavior of a system depends not only on the characteristics of individual neurons, but may also be determined primarily by the interaction and interconnection between cells. Associative memory is a typical example of a non-limitation by simulating the brain's non-limitations through a large number of connections between cells.

(3) Very qualitative artificial neural network has adaptive, self-organizing and self-learning ability. Neural networks can not only process information can vary, but also process information at the same time, nonlinear power system itself is constantly changing. Iterative processes are often used to describe the evolution of power systems.

(4) The evolution direction of a non-convex system will depend on a particular state function under certain conditions. For example, an energy function whose extreme value corresponds to the system's relatively stable state. Non-convexity means that this function has multiple extremely straight, so the system has more than one stable equilibrium state, which will lead to the diversity of system evolution.

### 1.2. BP Neural Network Model

The BP (Back Propagation) network, proposed in 1986 by a team of scientists led by Rumelhart and McClelland, is a multi-layered feed-forward network trained by error inverse propagation algorithms and is one of the most widely used neural network models. The BP network can learn and store a large number of input-output pattern mapping relationships without having to reveal mathematical equations that describe this mapping relationship in advance. Its learning rule is to use the fastest drop method, through reverse propagation to constantly adjust the weight and value of the network, so that the network error squared to the minimum. The BP neural network model topology includes the input layer, the hidden layer (hide layer), and the output layer (output layer) [9-10].

### 1.3. Radial base function network model

Radial Basis Function, RBF) neural network consists of three layers, with input layer nodes passing only input signals to the hidden layer, which consists of radiant functions like gaussian functions, and output layer nodes, which are usually simple linear functions. The function of action (base function) in the hidden layer node will respond to the input signal locally, that is, when the input signal is near the central range of the base function, the hidden layer node will produce a larger output, which shows that the network has local approximation capability, so the radial base function network is also called the local perceived field network.

## 2. Modeling

Step 1: According to the data to select the structure of BP neural network, the number of BP neural network model network layers used in this paper is 2, the number of hidden layer neurons is 10, the selection of hidden

layer and output layer neuron functions are tansig function and purelin function, the network training method uses gradient drop method, dynamic gradient drop method and adaptive lr gradient drop method respectively.

Step 2: The input and output data are normalized.

Step 3: There is a function newff ( ) to construct neural networks.

Step 4: Before training the neural network, set the relevant parameters, such as the maximum number of training, training requirements accuracy, learning rate and so on.

Step 5: Train BP neural networks.

Step 6: Repeat the training until the requirements are met.

Step 7: Save the trained neural network and make predictions with the trained neural network.

Step 8: Compare the predicted value with the actual output value to analyze the stability of the model.

### 3. Simulation

BP network neural network model simulation, matlab source code as follows:

```
x , . y=[ ]; p=x'; t=y'; [pn, minp,maxp, tn, mint , maxt]=premnmx(p,t); figure (1); plot (pn, tn,'-'); title ('Actual
input and actual output diagram', 'fontsize', 12) ylabel ('actual output', 'fontsize', 12)xlabel ('sample', 'fontsize',
12) net=newff (minmax(pn), '10 1', 'tansig'pure','traingd'); net. trainParam.epochs=50000 ;
```

```
net. trainPar am .goal=0.0001;
```

```
net.trainPar am .lr=0.01; %net .trainParam.min_ grad=1e-50; net=train(net,pn,tn); t2=sim(net,pn); figure(2);
plot(pn,tn,'r',pn,t2,'b'); legend ('expected output', 'predicted output') title ('predicted output vs. actual output',
'fontsize', 12)
```

```
ylabel (' P XIII', ' fontsize' , 12)xlabel ( ' f4', ' fontsize' , 12)figure (3)plot (pn, t2,' :og' );
```

```
hold onplot (pn, tn,'-*'); legend ('predicted output', 'expected output') title ('BP network prediction output',
'fontsize', 12)
```

```
ylabel ('function output', 'fontsize', 12) xlabel ('sample', 'fontsize', 12)errors t2-tn;
```

```
figure(4) plot (error, '-') title ('BP network prediction error,' 'fontsize', 12)ylabel ('error,' 'fontsize', 12)xlabel
('sample,' 'fontsize', 12)figure (5)plot (tn-t2)./t2,") title ('Percentage of neural network prediction error')
```

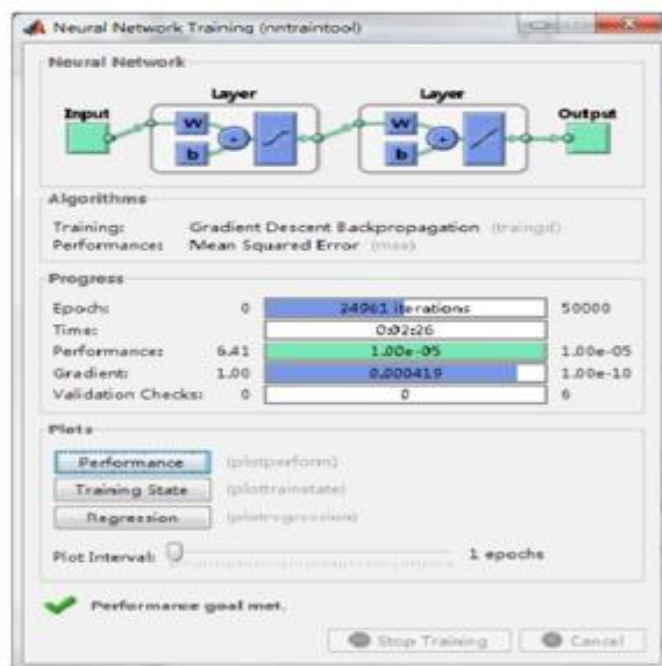
```
errorsum-sum (error)%output the trained weights and thresholds iw1-net. IW111;
```

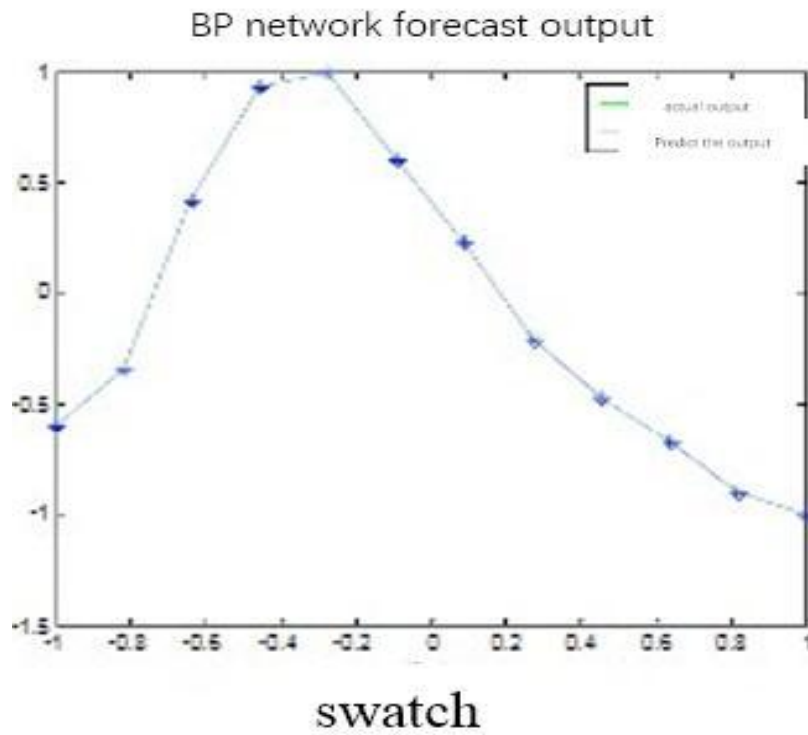
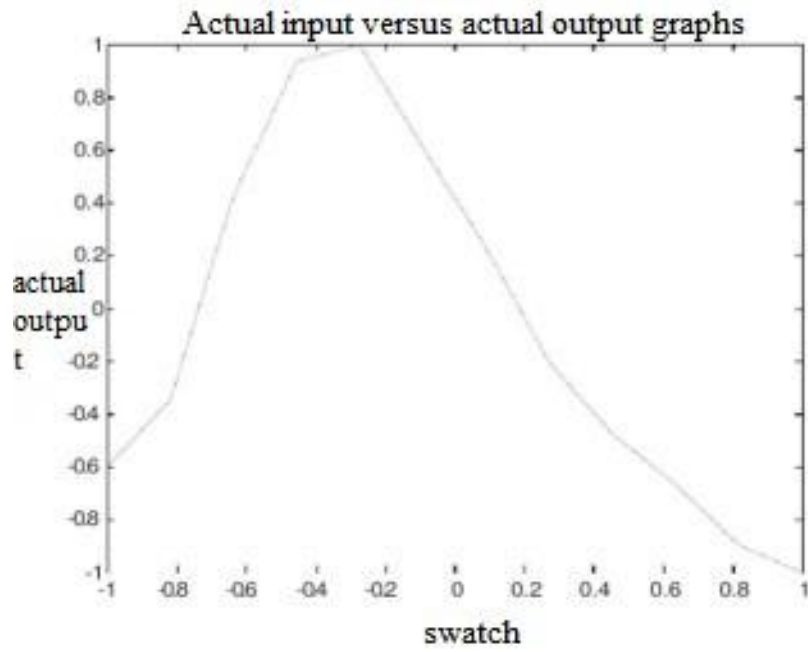
```
b1=net.b(1);
```

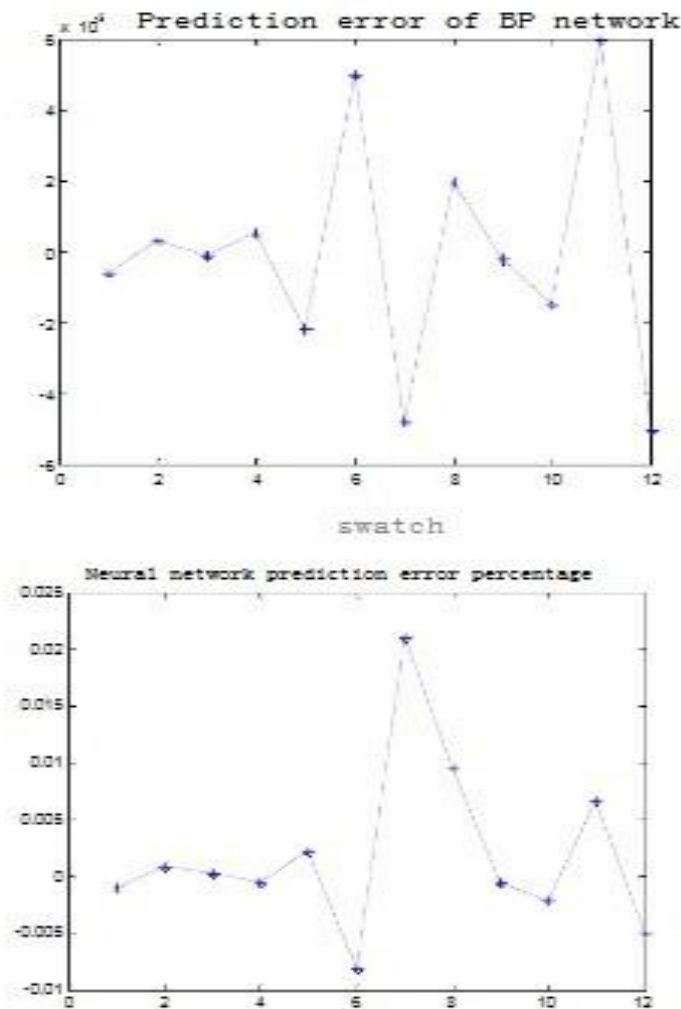
```
1w2=net. LMI21;
```

```
b2-net.bl21;
```

The simulation results are shown in the following image:







The weight of the first layer of the neural network is:

13. 6043215790013

-13.6781856692610

-14.0649206962947

-13.9848650272192

-13.9871960386982

-13.9951268043836

-14.0071081197679

-13 9888866256635

14. 1108167288415

13. 9653714912941

The bias of layer1 of the neural network is:

-14.3935492958342

11.2816707014364

7.64317226847366

4.70625608385651

1.69507957693330

-1.57106974025717

-4.61930431372367.

-7.79883621089532.

10.7542837046117



14.0333143274359

The weight of the second layer of the neural network is :

0.519149236262203 0. 251828769407092

0.102735654554234

0.150414\*\*\*\*\*

0.281458748263410

0.309838232092716

-0.0253911425777369 -0. 289400126603914

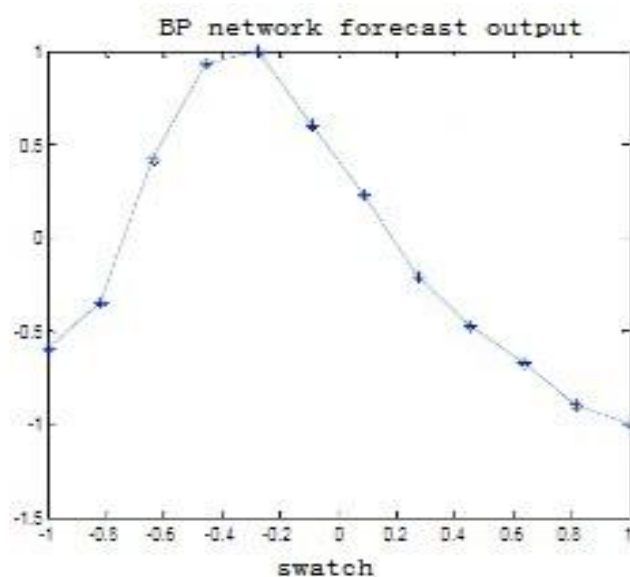
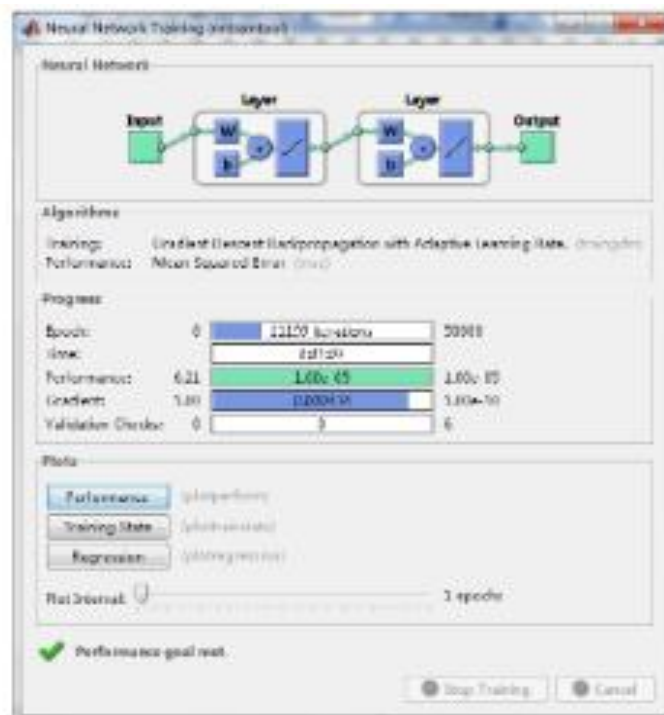
0.439990868328409

0.108195723319044

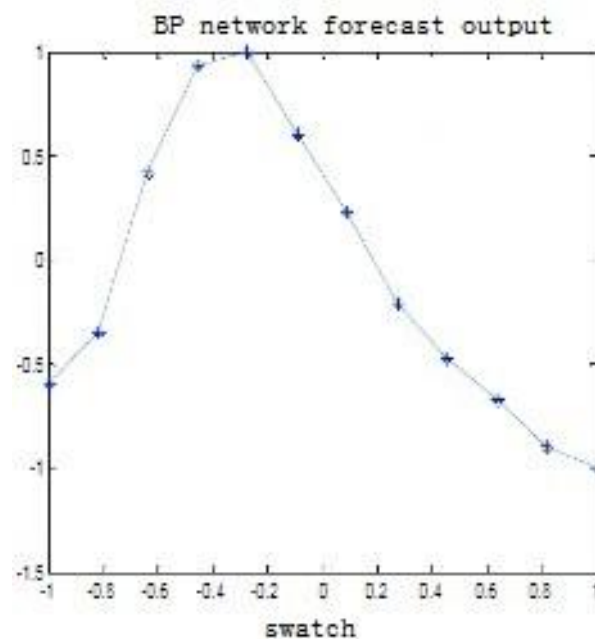
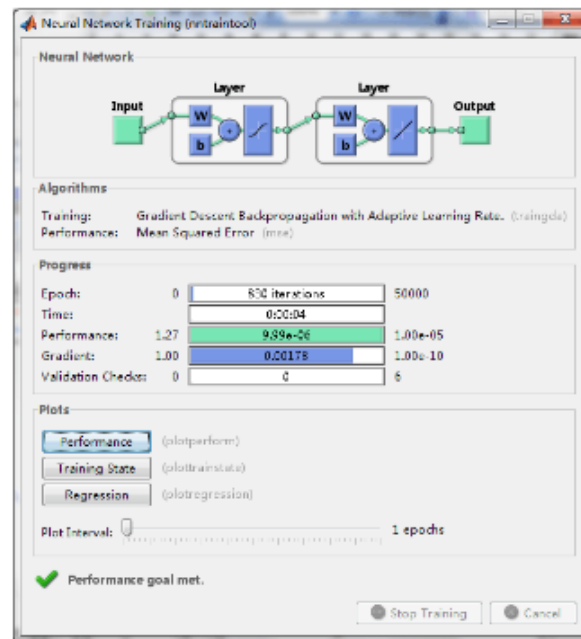
The bias of the second layer of the neural network is :

-0.427391472171392

(2) Themodel training method is the gradient drop method with momentum and the function is traingdm



(3) The model training method is the gradient drop method with adaptive lr, and the function is traingda



#### 4 Rating

(1) The effect of the initial value

In this paper, the BP neural network model is built using the newff() function, because of the randomness of newff(), so basically every training result is different. Feed-forward neural networks must initialize weights and thresholds prior to training, and newff() can automate this process, but when weights and thresholds are initialized randomly, this is not studied in detail.

(2) The effect of the training function

As can be seen from the results graph given above, the effect of training function on model training is significant, the network training method uses gradient drop method, dynamic gradient drop method and adaptive lr gradient drop method respectively, it can be seen that the gradient drop method is required to train 24961 steps to achieve the required training accuracy, while the use of momentum gradient drop method and adaptive



Its gradient drop method only needs 11199 steps and 830 steps, respectively. The number of training sessions is greatly reduced, but the trained models are as accurate as they are.

### (3) Stability

The BP neural network model has a few drawbacks:

First of all, because the learning rate is fixed, the convergence speed of the network is slow and requires a long training time. For some complex issues, the BP algorithm can take a very long time to train, mainly due to the small learning rate, which can be improved with a variable learning rate or adaptive learning rate.

Second, the BP algorithm converges the weight to a value, but does not guarantee that it is the global minimum of the error plane, because a gradient drop method may produce a local minimum value. For this problem, additional momentum method can be used to solve.

Third, there is no theoretical guidance on the choice of layers and cells of the network implied layer, which is generally determined by experience or by repeated experiments. Therefore, the network often has a lot of redundancy, to a certain extent, also increased the burden of network learning.

Finally, the learning and memory of the network is unstable. That is, if you add a learning sample, a well-trained network needs to be trained from scratch, and there is no memory of the previous weights and thresholds. However, you can save better weights for predictions, classifications, or clusters.

### (4) Generalization capability

The generalization ability of BP neural network model is very weak, and learning and memory are unstable, so in practical application, the original BP algorithm is difficult to do. Finally, the simulation results and source code of radial base function network are given, and compared with BP neural network model.

The simulated matlab source code for the radial base function network is as follows:

```
clear
clc
x=[]; % input
y=[]; % output
p=x';
t=y';
net=newrb(p,t);
Y=sim(net,p);
figure(1)
plot(p,Y,'og');
hold on
plot(p,t,'-*');
legend ('predicted output', 'expected output')
title ('Radial base function network output', 'fontsize', 12)
ylabel ('function output', 'fontsize', 12)
xlabel ('sample', 'fontsize', 12)
error=Y-t;
figure(2)
plot(error,'-*')
title ('Radial base function network prediction error', 'fontsize', 12)
ylabel ('error', 'fontsize', 12)
xlabel ('sample', 'fontsize', 12)
```

As can be seen from the figure above, it is also very good to simulate the radial base function network model with the newrb() function, and the model is simpler than the BP neural network model and requires fewer parameters to set. Stability is also much better than the BP model.

## References

- [1]. Wang Morran MATLAB610 with scientific calculations. Beijing: Electronic Industry Press, 2001, 1-6.
- [2]. Li Li, Wang Zhenli MATLAB Engineering Computing and Applications (M) Beijing: People's Post and





- Telecommunications Press, 2001, 1-3
- [3]. Han Liqun. Artificial Neural Network Theory, Design and Application of Human Nerve Cells, Artificial Neural Networks and Artificial Nervous System Beijing: Chemical Industry Press, 2002.
  - [4]. Zhang Xuegong. Research on Support Vector Machine Algorithms Based on Statistical Learning Theory, *Journal of Automation*, 2000, 78(1):322-42
  - [5]. Vapnik V. The Nature of Statistical Learning Theory. New York ; Springer Verlag, 1995.
  - [6]. Han Liqun, artificial neural network theory, design and application of human nerve cells, artificial neural networks and artificial neural systems. United States of China .M. Beijing: Chemical Industry Press, 2002.
  - [7]. So high. Artificial Neural Network Principles and Simulation Examples Beijing: Mechanical Industry Press 2003.
  - [8]. The ordinary, Zhang Changshui Artificial Neural Networks and Simulated Evolutionary Computing .M. Beijing: Tsinghua University Press 2000
  - [9]. Wang Yumei. Application of MATLAB program design in mathematical modeling.
  - [10]. Yu Shengwei. MATLAB Mathematical Modeling Classic Case Study ( M) Beijing: Tsinghua University Press, 2015.

