Research on House Price Prediction Based on Multi-Dimensional Data Fusion

Yang Yonghui
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: yangyh26@qq.com

***Abstract*—The price of commercial housing is related to the process of urbanization in China and the living standard of residents, so the prediction of the price of commercial housing is very important. A major difficulty in predicting regression problems is how to handle different attribute types and fuse them. This paper proposes a house price prediction model based on multi-dimensional data fusion and a fully connected neural network. The model building steps are: First, normalize the data involved in the sample; then, interpolate the normalized data to increase the data density; subsequently, the normalized sample data is converted into a pixel matrix; finally, a fully connected neural network model is established from the pixel matrix to the price of the commercial house. After the neural network model has been established, the price of house can be obtained by entering the attributes of the house into the neural network model.**

Keywords-Multi-Dimensional Data Fusion; Fully Connected Neural Network Model; House Price Prediction

I. INTRODUCTION

Urbanization[1], also known as urbanization and urbanization, refers to the process of population gathering towards cities, the expansion of cities, and the series of economic and social changes that result from it. The essence is the changes in economic, social, and spatial structures. Modernization is the core proposition of China's modernization process and sustained economic growth. In recent years, with the further progress of China's urbanization process, more and more young people have begun to enter second-tier, third tier and even first-tier cities. A major factor affecting young people's entry into big cities is the price of local commercial housing. In other words, a major factor affecting China's urbanization process is the price of urban house. This shows that it is necessary to forecast house prices. The attributes that affect house prices are transaction date, house age, distance from the subway station, the number of convenience stores in the walking circle, the dimension of the house, and the longitude of the house. This paper will build a data fusion model. The input information of this model is the seven factors that affect house prices, and the output information is the price of commercial housing. After the data fusion model has been established, only the attributes that affect house prices are entered into the data fusion model, and the price of the commercial house can be obtained.

A. Research Background and Significance

With the development of China's economy, people's living standards have gradually improved, and economic development has made people have a higher pursuit of living places. According to data from the National Bureau of Statistics[2]: from January to December 2018, the investment in real estate development nationwide was 12,266.4 billion yuan, an increase of 9.5% over the previous year, and the growth rate was 0.2 percentage points lower than the January-November period, an increase from the same period of the previous year. 2.5 percentage points. Among them, residential investment was 8,529.2 billion yuan, an increase of 13.4%, a 0.2 percentage point drop from January to November, and an increase of 4 percentage points from the previous year. The proportion of residential investment in real estate development investment was 70.8%. With the increase in housing sales, housing prices have also increased. According to relevant data, China's housing prices have at least doubled from 2015. With the increase of house prices, people pay more attention to the prediction of house prices. This paper will build a data fusion model. The input information of this model is six attributes that affect house prices: transaction date, house age, distance from the subway station, the number of convenience stores in the walking circle, the dimension of the house, and the longitude of the house; the output is the price of the commercial house. After the data

DOI: 10.21307/ijanmc-2020-001

fusion model has been established, only the six attributes that affect house prices are entered into the data fusion model, and the price of the commercial house can be obtained. The research of house price prediction based on multi-dimensional data fusion can provide reference for China's house price prediction and further promote the development of urbanization in China.

B. Data sources

The data in this paper comes from the Boston house price data provided by Kaggle, and the amount of data selected is relatively small. The data set contains 404 training samples and 102 test samples, for a total of 506 sample data. There are 6 attributes that affect house prices in house price forecasts. In the problem of house price prediction, the attributes that affect house prices are: transaction dateX_1, house ageX_2, distance from the subway stationX_3, the number of convenience stores in the walking circleX_4, the dimension of the houseX_5, and the longitude of the houseX_5; Dependent variable is house priceY.

II. KEY TECHNOLOGY

A. Research methods for regression problems

House price forecasting is a forecasting problem, and forecasting problems are regression analysis. This section aims to state the research methods of regression analysis. Regression analysis[3] is a method of statistically analyzing data. The purpose is to understand whether two or more variables are related, the direction, and strength of the correlation and establish a mathematical model to observe specific variables to predict the variables of interest to researchers. The roles in regression analysis are independent and dependent variables: the independent variable is a variable that actively changes, for example, several factors that affect house prices in this paper are independent variables; the dependent variable is a passively generated due to changes in independent variables, such as housing prices in this paper, are a dependent variable. Regression analysis can also be understood as a method for analyzing the relationship between independent and dependent variables. The regression analysis methods are linear regression, logistic regression, and polynomial stepwise regression.

Linear regression is a linear equation established between the independent variable and the dependent variable. This is the most well-known regression model. In this type of model, the independent variable may be discrete or continuous; the dependent variable must be continuous, and the nature of linear regression is linear. Logistic regression is a logistic equation built from independent variables to dependent variables. This is a regression model used to calculate the success or failure of an event. In this type of model, the independent variable may be discrete or continuous; the dependent variable must be in the interval $[0,1]$. Polynomial regression is a polynomial equation established between the independent variable and the dependent variable. This is a polynomial regression model commonly used in the field of deep learning. Under this model, a low polynomial degree leads to underfitting, and a high polynomial degree leads to overfitting. When dealing with multiple independent variables, stepwise regression is needed[4]. Standard stepwise regression does two things, adding or removing independent variables at each step. In this technique, the selection of independent variables is done by means of an automated process, which does not involve manual intervention.

B. Research methods for data fusion

Data fusion[5] is a technology that fuses attribute values from different attributes. Fusion of multiple attributes will get better performance results than a single attribute. Data fusion is widely used in multidisciplinary and multi-scenario integration fields. For example, you can monitor the patient's physiological and psychological information through different hardware devices, and finally obtain the patient's physical condition through data fusion. There are many similar examples. There are also many difficulties in data fusion. The first is how to deal with different attributes, and the second is how to fuse the data.

There are many difficulties in data fusion design. The first is how to handle different attribute types, and the second is how to fuse attributes. This thesis will detail the processing method of the attribute type in the "Handling of attribute types" Section and the data fusion method in the "Data Fusion" Section.

C. Handling of attribute types

The attribute type refers to the data type of the attribute. The attribute types are: Large_Attributes, Small_Attributes, Intermediate_Attributes, and Interval_Attributes[6].

1) Large_Attributes

The Large_Attributes are the larger the independent variable, the larger the dependent variable, that is, the independent variable will have a positive benefit on the dependent variable, in other words, there is a positive correlation between the dependent variable and the independent variable. The processing method for very large attributes is shown in (1).

$$\mathrm{x}' = \frac{\mathrm{x}-\mathrm{x}_{\min}}{\mathrm{x}_{\max}-\mathrm{x}_{\min}} \quad (1)$$

Among them, x_{max} is the maximum value of the attribute value; x_{min} is the minimum value of the attribute value; x is the original value of the attribute value; x' is the normalized attribute value.

2) Small_Attributes

The Small_Attributes refers to: the larger the independent variable, the smaller the value of the dependent variable, that is, the independent variable will have a negative benefit on the dependent variable, in other words there is a negative correlation between the independent variable and the dependent variable. The processing method of extremely small attributes is shown in (2).

$$\mathrm{x}' = \frac{\mathrm{x}_{\max}-\mathrm{x}}{\mathrm{x}_{\max}-\mathrm{x}_{\min}} \quad (2)$$

Among them, x_{max} is the maximum value of the attribute value; x_{min} is the minimum value of the attribute value; x is the original value of the attribute value; x' is the normalized attribute value. After processing by the above method, the extremely small attributes have been transformed into extremely large attributes.

3) Intermediate_Attributes

Intermediate_Attributes refer to the existence of a threshold. When the independent variable is smaller than the threshold, it displays the characteristics of Large_Attributes. When the independent variable is larger than the threshold, it displays the characteristics of Small_Attributes. Specifically, when the independent variable is less than the threshold, there is a positive correlation between the independent variable and the dependent variable; when the independent variable is greater than the threshold, there is a negative correlation between the independent variable and the dependent variable. The processing method of Intermediate_Attributes is shown in (3).

$$\begin{cases} \mathrm{x}' = \frac{\mathrm{x}-\mathrm{x}_{\min}}{\mathrm{x}_0-\mathrm{x}_{\min}}, \mathrm{x} < \mathrm{x}_0 \\ \mathrm{x}' = \frac{\mathrm{x}_{\max}-\mathrm{x}}{\mathrm{x}_{\max}-\mathrm{x}_0}, \mathrm{x} > \mathrm{x}_0 \end{cases} \quad (3)$$

Among them, x_{max} is the maximum value of the attribute value; x_{min} is the minimum value of the attribute value; x is the original value of the attribute value; x' is the normalized attribute value; x_0 is the threshold. After processing by the above method, the interval attribute has been transformed into Large_Attributes.

4) Enumerated_Attributes

Enumerated_Attributes means that the attribute value of the independent variable does not have real measurement characteristics, and the result of the dependent variable will be affected by the value of the independent variable, but this influence relationship is difficult to express. The processing method of Enumerated_Attributes is as follows:

Step1: List all the values of the input attributes;

Suppose the input attribute contains l attribute values: x_1、x_2、…、x_l;

Step2: Convert the attribute value to One-Hot [7] form;

Among them, x_1 is the $1st$ attribute value, so a vector with only the $1st$ position being 1 can be used instead. That is, x_1 can be expressed as: $(1 \quad 0 \quad \cdots \quad 0 \quad \cdots \quad 0 \quad \cdots \quad 0)^T_{1\times l}$;

Among them, x_2 is the $2nd$ attribute value, so a vector with only the $2nd$ position being 1 can be used instead. That is, x_2 can be expressed as: $(0 \quad 1 \quad \cdots \quad 0 \quad \cdots \quad 0 \quad \cdots \quad 0)^T_{1\times l}$; ……

Among them, x_l is the lst attribute value, so a vector with only the lst position being 1 can be used instead. That is, x_l can be expressed as: $(0 \quad 0 \quad \cdots \quad 0 \quad \cdots \quad 0 \quad \cdots \quad 1)^T_{1\times l}$;

So far, all values of the attribute have been expressed as One-Hot form.

D. Data Fusion

This section analyzes the problem of data fusion, that is, how to merge Large_Attributes, Small_Attributes, Interval_Attributes, and Enumerated_Attributes together. This thesis will propose a pixel-based data fusion method: first establish a pixel matrix; then use a fully connected neural network model to process the pixel matrix.

1) Create a pixel matrix

This section aims to transform multiple attributes into a pixel arrangement. Specifically, it is assumed that the sample contains m samples and each sample contain n attributes, that is,

All values for the $1st$ sample are: x_{11}, x_{12}, …, x_{1i}, …, x_{1j}, …, x_{1m};

All values for the $2nd$ sample are: x_{21}, x_{22}, …, x_{2i}, …, x_{2j}, …, x_{2m};

……

All values for the mnd sample are:x_{m1}, x_{m2}, …, x_{mi}, …, x_{mj}, …, x_{mm}.

Then, the $1st$ pixel matrix is:$(x_{11} \quad x_{12} \quad \cdots \quad x_{1i} \quad \cdots \quad x_{1j} \quad \cdots \quad x_{1m})^T$;

and the $2nd$ pixel matrix is:$(x_{21} \quad x_{22} \quad \cdots \quad x_{2i} \quad \cdots \quad x_{2j} \quad \cdots \quad x_{2m})^T$;

……

and the $2nd$ pixel matrix is:$(x_{m1} \quad x_{m2} \quad \cdots \quad x_{mi} \quad \cdots \quad x_{mj} \quad \cdots \quad x_{mm})^T$.

2) Processing pixel matrix

In "Create a pixel matrix", this article has already established the number of pixel matrices as the number of samples, and then we need to use the neural network to process the pixel matrix.

The choice of network structure: there are many neural network model structures, such as fully connected layer neural networks, convolutional neural networks, long-short-term memory networks, and Residual network. Because the application scenario in this paper is simple, it is more appropriate to choose a fully connected neural network model.

Selection of activation function: The activation function is a function that runs on the neuron and is responsible for mapping the input of the neuron to the output. The activation functions are: $Sigmoid$ function (Figure 1 $Sigmoid$), $Tanh$function (Figure 2 $Tanh$), $ReLU$ function (Figure 3 $ReLU$), $Leaky\ ReLU$ function (Figure 4 $Leaky\ ReLU$), where $Leaky\ ReLU$ is a special form of $ReLU$. Regarding the selection principle of the activation function, Andrew Ng gives the following reference scheme in “Neural Networks and Deep Learning”: $Tanh$ is very common in machine learning. The activation function is generally defaulted to $Tanh$. $Leaky\ ReLU$ is generally better than $ReLU$, but the scope of use of $ReLU$ Wider; the activation function used in the output layer of the binary classification problem is $Sigmoid$, and $Sigmoid$ was rarely used in other cases; $Tanh$ is almost always better than $Sigmoid$. $Tanh$ and $Sigmoid$ have a disadvantage that when the independent variable is large, the slope is small. The gradient descent method is limited; except for the output layer, linear activation functions are rarely used; neural network models use activation functions, which will lead to the final result being a linear combination of input feature vectors.

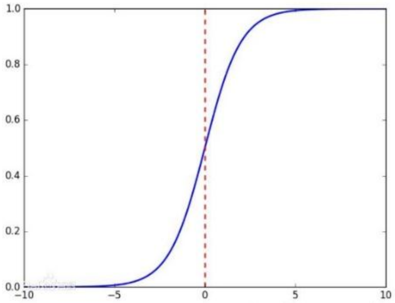


Figure 1. Sigmoid

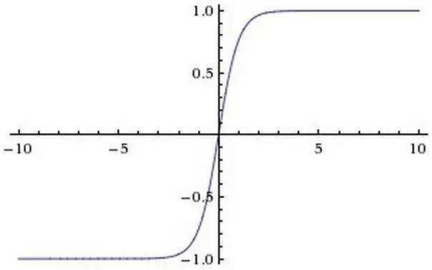


Figure 2. Tanh

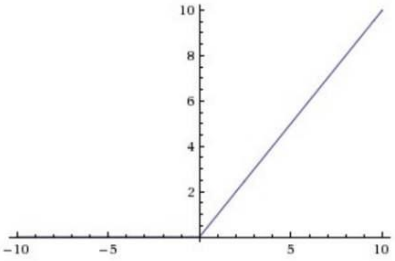


Figure 3. ReLU

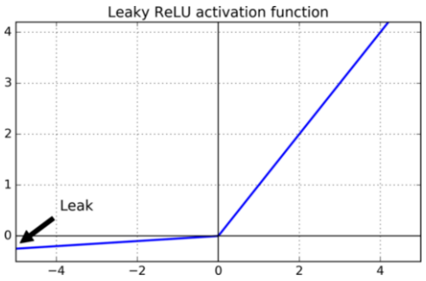


Figure 4. Leaky ReLU

III. NORMALIZATION OF ATTRIBUTES

This part needs to normalize the attributes involved in the data set: first analyze the data type of the attributes by "Attribute Analysis"; then normalize the attributes by "Normalization".

A. *Attribute Analysis*

As mentioned in "Data Sources", the data in this paper is derived from Boston house price data provided by Kaggle, and the amount of data selected this time is relatively small. The data set contains 404 training samples and 102 test samples, for a total of 506 sample data. In the problem of house price prediction, there are 6 attributes that affect house prices: transaction date X_1; house age X_2; distance from the subway station X_3; the number of convenience stores in the walking circle X_4; the dimension of the house X_5; the longitude of the house X_6;. dependent variable: house price Y.

Transaction date X_1 is a time variable; the house age X_2 is a Small_Attributes; distance from the subway station X_3 is a Small_Attributes; the number of convenience stores in the walking circle X_4 is a Large_Attributes; the dimension of the house X_5 and the longitude of the house X_6 are an Enumerated_Attributes.

B. *Normalization*

Transaction date X_1 is a time variable; the house age X_2 is a Small_Attributes; distance from the subway station X_3 is a Small_Attributes; the number of convenience stores in the walking circle X_4 is a Large_Attributes; the dimension of the house X_5 and the longitude of the house X_6 are an Enumerated_Attributes.

IV. DATA FUSION

In this part, the normalized data in "Normalization of attributes" needs to be fused: first, the pixel matrix is established by "Building a Pixel Matrix"; then the fully connected neural network model is established by "Building a Neural Network Model".

A. *Building a Pixel Matrix*

A pixel matrix can be established by "Data Fusion". As described in "Data sources", the data in this paper is derived from Boston house price data provided by Kaggle. The amount of data selected is small. The data set contains 404 training samples and 102 test samples, for a total of 506 sample data. Then there are:

All values for the $1st$ sample:$x_{11}, x_{12},\ldots, x_{17}$;

All values for the 2nd sample:$x_{21}, x_{22}, \ldots, x_{27}$;

……

All values for the 506st sample:x_{506_1}, x_{506_2}, …, x_{506_7}.

B. *Building a Neural Network Model*

The paper will eventually build a neural network model of house attributes to house prices: where the input attributes are house attributes: transaction date X_1; house age X_2; distance from the subway station X_3; the number of convenience stores in the walking circle X_4; the dimension of the house X_5; the longitude of the house X_6;output information is house price Y.

Step1: Design the network structure

Through the analysis of "Data Fusion", this paper will build a fully connected neural network model. The network model structure is shown in (Figure 5 Network structure): The input layer of the network structure contains 7 input nodes; the network structure contains 5 hidden layers, each of which contains 4 nodes; the output layer of the network structure contains 1 output node; all activation functions use $ReLU$ function; Training period: 50000; Target accuracy is: 10^{-5}; Learning rate: 0.01

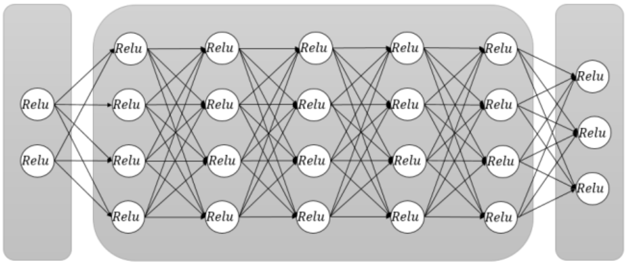


Figure 5. Network structure

Step2: Selection of training tools

There are many ways to train neural networks, such as Tensorflow, Caffe, MXNet, Torch, Theano in python, and nntool in Matlab. nntool is a network model training tool that is easy to deploy and simple in the environment. In this paper, the neural network model shown in (Figure 5 Network structure) is trained by nntool (Figure 6 nntool).

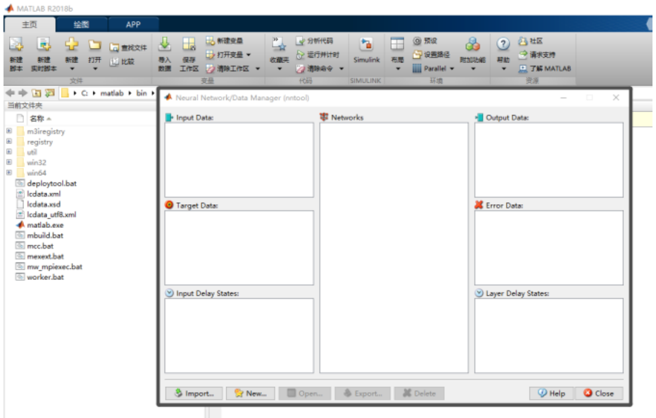


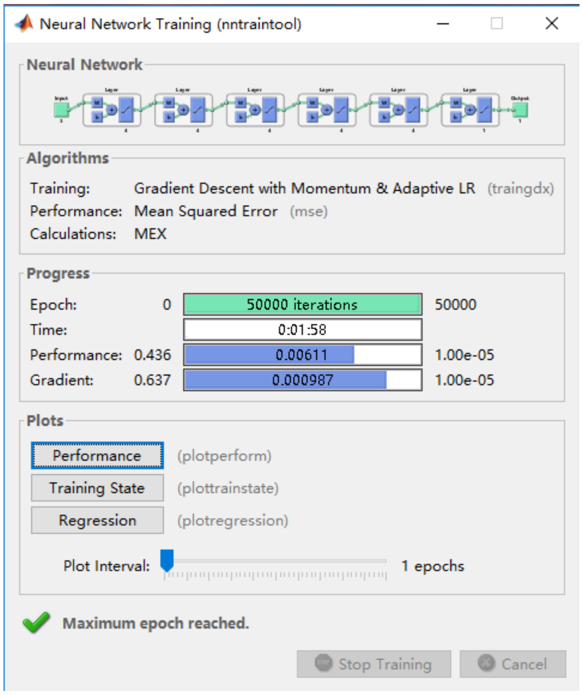
Figure 6. Nntool

Step3: Code design

See Appendix

Step4: Training process

In the process of neural network training using Matlab, part of the training process is shown in (Figure7 Training process). Among them, Performance is shown in (Figure 8 Performance); Training State is shown in (Figure9 Training State); Regression is shown in (Figure10 Regression).



Neural Network Training (nntraintool)

Neural Network

Input
Layer
Layer
Layer
Layer
Layer
Layer
Output

Algorithms

Training: Gradient Descent with Momentum & Adaptive LR (traingdx)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	50000 iterations	50000
Time:		0:01:58	
Performance:	0.436	0.00611	1.00e-05
Gradient:	0.637	0.000987	1.00e-05

Plots

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

✓ Maximum epoch reached.

Figure 7. Training process

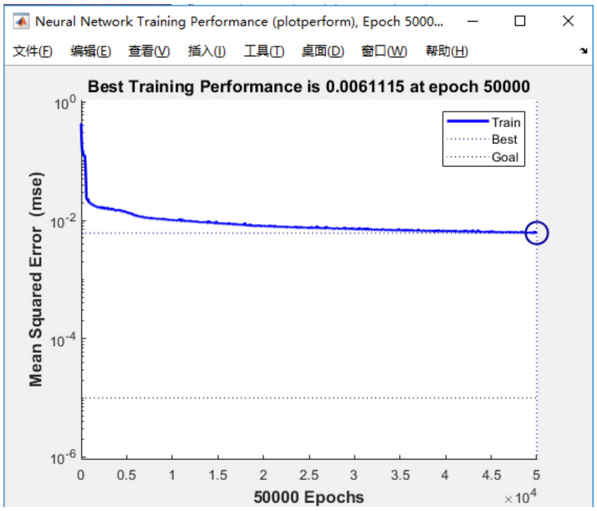


Figure 8. Performance

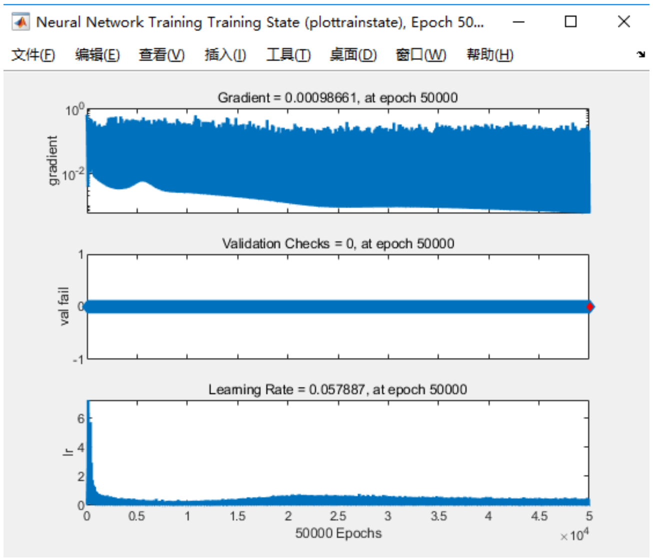


Figure 9. Training State

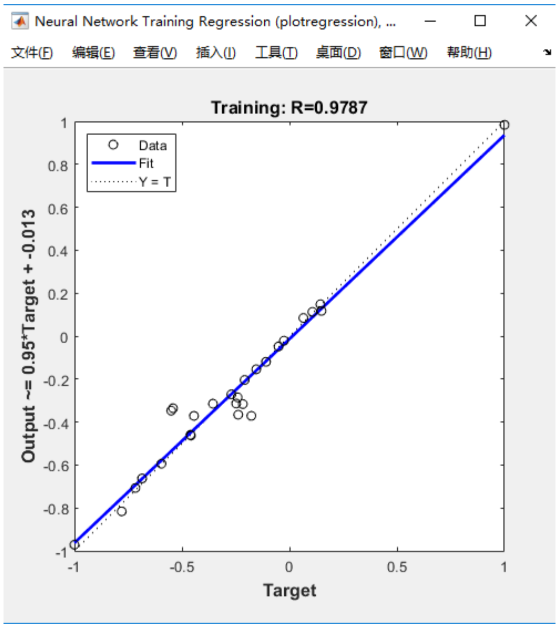


Figure 10. Regression

Step5: Results

The results of the neural network model include two parts: one is the partial result display, as shown in (Figure11 Result); the other is the error proportion chart, as shown in (Figure12 error_raph). As can be seen from the (Figure10 Regression), the accuracy of the network model is 97.87%.

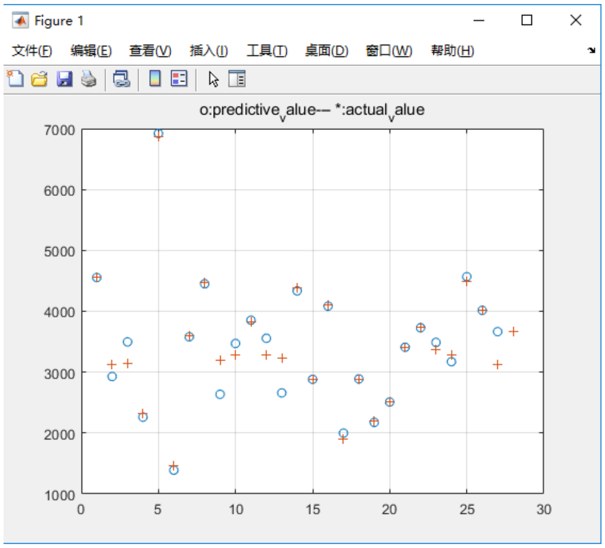


Figure 11. Result

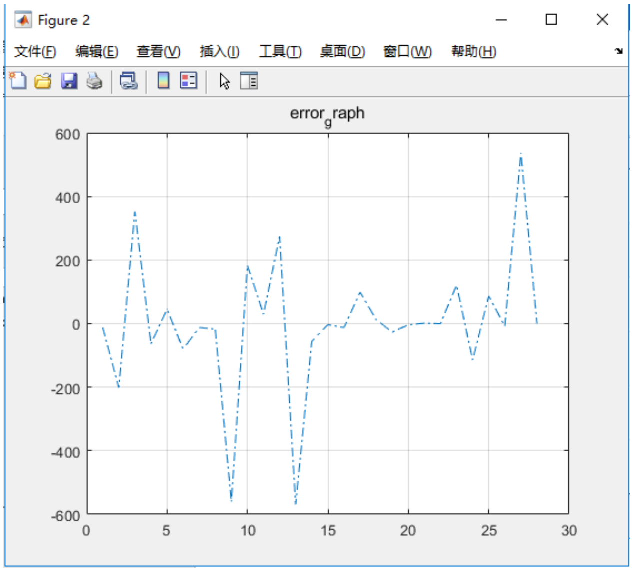


Figure 12. Error_raph

V. SUMMARY

This paper finally established a neural network model from house attributes to house prices: where the input attributes are commodity house attributes: transaction date X_1; house age X_2; distance from the subway station X_3; the number of convenience stores in the walking circle X_4; the dimension of the house X_5; the longitude of the house X_6 ;output information is house price Y.After the neural network model has been established, Enter the six attributes of the commercial house into this neural network model, and you can get the corresponding house price. The accuracy of the network model is 97.87%.

VI. APPENDIX

```
[pn,minp,maxp,tn,mint,maxt]=premnmx(p,t);
NodeNum1 =4;
NodeNum2=4;
NodeNum3=4;
NodeNum4=4;
NodeNum5=4;
TypeNum = 1;
TF1 = 'tansig';
TF2 = 'tansig';
TF3 = 'tansig';
TF4 = 'tansig';
TF5 = 'tansig';
TF6 = 'tansig';
net=newff(minmax(pn),[NodeNum1,NodeNum2,NodeNum3,NodeNum4,NodeNum5,TypeNum],{TF1 TF2 TF3 TF4 TF5 TF6},'traingdx');
%traingdm
net.trainParam.show=50;
net.trainParam.epochs=50000;
net.trainParam.goal=1e-5;
net.trainParam.lr=0.01;
net=train(net,pn,tn);
p2n=tramnmx(ptest,minp,maxp);
an=sim(net,p2n);
[a]=postmnmx(an,mint,maxt)
plot(1:length(t),t,'o',1:length(t)+1,a,'+');
title('o:predictive_value--- *:actual_value')
grid on
m=length(a);
t1=[t,a(m)];
error=t1-a;
figure
plot(1:length(error),error,'-.')
title('error_graph')
grid on
```

REFERENCES

[1] Lee W C, Cheong T S, Wu Y. The Impacts of Financial Development, Urbanization, and Globalization on Income Inequality: A Regression-Based Decomposition Approach [J]. SSRN Electronic Journal, 2017.

[2] Tan Paul. House prices have been stagnant [J]. Journalist observation, 2019 (4).

[3] Gogtay N J, Deshpande S P, Thatte U M. Principles of Regression Analysis [J]. The Journal of the Association of Physicians of India, 2017, 65(4):48-52.

[4] Gooch J W. Stepwise Regression [J]. Encyclopedic Dictionary of Polymers, 2011.

[5] Bleiholder J, Naumann F. Data fusion [J]. ACM Computing Surveys, 2008, 41(1):1-41.

[6] Han Zhonggeng. Mathematical model for comprehensive evaluation and prediction of Yangtze River water quality [J]. Journal of Engineering Mathematics (7): 69-79.

[7] Shuntaro Okada, Masayuki Ohzeki, Shinichiro Taguchi. Efficient partition of integer optimization problems with one-hot encoding[J]. Scientific Reports, 2019, 9(1).

[8] Wang Zhaoqing, Lu Xiaoyang. A Macro Element Method for Solving Potential Problems Based on Mean Value Interpolation [J]. Journal on Numerical Methods and Computer Applications (3): 21-29.

[9] Hershey S, Chaudhuri S, Ellis D P W, et al. CNN architectures for large-scale audio classification[C]// 2017.