Prediction of house price based on genetic algorithm to optimize BP neural network

Xiaohan Song[1], Xiaojing Zhang[2], Jingru Chen[3]

[1]Arts College, University of Waterloo, Waterloo, Ontario

[2]College of Economics and Management, Qingdao University of Science and Technology, Qingdao, Shandong, 266061

[3]School of Economics and Management, Nanjing Forestry University, Nanjing, Jiangsu, 210037

1393239448@qq.com

Keywords: Genetic algorithm; BP neural network; housing price; prediction model;

Abstract: With the rapid development of the real estate industry in recent years, the trend of housing prices has become faster and faster, and the factors affecting housing prices are complex. Therefore, studying the influencing factors of housing prices and predicting housing prices are of great significance to the development of the national economy. In this regard, this paper proposes a method that combines genetic algorithm and BP neural network, optimizes the weights and thresholds of BP neural network through genetic algorithm, and establishes a housing price prediction model based on genetic algorithm to optimize BP neural network. The simulation results show that the convergence speed and prediction accuracy of the BP neural network prediction model optimized by genetic algorithm have been greatly improved. With this method, it is possible to accurately predict the changes in housing prices in my country, which has very important reference value.

1. Introduction

For a long time, the real estate industry has become an indispensable part of the development of the national economy with its own advantages and an indisputable growth rate. Housing prices are affected by various factors and have been changing over time. The changes in housing prices are related to the country's policy adjustments and the vital interests of the people. Therefore, how to build a high-precision housing price prediction model has always been a very hot topic.

Ding Yuezhi, Zhang Haiyong, etc. [1] proposed a time series forecasting model. The ARMA-GARCH model is used to study the forecast and fluctuation of Jinan housing price index, and the short-term forecast and volatility analysis of Jinan housing prices are carried out. The AR model is first established, and then the GARCH model is added on the basis of the residual sequence with the ARCH effect, provides a new perspective for studying housing prices. Literature [2] adjusted the mean GM (1,1) model and the metabolic GM(1,1) model on the basis of MATLAB, and the error of the price prediction is small. Zhang Yao[3] proposed the use of neural networks to predict housing prices. In addition, a time series ARIMA model and a BP neural network model are established for the housing price index, and the possibility of predicting housing prices is analyzed from multiple angles. Literature [4] considered the differential impact of time on housing prices in Chongqing, used variable coefficient regression models to fit housing price data, and compared and analyzed the fitting effects of linear regression models, it is concluded that the variable coefficient model has a better fitting effect. Naalla Vineeth[5] and others established a housing price prediction model by using simple linear regression, multiple linear regression and neural networks in machine learning algorithms to help buyers and sellers find the best price for a house. Sun Shanshan[6] uses the relevance vector machine method with the kernel function as the radial basis to analyze the monthly price data. According to experiments, the fitting effect of this method is better than that of the traditional data mining method. Literature [7] analyzed the influencing factors of Turkish housing prices, using artificial neural network model and Hedonic model to predict housing prices,

and found that artificial neural network prediction effect is better than Hedonic model.

Through the above analysis, this article combines the genetic algorithm with the BP neural network, optimizes the threshold and weight of the BP neural network through the genetic algorithm, and assigns the optimal weight and threshold to the BP neural network for prediction, it can effectively avoid the local excellent situation of BP network. The simulation results show that the convergence speed and prediction accuracy of the optimized model have been greatly improved, which has very important reference significance for the prediction of housing prices in my country.

2. Genetic algorithm optimizes BP neural network

2.1 Basic Principles of BP Neural Network

BP neural network is a multi-layer feedforward neural network, which includes an input layer, a hidden layer and an output layer. The hidden layer can be one or more layers. It has strong learning, association and fault tolerance functions, and a high degree of non-uniformity. Linear function mapping function, high prediction accuracy, good generalization ability.

The topological structure of the BP neural network is roughly shown in Figure 1. Its working principle is mainly to non-linearly map the input variable to the output variable through the weight threshold and excitation function between the layers. In the training process, a signal is mainly adopted. The BP algorithm of forward pass and error back propagation. This algorithm is also called "negative gradient correction algorithm" and has been applied to the training of many other neural networks.

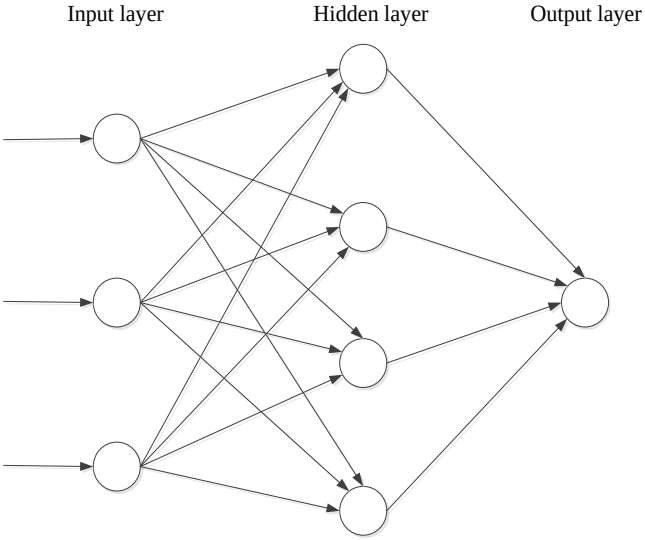

Fig 1 Topological structure of BP neural network

The prediction realization process of BP neural network algorithm is:

(1) Assign random numbers to the hidden layer weight matrix V and the output layer weight matrix M, the sample pattern counter p is set to 1, the training step counter t is set to 0, and the total error E_z is set to 0, The learning rate η is set to a decimal between 0 and 1, to determine the training accuracy ε and the maximum number of training steps.

(2) Input the vector value of the training sample pair X_p and the expected output result d_p, and calculate $Z=f(VX)$ and $y=f(WX)$.

(3) Assuming that there are P pairs of training samples, the cumulative error is calculated as:

$$E_z = \frac{1}{2}\sum_{p=1}^{p}\sum_{j=1}^{m}\left(d_{jp} - y_{jp}\right)^2 \qquad (1)$$

(4) Calculate the error signal of each layer, calculate the hidden layer output error δ_y and the output layer error δ_z.

(5) Adjust the weight of each layer, adjust the weight of the output layer $W(t+1)=W(t)+\eta\delta_y^T Z$, and adjust the weight of the hidden layer $V(t+1)=V(t)+\eta\delta_z^T X$.

(6) Check whether the pairing of all samples in the network is completed $p<P$, and the counters p and t increase by 1.

(7) If the total error is $E_z<\varepsilon$, the training ends, otherwise E_z is set to 0, p is set to 1, return to step (2), and repeat the above process calculation.

Assuming that the number of hidden layer neurons is m, the corresponding weight matrix is $W=(w_{ij})(i=1,2,\cdots,m;\ j=1,2,\cdots,p)$, and the threshold matrix is $B=(b_1,b_2,\cdots,b_m)^T$, then the input of the hidden layer:

$$I_2=W_{m\times p}X'_{p\times n}+Bones_{1\times n} \tag{2}$$

Among them, $ones_{1\times n}$ represents a matrix whose 1×n elements are all ones. The activation function usually used as the hidden layer is a unipolar sigmoid function, that is, the Sigmoid function, and its expression is:

$$f(x)=\left(1+e^{-x}\right)^{-1} \tag{3}$$

Then the output of the hidden layer is $O_2=f(I_2)$, and the input of the output layer is similar to the input of the hidden layer:

$$I_3=X_{jk}O_2+B_{jk}ones_{1\times n} \tag{4}$$

For the output of the third layer, the transfer function is a linear function, so it can be considered as $O_3=I_3$.

The following article uses the chain partial differential law to calculate the adjustment amount of the connection weight threshold between the output layer and the hidden layer and the hidden layer and the input layer. The specific calculation results are as follows:

$$\Delta W_{jk}=-\eta\frac{\partial E}{\partial W_{jk}}=-\eta(Y-O_3)O_2' \tag{5}$$

$$\Delta B_{jk}=-\eta\frac{\partial E}{\partial B_{jk}}=-\eta(Y-O_3)O_2'ones_{n\times 1} \tag{6}$$

Among them, $ones_{n\times 1}$ represents a matrix with n×1 elements all 1s.

From formula (3), we know $f'(x)=f(x)\left[1-f(x)\right]$, after further calculation, the weight threshold adjustment amount of the hidden layer is obtained as:

$$\Delta W_{jk}=-\eta\frac{\partial E}{\partial W_{jk}}=-\eta W'_{jk}(Y-O_3)O_2(1-O_2) \tag{7}$$

$$\Delta B_{jk}=-\eta\frac{\partial E}{\partial B_{jk}}=-\eta W'_{jk}(Y-O_3)O_2(1-O_2)ones_{n\times 1} \tag{8}$$

It can be seen from formulas (5) to (7) and (8) that the weight thresholds obtained from the second to n iterations of the BP algorithm are directly or indirectly related to the first weight threshold when adjusting the weight threshold. Therefore, the selection of the initial weight threshold is particularly critical for the training of the BP network. If it is set improperly, it may cause the network to not only converge slowly but also easily fall into the trap of local optimization during the training process. Therefore, how to optimize the initial weight threshold has always been one of the key issues in BP neural network research.

2.2 Basic principles of genetic algorithm

Genetic algorithm is a parallel random search optimization method that simulates genetic genetics and biological evolution, and it has the ability to find the optimal solution globally. Genetic algorithm mainly includes three basic steps: selection, crossover and mutation.

1) Selection: The selection operation is to select individuals from the population according to a certain probability, as the parent, for breeding offspring. The probability of selection is determined by fitness. The better the fitness, the greater the probability that an individual will be selected, so that excellent individuals can be retained, more excellent individuals will be reproduced, and the expected value will eventually be approached.

2) Crossover: The crossover process is to select two individuals and cross-swap one or more points on the individual's chromosomes to generate a new individual. The process of intersection embodies the idea of information interaction in the natural world.

3) Mutation: select individuals according to a certain probability and mutate a segment of chromosomes in the individual to enhance the fitness of the individual.

2.3 Genetic algorithm optimizes the algorithm flow of BP neural network

Genetic algorithm is characterized by global search, while BP neural network searches for optimal solutions locally. Therefore, the genetic algorithm can be used to determine the optimal solution range of the initial weight and threshold of the BP neural network, and then the BP neural network algorithm can be used to search for the local optimal solution. Since genetic algorithm is a heuristic global search algorithm that does not rely on auxiliary information, it is not easy to fall into the local optimal trap during the search process. This can just make up for the shortcomings of the BP network, so this article uses the genetic algorithm. A feature to optimize the setting of the initial weight threshold of the BP network. The process of genetic algorithm optimizing BP neural network is: chromosome expression is coding, individual fitness solving, genetic operation (mainly including selection, crossover, mutation).

The specific implementation process of genetic algorithm optimization of BP neural network mainly includes the following steps:

Step 1: Initialization of coding and population. First determine the topological structure of the BP neural network, and determine the length of the individual according to the network structure. All weights and thresholds in the network are encoded in real numbers (binary and decimal encoding methods can also be used) as a set of chromosomes, namely:

$$X = [\omega_{11}, \omega_{12}, \cdots, \omega_{mm}, \theta_1, \theta_2, \cdots \theta_m, \nu_{11}, \nu_{12}, \cdots \nu_{mn}, t_1, t_2, \cdots t_n] \tag{9}$$

Where: ω_{mm} is the weight between the input layer and the hidden layer; θ_m is the connection threshold between the hidden layers; ν_{mn} is the weight between the hidden layer and the output layer; t_n is the output layer threshold.

Step 2: Fitness evaluation. After the BP neural network determines the initial weight and threshold, it uses the training set to train the BP neural network and predicts the output of the system. Since the genetic algorithm is evolving in the direction of fitness reduction, the weight and threshold value that minimize the absolute value of the network error are found within the search area. Here, the sum of the absolute value of the predicted output and the expected output is used as the individual fitness value. which is:

$$F(X_i) = k\left(\sum_{i=1}^{n} |y_i - o_i|\right) \tag{10}$$

Where: $F(X_i)$ is the individual fitness value; k is the coefficient; $k = \frac{1}{2l}$, n is the number of neurons in the network; y_i and o_i are the predicted output and expected output of the system, respectively.

Step 3: Select operation. Calculate the individual fitness value from equation (10). The smaller the fitness, the better the individual, and the greater the probability of being selected. When calculating the probability of selection, the reciprocal is generally taken. The selection methods mainly include roulette method and tournament method. Here, roulette method is selected, that is, the strategy is selected based on the fitness ratio. The probability Pi of each gene being selected is expressed as:

$$f\left(X_i\right)=\frac{1}{F\left(X_i\right)} \tag{11}$$

$$p_i=\frac{f\left(X_i\right)}{\sum_{i=1}^{N} f\left(X_i\right)} \tag{12}$$

Where N is the number of populations.

Step 4: Crossover operation: The purpose of crossover operation is to improve individual coding structure from a global perspective by using crossover operators. Select two genes X_k and X_i by formula (5), and perform crossover operations on the j-th position on their chromosomes respectively, as follows:

$$\begin{cases} X_{kj}=X_{kj}\left(1+b\right)-X_{ij}b \\ X_{ij}=X_{ij}\left(1+b\right)-X_{kj}b \end{cases} \tag{13}$$

In the formula, b is a constant, and the value range is [0, 1].

Step 5: Mutation operation. It improves the local search ability of the algorithm and maintains the diversity of the population. It selects a gene point from the parent and replaces it with a uniformly distributed random number, so that the replaced individual is more suitable for the current system environment. The new gene point is expressed as:

$$X_{ij}=\begin{cases} X_{ij}+(X_{ij}-X_{\max})\times f(g) & r\geq 0.5 \\ X_{ij}+(X_{\min}-X_{ij})\times f(g) & r<0.5 \end{cases} \tag{14}$$

$$f(g)=r_2(1-g/G_{\max})^2 \tag{15}$$

Where: $X_{\max}$ and $X_{\min}$ are the upper and lower bounds of the initial individual genes, respectively; r and r_2 are random numbers between [0, 1]; $G_{\max}$ are the maximum number of evolutions; g is the number of current iterations.

Step 6: Bring the optimized weight threshold to the BP network for training, and then use it for prediction.

In this paper, genetic algorithm is used to optimize the BP neural network to obtain the optimal threshold and weight, and assign the optimal weight and threshold to the BP neural network for prediction, which can effectively avoid the local optimal situation of the BP network. Improve the prediction accuracy and achieve the purpose of optimization. The specific process of genetic algorithm optimizing BP neural network is shown in Figure 2 below.

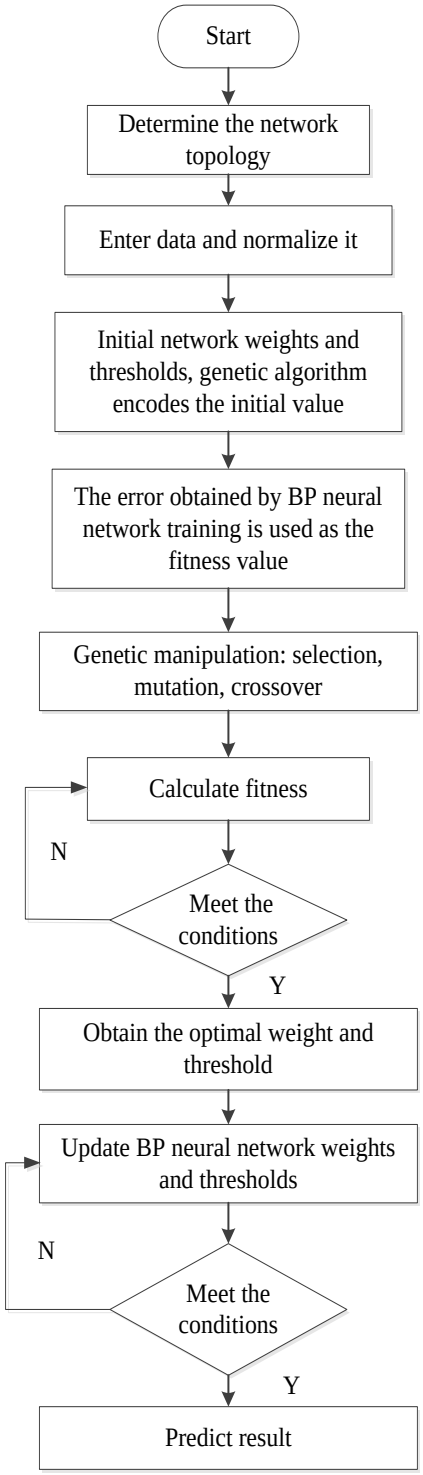

Fig 2 Flow chart of genetic algorithm optimization of BP neural network

3. Establishment of housing price prediction model based on genetic algorithm optimization by BP neural network

3.1 Establishment of housing price prediction model

In order to truly reflect the factors that affect housing prices, this article selects the average sales price of commercial housing, housing, resident consumption level, average salary of urban employees, disposable income of urban residents, GDP, and the number of urban employees through analysis. Seven main influencing factors are used as the input of this model, and the housing price is selected as the output of the model. The data in this article comes from a total of 11 sets of data from 2005 to 2015 in the "2016 China Statistical Yearbook".

In this paper, the model parameters of the BP neural network are set as: the maximum number of training times of the network is 200, the target training error is 10^{-4}, and the learning rate is 0.02. The specific settings of genetic parameters are: the population size is 50, the maximum number of iterations is 500, the selection probability is 0.9, the crossover probability is 0.5, and the mutation probability is 0.5.

3.2 Simulation results

This article selects a total of 6 sets of data from 2005 to 2010 to train the prediction model of this article, and uses a total of 5 sets of data from 2011 to 2015 to predict housing prices through the trained model. BP neural network and genetic algorithm optimize BP neural network to predict the value of housing price as shown in Table 1, the prediction curve is shown in Figure 3.

Table 1 The value of housing price predicted by BP neural networks

years	Actual value /100 million yuan	GA-BP	error/ %	BP	error/ %
2011	34.81	35.9	3.1	36.3	4.3
2012	40.1	41.1	2.5	44.5	10.9
2013	48.66	47.9	1.6	45.6	6.3
2014	54.38	53.8	1.1	50.5	7.1
2015	55.17	57.2	3.7	59.9	8.6

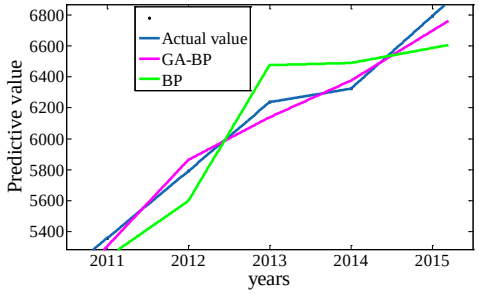

Fig 3 House price prediction result graph

According to the prediction results, it can be clearly seen from Table 1 that the highest error of the BP neural network prediction through genetic algorithm optimization is 1.6%, the lowest is 0.8%, and the average error is 1.22%. The highest error of BP neural network prediction is 3.8%, the lowest is 2.3%, and the average error is 23%. The genetic algorithm is used to find the better weights and thresholds of the BP neural network, and the prediction accuracy of the optimized BP neural network is higher than that of the single BP neural network prediction model.

4. Conclusion

In order to overcome the shortcomings of BP neural network, this paper proposes a prediction model based on genetic algorithm to optimize the weights and thresholds of BP neural network. Based on China's 2005-2015 housing prices and their main influencing factors data, a housing price prediction model based on genetic algorithm optimized BP neural network is established. The main conclusions are as follows:

1) This paper establishes a genetic algorithm to optimize the prediction model of BP neural network, which is suitable for the prediction of China's housing prices and has high prediction accuracy, which is of great significance to the development of the national economy and the vital interests of the people.

2) This paper uses genetic algorithm optimized BP neural network and a single BP neural network to predict housing prices. The results show that compared with a single BP neural network, its convergence speed and prediction accuracy are greatly improved.

References

[1] Ding Yuezhi. Housing price prediction and volatility analysis based on time series model [D]. Jinan: Shandong University, 2018.

[2] Wang Sihua, Zhao Zhiman, Li Guoliang. Prediction of house prices in Haikou City based on the GM (1,1) model of metabolism[J].China Water Transport (second half of the month), 2019,19(01):68-69.

[3] Zhang Yao. Analysis and prediction of my country's housing prices based on neural networks [D]. Suzhou: Soochow University, 2018.

[4] Liu Feng, Zhang Xing, Zhang Guangfeng. Modeling and analysis of variable coefficient regression model of housing prices in Chongqing [J]. Journal of Chongqing University of Technology: Natural Science Edition, 2014, 28(4): 150 -154.

[5] Vineeth N, Ayyappa M, Bharathi B. House Price Prediction Using Machine Learning Algorithms: Second International Conference, ICSCS 2018, Kollam, India, April 19–20, 2018, Revised Selected Papers[M]// Soft Computing Systems. 2018,425-433.

[6] Sun Shanshan. Real estate price prediction based on data mining [J]. Modern Electronic Technology, 2017(05):134-137.

[7] Selim H. Determinants of house prices in Turkey: Hedonic regression versus artificial neural network [J]. Expert Systems with Applications, 2009, 36(2):2843-2852.