ISSN 2616-5775 Vol. 7, Issue 8: 15-20, DOI: 10.25236/AJCIS.2024.070803

GA-BP-based Nonlinear Time Series Forecasting: Method and Applications

Jiakang Ma[1,*], Weipeng Xu[2]

[1]School of Traffic & Transportation Engineering, Central South University, Changsha, 410083, China
[2]School of Economics and Management, Tiangong University, Tianjin, 300387, China
[*]Corresponding author: maske2483@gmail.com

Abstract: *Accurate time series forecasting is crucial for decision-making in fields like financial analysis, weather prediction, energy consumption, and intelligent transportation. Traditional models often struggle with non-linear and complex data. This study aims to improve forecasting accuracy by integrating Genetic Algorithm (GA) with Back Propagation (BP) neural networks. BP neural networks are powerful but face issues like local minima entrapment and network structure selection difficulties. GA is employed to optimize the weights and thresholds of BP networks, enhancing their generalization and robustness. Testing on a non-linear, non-periodic time series dataset, the GA-BP model showed significant improvements in prediction accuracy and stability. Performance metrics such as RMSE,* R^2*, MAE, and MBE confirm the model's effectiveness. This hybrid approach not only boosts forecasting accuracy but also introduces innovative methods for addressing complex time series challenges, making a valuable contribution to the field.*

Keywords: *Back Propagation Neural Network, Genetic Algorithm, Prediction Model*

1. Introduction

With the rapid development of globalization and information technology, time series forecasting has become an increasingly important in various fields[1]. Whether it is for financial market analysis, weather forecasting, energy consumption prediction, or intelligent transportation systems, accurate time series forecasting is crucial for decision-making and resource management. Traditional forecasting model, such as statistical model, often rely on strong assumptions and are limited in their effectiveness when dealing with non-linear and complex time series data. Therefore, researchers have been continuously seeking more effective and flexible forecasting methods.

In recent years, the rapid development of artificial intelligence and machine learning has provided new avenues for time series forecasting. Among these,the Back Propagation(BP) neural network has become an important tool in the field of prediction due to its powerful non-linear mapping capabilities[2].However, traditional BP neural network have some inherent problems, such as the tendency to get trapped in local minima and the difficulty in selecting network structures, which limit their predictive performance in practical applications.

To overcome these limitations, researchers have started to explore the combination of Genetic Algorithm (GA) and BP neural networks[3]. As an efficient global optimization algorithm, Genetic Algorithms can search for the optimal solution across the entire search space, thereby avoiding getting trapped in local minima. By using Genetic Algorithms to optimize the weights and thresholds of BP neural networks, the generalization ability of the network can be improved, and the robustness of the prediction model can be enhanced[4]. This approach not only improves the accuracy of predictions but also brings new ideas and technical approaches to the field of time series forecasting.

The model combines GA with BP neural networks, using the GA to optimize the weights of the BP neural network to enhance the accuracy and stability of its predictions. It applies the optimized model to the forecasting of time series data during model testing, resulting in a BP neural network prediction model based on Genetic Algorithms that improves the performance of time series forecasting[5].

ISSN 2616-5775 Vol. 7, Issue 8: 15-20, DOI: 10.25236/AJCIS.2024.070803

2. Model establishment

2.1 Overview of the BP neural network

BP (Back Propagation) neural networks are sophisticated multi-layer feedforward neural networks with exceptional information processing capabilities. They can learn to predict outcomes without needing an explicit mathematical function to map inputs to outputs. This ability makes them particularly useful for handling complex decision-making tasks in uncertain environments. BP networks are extensively employed in predictive modeling across various fields.The network structure is shown in Figure 1.

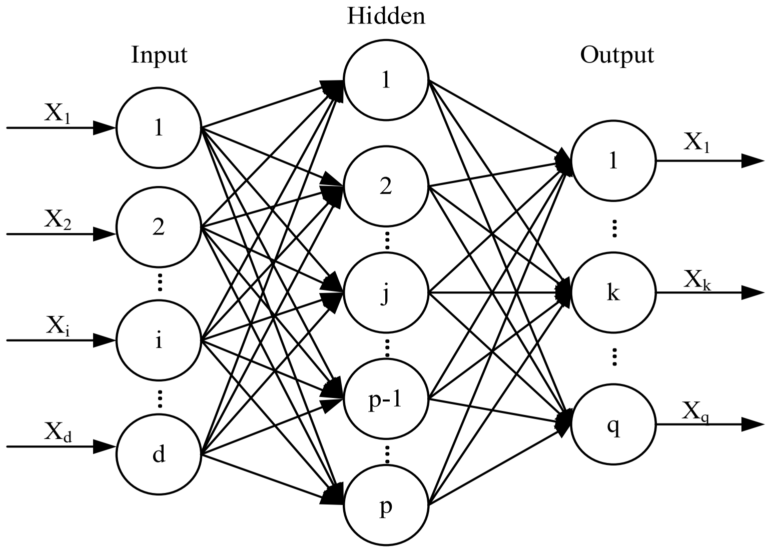


Figure 1: Neural Network structure

The advantage of the BP neural network lies in its learning ability. It can capture the complex non-linear relationship between input and output, and has the ability to generalize. This makes it widely used in fields such as function approximation, pattern recognition, classification, and prediction.

However, the BP neural network also has some limitations. Firstly, it may fall into a local minimum rather than a global minimum. Secondly, for large-scale networks and datasets, the training time may be long. In addition, the network may overfit the training data, leading to a decrease in the generalization ability for new data. Finally, the selection of the number of hidden layer nodes and layers lacks theoretical guidance and usually needs to be determined through experiments.

2.2 GA Parameter Optimization for BP Neural Network

To overcome the performance limitations and the issue of local optima inherent in traditional BP neural networks when using randomly generated initial weights and thresholds, this paper employs Genetic Algorithm (GA) for optimization. Genetic Algorithm, by simulating natural selection and genetic processes, demonstrates exceptional global search capabilities, effectively reducing the risk of converging to local optimal solutions. Optimizing the weights and thresholds of the BP neural network with GA can prevent prediction biases resulting from randomly assigned parameters. The specific implementation steps of the improved GA-BP neural network time series prediction model are depicted in Figure 2.

ISSN 2616-5775 Vol. 7, Issue 8: 15-20, DOI: 10.25236/AJCIS.2024.070803

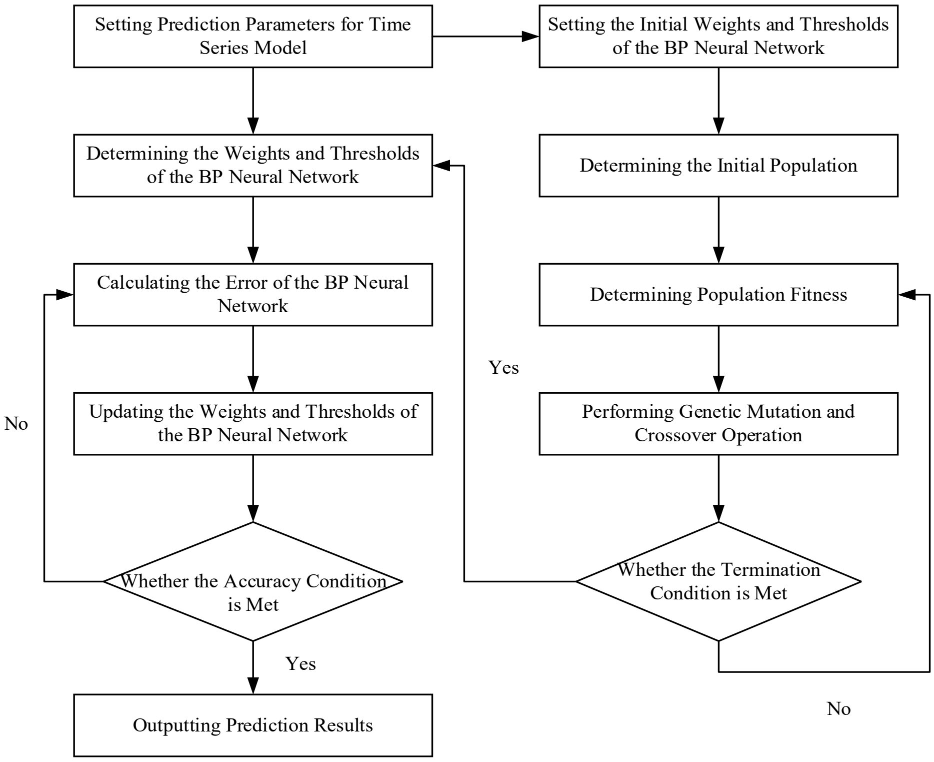


Figure 2: Algorithm Optimized BP Neural Network Flowchart

Addressing the issues of unstable learning speeds and slow convergence rates when dealing with large training sets in the BP neural network, we first harness the search capabilities of the genetic algorithm to optimize the initial weights and thresholds of the BP neural network, thereby enhancing network performance. Subsequently, we utilize training data to train the established traffic flow prediction model, aiming to achieve the optimal input-output mapping relationship. The algorithm flow for time series prediction based on GA-BP neural network is as follows:

Step 1: Define the chromosome by encoding the population using a real number encoding method. For the BP neural network model designed for time series prediction, compile all the weights and thresholds of the network into an N-dimensional vector, which serves as the chromosome for an individual. Determine the population size M and randomly generate an M×N matrix as the initial population. The calculation for N is as shown in Eq.(1):

$$N = n_{input} \times n_{hidden} + n_{hidden} + n_{hidden} \times n_{output} + n_{output}, \tag{1}$$

wheren_{input} represents the number of nodes in the input layer,n_{hidden} represents the number of nodes in the hidden layer,n_{output} represents the number of nodes in the output layer.

In our model, there are 15 nodes in the input layer, 5 nodes in the hidden layer, and 1 node in the output layer.

Step 2: Genetic algorithms assess the superiority or inferiority of individuals based on the value of the fitness function. By incorporating a selected fitness function, individuals are screened, selecting those with good fitness and eliminating those with poor fitness. Here, the genes of each individual are categorized, and their numerical values are used as weights and thresholds in the BP neural network model for traffic flow prediction. The formula for calculating the fitness value (val) is as shown in Eq.(2):

$$val = \frac{1}{RMSE} \tag{2}$$

Where RMSE represents the Root Mean Square Error.The formula for calculating RMSE is as shown in Eq.(3):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_{simi} - t_{traini})} \tag{3}$$

Where *N* represents the number of training samples,t_{simi} represents the simulated output of the neural network for the i^{th} training sample,t_{traini} is the actual output for the i^{th} training sample.

Step 3: The selection operation is based on the individual's fitness and uses the roulette wheel method to select superior individuals to be inherited into the next population. The main idea of the roulette wheel algorithm is that the selection probability of each individual is proportional to the size of its fitness function value. When the population size is M, the selection probability p_i for each individual is shown in Eq.(4) :

$$p_i = \frac{val_i}{\sum_1^M val_i} \tag{4}$$

Where p_i represents the selection probability of the i^{th} individual,val_i is the fitness value of the i^{th} individual.

Step 4: The critical aspect of genetic algorithms lies in the crossover step, which is the process of generating new offspring individuals that combine the features of the parent individuals, potentially resulting in excellent offspring. This article adopts the arithmetic crossover method, with the formula is shown in Eq.(5) and Eq.(6):

$$c_{1,i} = \lambda x_{1,i} + (1-\lambda)x_{2,i} \tag{5}$$

$$c_{2,i} = \lambda x_{2,i} + (1-\lambda)x_{1,i} \tag{6}$$

Where $c_{1,i}$ and $c_{2,i}$ are the i^{th} gene of two new individuals respectively, and $x_{1,i}$ and $x_{2,i}$ are the i^{th} gene of two parent individuals respectively.

Step 5: Perform the mutation operation, which involves randomly selecting an individual from the population and altering its genes to generate a wider array of combinations for the weights and thresholds of the time series prediction model. This process is aimed at maintaining the diversity of the population and enhancing the algorithm's search capabilities.

Step 6: Utilize the optimal individual obtained through genetic algorithm optimization as the ideal weights and thresholds for the neural network traffic flow prediction model established in this study, replacing the initial values set by the BP neural network. This results in a time series prediction model optimized by the genetic algorithm. Subsequently, the BP network undergoes a new round of training, enabling the acquisition of more precise time series data.

3. Experimentation and data analysis

3.1 Data preprocessing

The data for this paper is sourced from an open-source dataset on GitHub. The time series dataset exhibits clear nonlinear and non-periodic characteristics, with a total of 937 data points.

In order to enhance the generalization capability and accuracy of the model, the data was first segmented, namely dividing the data into a test set and a training set. Subsequently, the data was normalized, scaling all the data to fall within the range of [0,1]. The normalization formula is shown in Eq.(7):

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{7}$$

Where x_i represents an element from the original dataset, x_i' is the normalized value, and $min(x)$ and $max(x)$ are the minimum and maximum values of the data set respectively.

3.2 Algorithm simulation test

In this experiment, we conducted model simulation using Matlab to construct a three-layer BP neural network. The input layer consists of 15 nodes for receiving historical data; the hidden layer has 5 neurons capable of learning the non-linear relationships in the data; and the output layer has 1 node for predicting the next data point. During the training process with the training set, the maximum number of iterations is set to 1000, the learning rate to 0.01, and the error threshold to 10^{-6} to ensure that the training stops when the predetermined accuracy or the iteration limit is reached. After training, we tested the model using the test set.

ISSN 2616-5775 Vol. 7, Issue 8: 15-20, DOI: 10.25236/AJCIS.2024.070803

3.3 Performance evaluation

This paper employs root mean square error ($RMSE$), correlation index (R^2), mean absolute error (MAE), and mean bias error (MBE) as evaluation metrics for the algorithm, with the formulas are shown in Eq.(3), Eq.(8) ,Eq.(9) and Eq.(10).

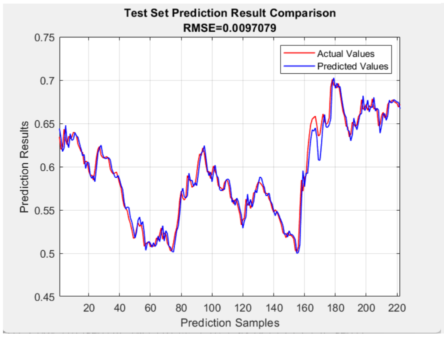
$$R^2 = 1 - \frac{\sum_{i=1}^{N}(t_{traini} - t_{simi})^2}{\sum_{i=1}^{N}(t_{traini} - \bar{t}_{traini})^2} \tag{8}$$

$$MAE = \frac{\sum_{i=1}^{N}|t_{traini} - t_{simi}|}{N} \tag{9}$$

$$MBE = \frac{\sum_{i=1}^{N}(t_{simi} - t_{traini})}{N} \tag{10}$$

Where $\bar{t}_{traini}$ represents the average of the training set.The formulas in this paper are all based on the training set, and the formulas for the test set can be obtained by simply changing 't_{traini}' to 't_{testi}' and 't_{simi}' to 't'_{simi}' .Where t'_{simi} represents the simulated output of the neural network for the i^{th} training sample,t_{testi} is the actual output for the i^{th} training sample.

This study validated the GA-BP neural network model on an open-source time series dataset and evaluated its predictive capabilities using multiple performance metrics. Figure 3 and Figure 4 show the prediction results comparison for the test set and training set, respectively. Figure 5 and Figure 6 illustrate the comparison between predicted and actual values for the test set and training set.



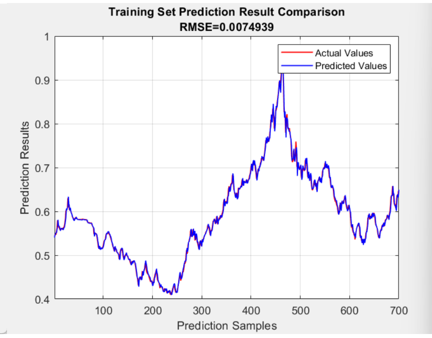
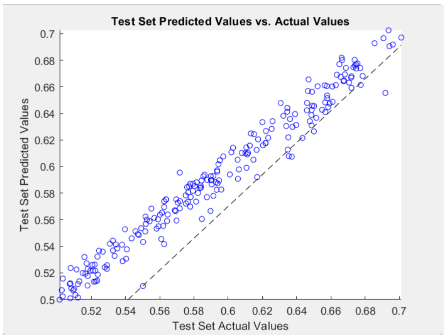


Figure 3: Test Set Prediction Result Comparison (Left)

Figure 4: Training Set Prediction Result Comparison (Right)



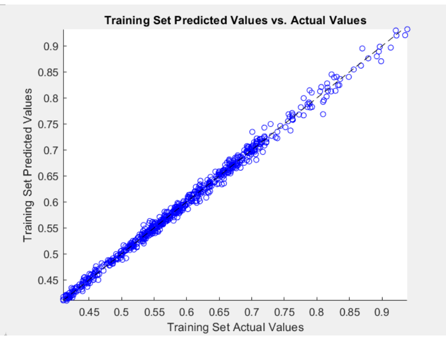


Figure 5: Test Set Predicted Values vs. Actual Values (Left)

Figure 6: Training Set Predicted Values vs. Actual Values (Right)

The evaluation metrics for the training set and test set include Root Mean Square Error *(RMSE)*, Correlation Coefficient *(R^2)*, Mean Absolute Error *(MAE)*, and *Mean Bias Error (MBE)*. The specific data are as follows, as shown in Table 1.

Table 1: Evaluation Metrics for Training set and Test Set

Dataset	*RMSE*	*R^2*	*MAE*	*MBE*
Training set	0.007513	0.99481	0.005482	0.00016827
Test set	0.010011	0.96575	0.0076372	-0.00018498

The experimental findings indicate that the GA-BP performs excellently in time series forecasting.

ISSN 2616-5775 Vol. 7, Issue 8: 15-20, DOI: 10.25236/AJCIS.2024.070803

The *RMSE* and *MAE* values for both the training and test sets are low, and the R^2 values are close to 1, demonstrating high prediction accuracy and stability. Additionally, the *MAE* and *MBE* values are near zero, indicating no significant systematic bias in the model.

These results confirm that the GA-BP neural network effectively handles non-linear and complex time series data. Unlike traditional BP neural networks, the GA-BP model significantly enhances prediction accuracy, stability, and generalization. This innovative approach offers valuable techniques for time series forecasting, making it crucial for decision-making and resource management in areas like financial market analysis, weather forecasting, energy consumption prediction, and intelligent transportation systems.

4. Conclusions

In this study, we have demonstrated the effectiveness of combining Genetic Algorithm (GA) with Back Propagation (BP) neural networks for time series forecasting. The GA-BP model addresses the limitations of traditional BP neural networks by optimizing their weights and thresholds, thus enhancing their prediction accuracy and stability.

The experimental results show that the GA-BP model performs exceptionally well in forecasting non-linear and complex time series data. Key performance metrics, including *RMSE*, R^2, *MAE*, and *MBE*, indicate significant improvements in both training and test sets. The model's ability to minimize systematic bias further underscores its robustness and reliability.

This research highlights the potential of GA-BP model as a powerful tool for time series forecasting. The innovative approach not only advances the field of predictive modeling but also offers practical benefits for decision-making and resource management in various domains, including financial market analysis, weather forecasting, energy consumption prediction, and intelligent transportation systems.

Future work could explore the application of this hybrid model to other types of forecasting problems and investigate further enhancements to the optimization process to achieve even greater accuracy and efficiency.

References

[1] Bawdekar A A, Prusty B R, Bingi K. Sensitivity analysis of stationarity tests' outcome to time Series facets and test parameters[J]. Mathematical Problems in Engineering, 2022, 2022.

[2] Hart A G, Hook J L, Dawes J H P. Echo state networks trained by Tikhonov least squares are L2 (μ) approximators of ergodic dynamical systems[J]. Physica D: Nonlinear Phenomena, 2021, 421: 132882.

[3] Duan Y, Liu M, Dong M. A metric-learning-based nonlinear modeling algorithm and its application in key-performance-indicator prediction[J]. IEEE Transactions on Industrial Electronics, 2019, 67(8): 7073-7082.

[4] Yang J, Hu Y, Zhang K, et al. An improved evolution algorithm using population competition genetic algorithm and self-correction BP neural network based on fitness landscape[J]. Soft Computing, 2021, 25: 1751-1776.

[5] Lu Y, Li T, Deng Y. Predication of Water Pollution Peak Concentrations by Hybrid BP Artificial Neural Network Coupled with Genetic Algorithm[J]. Applied Artificial Intelligence, 2024, 38(1): 2341356.