Academic Journal of Computing & Information Science
ISSN 2616-5775 Vol. 8, Issue 4: 92-97, DOI: 10.25236/AJCIS.2025.080411

Genetic Algorithm Optimization of BP Neural Network

Huaitian Zhang*

School of Mathematics and Statistics, Lanzhou University, Lanzhou, China, 730000
**Corresponding author: 183727869@qq.com*

Abstract: *In the context of the rapid development of artificial intelligence, optimizing the BP neural network model has become a key task to improve prediction performance. This paper studies the application of genetic algorithm in BP neural network optimization, which significantly improves the convergence speed and prediction accuracy of the model. On the test data, compared with the Standard BP Neural Network, the fitting effect and regression curve are significantly improved. The GA-BP model achieves a higher R-squared value (0.96804 vs. 0.84895), which verifies the effectiveness of this method. This study not only solves the limitations of Standard BP Neural Network such as slow convergence speed and easy to fall into local optimality, but also provides a practical solution for the optimization of neural networks in the context of big data and complex problem solving, which is of great significance to improving the performance and scalability of BP neural networks in practical applications, particularly in data-intensive applications.*

Keywords: *Neural Network, Prediction Model, Big Data*

1. Introduction

With the swift advancement of computer technology and the increasing pace of globalization, the volume and complexity of data across various industries are constantly on the rise. As a robust machine learning model, the back propagation (BP) neural network is capable of emulating the learning process of the human brain. It can progressively approximate the target function through training with a vast amount of data, thereby achieving nonlinear mapping. Particularly when confronted with complex nonlinear problems, traditional algorithms often fall short, whereas the BP neural network can effectively tackle these issues with its multi-layer architecture and back propagation training mechanism. This capability also enables it to demonstrate outstanding performance in addressing a new generation of complex problems, especially in the approximation of nonlinear mappings.

Since its inception, the BP neural network has garnered extensive attention and application in both academia and industry. Its fundamental concept involves utilizing the error back propagation algorithm to adjust the network weights, thereby enabling the nonlinear mapping of input data. The introduction of this algorithm has provided an effective solution for training multi-layer neural networks, allowing neural networks to handle more intricate pattern recognition and function approximation tasks[1].

With the widespread application of BP neural network, researchers began to pay attention to its own limitations and proposed various improvement methods. It has shown excellent performance in computer vision[2], photovoltaic power generation[3], transportation[4], finance[5-6], traffic management[7] and other fields. At the same time, some researchers have made various improvements and applied them to other aspects, such as combining BP neural network with deep belief network and applying it to environmental pollution prediction and haze prediction, and combining it with ant colony algorithm to predict object trajectory, and combining it with simulated annealing algorithm to predict network traffic.

Despite the notable achievements of the BP neural network in various domains, it still faces certain limitations, such as its tendency to fall into local optimal solutions and its sensitivity to initial weights. Given these challenges, this paper aims to explore and utilize the genetic algorithm to optimize the BP neural network. The study will further investigate the potential and impact of this approach in enhancing the training efficiency and optimizing the performance of the BP neural network, with the expectation of achieving superior results in addressing specific problems.

ISSN 2616-5775 Vol. 8, Issue 4: 92-97, DOI: 10.25236/AJCIS.2025.080411

2. Brief Introduction of Neural Network

2.1 The structure of BP neural network

The BP neural network, a type of multi-layer feedforward neural network, employs the error back propagation algorithm to learn and update its weights. It comprises an input layer, hidden layers, and an output layer, with neurons in each layer interconnected by weights. These layers are connected via distinct node coefficients. When the connection weights are positive, it signifies that the current link is in an excitatory state. Conversely, a negative link coefficient indicates a suppressive state.

During training, the BP neural network initially feeds data through each layer via forward propagation, computes the output result, and then compares this output with the desired outcome to calculate the error. Subsequently, the error is propagated backward from the output layer to the input layer, layer by layer, using the back propagation mechanism. The network then employs the gradient descent method to adjust the weights and thresholds of each connection, thereby progressively minimizing the error.

2.2 Traditional optimization methods and their limitations

The traditional optimization approach for BP neural networks primarily depends on the gradient descent method, which updates the weights by computing the gradient of the network output error. Given that the gradient descent method relies on local error information, it is prone to becoming trapped in local optima and failing to achieve global optimality. Additionally, when dealing with large-scale networks or complex data, the gradient descent method may necessitate a substantial number of iterations to converge, rendering it inefficient. Moreover, its performance is highly sensitive to the selection of initial weights. Poorly chosen initial weights can lead to slow convergence or even failure to converge. When the error landscape is more complex, gradient descent may also exhibit issues related to the oscillation of the learning rate, resulting in an unstable convergence process and difficulties in addressing complex problems.

3. Brief description of GA-BP, process and performance analysis

3.1 The basic principle of genetic algorithm

The Genetic Algorithm (GA) is a global optimization technique that mimics the process of natural evolution and is extensively applied to tackle complex optimization problems. Its fundamental principle is to approximate the optimal solution by emulating the biological processes of selection, crossover, and mutation. Unlike local search methods, GAs possess robust global search capabilities, enabling them to navigate vast solution spaces and avoid local optima.

In GAs, an initial population is generated by randomly creating a set of potential solutions. Each solution, referred to as an "individual," is evaluated using a fitness function to determine its quality. During each generation, the algorithm selects high-performing individuals based on their fitness and generates new candidates through crossover operations. These new individuals may then undergo mutation operations, introducing minor random changes to enhance population diversity and prevent local optima.

Over successive generations, the GA progressively refines the population, steering it toward solutions that more closely approximate the global optimum. This population-based evolutionary approach grants the GA a higher likelihood of discovering the global optimal solution, even when the initial solutions are suboptimal. As the population evolves, superior solutions are continuously accumulated and propagated, while inferior ones are gradually phased out. This dynamic allows the algorithm to explore a broader search space and converge toward the optimal or near-optimal solution.

3.2 Combination of Genetic Algorithm and BP Neural Network

3.2.1 Combination process

Initially, the weights and thresholds of the BP neural network are transformed into individual genes. All the weights and thresholds of the network are expanded into a vector and transformed into individuals that are utilized in the genetic algorithm. Subsequently, the fitness function is determined, and individuals with high fitness are selected to proceed to the next generation. Based on their fitness values, individuals with high fitness are chosen to advance to the next generation.

ISSN 2616-5775 Vol. 8, Issue 4: 92-97, DOI: 10.25236/AJCIS.2025.080411

Next, a crossover operation is performed to simulate the biological reproduction process. This operation combines the genes (weights and thresholds) of two individuals to produce a new individual. Through crossover, the new individual inherits some characteristics from the parent individuals.

To steer clear of local optima, a mutation operation is introduced. This operation randomly alters specific bits in the individual's genes (i.e., randomly fine-tunes the weights or thresholds) to generate a new solution. Given that the mutation probability is set at a low level, most solutions are derived from selection and crossover processes.

Through selection, crossover, and mutation operations, a new generation of the population is generated. After evaluating the fitness of the new population, if it does not meet the requirements, it replaces the previous generation and continues iterative evolution.

When the maximum number of iterations is reached or the fitness converges, the optimization process is terminated. The optimal weights obtained are then incorporated into the BP neural network to recalculate the error, update the biases and weights, and output the result. If the conditions are not met, the optimization process of the BP neural network continues.

3.2.2 Operation process

In order to address the issue of the BP neural network's tendency to fall into local optima, this paper integrates the genetic algorithm with the weight selection process of the BP neural network, leveraging the genetic algorithm to optimize the calculation of weights and thresholds in the BP neural network. The process of the GA-BP model is depicted in Figure 1.

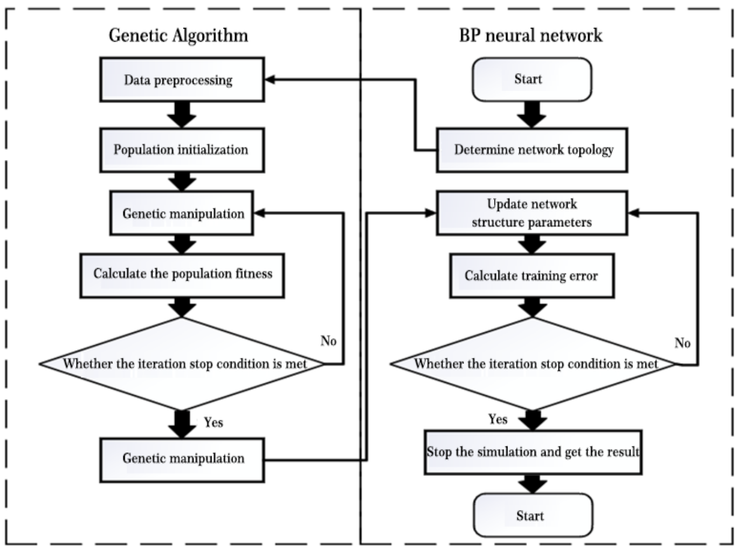

Figure 1 Operation process diagram

3.2.3 Prediction Convergence Analysis

Through the combination of genetic algorithm global search and BP neural network local fine-tuning, the model should have good convergence. When the error value is close to convergence, the difference between the model training and test errors should be small, and it has strong generalization ability. The global search capability of the genetic algorithm effectively improves the network weight, allowing the model to quickly reduce the prediction error. The model should have a faster convergence speed. The BP network finds a better initial solution through GA, so that the error function can more effectively descend along the gradient direction, avoiding excessive local oscillations during the iteration process. The model should have a smoother error curve. The optimized model will show higher stability in the convergence stage, and the prediction of data will be more accurate, which is suitable for complex prediction tasks.

ISSN 2616-5775 Vol. 8, Issue 4: 92-97, DOI: 10.25236/AJCIS.2025.080411

4. Experimental design and results analysis

4.1 Data selection

Data source: from the test data set on the github, which is used to verify the nonlinear regression problem.

Input features: 7 ,Output features: 1.

4.2 Introduction to evaluation indicators

Error calculation: Calculate the root mean square error RMSE and R, the formula is as follows:

$$\mathrm{R}MSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(y_i - \widehat{y_i})} \quad (1)$$

$$R = \sqrt{1 - \frac{\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{N}(y_i - \overline{y_i})^2}} \quad (2)$$

4.3 Data preprocessing and training

We preprocessed the data before the experiment, selected the first 80 samples as the training set and the last 23 samples as the test set, and normalized the data, scaling the range to [0, 1] to improve the convergence speed and enhance stability. After that, we created a feedforward neural network with 7 input layer neurons, 5 hidden layer neurons, and 1 output layer neuron, and trained it according to the gradient descent algorithm before the set maximum number of iterations or error threshold.

Upon finishing the training, we simulated and tested the model with the training set and the test set. We subsequently analyzed and evaluated the results. To evaluate the prediction accuracy of the model, we calculated the RMSE, R-squared, MAE, and MBE for both the training and test sets.

4.4 Experimental results and analysis and evaluation

The fitting effect diagram of pure BP neural network is shown in Figure 2.

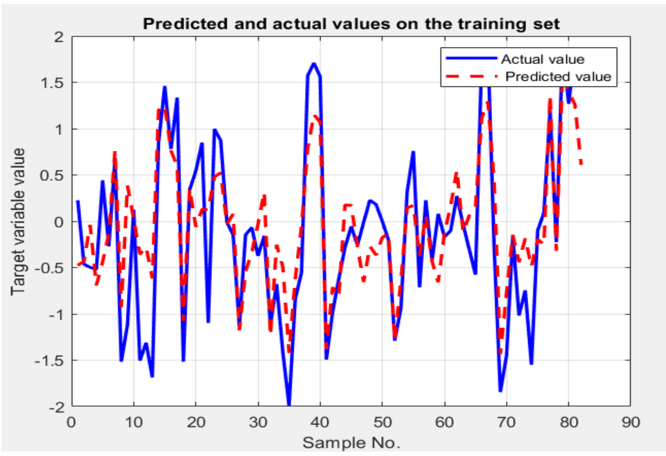

Figure 2 The fitting effect diagram of pure BP neural network

There is an obvious deviation between the actual value and the predicted value. The gap is large at some sample points. There is a certain error between the predicted result of the model and the true value. The fitting effect is not very ideal. This shows that the pure BP neural network has limitations in processing specific data sets.

The fitting effect diagram of GA-BP neural network is shown in Figure 3.

ISSN 2616-5775 Vol. 8, Issue 4: 92-97, DOI: 10.25236/AJCIS.2025.080411

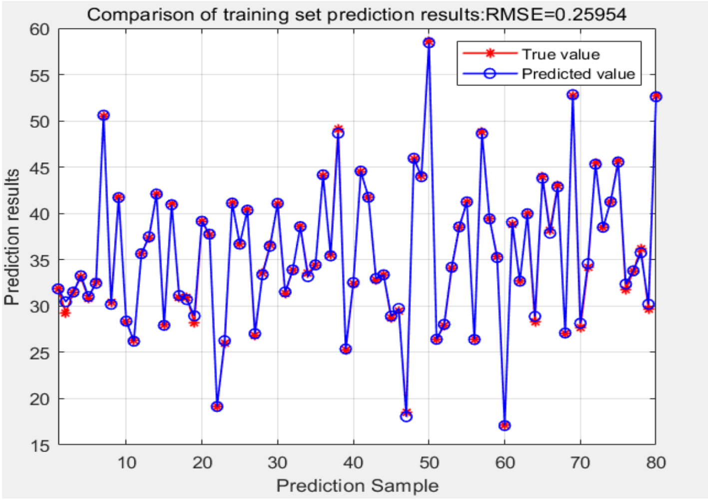

Figure 3 The fitting effect diagram of GA-BP neural network

The regression diagram of the GA-BP neural network is shown in Figure 4

The overall prediction accuracy of the GA-BP model has been significantly improved compared with the pure BP neural network. Although the data has certain fluctuations, the model can still capture the changing trend of the true value well, indicating that the optimized model has certain adaptability and stability when processing data with fluctuating characteristics.

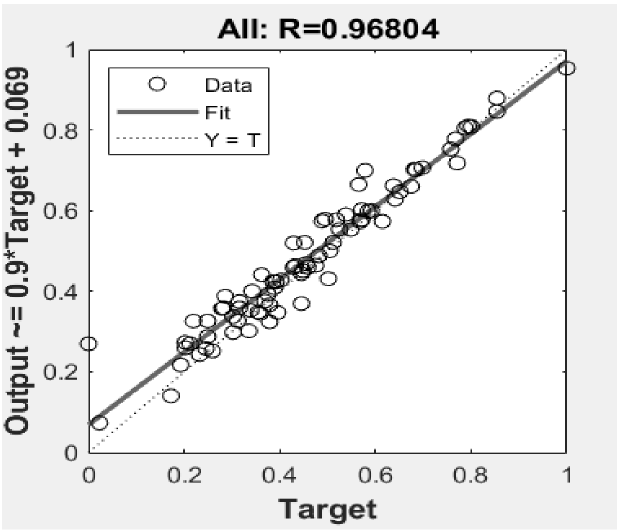

Figure 4 The regression diagram of the GA-BP neural network

5. Conclusion

This paper aims at the problems that BP neural network faces when it comes to initial weights and thresholds, such as overfitting and slow convergence speed. It uses genetic algorithms to optimize it and successfully improves the training efficiency and prediction performance of the model. Through experimental verification, the genetic algorithm not only effectively avoids the problem of local optimal solutions, improves the fitting effect, but also significantly shortens the training time, demonstrating its advantages in processing complex nonlinear problems. In the future, other methods can be explored. The combination of evolutionary algorithm and GA-BP compares the effects of different algorithms on performance improvement. The improved algorithm has better effects on problems such as roadbed settlement prediction and missing power data.

References

[1] Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating

ISSN 2616-5775 Vol. 8, Issue 4: 92-97, DOI: 10.25236/AJCIS.2025.080411

errors. Nature 323, 533–536 (1986).
[2] Song, Shaohui, et al. "Modeling the SOFC by BP neural network algorithm." International Journal of Hydrogen Energy 46.38 (2021)
[3] Lin, Xinyou, Zhaorui Wang, and Jiayun Wu. "Energy management strategy based on velocity prediction using back propagation neural network for a plug-in fuel cell electric vehicle." International Journal of Energy Research 45.2 (2021)
[4] Hosseinzadeh, Mehdi, et al. "A multiple multilayer perceptron neural network with an adaptive learning algorithm for thyroid disease diagnosis in the internet of medical things." The Journal of Supercomputing 77 (2021)
[5] Zhang, Dehua, and Sha Lou. "The application research of neural network and BP algorithm in stock price pattern classification and prediction." Future Generation Computer Systems 115 (2021)
[6] Li X, Wang J, Yang C. Risk prediction in financial management of listed companies based on optimized BP neural network under digital economy[J]. Neural Computing and Applications, 2023, 35(3): 2045-2058.
[7] Mall, Pawan Kumar, et al. "FuzzyNet-Based Modelling Smart Traffic System in Smart Cities Using Deep Learning Models." Handbook of Research on Data-Driven Mathematical Modeling in Smart Cities. IGI Global, 2023. 76-95.