Automation and Machine Learning (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/autml.2024.050204
ISSN 2516-5003 Vol. 5 Num. 2

Temperature Prediction Based on NEAT-Optimized GA-BP Neural Network

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

[1]*School of Economics and Management, Tiangong University, Tianjin, 300387, China*
[2]*School of Traffic & Transportation Engineering, Central South University, Changsha, 410083, China*
[*]*Corresponding author: AlbusSnape77@gmail.com*

Keywords: Temperature Prediction, Back Propagation Neural Network, NEAT Algorithm, Data Preprocessing, Model Optimization

Abstract: Accurate temperature forecasting is crucial for fields like meteorology, agriculture, energy management, and urban planning. Traditional models often fail to capture complex nonlinear patterns in temperature data. The BP neural network, while powerful, faces challenges such as local minima and network structure selection. To address these, this paper integrate Genetic Algorithms (GA) with BP neural networks using the NEAT algorithm, which evolves network topologies and weights. This hybrid GA-BP: NEAT model enhances prediction accuracy and stability by avoiding local minima and optimizing network complexity. Experimental results show significant improvements in forecasting nonlinear time series data, offering valuable insights for various applications.

1. Introduction

In recent years, the rapid development of artificial intelligence and machine learning has offered new avenues for temperature prediction. Among these methods, the backpropagation (BP) neural network has become a significant tool in the forecasting arena due to its powerful nonlinear mapping capabilities[1]. However, traditional BP neural networks face inherent challenges, such as susceptibility to local minima and difficulties in selecting network structures, which limit their predictive performance in practical applications.

To overcome these limitations, researchers have begun exploring the integration of genetic algorithms (GA) with BP neural networks. Among these methods, the NeuroEvolution of Augmenting Topologies (NEAT) algorithm, with its unique evolutionary strategy and capabilities, has shown tremendous potential in the forecasting domain[2].The NEAT algorithm evolves the topology and weights of neural networks, automatically discovering optimal network structures, thus circumventing the challenge of selecting network structures inherent in traditional BP neural networks.

A key advantage of the NEAT algorithm is its ability to avoid the problem of local minima through a population-based global search strategy. Additionally, the NEAT algorithm can adaptively adjust the complexity of the network, ensuring that the model is neither overly simplistic nor overly complex, thereby enhancing the accuracy and stability of temperature predictions[1].

This paper's main contributions are as follows:

a. Novel Integration of NEAT with GA-BP Neural Networks: This study uniquely combines the NEAT algorithm with GA-BP neural networks to optimize both network structure and weights, addressing the common issues of local minima and suboptimal network configuration in traditional BP networks.

b. Enhanced Prediction Accuracy and Stability: By leveraging NEAT's evolutionary capabilities, the proposed model significantly improves the predictive performance for nonlinear and complex time series data.

c. Comprehensive Model Evaluation: Detailed experimental results demonstrate the model's effectiveness in various scenarios, highlighting its practical applicability in fields like meteorology, agriculture, and energy management.

d. Innovative Approach to Data Preprocessing and Network Optimization: The study introduces advanced techniques for data preprocessing, including normalization and sliding window methods, ensuring robust model training and evaluation.

By applying the NEAT algorithm to temperature data prediction, researchers have developed a new temperature prediction model. This model not only improves the performance of temperature predictions but also introduces new ideas and technological approaches to the field[3].

2. Model Development

2.1 Overview of BP Neural Networks

The backpropagation (BP) neural network is a complex multilayer feedforward neural network with excellent information processing capabilities. It can learn to predict outcomes without an explicit mathematical function mapping inputs to outputs. This capability makes it particularly suitable for handling complex decision-making tasks in uncertain environments. BP networks are widely used in various fields for predictive modeling. The network structure is shown in Figure 1.

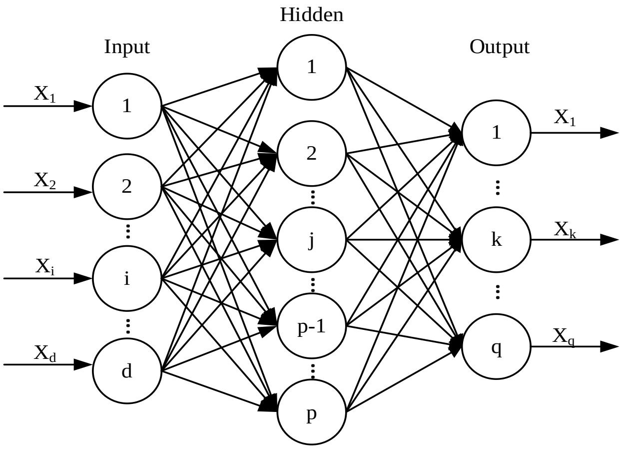

Figure 1: Neural Network Structure

The strength of BP neural networks lies in their learning capability. They are able to capture complex nonlinear relationships between inputs and outputs and have generalization abilities. This makes them extensively used in function approximation, pattern recognition, classification, and prediction tasks[4].

However, BP neural networks also have limitations. Firstly, they can fall into local minima rather than global ones. Secondly, for large-scale networks and datasets, training time can be lengthy. Additionally, the network may overfit the training data, thus reducing its generalization ability on new data. Lastly, the selection of the number of nodes and layers in the hidden layers lacks theoretical guidance and is often determined experimentally.

2.2 Optimizing BP Neural Network Parameters Using NEAT

To overcome the performance limitations and local optima issues of traditional neural networks during initialization of weights and thresholds, this study employs the enhanced GA-BP algorithm—NEAT[5]. By simulating the natural selection, mutation, and heredity mechanisms of biological evolution, NEAT demonstrates superior global search capabilities, effectively reducing the risk of convergence to local optima. Optimizing the structure and parameters of the neural network using NEAT enhances the model’s predictive performance and generalization abilities. The specific implementation steps of the improved GA-BP neural network Temperature Prediction model are depicted in Figure 2.

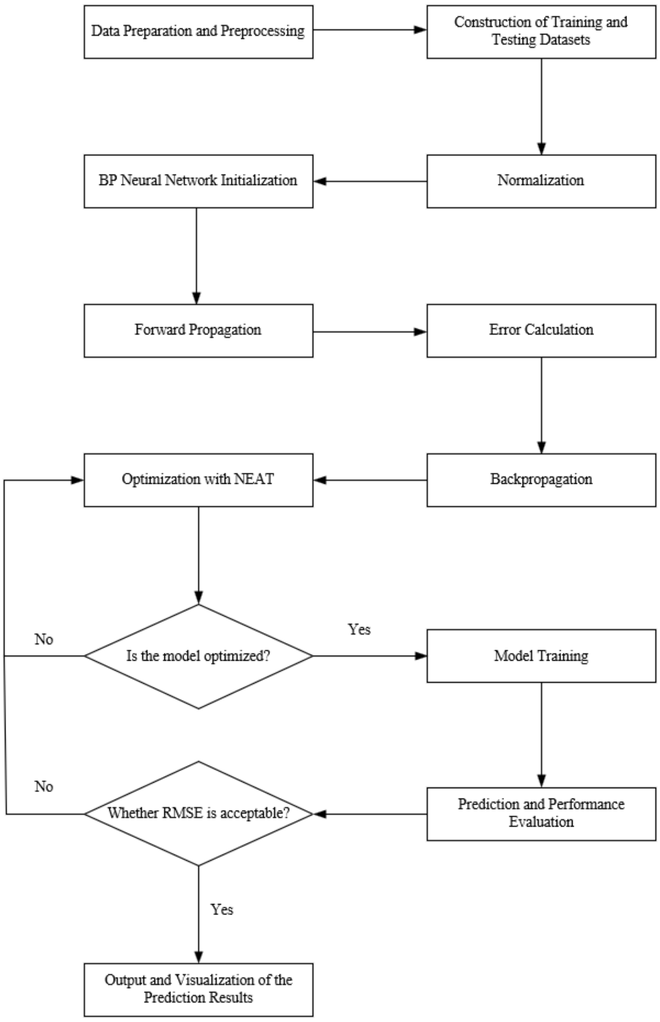

Figure 2: Algorithm Optimized BP Neural Network Flowchart

Addressing the issue of unstable learning speeds and slow convergence rates in handling large-scale training sets, this paper first utilize NEAT's search capabilities to optimize the network's structure and weights, thereby improving network performance. Subsequently, this paper train the established temperature prediction model using training data to achieve optimal input-output mapping relationships. The NEAT neural network structure significantly reflects the optimization of BP neural network parameters through the network weight adjustment formula, presented in Eq.(1):

$$weight = weight + mutation\_rate \times random\_value \quad (1)$$

Where: $mutation\_rate$ is the mutation rate, and $random\_value$ is a random value (e.g., sampled from a normal distribution).

The temperature prediction algorithm process based on the NEAT neural network is described below:

Step 1: Data Preparation and Preprocessing

Using the pandas library, temperature data is read and converted into a NumPy array to facilitate efficient numerical computations. This preprocessing ensures the data format is suitable for subsequent analysis and model training.

Step 2: Construction of Training and Testing Datasets

The dataset is divided into training and testing sets with an 80/20 split. Then a function generates a set of inputs (features) and outputs (labels) using a sliding window method. This method helps the model learn time dependencies by considering a sequence of consecutive data points for predicting the next point.

Step 3: Normalization

Normalization scales all feature values to the same numeric range, typically [0,1]. This step is crucial for preventing issues such as gradient vanishing or explosion during training, thus speeding up learning and enhancing model performance.

Step 4: BP Neural Network Initialization

Initialize the BP neural network with random weights and biases. The network consists of an input layer, more hidden layers, and an output layer. The number of neurons in each layer is determined based on the complexity of the prediction task.

Step 5: Forward **Propagation**

In the forward propagation step, input data is passed through the network. Each neuron's output is computed using the following formula, presented in Eq.(2):

$$[output = f(\sum_{i=1}^{n}(w_i \cdot x_i) + b)] \quad (2)$$

Where f is the activation function, w_i are the weights, x_i are the input values, and b is the bias.

Step 6: Error Calculation

Calculate the error at the output layer using the mean squared error (MSE) between the predicted and actual values, presented in Eq.(3):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (3)$$

where y_i are the actual values and $\hat{y}_i$ are the predicted values.

Step 7: Back propagation

Adjust the weights and biases to minimize the error using the back propagation algorithm. The weight adjustment formula is presented in Eq.(4):

$$w_{new} = w_{old} - \eta \cdot \frac{\partial MSE}{\partial w} \quad (4)$$

Where η is the learning rate.

Step 8: Optimization with NEAT

To enhance the BP neural network's performance, employ the NEAT algorithm to optimize the network's structure and weights. NEAT's evolutionary process involves the following steps:

Population Initialization: Generate a population of random network structures and weight configurations.

Fitness Evaluation: Define a fitness function based on the MSE to assess each genome's performance.

Selection: Select the top-performing genomes based on their fitness scores.
Crossover and Mutation: Apply crossover and mutation operations to produce new offspring, introducing variations in the network structures and weights.
Evolution: Repeat the evaluation, selection, crossover, and mutation steps for multiple generations until an optimal network structure is found.

Step 9: Model Training

Train the optimized BP neural network using the training dataset. The NEAT algorithm's optimized weights and structure enhance the network's predictive capabilities and generalization abilities.

Step 10: Prediction and Performance Evaluation

Use the trained neural network to make predictions on the test dataset. Evaluate the model's performance using the root mean squared error (RMSE), presented in Eq.(5):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

Where n is the number of samples, yi are the actual values, and $\hat{y}_i$ are the predicted values.A low RMSE value indicates high prediction accuracy.

3. Experiment 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.

To prepare the data for temperature prediction analysis, this paper implemented the following steps:

- Data Series Decomposition: This paper separated the trend and seasonal components from the raw data to better understand the structure of the data.
- Sliding Window: The `create_dataset` function was used to create a sliding window dataset, capturing patterns within the sequence. This function takes data and window size as inputs and outputs a series of feature and label pairs for training. The choice of the `window_size` parameter depends on the characteristics of the data and the requirements of the prediction task.
- Dataset Splitting: This paper divided the dataset into training and testing sets, with 80% of the data used for training and the remaining 20% for testing.
- Data Normalization: This paper utilized MinMaxScaler to normalize the data, scaling all data to the [0,1] range to eliminate the impact of different magnitudes. In the code, MinMaxScaler operates using the following formula, presented in Eq.(6):

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{6}$$

Where X is an original data point, and $Xmin$ and $Xmax$ are the minimum and maximum values in the dataset, respectively. The normalized values are confined within the [0,1] interval.

3.2 Algorithm Simulation Testing

This paper built a neural network model based on the NEAT algorithm using the neat-python library. A key feature of the NEAT algorithm is its ability to evolve both the topology and weights of the network. This paper defined a fitness function based on mean squared error to assess the ASperformance of each genome. Through the training process, this paper identified the optimal neural network structure, which was used to make predictions on the test data.

3.3 Performance Evaluation

Performance was assessed by calculating the root mean squared error (RMSE) on the test dataset. RMSE is a commonly used metric to measure prediction accuracy, providing a quantification of the difference between predicted and actual values. This paper also visualized the actual and predicted temperature values to visually demonstrate the model's predictive performance.

3.4 Visual Analysis

Using visualization tools to analyze prediction results effectively demonstrated the model's predictive capabilities and identified areas for improvement. By comparing actual and predicted values, this paper detected specific points where the model showed prediction biases, offering insights for further model refinement.

Figure 3 shows a line graph comparing the predicted temperature values with the actual temperature values for the training set. The x-axis represents the prediction samples, while the y-axis represents the prediction results in terms of temperature values. The blue line indicates the actual temperature values, and the orange line indicates the predicted values by the model. The close alignment of the predicted values with the actual values demonstrates the model's ability to capture the underlying patterns in the training data, indicating effective training and a good fit.

Figure 4 presents a line graph comparing the predicted temperature values with the actual temperature values for the test set. Similar to Figure 3, the x-axis represents the prediction samples, and the y-axis represents the prediction results. The blue line represents the actual temperature values, while the orange line represents the predicted values. The predicted values closely follow the trend of the actual values, showing that the model generalizes well to unseen data and maintains good prediction accuracy.

Figure 5 illustrates a line graph showing the trend of the mean squared error (MSE) over training epochs. The x-axis represents the number of epochs, and the y-axis represents the MSE. The decreasing trend in MSE over epochs indicates that the model is learning and improving its predictions as training progresses. The eventual stabilization of the error values suggests that the model has reached a point where further training yields diminishing improvements in accuracy.

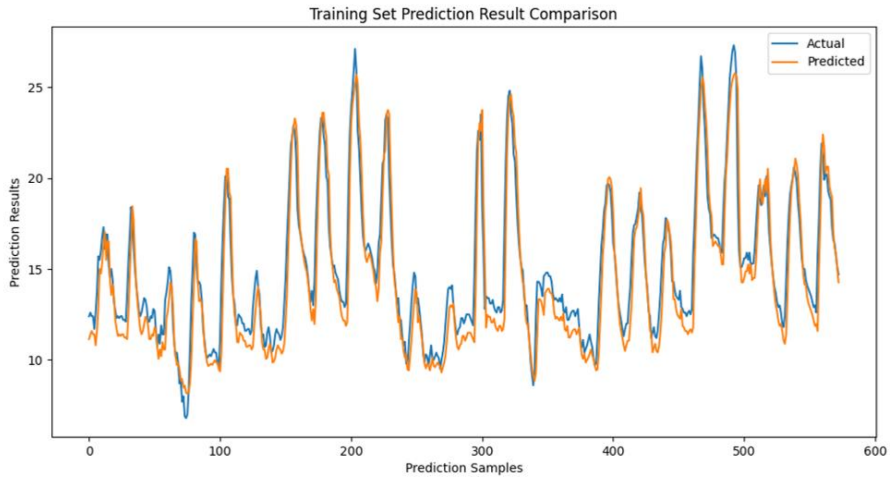

Figure 3: Training Set Prediction Result Comparison

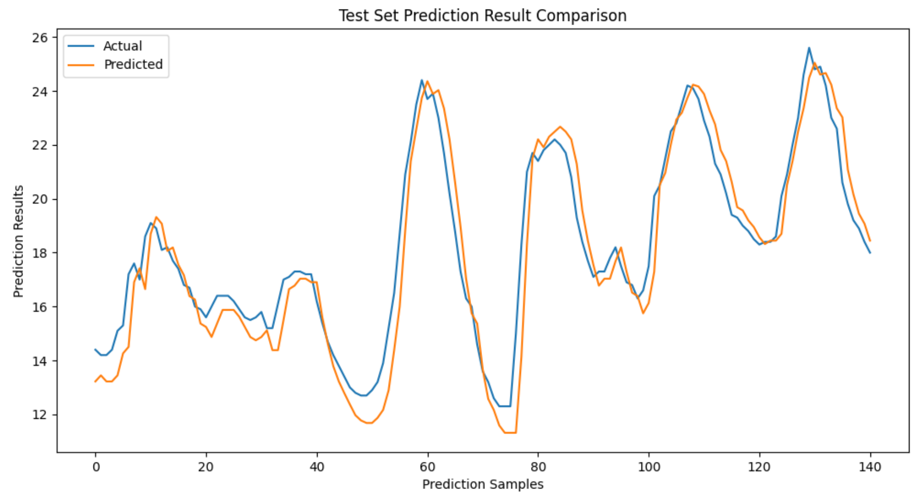

Figure 4: Test Set Prediction Result Comparison

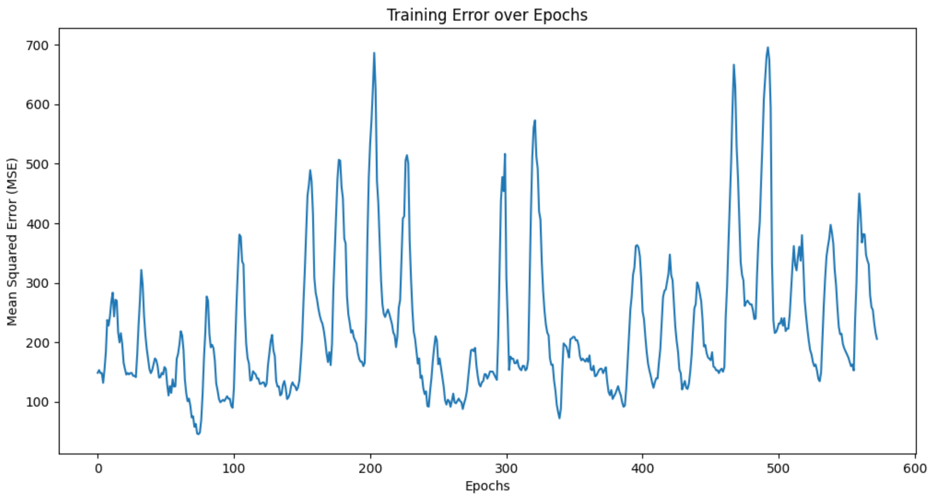

Figure 5: Training Error over Epochs

4. Conclusions

In this study, this paper have demonstrated the effectiveness of combining an improved Genetic Algorithm (GA) with Back Propagation (BP) neural networks, referred to as GA-BP: NEAT, for temperature prediction. The improved GA-BP: NEAT 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 improved GA-BP: NEAT 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 the improved GA-BP: NEAT model as a powerful tool for temperature prediction. 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] Jianan Liu. Research on Short-term Climate Prediction in China Based on Machine Learning Methods [D]. Nanjing University of Information Science & Technology, 2023.
[2] Zitong Yu. Stock Selection Model Based on BP Neural Network Optimized by Genetic Algorithm [D]. Dongbei University of Finance and Economics, 2022.
[3] Pang P ,Zheng J ,Zhao Y , et al. Thermal-vibration correlation study for high-temperature superconducting maglev intelligent monitoring based on back propagation neural network analysis[J].Superconductor Science and Technology, 2024, 37(2)
[4] Yewei Cheng. Research on Prediction of Coal and Gas Outbursts Based on SKPCA and NEAT Algorithms [D]. Liaoning Technical University, 2021.
[5] Shuxian Yang. Research on Seasonal Prediction of Summer Precipitation and Temperature in the Middle and Lower Reaches of the Yangtze River Based on Deep Learning Methods [D]. Nanjing University of Information Science & Technology, 2023.