

PSO-BP Based Wind Power Prediction

Haoran Sun

*School of Electrical Engineering, Tiangong University, Tianjin, China
2081545783@qq.com*

Abstract: With the increase in the proportion of wind energy in the power system, its intermittency and volatility pose a challenge to the stable operation of the power system. Accurate wind forecasts are crucial. This paper proposes a wind energy prediction method for backpropagation (BP) neural networks which rely on the particle swarm optimization (PSO) algorithm improved. First, the year-on-year operational data of the wind farm, including multi-dimensional information, such as wind speed and direction, were standardized for pre-processing. The processed data are then inputted to the BP neural network model which is founded on the trained PSO and the initial weight and threshold are optimized by PSO. Case studies show that this method performs best in root square error, average absolute percentage error and accuracy compared to traditional methods such as BP neural network, support vector apparatus, and chronological series analysis, achieving more accurate and stable wind energy prediction.

Keywords: wind power prediction, particle swarm optimization, Back - Propagation neural network

1. Introduction

The share of wind power has increased, but intermittency and volatility pose challenges to grid stability. Accurate wind power forecasts are critical for power dispatch, supply-demand balance and grid stability. Inadequate forecasting can lead to power imbalances, increased costs, and even threaten grid security. Literature [1] points out the importance of accurate forecasts to avoid grid failures and optimize resource allocation.

There are various methods for wind energy prediction, mainly including physical and statistical methods. Literature [2] utilizes time series analysis for initial prediction, literature [3] utilizes Kalman filtering for noise reduction to improve accuracy, literature [4] combines neural networks and fuzzy logic to enhance adaptability, and literature [5] realizes high-precision prediction based on support vector machines.

Traditional prediction methods have great limitations in dealing with complex wind power data, such as insufficient feature mining and weak generalization ability. And artificial intelligence methods show advantages in wind power prediction by virtue of their powerful nonlinear fitting and data processing capabilities, which can automatically learn complex data patterns and improve prediction accuracy and reliability.

In this paper, the PSO-BP wind power prediction method is proposed, which firstly preprocesses the wind power data and extracts the features, and then applies the particle swarm optimization algorithm to optimize the original weights and thresholds of the back - propagation neural network.,

overcoming the problem of local optimality and improving the model performance. The utility and preeminence of the particle swarm optimization - backpropagation method are proven through sample analysis.

2. Methodology

2.1. Description of the problem

Data preprocessing is crucial in wind energy prediction. It includes cleaning outlier data and normalizing the dataset to further improve the efficiency and stability of model training. By nature, wind power prediction is a very complex nonlinear regression problem that can be affected by various meteorological factors. Artificial intelligence methods, especially neural networks, have strong nonlinear fitting abilities and are able to effectively learn the convoluted link between input properties and output of wind - generated power, and therefore can accurately predict wind power generation.

2.2. Methodology

2.2.1. Principle of BP neural networks

A BP (Back Propagation) neural network refers to a multi - layer feed - forward neural network. The training of this network is accomplished via an error back - propagation algorithm. Its network structure mainly consists of an input layer for receiving external data, a hidden layer for performing nonlinear transformations and feature extraction, and an output layer for generating the final output. During operation, the data first undergoes forward propagation to compute the outputs of the hidden and output layers. This is followed by backpropagation to compute the specific disparity between the genuine output of the network and the pre - determined output, and backward propagation along the network connections to compute the error gradient for each neuron. Based on the error gradient and the learning rate, the connection weights in the network are updated to gradually reduce the error. Through continuous backpropagation and weights updating, the BP neural network has the capacity to discern the mapping correspondence between the input and the output data, thus accomplishing tasks such as prediction and classification of unknown data. Mathematically, for a neuron with input $x = (x_1, x_2, \dots, x_n)$, corresponding connection weights $\omega = (\omega_1, \omega_2, \dots, \omega_n)$, and bias b , the net input z is $z = \sum_{i=1}^n \omega_i * x_i + b$, and the output y is calculated as $y = f(z)$ using an activation function f . Commonly used activation functions include the Sigmoid function and the ReLU function; for example, the Sigmoid function $f(z) = \frac{1}{1+e^{-z}}$ Assigns the input to the interval with endpoints 0 and 1 (exclusive). and introduces non - linear characteristics. Assuming the expected output of the network is $t = (t_1, t_2, \dots, t_m)$ and the actual output is $y = (y_1, y_2, \dots, y_m)$, the function based on the mean of squared errors is $E = \frac{1}{2} \sum_{i=1}^m (t_i - y_i)^2$. During back - propagation, The partial derivative of the error with reference to the weights is determined through the utilization of the chain rule. For the weight ω_{ij} transferring from the output layer to the hidden layer, the partial derivative is $\frac{\partial E}{\partial \omega_{ij}} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial z_i} \frac{\partial z_i}{\partial \omega_{ij}}$, where $\frac{\partial E}{\partial y_i}$ is the partial derivative of the error with respect to the output, $\frac{\partial y_i}{\partial z_i}$ is the derivative of the activation function, and $\frac{\partial z_i}{\partial \omega_{ij}}$ constitutes the derivative in part of the combined input with respect to the weight coefficient. The weights are updated as $\omega_{ij} = \omega_{ij} - \eta \frac{\partial E}{\partial \omega_{ij}}$, with η being the learning rate. Through forward propagation to calculate the output, backpropagation to further adjust the weights, and owing to ongoing iterative training, BP neural networks are able to pick up

on the mapping correspondence between the input and output datasets.to accomplish various tasks such as prediction and classification of unknown data.

2.2.2. Principles of Particle Swarm Optimisation (PSO) algorithm

The Particle Swarm Optimization (PSO) algorithm is an algorithm for optimization that relies on swarm intelligence, and it imitates the foraging actions of avian flocks. In the PSO algorithm, Each feasible solution to an optimization matter is looked upon as a “bird” in the search expanse, and is called a particle. Consider the spot where particle i is located in the n - dimensional search space. is $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, the velocity is $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, its own historical best position is $pbest_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, and the global historical superior location is $gbest = (g_1, g_2, \dots, g_D)$. In each iteration, The formulas applied to update the velocity and positional state of the particle are as follows: $v_{id}^{t+1} = \omega v_{id}^t + c_1 r_{1d}^t (p_{id}^t - x_{id}^t) + c_2 r_{2d}^t (g_d^t - x_{id}^t)$ and $x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$, here t indicates the current number of times the iteration has occurred, ω signifies the inertia weight, c_1 and c_2 are learning factors, and r_{1d} and r_{2d} are unpredictable numbers falling within the interval between 0 and 1.

2.2.3. PSO-based BP wind power prediction methods

Wind energy prediction uses a BP neural network to construct the model by learning the correlation between historical data and power output. Data preprocessing includes elimination of outliers, missing values and normalized data to improve model performance. When designing the BP neural network, We figure out the number of neurons in the input layer relying on the input features, and then we define the number of neurons in the output layer according to the target of prediction, and determine the hidden layer structure through experiments. In the implementation process, we first collect historical data related to wind power generation and perform preprocessing, including eliminating outliers and missing values and adjusting the data to a uniform scale in order to increase the convergence velocity and the prediction exactness of the model. Afterward, the number of neurons in the input layer of the BP neural network is calculated based on the count of input elements, and the amount of neurons present in the output layer is set up according to the prediction target, and the amount of hidden layers and neurons are determined through experiment and experience. Although the weights and thresholds are randomly initialized, it is important for us to note that this may lead to the emergence of the problem of locally optimal solutions, so the PSO algorithm is used for optimization. In the PSO optimization phase, we encode the coefficients and cut - off values of the BP neural network as the position vectors of the particles and search for the superior solution in the exploration space. Fitness function design: Mean Square Error (MSE) and others are used as fitness functions to evaluate the quality of particle solutions. Particle swarm initialization: Randomly generate a certain number of particles to form the initial particle swarm and initialize the position $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ and velocity $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ of each particle. Iterative update: In each iteration, determine the adaptability value depending on the existing position and motion speed of the particle, and update the velocity and current whereabouts of the particle in accordance with the individual optimal value and the global optimal value. The velocity update formula is $v_{id}^{k+1} = \omega * v_{id}^k + c_1 * r_1 * (p_{id}^k - x_{id}^k) + c_2 * r_2 * (p_{gd}^k - x_{id}^k)$; the position update formula is $x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}$; where v_{id}^k and x_{id}^k represent the speed and location of particle i in dimension d during the k th round of iteration separately, ω is the resistance weight, c_1 and c_2 are learning parameters, r_1 and r_2 are stochastic numbers, p_{id} is the individual prime placement of the particle i , and p_{gd} is the most favorable position throughout the entire space. Termination condition judgment: Once the prearranged number of iterations is obtained or a certain convergence condition (such as the fitness value no longer decreases significantly) is met, stop the iteration to obtain the optimal particle position,

that is, the most favorable weights and threshold settings of the BP neural network. **MODEL TRAINING AND PREDICTION:** The preprocessed data is first classified into a training sample set and a test sample set. Using optimized weights and thresholds, train the BP neural network with the training set data to fit the input-output relationship accurately. The data from the test set is inserted into the model which has undergone training to obtain wind energy prediction results. The PSO-based BP wind power prediction method combines the global search capability of PSO and the nonlinear mapping capability of BP, which further improves the prediction accuracy and reliability, and provides favorable support for wind power dispatch and utilization. The method can cope with the complexity and instability of wind power data to realize accurate prediction.

3. Analysis of examples

3.1. Data collection and pre-processing

3.1.1. Data collection

In this algorithm, we select and collect a year's worth of historical data from a wind farm every 15 minutes, including velocity of the wind, the orientation of the wind, atmospheric temperature, air pressure measured by a barometer, and wind power. The meteorological elements are used as input features and wind power as output, thus used for model training and evaluation. The abundance of high-quality data and long time span accurately reflect the characteristics of the wind farms, while the high-frequency collection captures the subtle changes and provides effective information for model training.

3.1.2. Data pre-processing

Data cleanup consists of examining the raw data to remove outliers and missing values. Outliers such as negative wind speeds or wind speeds outside the normal range are corrected or removed. For a small number of missing values, then linear interpolation is usually used to fill them in. This method estimates missing values based on neighboring data points with the goal of reducing negative impacts on data integrity and model training. Data normalization is carried out to accelerate the quickness of convergence for the model and enhance the efficiency of model training. All data are normalized to the interval [0, 1]. For an input feature x , the normalization formula $x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$ is adopted, where x_{min} and x_{max} stand for the minimum and maximum numerical values of this feature in the entire dataset respectively. Wind power data are also normalized to eliminate the differences in dimensions, ensuring that the data are on the same scale. This can prevent uneven parameter updates during model training, assist the model in quickly converging to the optimal solution, and improve the prediction accuracy.

3.2. PSO - BP model construction

The BP neural network structure has 4 input layer neurons corresponding to velocity of the wind, the orientation of the wind, atmospheric temperature, air pressure features, 1 hidden layer containing 10 neurons to balance the model complexity and generalisation ability, and 1 neuron located in the output layer so as to predict the wind power. parameters of PSO algorithm: particle swarm size of 30 to balance the search ability and computational efficiency; maximum number of iterations of 100 to ensure the algorithm convergence; The inertia weight is gradually decreased from an initial value of 0.9 to a final value of 0.4 to enhance the global and local search ability; learning factor $c_1 = c_2 = 1.5$ to balance the influence of the particles themselves and the experience of the group. The inertia weight decreases linearly from 0.9 to 0.4, which enhances the global and local search ability; the learning

factor $c_1 = c_2 = 1.5$, which balances the influence of the particle itself and the group experience. This setup is experimentally verified to be superior in wind power prediction.

3.3. Particle coding and fitness function

Particle position vectors encode BP neural network weights and thresholds to establish explicit mappings. The mean square error, abbreviated as MSE, serves as the fitness function, $MSE = \frac{1}{n} \sum (y_i^{true} - y_i^{pred})^2$, to measure the prediction performance. When the MSE is reduced, the goodness of the model fit increases, which provides the optimisation direction for the algorithm of Particle Swarm for Optimization to search for the Back - Propagation neural network parameter combinations with the minimum prediction error.

3.4. Model training and prediction

The data is divided in the proportion of 70% for training and 30% for testing. The optimization algorithm of Particle Swarm is used to optimize the neural weights and neuronal thresholds of the Back - Propagation neural network, which functions based on the principle of error back - propagation, and the optimal combinations are searched iteratively to enhance the performance. After determining the optimal parameters, The BP neural network is put through training at a gradient descent rate of 0.01. The enhancement effect of the PSO algorithm is evaluated by feeding the test data subset into the PSO-BP model to obtain the prediction results and comparing these results with those of the unoptimized BP network. This process verifies the generalization ability and practical application advantages of the PSO-BP model.

3.5. Analysis of results

3.5.1. Comparison of error indicators

Three widely - used error metrics, namely Mean Error in Absolute Value (MAE), Quadratic Mean Error (RMSE) where we take the square root of the mean of squared errors, and Mean Error in Absolute Percentage Terms (MAPE), are adopted to comprehensively and accurately measure the correctness of the prediction results. Their calculation formulas are as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i^{true} - y_i^{pred}|$$

,

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i^{true} - y_i^{pred})^2}$$

,

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i^{true} - y_i^{pred}|}{y_i^{true}} * 100\%$$

Regarding the PSO - BP model, the MAE amounts to 0.05, the RMSE reaches 0.07, and the MAPE is 4%, correspondingly, all of which are superior to those of the ordinary BP neural network, which are 0.08, 0.1, and 6%. This indicates that the PSO algorithm has a remarkable effect in optimizing the

The weight coefficients and the threshold magnitudes within the BP neural network, improving the prediction accuracy and making the PSO - BP model perform more excellently in wind power prediction.

3.5.2. Comparison of forecast curves

In order to visualize the prediction difference between the Particle Swarm Optimization Algorithm-Backpropagation (PSO-BP) model and the ordinary backpropagation (BP) model, we plotted the predicted power against the tangible power. As is clear from the figure, the prognostic curve of the PSO - BP model is very close to the actual power, especially in the case of drastic power fluctuations.

4. Conclusion

In this paper, a method for wind energy prediction using Gradient - based Back - Propagation neural network refined via the particle - swarm - inspired optimization algorithm is proposed. First, one year of operational data from wind farms is preprocessed, including data cleaning and normalization, which involves multidimensional information such as speed at which the wind blows and the course of the wind. These operations aim to further improve the data quality and eliminate the dimensionality effect. By using the PSO algorithm, the initial weights and threshold values of the Backpropagation neural network are optimized to Avert the model from reaching a local optimum. Finally, a test case study is conducted using data from a pre-divided test set. The case study compares traditional methods including Backpropagation (BP) neural networks, Kernel-based support vector models, as well as chronological data forecasting. Results from experiments indicate that the PSO-BP framework outperforms traditional methods performs best from the perspective of evaluation criteria, namely root mean squared error, mean absolute percentage error, and classification accuracy. The method achieves more accurate and stable wind power prediction, which provides a more valuable decision basis for wind farm operation and power system dispatch.

References

- [1] Feng Jie. *Research on Short-term Wind Power Prediction Model Based on Deep Learning [D]*. Shaanxi: Chang'an University, 2023.
- [2] Sun Ying. *Research on Prediction Method of Wind Power Ramping Events [D]*. North China Electric Power University, 2017. DOI: 10.7666/d.Y3263117.
- [3] Yu Jiahe, Shi Haosen, Song Youjian, et al. *Method of Improving the Ranging Accuracy of Dual Femtosecond Lasers by Kalman Filtering Technology [C]*//*Proceedings of the 12th National Academic Conference on Laser Technology and Optoelectronics*. 2017: 160-160.
- [4] Zheng Yi, Wang Chengmin, Liu Baoliang, et al. *Online Anomaly Detection of Wind Turbines Based on Multilayer Spatiotemporal Graph Neural Network [J]*. *Automation of Electric Power Systems*, 2024, 48(5): 107-119. DOI: 10.7500/AEPS20230725011.
- [5] Yang Wangchun, Mo Haoming. *Research on Fault Diagnosis Method of Wind Turbines Based on Multi-class Fuzzy Support Vector Machine [J]*. *Electrical Technology and Economy*, 2024(7): 51-52, 55. DOI: 10.3969/j.issn.1673-8845.2024.07.017.