Power Generation Prediction Model Based on Improved PSO-BPNN

Yan Yu
School of Automation, Chongqing University of Posts and Telecommunications, Chongqing, China
yuyan@cqupt.edu.cn

Abstract. Electricity generation greatly impacts economic development, and electricity is indispensable for production, transportation, and living. Therefore, forecasting electricity generation accurately is of great research significance for the development of the country and the livelihood of the people. Because of the nonlinear relationship between electricity generation and the influencing factors, this paper, supported by the above data in China over the past 20 years, describes a prediction model based on Improved Particle Swarm Optimization (PSO) -- Back Propagation Neural Network (BPNN) to optimize the algorithm about forecasting electricity generation. The experimental results have shown that the accuracy and stability of the prediction model were constructed in this paper, which was improved by about 2%-6% compared with the traditional ones. In addition, the application of this model could provide a constructive theory for some relevant works in the electric-power industry.

Keywords: Improved PSO; BPNN; electricity generation; prediction; optimization.

1. Introduction

Electricity generation refers to the amount of electric energy produced by the energy conversion of generators, which has an important impact on economic development. Forecasting electricity generation can predict the demand for power production and power generation equipment and help relevant enterprises make overall preparations, ensuring the country's power demand and people in all aspects, including production, transportation, living, etc. Consequently, accurate prediction of power generation is an important issue concerned by scholars.

At present, many domestic and foreign experts have carried out related research on power generation prediction. For example, Lu et al. [1] proposed an ARIMA prediction model based on time series to predict power generation. The model is based on its historical data fitting and prediction. It is easy to operate and has low requirements for data collection. However, it requires high stability of data and is often used to capture linear relationships. Moreover, our country's power generation data is not stable and does not have good linear conditions. Even using data differential processing, the prediction error is still large. Dai [2] used the BPNN model to predict photovoltaic power generation. The nonlinear relationship of power generation was predicted by analyzing environmental temperature, solar radiation, and other factors. This model may be a successful prediction model. It had a strong self-learning ability and can deal with nonlinear data well. However, during the construction of a neural network, the
parameters, such as weights and thresholds, have great randomness, causing the training results to fall into local minimum values easily. Ma [3] pointed out the prediction model based on Improved PSO-BPNN and applied it to the prediction of short-term traffic flow, which effectively improved the prediction accuracy and stability of the algorithm.

Therefore, aiming at the nonlinear relationship between our country's power generation and the number of industrial enterprises, current industrial assets, and other factors in the past 20 years, this paper, based on predecessors' shortcomings, proposes a power generation prediction model based on the shortcomings of predecessors' on Improved PSO-BPNN. When the algorithm does not converge, genetic algorithms such as crossover and mutation operate some not optimal particles to avoid the algorithm falling into local optimization. Additionally, it improves the prediction accuracy of the algorithm through searching and determining the weights and thresholds of each layer of BPNN by this improved particle swarm algorithm.

2. Preliminary
At the beginning of this study, we need to search and collect some data online, preprocess some missing values and normalize these data.

2.1. Data Sources
By searching the RESSET database and consulting the official website of the National Bureau of Statistics, four groups of monthly data were obtained from February 2000 to December 2020, including the number of enterprises per month, the total current assets per month (billion yuan), the total assets per month (billion yuan), and the current value of power generation (billion kWh). Then, we imported them into Excel tables and established an experimental database.

2.2. Missing Value Processing
Due to some problems, such as inconsistent statistical cycles, there are missing data for each part. To ensure the reliability of the implementation of the subsequent algorithm, Firstly, we deleted the months without statistical data (for example, January) directly. And then, we used the average values of the two data before and after this missing value to fill it.

2.3. Data Normalization
Since the units of each parameter vary greatly, the data needs to be normalized. We adopted max-min normalization to handle these data in this experiment. In other words, we processed all data and converted them to the range of [0, 1] by Python through Formula (1) to improve the operation and convergence speed of subsequent iterative solutions. [4]

\[ X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \]  \hspace{1cm} (1)

3. Model Establishment
3.1. Particle Swarm Optimization
PSO algorithm is a global random search algorithm based on simulating a biological mechanism of birds swarm foraging. Moreover, it searches for the optimal solution by iteration of the random solution, which is commonly used to solve optimization problems. Moreover, the main idea is to determine the fitness function of particles according to the objective function of the optimization problems. The particle in the search space is the candidate solution to the problems. Then, initialize the velocity (vi) and position (xi) of each particle, evaluate its fitness value by the position of each particle, and compare the optimal global position (gBesti) and the optimal historical position (pBesti) of all particles. Thereby, the velocity and position of the particles are updated according to Formula (2) (3), which is used for iteration until the set conditions are met. [5]
Where $w$ is the inertia weight coefficient to control the influence of the previous velocity on the current velocity. $c_1$ and $c_2$ are learning factors that control their movements to the individual particle's optimal historical position and global position. $r_1$ and $r_2$ are random numbers between 0 and 1.

However, PSO has the defects of slow convergence speed and is easy to fall into local optimum. Therefore, this paper conducts the dynamic decrease of inertia weight for PSO and improves genetic algorithm crossover and mutation operation for some particles in the iteration.

3.2. Improved PSO

1) Dynamic decreasing inertia weight $w$: $w$ (the inertia weight coefficient) controls the proportion of the effect of the previous speed on the current speed. Formula (2) shows that when $w$ is larger, the previous speed has a greater impact on the current speed, which is beneficial to local search, but it is easy to fall into local optimum. Conversely, it is more beneficial to global search, but the convergence efficiency is low. Therefore, this paper uses dynamic linear decreasing inertia weight to balance global and local search to seek the optimal solution. $w$ updates with Formula (4) at each iteration.

$$w' = w_{\text{max}} - \frac{t}{t_{\text{max}}} (w - w_{\text{min}})$$

2) Self-mutation algorithm: To avoid the algorithm falling into local optimum preferably, this paper also improves the self-mutation algorithm on PSO. In the iterative process of the algorithm and without convergence, some non-optimal particles would start the operation of the genetic algorithm for mutation. However, if only all non-global optimal particles do this, the particles with a high adaptive value near the optimal particle still have the same mutation probability ($P_m$) as the particles far from the optimal particle with a low adaptive value. Then the particles near the global optimal will be degraded, which is not conducive to convergence. Therefore, this paper introduces the self-mutation operator to effectively reduce the probability of the algorithm falling into the local optimum.

First, through Formula (5), which means the energy of a single particle is equal to the reciprocal of the adaptive value (where the adaptive value is the difference between the predicted output and the actual value), the smaller the adaptive value, the greater the energy of the particle, and the closer to the optimal global position.

$$E = \frac{1}{f}$$

Second, we could obtain the mutation probability ($P$) of each particle by Formula (6), and then the roulette calculation in the same genetic algorithm is used to determine whether the particle is mutated. If the particle is mutated, the position value and velocity value of the particle are randomly assigned. On the contrary, the original values are maintained without mutation operation. The roulette wheel algorithm can make the probability of mutation large. Namely, the energy and the fitness value are small, and the possibility of particle mutation far from the optimal global particle is high.

$$P = \frac{E_{\text{max}} - E}{E_{\text{max}} - E_{\text{min}}}$$
3.3. Network Structure Setting

The input and output of this study have a high nonlinear relationship, so the prediction method adopts BPNN with strong calculation ability and can solve the solid nonlinear input and output relationship. The typical BPNN has a three-layer network structure, including input layer, hidden layer and output layer. For the power generation prediction problem, three influencing factors are three nodes of the input layer, and the power generation is the output node of the output layer. According to Formula (7), the number of nodes in the hidden layer L can be calculated. In addition, the activation function of the hidden layer and the output layer is the Sigmoid function.

\[ L = \sqrt{M + N} + P \] (7)

Where M is the number of nodes in the input layer, N is the number of nodes in the output layer, and P is an integer in the range of [0, 10]. The network structure is shown in Fig. 1.

In addition, the initial weights and thresholds in the network are obtained by iterative optimization of Improved PSO. Each particle needs to contain all the weights and thresholds of the entire network. So, according to Formula (8), the dimension D of the particle can be calculated.

\[ D = M \times L + L \times N + L + N \] (8)

Then, the fitness function of the particle swarm is the error between the output value of BPNN and the output value of the sample, and the smaller the value, the better.

Then, the Improved PSO is used to iterative and update the parameters of BPNN until the termination condition is satisfied and the iteration is stopped. The optimal solution is the particle with the smallest fitness value, and then the initial value of all network parameters is given, and the network is trained according to the sample.

3.4. Power Generation Prediction Model Based on Improved PSO-BPNN

First, multiple groups of factors that have a greater impact on power generation are selected for research and comparison, such as seasons and factory emissions. Finally, three groups of data, such as the number of enterprises at the end of this month, are selected as the input of this study to predict power generation.

Second, the data are preprocessed. 12 groups are randomly selected from 120 groups as a test group and the remaining training group.

Finally, Improved PSO-BPNN is used to predict the power generation, and the flow chart is shown in Fig. 2.
The specific steps are as follows:

- **Step 1**: Determine the structural parameters of BPNN and select the activation function of the hidden layer and the output layer.

- **Step 2**: Initialize the parameters of the Improved PSO. For example, determine particles' dimension according to the number of nodes in each layer of the network and initialize the learning factor, inertia weight, particle swarm size, maximum iteration number, particle speed and position.

- **Step 3**: Calculate the fitness value. The value of each particle dimension is used as the corresponding weights and thresholds of BPNN. The corresponding output is obtained according to the input. The difference between the actual output and the output is the most adaptive value. Obviously, the better the particle is the smaller the fitness value.

- **Step 4**: Update particle and particle swarm parameters. Comparing whether the fitness value of the current particle is less than the optimal global particle if it is, the particle is updated to the optimal global particle. For non-global optimal particles, the self-variation algorithm and the roulette wheel in 3.2 (2) above are used for updating judgment and operation, and the possibility of variation of the particle far from the global optimal is great. In addition, for each iteration of particle swarm, a linear decreasing inertia weight w operation is needed. Loop iteration until the termination condition is satisfied.

- **Step 5**: Determine the weights and thresholds of BPNN. The optimal particles obtained by Improved PSO are used as the weights and thresholds of the corresponding nodes in BPNN.

- **Step 6**: Train the network to obtain the predicted value. The input data propagates forward to get the predicted output in the network, and then the error propagates backwards to update the weights and thresholds of the network. Loop iteration until the termination condition is satisfied. Then input the test group, we can get its output value, the predicted value.

### 4. Model Solution

#### 4.1. ARIMA

The ARIMA model is a time series prediction model and requires stable data, including three parameters: p, d, and q. So, firstly, we needed to test the stability of the original data. It could be seen from Fig. 3...
that the data did not fluctuate uniformly on the horizontal axis, so the data were not stable. Then, we carried out a first-order difference on the original data. Moreover, we could conclude that the ADF parameter was 1 and the KPSS parameter was 0 through the ADF unit root test and KPSS test. Therefore, we could obtain the smooth data, and we also determined that d is 1. Then the data were analyzed by autocorrelation and partial correlation to determine p and d were both 13.

**Figure 3.** Original and first-order differential power generation data

After determining the parameters, we performed the residual test on the model and concluded that the residual was close to normal distribution. Furthermore, the ACF and PACF test results in residual autocorrelation and partial autocorrelation were almost within the threshold range. Therefore, it was considered that this model meets the requirements of ARIMA. Finally, the ARIMA model was used to input the data from 1 to 195, and the power generation of 12 months from 196 to 207 was directly predicted. The predicted results are shown in Fig. 4.

**Figure 4.** Forecasting power generation by ARIMA model

4.2. **BPNN**

First, we set the network structure of BPNN, which means the number of nodes in the input layer, the hidden layer and the output layer was 3, 10 and 1, respectively. The maximum number of iterations was 20, the learning rate was 0.9, and the expected minimum error was 0.00001. The weights and thresholds of the network were randomly initialized, and Gradient descent with Momentum was used to train the network. The termination condition of network training was set to achieve the expected error. If not, the training was terminated when the maximum number of iterations was reached so that the network could achieve a better fitting effect.
Differing from the ARIMA model, the prediction of BPNN needs to input data sets. So, we selected 12 sets of data randomly as the calibration sets of the algorithm. The predicted results are shown in Figure 5, and it could be found that the error magnitude of the prediction is $10^3$ and $R$ is 0.94414, which indicates that the fitting effect of this algorithm is promising.

![Figure 5. The predicted results of BPNN](image1)

4.3. Improved PSO-BPNN

First, we set the parameters of the particle swarm. The population size was 50, the maximum number of evolutions was 120, the maximum inertia factor was 0.9, and the minimum inertia factor was 0.2. The inertia factor was dynamically linearly decreasing at each iteration. The dimension of each particle was 51, and the sum of the weights and thresholds of the whole network. Every particle's position and velocity interval was between -1 and 1. Learning factors, $c_1$ and $c_2$, were 2. $r_1$ and $r_2$ were random numbers between 0 and 1. The self-adaptive function is the difference function between output value and actual value. We calculated the self-adaptive function value and mutation probability of each particle. And then, we made iteration circulation to update the optimal values of particles and populations until we achieved the maximum iteration value.

Finally, we could get the optimal particles in the population and take the values in each dimension as the corresponding weights and thresholds on BPNN. To compare the superiority of the improved algorithms, we kept the remaining parameters of this model regarding the BPNN part being the same as
the above BPNN's. Likewise, we selected 12 groups of data randomly as the calibration set of the algorithm. The predicted results are shown in Figure 7. It could be found that the curve of the predicted results is consistent with the actual results, and the error magnitude is $10^2$. Compared with BPNN, its prediction accuracy is significantly higher.

**Figure 7.** The predicted results of Improved PSO-BPNN

Moreover, for the training results, the regression fitting graph is shown in Fig.8. We can see $R$ is 0.98673, which means the deviation between the output curve and the target output curve is small, and most of the data points are concentrated near the fitting curve, so the fitting degree is high and the overall fitting effect is good.

**Figure 8.** The regression fitting graph of Improved PSO-BPNN

This study uses the above three algorithms to predict the results in the same input data set to achieve better comparison results. The comparison between predicted values and actual values is shown in Figure 9. It can be found that the error of the prediction results of BPNN is relatively more extensive than that of the other two, and the results of the Improved PSO-BPNN and ARIMA model based on time series are almost close to the actual value. Further, from the error image of Figure 10, it can be seen that the error of Improved PSO-BPNN is smaller in most cases.
4.4. Comparative Analysis of Experimental Results Improved PSO-BPNN

1) Parameter declaration: This paper is to predict the power generation, so we introduced two parameters, the Relative Error and Absolute Error, to analyze the results of three different algorithms (Absolute Error is the absolute value of the difference between Predicted Value and True Value; Relative Error is Absolute Error divided by True Value). In addition, Mean Absolute Percentage Error (MAPE) is used to evaluate the prediction accuracy. The smaller MAPE, the higher the prediction accuracy. Similarly, Standard Deviation of Absolute Percentage Error (SDAPE) is used to evaluate the prediction stability. The smaller SDAPE, the higher the prediction stability.

2) Results: First, we calculated and compared the absolute and relative errors of the output data of twelve test sets randomly selected, namely the predicted power generation. The results are shown in Table 1. By comparison, it can be found that the error of Improved PSO-BPNN is within the acceptable range, and its maximum error is the smallest of the three algorithms.
Table 1. Error Comparison of Three Prediction Algorithms

| Number | ARIMA Predicted value | Absolute error | Relative error | BPNN Predicted value | Absolute error | Relative error | Improved PSO-BPNN Predicted value | Absolute error | Relative error | True value |
|--------|-----------------------|----------------|---------------|----------------------|----------------|---------------|----------------------------------|----------------|---------------|------------|
| 1      | 1667.61               | 11.29          | 0.67%         | 1456.07              | 222.83         | 13.27%        | 1692.80                          | 13.90          | 0.83%         | 1678.9     |
| 2      | 2500.88               | 533.58         | 21.12%        | 2817.23              | 849.93         | 43.20%        | 2304.93                          | 337.63         | 17.16%        | 1967.3     |
| 3      | 4990.44               | 82.54          | 1.68%         | 4990.40              | 82.50          | 1.68%         | 4960.77                          | 52.87          | 1.08%         | 4907.9     |
| 4      | 1101.73               | 86.77          | 7.30%         | 1193.02              | 4.52           | 0.38%         | 1374.50                          | 186.00         | 15.65%        | 1188.3     |
| 5      | 5616.30               | 81.60          | 1.43%         | 6510.00              | 812.10         | 14.25%        | 5746.39                          | 48.49          | 0.85%         | 5697.9     |
| 6      | 4484.16               | 475.14         | 9.58%         | 4057.52              | 901.78         | 18.18%        | 4709.19                          | 250.11         | 5.04%         | 4959.3     |
| 7      | 3779.79               | 258.31         | 6.40%         | 3030.24              | 467.56         | 13.37%        | 3051.35                          | 446.45         | 12.76%        | 3497.8     |
| 8      | 2909.69               | 588.11         | 16.81%        | 4360.24              | 322.14         | 7.98%         | 3818.48                          | 219.62         | 5.44%         | 4038.1     |
| 9      | 5429.90               | 61.10          | 1.11%         | 6094.31              | 603.31         | 10.99%        | 5389.02                          | 101.98         | 1.86%         | 5491       |
| 10     | 4277.69               | 83.39          | 1.99%         | 4272.66              | 78.36          | 1.87%         | 4265.99                          | 71.69          | 1.71%         | 4194.3     |
| 11     | 4698.64               | 744.66         | 13.68%        | 6147.07              | 703.77         | 12.93%        | 5598.83                          | 155.53         | 2.86%         | 5443.3     |
| 12     | 1865.42               | 117.32         | 6.71%         | 1594.78              | 153.32         | 8.77%         | 1721.52                          | 26.58          | 1.52%         | 1748.1     |

Second, we evaluated the prediction accuracy and stability of the three algorithms. As shown in Table 2, the MAPE and SDAPE of Improved PSO-BPNN are the smallest. So it can be considered that the prediction accuracy and stability are relatively good.

Table 2. Accuracy Evaluation of Three Prediction Algorithms

| Algorithms | MAPE | SDAPE |
|------------|------|-------|
| ARIMA      | 7.87%|       |
| BPNN       | 12.24%|      |
| Improved PSO-BPNN | 5.56%| 6.07% |

5. Conclusion

This study is collected and preprocessed monthly power generation and its related influencing factors in China in the past 20 years. Eventually, the parameters of weights and thresholds of BPNN were determined by PSO improved through self-variation and inertia weight decreasing dynamically, then it established the prediction model based on Improved PSO-BPNN. The experimental results showed that compared with the traditional single BPNN, the prediction accuracy and stability of the prediction model in this study were improved by 6.68% and 5.18%, respectively. Meanwhile, compared with the ARIMA model based on time series, the prediction accuracy and stability were improved by 2.31% and 1.92%, respectively.

Moreover, in future work and research, we would consider more sufficient influencing factors and the predicted time length ranges.

6. Reference

[1] Lu J.C., Zhang S.Y., Niu X. D., "Generation forecasting method based on ARIMA," Journal of North China Electric Power University, vol.3, pp.78-80, 2004.
[2] Dai J., "Power generation forecasting method based on ARIMA research on photovoltaic power generation forecasting based on Improved BP neural network", Journal of Nanchang Aviation University (NATURAL SCIENCE EDITION), vol.29(03), pp.91-97, 2015.
[3] Ma Q.F., "BP neural network short-term traffic flow prediction based on Improved PSO optimization", Computer Simulation, vol. 36(04), pp.94-98+323, 2019.
[4] Niu D. L., Chen Y.Y., Cheng Z.Z., "Research on short-term prediction of egg price in Beijing agricultural products wholesale market based on BP neural network", China Poultry, vol.39(24), pp.35-40, 2017.
[5] Li X., Dai W., Gao H.J., Xu W.P., Wei X.H., "Study on the correlation between grain yield and chemical fertilizer consumption based on BP neural network", Journal of Agricultural Machinery, vol. S1, pp.186-192, 2017.