Power Load Forecasting Model Based on Deep Neural Network

Jian Yuan1*, Jiaying Wang1, Qing Cheng2, Jianqiao Sun1

1State Grid Zhejiang Marketing Service Center, Hangzhou, Zhejiang, China, 310014

2Inner Mongolia Electric Power Marketing Service Company, Hohhot, Inner Mongolia, China, 010000

*Corresponding author e-mail: youki1100@sgzmsc.org

Abstract. Aiming at the problems of the traditional power load forecasting model, based on the analysis of the traditional DNN neural network model, a PSO dual improvement and optimization power load forecasting model is proposed. In this model, the discrete particle swarm algorithm is used to determine the DNN network architecture, and then the particle swarm algorithm is used to optimize the parameters of the neural network to obtain a model with the best structure and parameters. Finally, it is verified by simulation, and the results show that the above method is feasible and has high prediction accuracy.

Keywords: Deep Neural Network, PSO Double Improvement, Prediction

1. Basic method

Scholars at home and abroad have done more research on the NN model, and used the NN model to predict the short-term load of the power system earlier. In contrast, although the academic community has more heated discussions on the deep neural network (DNN), but so far there is no example of research on short-term load forecasting of power system based on DNN model. The DNN model has a compact non-linear mapping relationship and has the ability to deal with a large set of functions. In the case of sufficient samples, iterative training of the built-in multi-hidden layer model can achieve satisfactory prediction accuracy. [1-2] In this context, this article attempts to use the DNN model to scientifically predict the short-term load of the power system, so as to provide convenience for the management of the power system. The DNN model architecture is shown in Figure 1:
Figure 1. The basic architecture of the DNN model

According to the analysis in Figure 1 above, the DNN model belongs to a three-layer architecture, namely the input layer, the hidden layer and the output layer. What is particularly special is that the DNN model has a multi-hidden layer structure, which is the biggest feature that distinguishes it from traditional feedforward networks. The $X$ in Figure 1 above refers to the input of the DNN network, represented as $X = [x_1, x_2, x_3, ..., x_m]$, and the input is an $m$-dimensional column vector.\[3-5\] Apply the DNN model to the field of power system load forecasting. At this time, $X$ refers to a multi-dimensional column vector composed of specific factors (such as temperature conditions, load data, etc.). Generally speaking, the more complex the operating conditions of the power system, the larger the dimension of the vector. A linear identity function is used as the activation function of the input layer, the $X$ vector completes the identity transformation in the input layer, and then is transmitted to the first hidden layer. \[6\] The activation function of the first hidden layer performs a nonlinear transformation on the input vector, and then the result of the operation is transmitted to the next layer of neurons, and so on, the output $y$ is finally obtained. They represent the weight parameters and threshold parameters between the hidden layer $l-1$ and the hidden layer $l$ in turn.\[7-8\]

2. PSO double optimization of DNN network

2.1. Optimization ideas

In the process of applying the DNN model, the first task is to determine the structural parameters of the network, including the number of hidden layers and the number of nodes in each hidden layer, and then determine the optimal solution $(w_i, b_i)$ between each layer. If the DNN model contains $n$ hidden layers, then the number of layers of the DNN network is $n+2$; the number of neurons in the $i$-th hidden layer is $m(i=1,2,...,n)$, the corresponding transfer function is $f_i$; the number of elements in the input vector is $z$. Based on the above settings, the dimensions of the weight matrix between the input layer and the first hidden layer, between the adjacent hidden layers, and between the output layer and the last hidden layer are $[m_1, z]$, $[m_i, m_{i+1}]$, $[y, m_n]$. \[9-10\] Considering that only the load prediction result of the power system at a certain moment can be obtained in one training, the number of neurons in the output layer is $y = 1$. In addition, there is a threshold $b$ corresponding to the layer of neurons between each network structure. The number of thresholds between adjacent networks is $b_i$, at this time, the number of thresholds to be optimized is $\sum_{i=1}^{n+2} b_i$. According to the above analysis, when using the DNN model to perform power system load forecasting, the structural parameters of the DNN model include the number of hidden layers, the number of neurons in each hidden layer, the transfer function of each hidden layer, and the number of elements in the input vector. The optimization of the DNN model is divided into two stages: the first stage is to determine the number of hidden layers and the number of neurons in each hidden layer, and the second stage is to determine the optimal weights and thresholds of the network.
model for power system load forecasting, in order to ensure the final forecast accuracy, sufficient structural parameters must be optimized.

The number of parameters that need to be optimized during DNN training can be determined by the following formula (1):

\[
\text{paranum} = n + \sum_{i=1}^{a} m_i + z \times m_1 + \sum_{i=1}^{a-1} (m_i \times m_{i+1}) + m_n + \sum_{i=1}^{a+1} b_i
\]  

(1)

Use the above formula (1) to calculate the number of unknown parameters that need to be optimized, and ensure that the above parameters are optimized during DNN training, so as to ensure the prediction accuracy of the DNN model. In order to obtain the best prediction effect, this article uses the PSO dual optimization algorithm to train the DNN network. In the implementation process, the first step is to use the discrete particle swarm algorithm to determine the DNN architecture, and the second step is to use the particle swarm algorithm to determine the DNN \((w_i, b_i)\), after optimization, a complete DNN model is finally constructed.

2.2. Model specific construction

2.2.1. Confirm the structure of DNN

According to the above analysis, the structural parameters of the DNN network mainly include the number of hidden layers and the number of neurons inside, confirming that the structure of the DNN is actually to solve the discretization problem. In this regard, this article quotes the B-PSO algorithm for calculation, and uses the update iteration formula of the B-PSO algorithm to update the particle state, and establishes the mapping relationship between the DNN structure and the B-PSO particle state. If the particle position is 1, then 2 hidden layers or 2 neurons in the DNN network are connected; if the particle position is 0, there is no connection between the 2 hidden layers or 2 neurons in the DNN network. According to the above judgment mechanism, the number of hidden layers and neurons in the DNN network is tested.

2.2.2. Using PSO to optimize the weights and thresholds of DNN

The PSO algorithm first retrieves random particle positions in different solution spaces, then compares and analyzes the fitness of different particle positions, and finally takes the particle position with the highest fitness as the optimal solution. Establish the mapping relationship between the weights and thresholds of each hidden layer of the DNN and the states of all particles in the PSO algorithm, and only need to update the particle states to achieve the effect of updating the DNN network parameters. In the PSO algorithm, each particle corresponds to its own fitness. After determining the two optimal positions (pbest, gbest), iterative operations are performed to reduce the error, thereby confirming the optimal solution. Compared with the B-PSO algorithm, although the PSO algorithm uses the speed update formula of the B-PSO algorithm, its location update company deletes the probability constraint conditions, indirectly strengthens the competition-coordination relationship between particles, the overall multi-space search function is improved, which can better solve the optimization problem of continuous space.

When solving the structural parameters of the DNN network structure, this paper sets the mean square error \(F\) between the real value of the load and the predicted value as the fitness of the PSO particles, and the formula is as follows:

\[
F = \frac{1}{n} \sum_{i=1}^{N} \left( \frac{\overline{y}(i) - y(i)}{y(i)} \right) \times 100
\]

(2)

Among them, \(\overline{y}(i), y(i)\) are the predicted value and actual value of the load in turn, and \(N\) refers to the number of test samples.

2.2.3. Coding method
In order to simplify the operation, this paper chooses NATLAB software for load forecasting. In terms of program coding, this paper adopts binary coding method to encode DNN structural variables, and real value coding method to encode the transformation parameters of the network transmission process.

The first step is to initialize the random particles to construct the mapping relationship between the dimensional space of the random particles and the DNN transfer weight. Customize the initial state of each particle and confirm the connection variables between random particles. Afterwards, the real number set is updated to confirm the position information of each particle and ensure that the particle state parameter is within the weight range.

The second step is to use the B-PSO algorithm to reverse-encode the state of the encoded particles, thereby constructing a mapping of the network structure, and confirming the final particle position in combination with the state transition probability. If the confirmation result is 0, delete the connection and its hidden layer; if the confirmation result is 1, keep the hidden layer. After confirming the number of hidden layers, perform the same operation on other neurons according to the above coding process, and finally obtain the number of neurons in different hidden layers.

The third step is to input randomly established training samples into the DNN model, and use equation (2) to solve the particle fitness. If the calculated fitness value fails to meet the accuracy standard, the particle state needs to be updated, and so on until the accuracy requirement is met. After this training process, the optimal particle position is finally confirmed. According to the established mapping relationship, the weight threshold of the DNN network can be obtained according to the particle state variables.

In summary, this paper uses the B-PSO algorithm to optimize the DNN structure parameters, and simultaneously uses the PSO algorithm to optimize the DNN weight threshold, which improves the overall load prediction accuracy of the DNN network. The test results show that the method in this paper is practical and effective.

The PSO dual optimization process of the DNN prediction model is shown in Figure 2 below.
3. Calculation example verification

3.1. Forecast data processing

In order to test the prediction accuracy of PSO-DNN, this paper selects the historical power load data of a certain place within 30 days as a sample, and at the same time collects the meteorological data of the same period as the influence parameter, and carries out simulation research on this basis. In terms of data processing, if 15 minutes is set as the unit time for sample collection, there are 96 sample collection nodes in one day. The input dimension is set to 25 dimensions, including 9 dimensions of load data and 16 dimensions of weather conditions, the output dimension is set to 1, and the output is the predicted value at the target time. In addition, this article categorizes weather conditions into three categories, and sets the impact factors of various weather conditions, namely: the impact factor of “sunny” is 1, the impact factor of severe weather represented by rain and snow is 0, The impact factor of sky or cloudy is 0.5.

Considering that the dimensions of different input variables are different, for example, the dimension of temperature is °C, and the dimension of load is W, so it is necessary to perform normalization processing on each input quantity, and transform each input quantity into the amount of data within [0,1].

The input is normalized, and the formula is as follows:

$$\hat{x} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Among them, x represents the original data; $\hat{x}$ represents the result of the normalization process,
and the value interval is \([0,1]\); \(x_{\text{max}}\) and \(x_{\text{min}}\) are the maximum and minimum values in the same dimension data in sequence.

### 3.2. Forecast results and error analysis

To test the prediction effect of the DNN network, this article retrieves the load time series within 10 days of a certain place, which covers working days and rest days. The above data is used to construct training samples, and then the PSO-DNN prediction model is used for working days and rest days, forecast the daily load data. After the prediction is finished, the difference between the predicted value based on PSO-DNN and the predicted value based on RBF-NN or traditional BP-NN is compared and analyzed.

In order to ensure comparability, the structural parameters of the traditional BP-NN model and RBF-NN model are set to 25-10-1. In addition, other parameters of the traditional BP-NN network and their values are: the target error is 0.02, the learning rate is 0.05, the momentum factor is 0.95; the other parameters of the RBF-NN model and their values are: the target error is 0.02, the RBF distribution density is 2.0, and the maximum number of neurons is 960; the other parameters of the PSO-DNN model and their values are: the fitness function error is 0.01, the maximum number of hidden layers is 10, the maximum number of neurons is 50, the learning factor is 1.4962, the random particle is 40, the iteration peak is 200, the maximum and minimum speeds are respectively 0.4, -0.4. After completing the simulation training, the number of hidden layers corresponding to the best prediction result is 3, so the network structure parameter is 25-34-30-26-1.

The above three prediction models are used to predict the load data on working days and rest days. The average absolute error and maximum relative error of the different models are summarized in the following table 1:

| Date        | \(E_{\text{MAPE}}\)/% | \(E_{\text{MAPE}}\)/% |
|-------------|------------------------|------------------------|
| BP-NN       | RBF-NN                 | PSO-DNN                | BP-NN       | RBF-NN                 | PSO-DNN |
| Working day | 2.85                   | 2.54                   | 2.02       | 8.59                   | 9.20    | 5.89    |
| Off day     | 3.14                   | 2.97                   | 2.13       | 10.63                  | 11.31   | 7.32    |

Based on the analysis in Table 1 above, it can be found that the three forecasting models can predict the power load on weekdays more accurately, but the prediction accuracy for rest days is slightly lower. This is mainly because the load fluctuation on rest days is significantly higher than that on working days, thereby increasing the difficulty of prediction. In terms of forecasting the power load on weekdays, the forecast accuracy of the above three forecasting models is compared. Among them, the average absolute error and maximum relative error of PSO-DNN are 2.02% and 5.89%, respectively, and the average absolute error of RBF-NN, the maximum relative error is 2.54%, 9.20%, the average absolute error of traditional BP-NN, the maximum relative error is 2.85%, 8.59%, according to which the prediction accuracy of PSO-DNN is the highest, RBF-NN is the second, and traditional BP-NN has the worst prediction accuracy. Compared with the traditional BP-NN model and RBF-NN model, the average absolute error of the PSO-DNN model decreased by 0.83%, 0.53%, and the maximum relative error of the PSO-DNN model decreased by 2.70%, 3.31%, which is sufficient. It is confirmed that the PSO-DNN model established in this paper is significantly better than the traditional BP-NN model and RBF-NN model in terms of load prediction accuracy.

### 4. Conclusions

According to the analysis of the above results, the important drawback of the traditional network is that the calculation process is easy to fall into the shackles of solving the local optimal solution, and the nonlinear structure of the DNN network sends the input information to the hidden layer, and the PSO algorithm in the hidden layer optimizes the input information, and progresses layer by layer in
this way, and finally improves the accuracy of load forecasting. The test results confirm the feasibility and superiority of the method proposed in this paper, which provides a powerful tool for the actual power system load forecasting work and is beneficial to improve the power system management effect. However, the load sequence is a typical dynamic nonlinear time sequence, and the DNN model lacks a feedback mechanism and cannot accurately reveal the dynamic characteristics of the load sequence, which directly weakens the prediction accuracy of the DNN network. Increasing the number of hidden layers and neurons of the DNN network can solve the above problems, but this is at the cost of extending the program running cycle. Follow-up research should focus on overcoming the above-mentioned problems, and on the premise of ensuring prediction accuracy, greatly reducing the program running cycle, thereby promoting the popularization and application of DNN networks.

References

[1] Yang Zeng, Ding Shiyin, Ye Meng, Li Jing, Xue Shuqian, Wu Haotian. Short-term Power Load Forecasting Model Based on Variational Modal Decomposition and Deep Learning [J/OL]. Electrical Measurement and Instrumentation: 1-8 [2020-12-16].

[2] Jeong Dongyeon, Park Chiwoo, Ko Young Myoung. Short-term Electric Load Forecasting for Buildings Using Logistic Mixture Vector Autoregressive Model with Curve Registration [J]. Applied Energy, 2021, 282(PB).

[3] Liu Hui, Li Houjun, Liu Yuwei, Zou Qixiao. Power Load Forecasting Method Based on VMD and GWO-SVR [J]. Modern Electronic Technology, 2020, 43(23): 167-172.

[4] Lu Wei, Hu Junjian. Research on Yangzhou Electric Power Load Forecast Based on Neural Network Model [J]. Shanghai Energy Conservation, 2020(11): 1279-1283.

[5] Wang Guojuan, Leng Jianwei. Short-term Power Load Forecasting Based on ESMD-PE and ADBN [J/OL]. Electrical Measurement and Instrumentation: 1-8 [2020-12-16].

[6] Xiao Bai, Xiao Zhifeng, Jiang Zhuo, Zhao Xu, Kan Zhongfeng, Qi Xuesong, Bai Yiran. Space Power Load Situation Awareness Based on Noise Reduction Autoencoder, Singular Spectrum Analysis and Long-Short-Term Memory Neural Network [J/OL]. Proceedings of the Chinese Society for Electrical Engineering: 1-10 [2020-12-16].

[7] Zhang Xu, Liu Wenjun, Wang Jianyuan, Dong Ming, Yang Wenyu, Shi Ning, Guo Xin. Big Data Analysis of Short-Term Power Load in Summer School in Shaanxi During the New Crown Epidemic [J]. Power System and Clean Energy, 2020, 36(11): 100-105.

[8] Jiang Shan, Zhou Qiupeng, Dong Hongchuan, Ma Xu, Zhao Zhenyu. Long-term Power Load Combined Forecasting Method Considering Periodicity and Trend Characteristics of Data [J/OL]. Electrical Measurement and Instrumentation: 1-10 [2020-12-16].

[9] Yuan Feng, Wang Xusheng, Liang Yujia. Research on Power Load Forecasting Method Based on DBN-ELM [J]. Science and Technology Innovation, 2020(33): 1-5.

[10] Nie Ying, Jiang Ping, Zhang Haipeng. A Novel Hybrid Model Based on Combined Preprocessing Method and Advanced Optimization Algorithm for Power Load Forecasting [J]. Applied Soft Computing, 2020, 97 (prepublish).