Big data analysis of smart grid based on back-propagation neural network algorithm in Hadoop environment

Big data analysis of smart grid based on back-propagation neural network algorithm in Hadoop environment

Linli Fan, Changmao Li, Zhenping Lan and Li Liu[1]

School of Information Science and Engineering, Dalian Polytechnic University, Dalian, China.

[1] E-mail: liu.li@dlpu.edu.cn

Abstract. Smart micro-grid, as an important way for the best use of renewable energy, can help to solve the problem of energy crisis. However, smart micro-grid has limitations to the impact of natural geographical environment conditions, which increases the instability of productions. In this paper, the back-propagation neural network algorithm is used to analyse the data of wind, and the future wind speed is forecasted through the established network model, experimental data is stored and managed in Hadoop framework. The accuracy of predicted results is presented by Relative Error (RE).

1. Introduction

Modern society has been facing an energy crisis due to increasing demand of unsustainable energy [1]. People will also confront with the climate problems caused by excessive energy consumption. In order to protect the environment and improve energy security as well as helping to solve the growing electricity demand, more renewable energy should be used to generate electricity [2].

Nowadays, with the transforming from conventional grid system to the smart grid [3], more renewable energy productions are used in the systems, meanwhile generation costs is reduced [4]. Smart grid is a miniature power system which can be controlled and managed in distributed generation [5]. Smart grid can identify user needs and use renewable energy to achieve productions. Intelligent energy management and allocation are carried out through energy management system and transmission and distribution system to achieve the optimal operation and control effect. However, under the influence of natural geographical environment, the capacity of renewable energy production is instability. This not only brings risk to the operation of the whole micro-grid, but also makes the energy control of the micro-grid become difficult. At the same time, with the rapid development of smart micro-grid, a large number of data information has been generated in the operation process. Therefore, it is necessary to deeply carry out analysis and application of renewable energy data.

This article selects wind energy data as the representative of renewable energy. As one of the energy source for power generation [6], it is renewable, abundant and pollution-free [7]. The wind energy data is stored and managed in Hadoop framework. The MapReduce data processing is designed according to the characteristics of large data distributed process. Back-Propagation (BP) neural network algorithm is used to predict the wind speed in the future, so as to eliminate the impact of natural geographical environment on the stability of renewable energy production capacity.

2. Hadoop distributed platform

Hadoop is distributed system with open source architecture. It uses multiple computers to process large amounts of data in parallel. Once a distributed architecture is formed, many devices will work together, greatly reducing data processing time. The MapReduce process can be designed according to its application. The kernel module of Hadoop contains two parts, one is Hadoop Distributed File System (HDFS) and the other is MapReduce.

HDFS is used to store and manage files through a unified namespace. Each machine in the Hadoop cluster has HDFS for storing massive data. It has high fault tolerance and provides high throughput capacity data access. The client visits HDFS by accessing the operating system files like windows or Linux. Figure 1 shows the structure of HDFS consisting of a NameNode and many DataNodes.

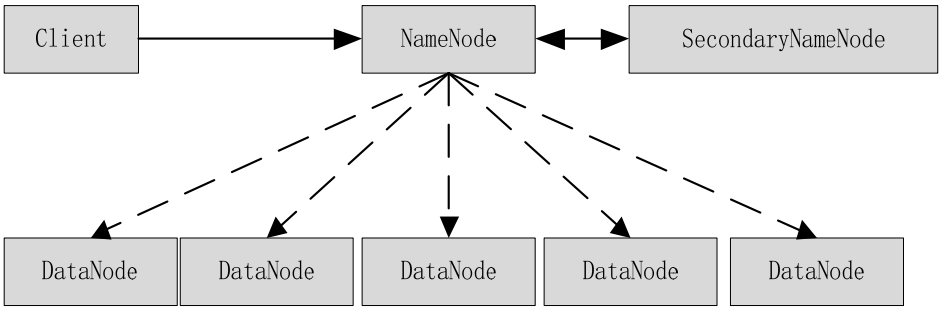

Figure 1. The structure of HDFS.

The upper level MapReduce of HDFS is responsible for distributed processing of massive data, including JobTracker and TaskTracker. It can be used for analysis and statistics large amounts of data on HDFS and is a distributed computing framework provided by Hadoop for data processing. In distributed computing, the MapReduce framework parallel programming complex problems such as distributed storage, job scheduling, load balancing, fault-tolerant balancing, fault-tolerant processing, and network communication. The process can be highly divided into two functions: Map and Reduce. Map is used for decomposing the overall into multiple tasks. In contrast, Reduce is responsible for aggregating the results of multitasking. Figure 2 shows the structure of MapReduce.

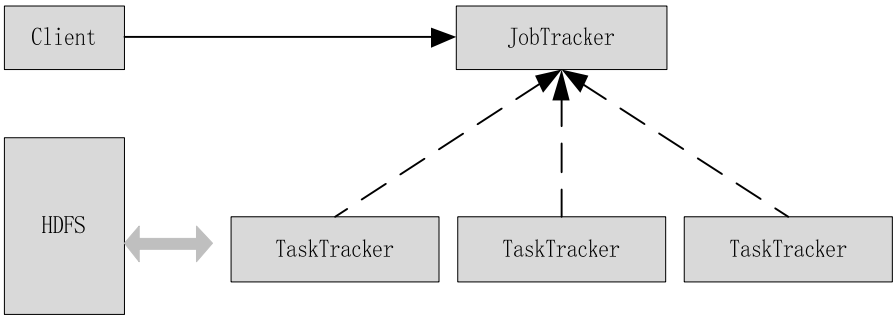

Figure 2. The structure of MapReduce.

In this paper, Hadoop distributed framework is applied to large data processing and analysis of micro-grid system. Large amounts of wind energy data are stored and read through the HDFS. The Map process is used to read data and classify it into four categories by collection time, including spring, summer, autumn and winter. Different classes are assigned to different nodes for BP neural network training. Multiple nodes run at the same time, and then the results of each run are sent to the Reduce process. Reduce merges data that from different Maps process together for final processing, resulting in a new data block.

3. Back-propagation neural network algorithm

The back-propagation neural network is a multi-layered nonlinear feed-forward network trained by the back-propagation learning algorithm [8]. Data normalization is necessary before using BP neural network algorithm to train data, because the magnitude of the data is inconsistent. Normalization can eliminate the inaccuracy of prediction caused by different dimensions. The min-max standardization is used in this paper, which assigns the data level to the interval of [0,1].

$$x_i = \frac{(x - x_{max})}{(x_{max} - x_{min})} \quad (1)$$

In the formula, x is primary data, x_{max} and x_{min} are the maximum and minimum values in data.

Figure 3 shows the neural network consists of three layers, including input layer, hidden layer and output layer [9].

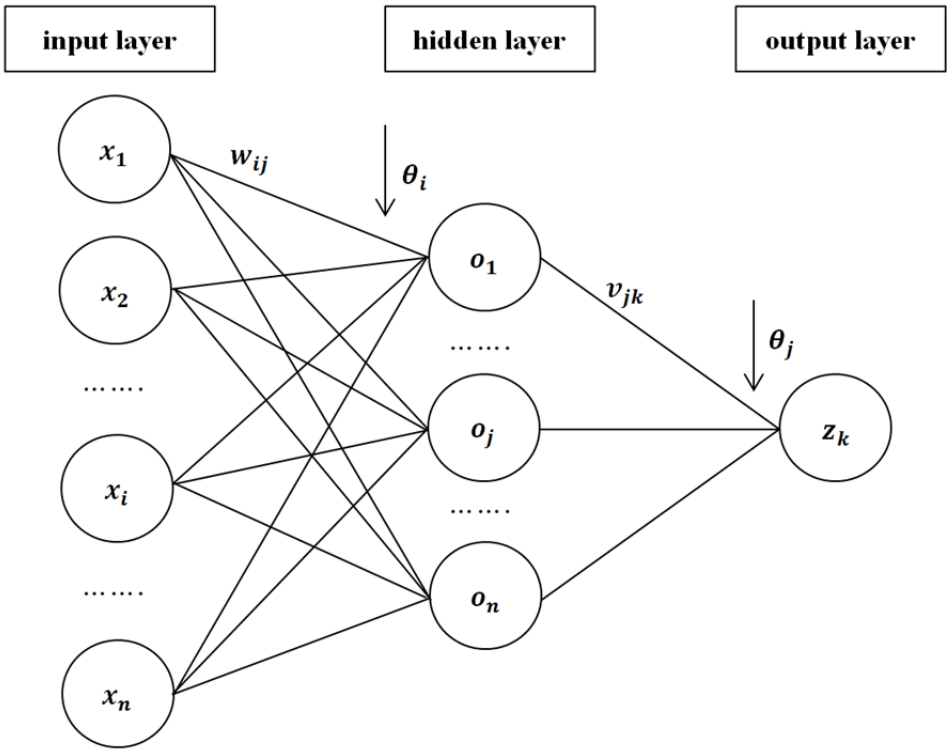

Figure 3. Basic structure of back-propagation neural network.

The training process of neural network is continuous learning process from the samples. The purpose of learning is to get a smaller prediction error. The training is ended by limiting the number of iterations or satisfying the termination condition of learning. The algorithm learning process includes forward direction and back-propagation processes [10]. The output value is obtained by forward propagation, and then the error is propagated back to calculate the weight adjustment value until the output is acceptable.

The forward propagation stage means that the sample information starts from the input layer, from top to bottom through the hidden layer node calculation processing. The output of the upper layer node is the input of the lower layer, and the final sample information is propagated to the output layer node to get the predictions. The activation function can be applied to obtain more accuracy in the predicting process. Sigmoid activation function is used to activate the output and get the prediction result, which limits the output of nodes to 0-1 range, and it can better reflect the gradual transformation process from approximate linearity to non-linearity in the process of network weight correction. The activation function formula:

$$f(x) = \frac{1}{1+\exp(-x)} \tag{2}$$

The output value of hidden layer o_j in Figure 3 is calculated by multiplying all of the input nodes x_i and weight w_{ij} between input and hidden layer, in addition add the threshold value θ_i.

$$y_j = \sum_{i=1}^{n} (w_{ij}x_i) + \theta_i = w_{1j}x_1 + w_{2j}x_2 + \cdots + w_{ij}x_i + \cdots + w_{nj}x_n + \theta_i \tag{3}$$

$$o_j = f(y_j) = \frac{1}{1+\exp(-y_j)} \tag{4}$$

Next, the hidden layer is used as input. The export value of output layer z_k is obtained by input of hidden layer o_j with weight v_{jk} between hidden and output layer, and threshold value θ_j.

$$o_k = \sum_{j=1}^{n} (v_{jk}o_j) + \theta_j = v_{1k}o_1 + v_{2k}o_2 + \cdots + v_{jk}o_j + \cdots + v_{nk}o_n + \theta_j \tag{5}$$

$$z_k = f(o_k) = \frac{1}{1+\exp(-o_k)} \tag{6}$$

In back propagation stage, the prediction value of output node is propagated to the hidden and the input layer node in the opposite direction. The weights are adjusted layer by layer until all the weights are adjusted. Whether the training of neural network is accomplished or not is commonly measured by using the error function. The error function is calculated from the actual value z_i and the predicted value z_k.

$$E = \frac{1}{2}(z_i - z_k)^2 \tag{7}$$

If the termination conditions are not satisfied, the error is propagated backwards to the hidden layer and continues to the input layer.

$$E = \frac{1}{2}(z_i - z_k)^2 = \frac{1}{2}(z_i - f(o_k))^2 = \frac{1}{2}\left(z_i - f\left(\sum_{j=1}^{n}(v_{jk}o_j) + \theta_j\right)\right)^2 \tag{8}$$

$$E = \frac{1}{2}\left(z_i - f\left(\sum_{j=1}^{n}(v_{jk}o_j) + \theta_j\right)\right)^2 = \frac{1}{2}\left(z_i - f\left(\sum_{j=1}^{n}\left(v_{jk}f(y_j)\right) + \theta_j\right)\right)^2$$
$$= \frac{1}{2}\left(z_i - f\left(\sum_{j=1}^{n}\left(v_{jk}f\left(\sum_{i=1}^{n}(w_{ij}x_i) + \theta_i\right)\right) + \theta_j\right)\right)^2 \tag{9}$$

The weight v_{jk} and w_{ij} is adjusted according to the error function, and weights of the next round t+1 are obtained for the weights of the round t and the weight adjust value. The negative sign in the expression indicates gradient descent, and constant η shows the coefficient of proportionality.

$$\Delta w_{ij} = -\eta\frac{\partial E}{\partial w_{ij}} = -\eta\frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial w_{ij}} = -\eta\frac{\partial E}{\partial y_j}x_i \tag{10}$$

$$\Delta v_{jk} = -\eta\frac{\partial E}{\partial v_{jk}} = -\eta\frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial v_{jk}} = -\eta\frac{\partial E}{\partial o_k}o_j \tag{11}$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \tag{12}$$

$$v_{jk}(t+1) = v_{jk}(t) + \Delta v_{jk} \tag{13}$$

After repeated iterations of the same sample data, the training is stopped until the convergence criterion is reached, so that the predicted value of network output approximates to the actual value.

Under the Hadoop framework, the normalized data is read in the Map phase, which is trained by BP neural network to adjust the weight output prediction value. BP neural network model is built to find the connection between the attributes through the training set data. Then, use the test set data to evaluate the established model. The data are integrated by Reduce process, and the final result can be saved in HDFS.

4. Results and discussion

In this paper, the wind energy data was divided into four categories by collection time including spring, summer, autumn and winter. BP neural network training was carried out in each category. Besides, in order to effectively judge the performance of the prediction model, it is necessary to make a separate dataset that without participate training in advance. Therefore, some data should be stored into testing sets and all the original data are randomly distributed into these two sets by percentage of 80% and 20% separately.

The results of the new data block from Reduce process is exported from HDFS. The data file contains the actual and the predicted values. In the end, Relative Error (RE) is employed to determine the prediction results. The RE refers to the ratio of Absolute Error (AE) to actual value z caused by prediction. AE is calculated by subtracting the predicted value z_k from the actual value z. Calculating formula for solving RE is:

$$\mathrm{RE} = \frac{z - z_k}{z} \times 100\% \tag{14}$$

The calculated RE was divided into 10 classes from ascending sort. Figure 4 shows the frequency distribution of RE.

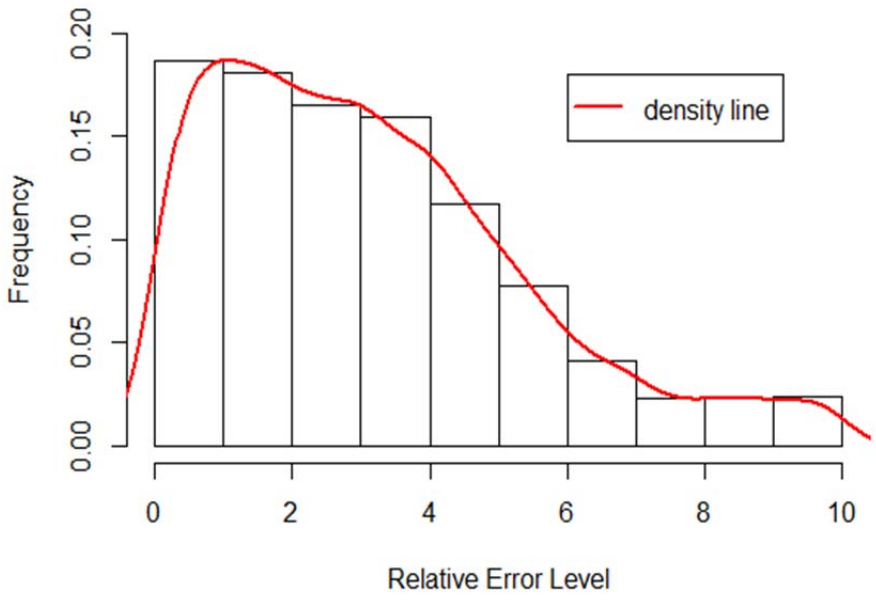

Figure 4. The frequency distribution of Relative Error.

The horizontal coordinate represents the 10 levels of RE, and vertical denotes the frequency distribution of RE. The reference line is decreasing which the frequency distribution trend line. As shown in Figure 4 that the relative error between actual and predicted data is low. The total frequency of RE is approximate 0.7 for the relative error levels less than 4. When the RE level is greater than 6, the frequency is very small, almost close to zero.

5. Conclusions

In this paper, BP neural network algorithm is applied to predict the data of wind speed in Hadoop framework. The results indicate that the historical wind data are successfully used to predict the future wind speed data, and the error between the actual data and the predict data is low, which provides theoretical basis for the safe operation of the smart micro grid. In the future work, further predictive methods will be investigated to solve the problem of energy input instability.

Acknowledgment
Authors thank for the financial support by the fundamental research funds for educational committee of Liaoning, grant number 2016J064.

References

[1] Yeliz Yolda, Ahmet Önen, S M Muyeen, Athanasios V Vasilakos, İrfan Alan 2017 Enhancing smart grid with microgrids: challenges and opportunities *Renewable & Sustainable Energy Reviews* 72
[2] A Ali, W Li, R Hussain, X He, B Williams 2017 Overview of current microgrid policies, incentives and barriers in the european union, united states and china *Sustainability* **9** 7
[3] P Singh, B Khan, P Singh, B Khan 2017 Smart microgrid energy management using a novel artificial shark optimization *Complexity* **2017** 1
[4] Azmi Hashim, Kwok L Lo 2018 Thermal effect of wind generation on conventional generator in a microgrid *Indonesian Journal of Electrical Engineering and Computer Science* **10** 3
[5] PD Diamantoulakis, VM Kapinas, GK Karagiannidis 2015 Big data analytics for dynamic energy management in smart grids *Big Data Research* **2** 3
[6] J Feng, WZ Shen 2015 E sciubba modelling wind for wind farm layout optimization using joint distribution of wind speed and wind direction *Energies* **8** 4
[7] AA Abdullah, AE Saleh, MS Moustafa, KM Abo-Al-Ez 2014 A proposed framework for a forecasting system of wind energy power generation *International Journal of Advanced Research in Computer Engineering & Technology* **3** 4
[8] J Mccullagh 2010 Data mining in sport: a neural network approach *Internation Journal of Sports Science and Engineering* **4** 3
[9] Biao X 2012 Prediction of sports performance based on genetic algorithm and artificial neural network *International Journal of Digital Content Technology and its Applications* **6** 22
[10] J Wang 2014 BP neural network-based sports performance prediction model applied research *Journal of Chemical and Pharmaceutical Research* **6** 7