Short-Term Load Forecasting Model Based on Multi-label and BPNN

Xiaokui Sun(✉), Zhiyou Ouyang, and Dong Yue

Institute of Advanced Technology,
Nanjing University of Posts and Telecommunications, Nanjing 210046, China
260602812@qq.com

Abstract. With the rapid development of smart grid, the importance of power load forecast is more and more important. Short-term load forecasting (STLF) is important for ensuring efficient and reliable operations of smart grid. In order to improve the accuracy and reduce training time of STLF, this paper proposes a combined model, which is back-propagation neural network (BPNN) with multi-label algorithm based on K-nearest neighbor (K-NN) and K-means. Specific steps are as follows. Firstly, historical data set is clustered into *K* clusters with the K-means clustering algorithm; Secondly, we get *N* historical data points which are nearest to the forecasting data than others by the K-NN algorithm, and obtain the probability of the forecasting data points belonging to each cluster by the lazy multi-label algorithm; Thirdly, the BPNN model is built with clusters including one of *N* historical data points and the respective forecasting load are given by the built models; Finally, the forecasted load of each cluster multiply the probability of each, and then sum them up as the final forecasting load value. In this paper, the test data which include daily temperature and power load of every half hour from a community compared with the results only using BPNN to forecast power load, it is concluded that the combined model can achieve high accuracy and reduce the running time.

Keywords: BP neural network · K-means algorithm · K-nearest neighbor · Multi-label · Short-term load forecasting

1 Introduction

With the trend of smart grid, it is very important to forecast the user-side power load. High quality and accuracy load forecasting is essential in the smart grid [1, 2]. In recent years, the residents ladder price is implemented by the State promoting to save electricity and use electricity reasonably. The smart grid of China starts too late and the relatively technology is backward and it results in a number of power supply and demand imbalance [3]. In this case, the state encourages the power plants to generate more power. On the one hand, it meets the electricity consumption in some power-needed areas. On the other hand, it also leads to a lot of electricity vastly waste in many areas.

STLF usually refers to the load demand for the next 24 h. Its methods include time series analysis [5, 6], grey theory method [7] and neural network method [8]. Among

M. Fei et al. (Eds.): LSMS/ICSEE 2017, Part I, CCIS 761, pp. 263–272, 2017.
DOI: 10.1007/978-981-10-6370-1_26

them, an effective mathematical model of the time series analysis method is difficult to be built and the prediction accuracy is generally not high. Grey prediction model is theoretically applicable to any nonlinear load forecasting and the differential equation is suitable for load forecasting with exponential growth trend, but it is difficult to improve the accuracy of fitting gray with other indicators of trend. The neural network usually trained by back-propagation (BP) of errors is one of the popular network architecture in use nowadays [9]. BPNN, due to its excellent ability of non-linear mapping, generalization and self-learning, has been proved to be widely used in engineering optimization field [10]. Due to too much training data, BPNN training time is too long [11]. Because the BP algorithm is essentially a gradient descent method, and the objective function to be optimized is very complex, therefore, there will be a "saw-tooth" phenomenon, which makes the BP algorithm inefficient [11]. Nowadays, in the trend of big data, we usually deal with mass of data. Therefore, we should minimize the size of the training data without losing effective data before using the BPNN. [13] uses the BPNN with K-means clustering algorithm to forecast power load. It regards the center of clustering as the label and just train the BPNN model with the one cluster. So, it is possible to lose valid information in other clusters.

In this paper, our contribution is to propose the BPNN with multi-label algorithm based on K-NN and K-means clustering algorithm. We give different weight of each cluster for the forecasting points by Multi-label algorithm based on K-NN and K-means proposed by us and build models by BPNN. This method can avoid losing valid information of other clusters. Compared with the traditional BPNN algorithm, the combined algorithm can obtain better performance in accuracy and running time.

This paper is organized as follows. The introduction is described in Sect. 1. In Sect. 2, the BPNN is discussed. BPNN with multi-label algorithm based on K-NN and K-means is given in Sect. 3. The experiment on real data are given in Sect. 4. In Sect. 5, we draw conclusions and give insightful discussions.

2 BP Neural Network

The BP algorithm [14, 15] uses the gradient of the performance function to determine how to adjust the weights to minimize errors that affect performance.

Figure 1 shows a multi-layer feedforward network structure with d input layer neurons, l output layer neurons, q hidden layer neurons, where the threshold value of the j th neurons in the output layer is represented by θ_j, The threshold of the h th neurons is denoted by γ_h. The connection weight between the i th neurons of the input layer and the h neurons of the hidden layer is v_{ih}, and the connection weight between the h th neurons of the hidden layer and the j th neurons of the output layer is w_{hj}. $\alpha_h = \sum_{i=1}^{d} v_{ih}x_i$ represents the input received by the h th neurons in the hidden layer, and $\beta_j = \sum_{h=1}^{q} w_{hj}b_h$ represents the input received by the j th neurons in the output layer, where b_h is the output of the h th neurons in the hidden layer.

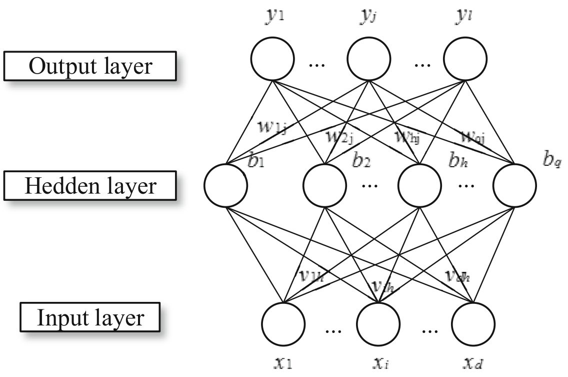

Fig. 1. Architecture of BP neural network

There are many forms for activation function, but the most popular is sigmoid function. In this paper, we use the sigmoid function as output function of the hidden neuron and the output neuron. The sigmoid function is as follows,

$$f(x) = \frac{1}{1 - e^{-x}} \tag{1}$$

For training sample (x_k, y_k), assuming that $\hat{y} = (\hat{y}_1^k, \hat{y}_2^k, \ldots, \hat{y}_l^k)$ is the output of the neural network, where

$$\hat{y}_j^k = f(\beta_j - \theta_j) \tag{2}$$

So we can obtain the mean square error E_k of the neural network in training sample,

$$E_k = \frac{1}{2} \sum\nolimits_{j=1}^{l} \left(\hat{y}_j^k - y_j^k\right)^2 \tag{3}$$

The neuron gradient term is

$$g_j = \hat{y}_j^k (1 - \hat{y}_j^k)(y_j^k - \hat{y}_j^k) \tag{4}$$

The updating formula of w_{hj} is

$$\Delta w_{hj} = \eta g_j b_h \tag{5}$$

where η is the leaning rate.

The updating formula of θ_j is

$$\Delta \theta_j = -\eta g_j \tag{6}$$

The updating formula of v_{ih} is

$$\Delta v_{ih} = \eta e_h x_i \tag{7}$$

where $e_h = b_h(1-b)\sum_{j=1}^{l} w_{hj} g_j$, x_i is the input data.

The updating formula of γ_h is

$$\Delta \gamma_h = -\eta e_h \tag{8}$$

BPNN is an iterative learning algorithm that arbitrary parameter v update that are

$$v \leftarrow v + \Delta v \tag{9}$$

And the iterative process of w, θ and γ are the same to v. Updating w and v will consume a lot of time, and we reduce the running time by the mean of the combined model.

The goal of the BPNN is to minimize the cumulative error of the historical set. During the training process, historical set that differ greatly from the predicting load are involved in the training model, which not only increase the training time but also reduce the accuracy of the prediction. Therefore, in this paper, we introduce a method to reduce the impact of these irrelevant data sets.

3 BPNN with Lazy Multi-label Algorithm Based on K-NN and K-means

3.1 K-means Clustering Algorithm

K-means [16] is an unsupervised learning algorithm, and it is an algorithm for clustering data points by the mean values. The K-means algorithm divides similar data points into K clusters, that the K is a pre-setting value, and each cluster has an initial center which is selected from data set randomly. In this paper, Euclidean distance is used to represent the similarity between data. For instance, the Euclidean distance of $\vec{X} = (x_1, x_2, \ldots, x_n)$ and $\vec{Y} = (y_1, y_2, \ldots, y_n)$ is

$$d_{XY} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{10}$$

K-means algorithm general steps:

Step 1: select K data from the data set as the initial cluster center;

Step 2: calculating the Euclidean distance between each data and K cluster centers and dividing data into the nearest cluster center;

Step 3: recalculating the cluster center by mean values;

Step 4: calculating the standard measure function if it reaches the maximum number of iterations, and then stop, otherwise go to step 2.

After the iteration is completed, we can get K clusters which high similarity within cluster and low similarity between clusters. In this paper, K-means algorithm is used to divide the data set into K clusters. Then, these clusters are regarded as original labels which will be used in the next step.

3.2 Lazy Multi-label Algorithm Based on K-NN and K-means

In Sect. 3.1, the historical data set is clustered into K clusters. In [17], the cluster center is regarded as feature of each cluster, and the forecasting load are divided into the cluster whose cluster center is the shortest from forecasting load. Because the K-means clustering algorithm is an unsupervised learning algorithm, we cannot get the specific characteristics of each cluster. [17] will lose some information of other clusters. We solve this problem by the lazy multi-label algorithm based on K-NN and K-means.

ML-KNN, i.e. multi-label K-NN, was proposed in [17]. It is mainly used for pattern recognition. On this basis, we put forward a lazy multi-label algorithm based on K-NN and K-means. Specific description is following.

Define the result of K-means as.$C = \{\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_i, \ldots, \mathrm{C}_\mathrm{k}\}$, where C_i represents the i th cluster.

Step 1: compute and finding N train points which are the nearest distance from the forecasting load through the K-NN algorithm. The distance is computed by formula (10).

Step 2: count that each cluster contains the number of N training points and the result is $\vec{n}$, where $\vec{n} = (n_1, n_2, \ldots, n_i, \ldots, n_k)$, and n_i represents the number of N training points in the i th cluster.

Step 3: use lazy multi-label theory, we can obtain a set of weights $\vec{p}$, where $\vec{p} = (p_1, p_2, \ldots, p_i, \ldots, p_k) = (\frac{n_1}{N}, \frac{n_2}{N}, \ldots, \frac{n_i}{N}, \ldots, \frac{n_k}{N})$, and p_i is the probability that the forecasting load belongs to the i th cluster.

We can observe that the proposed algorithm will reduce some training data. We think that the reduced data are low related to the forecasting load and if they are used to build the BPNN model, it abates accuracy of the BPNN.

3.3 BPNN with Multi-label Algorithm Based on K-NN and K-means

As shown in Fig. 2, The date set is clustered into K clusters, such as $\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_i, \ldots, \mathrm{C}_\mathrm{k}$, by the K-means algorithm. In this paper, our historical data include temperature and power load and the temperature of the forecasting points can be known from weather forecasting. Therefore, the temperature of the forecasting points is regarded as the known element of forecasting data. Using the temperature, we can get the weights of each cluster by Multi-label based on K-NN and K-means, such as $p_1, p_2, \ldots, p_i, \ldots, p_k$. If p_i isn't zero, we build BPNN of the C_i and obtain forecasting power load of the C_i which multiply p_i is $Load_i$. Then we sum the forecasting load of each cluster, they are $Load_1, Load_2, \ldots, Load_i, \ldots, Load_k$, and get the forecasting load.

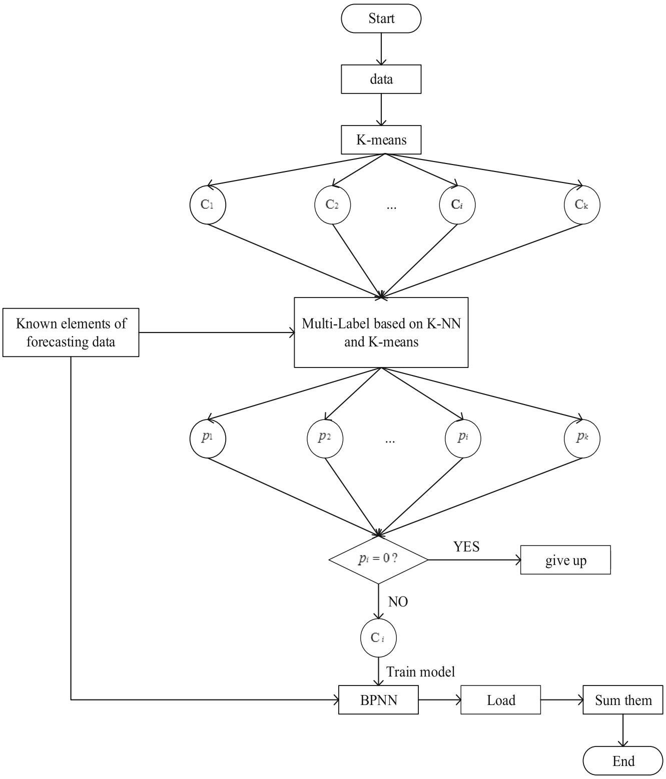

Fig. 2. Flow chart of the combined model

4 Experiment

4.1 Data Preprocessing

In this paper, the historical data set is a real data of a community from January to December. The data set has one temperature per day and have 48 load data points, that is, a load value point is the past half an hour. Temperature is the only information related to the power load in this data sets. We can get the future temperature information from weather forecasting web.

In order to facilitate observation, the community total daily power load is painted as Fig. 3. It can be seen from the Fig. 3 that the seasonal variation of the load is very obvious. The load of May to September compared to other months, there is a significant decline, but the volatility is periodic.

In order to speed up the convergence of clustering, avoid data annihilation and reduce the sensitivity of the singular data to the algorithm, the historical data needs to be normalized. There are many normalized methods, for instance, Min-Max scaling and Z-score standardization. In this paper, we use Min-Max scaling.

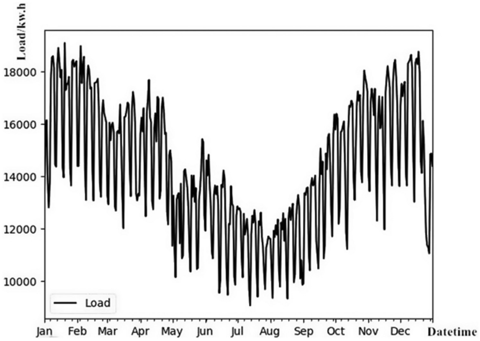

Fig. 3. Total power load

The method realizes the equal scaling of the historical data. The Min-Max scaling is following,

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{11}$$

where X_{norm} is the normalized data, X is the original data, and $X_{\max}$ and $X_{\min}$ are the maximum and minimum of the original data set.

There are two evaluation standards. One is RMSE which is root-mean-square error, the other is MAPE where is mean absolute percentage error.

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n} \left(x_{pi} - x_{ri}\right)^2} \tag{12}$$

$$MAPE = \frac{1}{n}\sum\nolimits_{i=1}^{n} \left|x_{pi} - x_{ri}\right| \times 100\% \tag{13}$$

where x_{pi} is the i th forecasting data, and x_{ri} is the i th real data. In this paper, we use the data of 1–201 days to forecast the load of the 202th day.

4.2 BPNN Experiment

The data is 49 dimensions containing temperature and 48 load points. BPNN is built with one input layer neuron, five hidden layer neurons and 48 output layer neurons. The temperature is the input layer neuron and the 48 load points is the output layer neurons. Then, normalize the data set by formula (11) and set the learning rate to 0.05, the training times to 1000 and the correct to 0.01.

We use the normalized data of 1–201 days to build BPNN model and forecast the load of the 202th day. This article uses python to implement it. The results of the 202th day's load are in Fig. 4 and the RMSE, MAPE and running time are in Table 1.

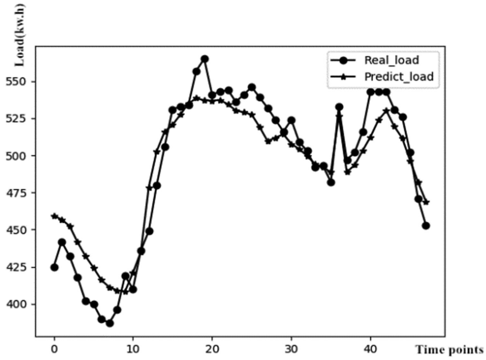

Fig. 4. BPNN forecasting load compared with real load

Table 1. BPNN results

Model	RMSE	MAPE(%)	Running time(s)
BPNN	18.11234	3.32004	1002.28238

4.3 BPNN with Lazy Multi-label Based on K-NN and K-means (BPNN-L-ML-K-K) Experiment

The theory and implementation steps of the algorithm have been introduced in Sect. 3.3 and BPNN is built with one input layer neuron, five hidden layer neurons and 48 output layer neurons. We set the learning rate to 0.05, the training times to 1000 and the correct to 0.01. The temperature is the input layer neuron and the 48 load points is the output layer neurons. We use the normalized data of 1–201 days to build BPNN-L-ML-K-K model and forecast the load of the 202th day.

Firstly, we normalize the data. Secondly, in this paper, the historical data are clustered into five clusters by K-means clustering algorithm, recorded as $C = \{\mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3, \mathrm{C}_4, \mathrm{C}_5\}$. Thirdly, make $N = 60$, and count that each cluster contains the number of 60 training samples and the result is $\vec{n} = (0, 0, 22, 14, 24)$. Fourth, using lazy multi-label theory, we can obtain a set of weights $\vec{p} = \left(\frac{0}{60}, \frac{0}{60}, \frac{22}{60}, \frac{14}{60}, \frac{24}{60}\right) = (0, 0, 0.37, 0.23, 0.40)$. Finally, the BPNN is built with the non-zero weight clusters and the forecasting load is given each other, where the forecasting load is represented by L. The last forecasting load is $L_{total} = L \times \vec{p}^T$. The results of the 202th day's load are in Fig. 5 and the RMSE, MAPE and running time are in Table 2.

4.4 Comparison of Experimental Results

Table 3 shows that the BPNN-L-ML-K-K model performs better than BPNN in RMSE, MAPE and running time. Through the processing of clustering and multi-label algorithm, the historical data with high correlation with the predicting load are used to build the model and get the weights of different clusters. Some lower correlation data are deleted. Thus, the training time is reduced and the efficiency of the model is

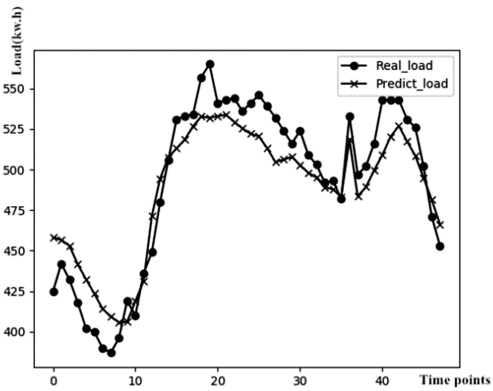

Fig. 5. BPNN-L-ML-K-K forecasting load compared with real load

Table 2. BPNN-L-ML-K-K effect

Model	RMSE	MAPE(%)	Running time(s)
BPNN-L-ML-K-K	16.42211	2.91566	469.55906

Table 3. Models comparison

Model	RMSE	MAPE(%)	Running time(s)
BPNN	18.11234	3.32004	1002.28238
BPNN-L-ML-K-K	16.42211	2.91566	469.55906
Improvement	9.3%	12.2%	53.2%

improved. The experimental results show that BPNN-L-ML-K-K can improve the accuracy of prediction.

5 Conclusion

In view of the characteristics of the STLF, we put forward a BPNN with multi-label based on K-NN and K-means model. Based on the analysis of a community power load, the forecast results show that the new model not only can improve the prediction accuracy, but also can reduce the running time. In the trend of big data, a lot of data has been collected and stored. We should use highly related data for us and delete irrelevant data. In this paper, the combined model can achieve it with better accuracy and less running time.

References

1. Javed, F., Arshad, N., Wallin, F.: Forecasting for demand response in smart grids: An analysis on use of anthropologic and structural data and short term multiple loads forecasting. Appl. Energy **96**, 150–160 (2012)
2. Borges, C.E., Penya, Y.K.: Fernandez I. Evaluating combined load forecasting in large power systems and smart grids. IEEE Trans. Industr. Inf. **9**(3), 1570–1577 (2013)
3. Beccali, M., Cellura, M., Brano, V.L.: Forecasting daily urban electric load profiles using artificial neural networks. Energy Convers. Manag. **45**(18), 2879–2900 (2004)
4. Hagan, M.T., Behr, S.M.: The time series approach to short term load forecasting. IEEE Trans. Power Syst. **2**(3), 785–791 (1987)
5. Papalexopoulos, A.D., Hesterberg, T.C.: A regression-based approach to short-term system load forecasting. IEEE Trans. Power Syst. **5**(4), 1535–1547 (1990)
6. Niu, D.: Adjustment gray model for load forecasting of power systems. IEEE Trans. on PWRS. **5**(4), 1535W–1547 (1990)
7. Chow, T.W.S., Leung, C.T.: Neural network based short-term load forecasting using weather compensation. IEEE Trans. Power Syst. **11**(4), 1736–1742 (1996)
8. Yu, S., Zhu, K., Diao, F.: A dynamic all parameters adaptive BP neural networks model and its application on oil reservoir prediction. Appl. Math. Comput. **195**(1), 66–75 (2008)
9. He, Y., Xu, Q.: Short-term power load forecasting based on self-adapting PSO-BP neural network model. In: IEEE 2012 Fourth International Conference on Computational and Information Sciences (ICCIS), pp. 1096–1099 (2012)
10. Wang, H., Shan, G., Duan, X.: Optimization of LM-BP neural network algorithm for analog circuit fault diagnosis. In: IEEE 2013 International Conference on Information Science and Technology (ICIST), pp. 271—274. Karnataka (2013)
11. Huang, H.C., Hwang, R.C., Hsieh, J.G.: A new artificial intelligent peak power load forecaster based on non-fixed neural networks. Int. J. Electr. Power Energy Syst. **24**(3), 245–250 (2002)
12. Comité, Francesco, Gilleron, Rémi, Tommasi, Marc: Learning multi-label alternating decision trees from texts and data. In: Perner, Petra, Rosenfeld, Azriel (eds.) MLDM 2003. LNCS, vol. 2734, pp. 35–49. Springer, Heidelberg (2003). doi:10.1007/3-540-45065-3_4
13. Frean, M.: The upstart algorithm: A method for constructing and training feedforward neural networks. Neural Comput. **2**(2), 198–209 (1990)
14. Lu, Y.W., Sundararajan, N., Saratchandran, P.: Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. IEEE Trans. Neural Netw. **9**(2), 308–318 (1998)
15. MacQueen, J.: Some methods for classification and analysis of multivariate observa-tions. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, pp. 281–297 (1967)
16. Huang, L., Cheng, H., Yi, Q.M.: OLoad forecast based on hybrid model with k-means clustering and BP neural network. Power Energy **1**, 56–60 (2016)
17. Zhang, M.L., Zhou, Z.H.: ML-KNN: A lazy learning approach to multi-label learning. Pattern Recogn. **40**(7), 2038–2048 (2007)