Short Term Load Forecasting Method Based on Full Convolution Deep Learning

Hai-hong Bian[1](✉), Xing-jian Shi[1], Qian Wang[1], and Li-kuan Gong[2]

[1] Nanjing Institute of Technology, Nanjing 211167, China
llmn0002@sina.com
[2] Power Dispatching Control Center of Shenzhen Power Supply Bureau, Shenzhen 440304, China

Abstract. The traditional load forecasting methods can not take into account the time and space characteristics of load data at the same time, which leads to the low application efficiency of load forecasting methods. In order to solve this problem, a short-term load forecasting method based on full convolution deep learning is proposed. Preprocess the power load data, delete the abnormal samples, unify the load data format through normalization processing, design the relevant network parameters, determine the loss function, complete the design of the prediction model, use the sample data to train the prediction model, and predict the short-term power load after the model meets the prediction requirements. The experimental results show that: in the same experimental environment, the short-term power load forecasting method based on full convolution deep learning has high prediction accuracy, wide prediction range, and its application efficiency has been improved.

Keywords: Full convolution deep learning · Short term electricity · Load forecasting · Neural network

1 Introduction

Power system load forecasting is based on the historical data of power sector, economy, policy, meteorology, etc., to explore the influence of the historical data change characteristics of power load on the future power load, and to find the possible relationship between power load and various related factors, so as to correctly predict the future power load. According to the length of forecasting time, power sector load forecasting can be divided into annual forecasting, monthly forecasting and daily forecasting, as well as long-term, medium-term, short-term and ultra short-term load forecasting [1]. The coverage time of long-term forecast varies from the next few years to more than ten years, which is mainly used for power development planning and power grid development planning; Medium term forecast refers to the power load forecast in the next year (12 months), which is used to arrange the overhaul plan and the economic operation of hydropower plants; Short term load forecasting usually refers to the daily load forecasting in the next 24 h and the weekly load forecasting in the next 168 h.

S. Wang et al. (Eds.): IoTCare 2021, LNICST 415, pp. 461–476, 2022.
https://doi.org/10.1007/978-3-030-94182-6_37

The goal is to arrange daily and weekly power generation tasks for each power plant, including the determination of unit startup and shutdown, coordination of thermal power and hydropower, fuel procurement and supply, tie line exchange power, economic distribution of power load, reservoir dispatching and equipment repair, etc.; Ultra short term forecast refers to the forecast of 1 h, 0.5 h or even 10 min in the future, which is mainly used for safety monitoring, preventive control and emergency handling [2–4].

With the development of the power market in the world, people pay more and more attention to the field of load forecasting. Some scholars have made statistics on the works published in IEEE Trans. And other world-famous journals. The results show that people's interest in load forecasting has been increasing year by year since the early 1970s. In the mid-1980s, due to the shortage of energy, there is an urgent need for scientific load management and the expectation of accurate and adaptive load model. It makes the research investment of load forecasting more and more high [5]. Since the 1990s, with the vigorous development of power markets in various countries, power load forecasting has been paid more and more attention by scholars.

Power system short-term load forecasting has been studied for a long time. It is developed with the gradual development of EMS system in power system. With the introduction of data analysis and artificial intelligence technology, people have proposed prediction methods based on different principles. Each of these methods has its own advantages and disadvantages. No method is absolutely superior to the prediction method in any region or at any stage [6]. For short-term load forecasting, it is necessary to fully study the characteristics of power grid load change and analyze the factors related to load change, especially the weather factors, date types and other factors that have a greater relationship with short-term load change [7]. Many literatures have done a lot of research on load forecasting, mostly based on more advanced theory to improve the accuracy of load forecasting, which provides a strong guarantee for the economy and security of power system operation. At present, the research of load forecasting mainly focuses on the improvement of forecasting methods and the proposal of new methods [8]. According to the applied mathematical model and the development process of load forecasting, it can be divided into traditional classical methods and new methods of artificial intelligence technology [9–11]. Classical methods mainly refer to time series models of various statistical theories, while artificial intelligence technology is a modern forecasting method represented by soft computing. After application research, it is found that some traditional prediction methods have poor application efficiency [12–15].

Therefore, a short-term power load forecasting method based on full convolution deep learning is proposed. The full convolution neural network in deep learning is used to obtain the load data in real time, so as to achieve accurate forecasting and solve the problems in the traditional forecasting methods mentioned above.

2 Design of Short-Term Load Forecasting Method Based on Full Convolution Deep Learning

2.1 Power Load Data Preprocessing

Among the various attributes of historical load data and historical meteorological information, the data may be abnormal due to the unknown records or abnormal events.

These data not only have no reference value for forecasting itself, but also affect the load forecasting results to a large extent [16]. Therefore, it is necessary to determine which abnormal samples are and exclude them from the data input. The following abnormal data detection algorithm is based on the combination of distance and clustering. Firstly, the samples are divided by clustering algorithm; Secondly, for each sample, calculate its clustering distance, and then calculate its anomaly index. Then all samples are arranged in descending order according to the size of the anomaly index. Finally, the sample with the largest outlier index is determined as the outlier sample.

If the value of some continuous attributes of a sample is too large (greater than 10150), delete the sample. If the value of each attribute of a sample is missing, delete the sample. If some attributes in all sample data are the same or missing, delete this attribute. For each attribute U_{ok} and $k = 1, 2, \ldots, K$ in the dataset, if U_{ok} is a continuous attribute, the total average value Q_k and the total standard deviation SD_k of the attribute are calculated by using all the valid values of the attribute. Replace all missing values with the calculated overall average. If U_{ok} is a discrete attribute, a new missing value is used instead of all missing values, and it is regarded as a valid attribute value.

After the above processing, the attribute variables are used as input variables, and a clustering model is constructed by clustering algorithm, so that the samples are allocated to a cluster [17–19].

For each continuous attribute U_k, calculate and store its overall mean Q_k and overall standard deviation SD_k. For each cluster $p = 1, 2, \ldots, P$, calculate its sample size n_p. If U_k is a continuous variable, the average value Q_{pk} and standard deviation SD_{pk} of U_k in cluster p are calculated; If U_k is a discrete attribute with m different values, calculate the frequency η_{nkm} of each value in cluster p, and save the mode Q_{pk} of the attribute in cluster p. These statistics will be used to calculate the log likelihood distance $d(p, s)$ between a cluster p and a given sample s.

Given the group deviation index GDI of a sample s, it is actually the log likelihood distance $d(p, s)$ between s and its nearest sample p, which is used to measure the similarity between the sample and cluster p. The larger the distance is, the smaller the similarity is; otherwise, the greater the similarity is. $d(p, s)$ is the sum of the distance component $d(p, s)$ between each attribute value U_k and the corresponding attribute value in the cluster. $d_k(p, s)$ is called the variable deviation index VDI_k of the variable U_k.

When the attribute variable U_k is a continuous attribute, the deviation index VDI_k of attribute U_k relative to cluster p is calculated as follows:

$$d_k(p, s) = \frac{1}{2}\left[-N_p log\left(\Delta_k + \alpha_{pk}^2\right) - N_s log\left(\Delta_k + \alpha_{sk}^2\right) + N_{(p,s)} log\left(\Delta_k + \alpha_{\langle p,s\rangle k}^2\right)\right] \tag{1}$$

In the formula, N_p is the number of samples in cluster p, $\Delta_k = \alpha_k^2/6$ is a regulating factor to avoid the situation that the logarithm is meaningless when there is only one sample in the cluster. In the formula, α_k^2 is the sample variance of U_k in the whole data set, α_{pk}^2 is the sample variance of U_k in cluster p, N_s is the number of samples in the cluster, and α_{sk}^2 is the sample variance of U_k in cluster s, $N_{(p,s)}$ is the number of samples after cluster p contains sample s. obviously, $N_{(p,s)} = N_p + 1$.

$\alpha_{\langle p,s\rangle k}^2$ means that cluster p contains the sample variance of U_k after sample s.

After calculating the deviation index VDI_k of all attribute variables, the group deviation index of the sample can be calculated

$$GDI = d(p, s) = \sum\nolimits_{K=1}^{K^A+K^B} d_k(p, s) \tag{2}$$

In the formula, K^A represents the number of continuous attributes of the sample, and K^B represents the number of discrete attributes of the sample. After obtaining the above calculation results, the anomaly index is calculated to measure the anomaly of the sample. It is the ratio of GDI of sample s divided by the average GDI of all samples in cluster p.

$$AI = \frac{GDI_s}{mean\left(GDI_p\right)} \tag{3}$$

The larger the calculated anomaly index AI is, the more abnormal the data sample is. Generally, observations with an anomaly index less than 1.5 will not be regarded as outliers, but observations with an index value greater than 2 may be outliers. After clearing the outliers in the short-term load data, the data samples are normalized [20].

Generally speaking, in the training and learning of historical load data, neural network will not directly use the original data, otherwise it will appear the phenomenon of neuron saturation, so it is necessary to normalize the load data. Before inputting the historical data into the neural network, it is normalized by formula (4). In order to convert it into the value of $[-1,\ 1]$ interval; At the end of the forecast, the result is changed back to the actual load value through formula 5.

$$y_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{4}$$

$$x_i = y_i(x_{\max} - x_{\min}) + x_{\min} \tag{5}$$

In the formula, $x_{\max}$ and $x_{\min}$ represent the maximum and minimum values of the load in the load data sample, and x_i and y_i represent the values of the sample before and after normalization. After the processing, the features of power load data are extracted, and the forecasting model is designed. The extracted features are used as the input of the forecasting model to realize the short-term load forecasting.

2.2 Prediction Model Design Based on Deep Learning

The prediction model design combines the full convolutional neural network, and uses several parallel u-net neural networks to process short-term power load data. The structure of the full convolutional neural network is shown in Fig. 1.

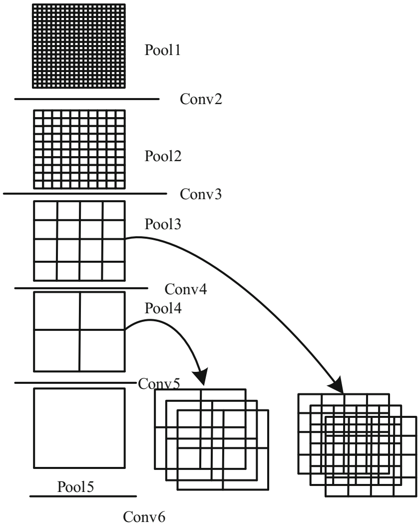


Fig. 1. Full Convolutional neural network structure diagram

The related parameter settings are shown in Table 1.

Table 1. Parameters of fully convolutional neural network

Modular	Level	Layer name	Convolution kernel/window size	Output feature map size
Search path	Level 1	Convolution layer 1	3 * 3	80 * 8 * 240/32
		Convolution layer 2	3 * 3	80 * 8 * 240/32
		Pooling layer	2 * 2	40 * 8 * 120/32
	Level2	Convolution layer 3	3 * 3	40 * 8 * 120/064
		Convolution layer 4	3 * 3	40 * 8 * 120/064
		Pooling layer 2	2 * 2	20 * 8 * 60/64
	Level3	Convolution layer 5	3 * 3	20 * 8 * 60/128
		Convolution layer 6	3 * 3	20 * 8 * 60/128
		Pooling layer 3	2 * * 2	20 * 8 * 30/128

(*continued*)

Table 1. (*continued*)

Modular	Level	Layer name	Convolution kernel/window size	Output feature map size
		Bidirectional convolution	3 * 3	10*8 * 30/128
Convolution	Level3	LSTM layer 1		10 * 8 * 30/256
		Convolution layer 7		10 * 8 * 30/256
		Bidirectional convolution		10 * 8 * 30/256
		LSTM layer 2		10 * 8 * 30/256
		Upper sampling layer 1	2 * 2	20 * 8 * 60/372
		Convolution layer 8	3 * 3	20 * 8*60/128
		Convolution layer 9	3 * 3	20 * 8 * 60/128
		Upper sampling layer 1	2 * 2	40 * 8 * 120/192
Expansion path	Level2	Convolution layer 10	3 * 3	40 * 8 * 120/64
		Convolution layer 11	3 * 3	40 * 8 * 120/64
		Upper sampling layer 3	2 * 2	80 * 8*240/96
	Level1	Convolution layer 12	3 * 3	80 * 8 * 240/32
		Convolution layer 13	3 * 3	80 * 8 * 240/32
Output		Convolution layer 14	2 * 2	80 * 8 * 240/32

The whole structure of the neural network includes three parts: downward contraction path, convolution LSTM module and upward expansion path. The main function of contraction path is to learn and extract important features from the input data. Convolution LSTM layer receives these features and fuses the adjacent features of Y axis to enhance and improve its own features. After expanding the path, the predicted data block is obtained. Because the y-axis dimension of the input data block is 8, the proposed neural network is equivalent to 8 parallel u-net neural networks [21, 22].

It is proposed that the shrinking path and expanding path of neural network consist of three different levels of shrinking module and expanding module respectively, and the characteristic dimension and channel number of different levels of modules are different. Each shrinking module contains a residual module and a pooling layer. Each residual module contains two convolution layers with the size of 3×3 convolution kernel [23, 24]. The convolution layer uses the strategy of filling 0 value to avoid dimension reduction. Each convolution layer is followed by a batch normalization layer and a relu activation function. The template size of pooling layer is 2×2, and the maximum pooling strategy is adopted. Every time the pooling layer passes through, the number of channels of the next feature is doubled. More feature channels can get more multi-scale features. The number of channels in the three levels of contraction path is 32, 64 and 1280 respectively.

The convolution LSTM layer is located between the contraction path and the expansion path. This algorithm uses two bidirectional convolution LSTM layers with 256 channels. This structure effectively connects eight parallel u-net neural networks in series. The features extracted from different u-net after shrinking path interact and fuse in convolution LSTM layer to improve the segmentation effect of network. There is a convolution layer with the size of 3×3 convolution kernel between two bidirectional convolution LSTM layers. The features enhanced by convolution LS TM layer will be sent to the expansion path.

The expansion path and contraction path are just opposite. It includes three expansion modules. Each expansion module contains an upper sampling layer and a residual module. The structure of each residual module is the same as that of the shrinkage path. The upsampling layer is equivalent to the reverse process of the pooling layer, and the template size is also 2×2. The maximum strategy is adopted. Every time it passes through the up sampling layer, the number of the next channels is halved. The number of channels in the three levels of expansion path is 128, 64 and 32 respectively. Like the original u-net structure, this network also adds three jump connections to connect the same level of contraction module and expansion module. It is helpful to learn segmentation details. Finally, a 1 * 1 convolution layer is used to reduce the dimension, and a three channel characteristic graph matrix is obtained. Finally, softmax layer is used for classification. Softmax layer can classify the known data and map the probability to the range of (0, 1). The calculation formula is as follows:

$$c_i = \frac{exp(b_i)}{\sum_{k=1}^{3} exp(b_k)} \tag{6}$$

In the formula, *exp* represents the exponential constant with natural number as the base, b_i and b_k represent the output of 1 * 1 convolution layer classified as class i and class k respectively, and c_i represents the probability of class i after classification. In the training step, the loss function uses the daisy loss function to minimize the difference between the predicted result and the labeled result.

$$E_{dice} = 1 - \frac{2 \times \sum_{i=1}^{3} \sum v_i(x) g_i(x)}{\sum_{i=1}^{3} \sum \left(v_i^2(x) g_i^2(x)\right)} \tag{7}$$

In the formula, E_{dice} refers to the value of the day's loss function, $v_i(x)$ represents the probability that the prediction data x belongs to class i, and $g_i(x)$ represents the value that the labeled data x belongs to class i. Before forecasting, the forecasting model is trained, and the parameters of the forecasting model are adjusted continuously through the loss function until the forecasting requirements are met.

2.3 Short-Term Power Load Forecasting

Before the prediction, the bottom-up unsupervised pre training is used to initialize the parameters of the full convolution neural network, and then the top-down supervised training is used to fine tune the parameters of the whole network. The connection weight β and bias ε of the visible layer and the hidden layer in the model are continuously adjusted to fit the functional relationship between the input and output data, Then the overall cost function of the prediction model is as follows:

$$H(\beta, \varepsilon) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - y_i^{'}\right)^2 + \frac{\lambda}{2} \sum_{i} \sum_{j} \sum_{l} \left(\beta_{i,j}^{l}\right)^2 \tag{8}$$

In the formula, y_i is the target output value of input x_i, $y_i^{'}$ is the corresponding prediction value, and λ is the weight attenuation parameter.

BP back propagation algorithm based on gradient descent is used to update the parameters of a single self encoder. However, when it is directly used in prediction model training, gradient dispersion is easy to occur and fall into local optimum. The core idea of the algorithm is to train only the AE with one hidden layer in the network each time. When the AE is optimal, the reconstructed visual layer is removed, and the output of the hidden layer is used as the input to train the next AE, until the last AE is optimal, the unsupervised pre training of SAE is completed. In the process of layer by layer training, the reconstructed data gets an output through a single AE, and then the cost function is minimized to update $\{\beta, \varepsilon\}$. the weight update rules are as follows:

$$\beta^l(k+1) = \beta^l(k) - \sigma \nabla_{\beta^l} C(\beta, \varepsilon) \tag{9}$$

$$\varepsilon^l(k+1) = \varepsilon^l(k) - \sigma \nabla_{\varepsilon^l} C(\beta, \varepsilon) \tag{10}$$

In the formula, $l = 1, 2, \ldots, N$, σ is the learning step of each iteration. If l is the l-compiler, the error of each node in the hidden layer is τ^l, then the partial derivative of the cost function C to the weight parameter is calculated as follows:

$$\nabla_{\beta^l} C(\beta, \varepsilon) = \tau^l + \lambda \beta^l \tag{11}$$

$$\nabla_{\varepsilon^l} C(\beta, \varepsilon) = \tau^l \tag{12}$$

$$\tau^l = \left[\beta^l \left(x^l - y\left(x^l\right)\right) g'(z)\right] f'(z) \tag{13}$$

In the formula, $y\left(x^l\right)$ represents the reconstructed output of x^l, $f'(z)$ and $g'(z)$ are the derivative values of the activation functions $g(z)$ and $f(z)$ respectively. After the unsupervised pre training, the prediction model is given initial weights, and the supervised parameters of the whole network are fine tuned by combining the connection weights β and bias ε of the visible layer and the hidden layer. The activation function value and the overall cost function of each layer of neuron node are calculated by using the above formula, and then the BP algorithm is used to calculate backward layer by layer. At this time, the cost function becomes $H(\beta, \varepsilon)$, The error of each node in the output layer and hidden layer of the prediction model is expressed as:

$$\begin{cases} \tau^l = -\left(y - y'\right) f'(z) \\ \tau^l = \beta^l \tau^{l+1} f'(z) \end{cases} \tag{14}$$

The gradient descent method is used to update parameters repeatedly to minimize $H(\beta, \varepsilon)$. After the training, the training samples are used as the input of the forecasting model for feature learning, and the high-order feature representation of the reconstructed load, weather and day type data is obtained. After unsupervised network training and supervised parameter fine-tuning, the forecasting model is optimized, and the test samples are input into the forecasting model for short-term load forecasting. So far, the design of short-term load forecasting method based on full convolution deep learning is completed.

3 Experimental Research on Short-Term Load Forecasting Method Based on Full Convolution Deep Learning

3.1 Setting up the Experimental Environment

The Hadoop cluster built in the experiment consists of six PCs with the same configuration. Each PC has 2 G memory, soogb hard disk, dual core inter E7500 CPU, 2.93 GHz main frequency, and runs several CentOS Linux operating systems. One machine is the master node of master, which is responsible for the resource allocation and job scheduling of the whole cluster. The other five machines are slave nodes, which are mainly used to store data and run tasks. If the number of HDFS copies is 3, the slave node can run 4 map/reduce tasks at the same time. The Hadoop cluster topology is shown in Fig. 2.

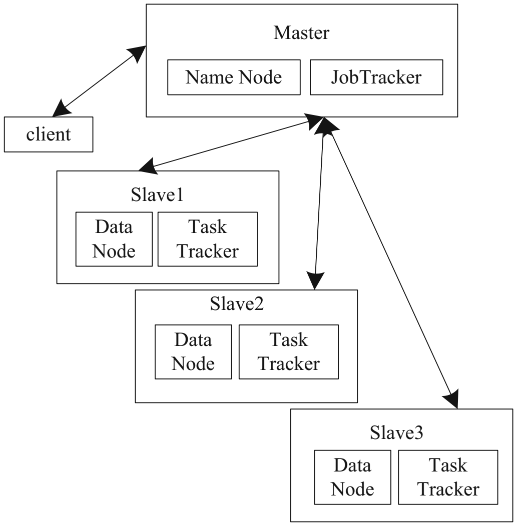


Fig. 2. Hadoop Cluster topology diagram

In the above experimental environment, the traditional three prediction methods are introduced and compared with the designed prediction method in the same experimental environment. According to the experimental results, each prediction method is compared and analyzed.

3.2 Preparation of Experimental Data Sets

The experimental data comes from the load data and influencing factor data collected by a power grid in a certain area. The amount of data is TB level, and the latitude is high. It is mainly structured and semi-structured data, which is in line with the characteristics of power big data. The influencing factors include daily maximum temperature, average temperature, minimum temperature, average wind speed, average humidity, average precipitation, sunshine type and season. The calculation of the correlation degree between each influencing factor and load is shown in Table 2.

Table 2. Results of the calculation of the correlation of each factor

Project	Parameter	Project	Parameter
Daily maximum temperature	0.87	Average humidity	0.66
Average temperature	0.71	Average precipitation	0.32
Minimum temperature	0.57	Sunshine type	0.60
Average wind speed	0.28	season	0.53

According to the calculation results, the daily maximum temperature, average temperature, sunshine type, average humidity and minimum temperature are selected as the five key factors affecting the load. The training sample is the power consumption data of a whole year, and the sampling interval is 1H. The composition of the training set is shown in Table 3.

Table 3. Training sample set

Parameter setting		Meaning
Input	x_1	Predicted daily maximum temperature
	x_2	Forecast daily average temperature
	x_3	Forecast daily average humidity
	x_4	Forecast daily sunshine type
	x_5	Forecast daily minimum temperature
	x_6–x_8	Forecast the load value of $t-1$, t and $t+1$ on the day before the day
	x_9–$x11$	Load values of $t-1$, t and $t+1$ in the same week before the forecast day
Output	y	Load forecast value of forecast day t

Taking 24-h load data as test samples, relative error, average error, root mean square error and average absolute percentage error are used to measure load forecasting effect. The calculation formula of each index is as follows:

$$\chi_{ME} = \frac{1}{n+1}\sum_{i=0}^{n}(u(i) - v(i)), n = 23 \tag{15}$$

$$\chi_{RMSE} = \sqrt{\frac{\sum_{i=0}^{n}(u(i) - v(i))^2}{n+1}}, n = 23 \tag{16}$$

$$\chi_{MAPE} = \frac{1}{n+1}\sum_{i=0}^{n}\left|\frac{u(i) - v(i)}{v(i)}\right| \times 100\%, n = 23 \tag{17}$$

In the formula, $u(i)$ represents the actual load value of time i and $v(i)$ is the predicted value of time i. Taking the above indexes as the standard to measure the prediction accuracy, only one group of experiments is not enough to predict the practical application level of the method. Therefore, the second group of experimental plans is designed, and the second group of experiments takes the size of the prediction range as the measurement index, the larger the prediction coverage, the better the prediction ability.

3.3 Experimental Results and Analysis of Load Prediction Accuracy

In the load forecasting accuracy experiment, four different forecasting methods are used to forecast the daily load. The historical load value, weather attribute and day type data of similar days are used at each time point. The forecasting method is used to forecast the power load change for five consecutive days, and the load forecasting error index is counted. The specific experimental results are shown in Table 4.

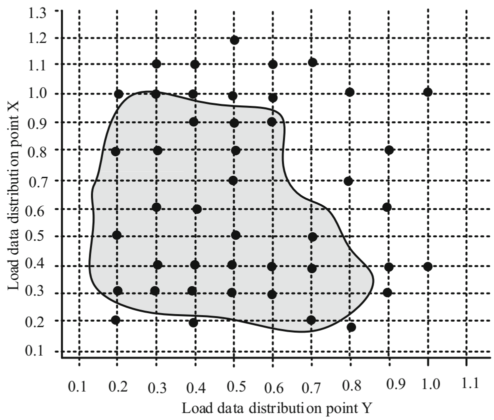
Table 4. Experimental results of prediction accuracy of different load forecasting methods

	Date	Relative error of daily maximum load/%	Average error/MW	Root mean square error/%	Percentage error of mean absolute value/%
Traditional prediction method 1	1 day	3.15	0.0850	4.11	21.05
	2 days	4.05	0.0904	3.97	23.53
	3 days	3.67	−0.0793	5.02	22.58
	4 days	4.26	−0.0993	4.95	19.75
	5 days	4.35	0.876	4.55	18.22
Traditional prediction method 2	1 day	3.66	0.0941	3.94	20.63
	2 days	4.05	0.0826	4.11	22.64
	3 days	4.91	0.0791	6.23	23.74
	4 days	4.52	−0.823	5.24	24.06
	5 days	3.86	0.0741	4.81	19.54
Traditional prediction method 3	1 day	4.62	0.0873	6.26	23.22
	2 days	3.91	0.0562	5.12	21.45
	3 days	3.87	0.0611	4.29	19.87
	4 days	4.03	−0.0438	3.94	26.52
	5 days	4.59	0.0506	4.01	20.47
The prediction method is proposed	1 day	1.72	−0.0141	2.09	16.21
	2 days	1.35	0.0106	2.62	17.32
	3 days	1.78	0.0151	2.50	17.26
	4 days	1.06	−0.0137	2.68	18.51
	5 days	1.14	0.0171	2.35	16.82

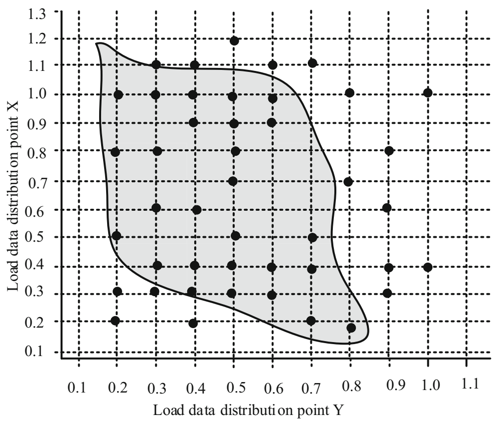
Comparing and observing the data in the table, it can be seen from the data in the table that the relative error, average error, root mean square error and absolute error of the three traditional prediction methods in the short-term prediction are higher than those of the proposed prediction methods, which are quite different from the actual load. The maximum relative error, average error, root mean square error and absolute error of the proposed prediction methods are 1.78%, 2.68 MW, 18.51% respectively 0.0171%. Therefore, the load forecasting method based on full convolution deep learning has high forecasting accuracy. The reason is that the input data is convoluted and pooled by full convolution deep learning, which improves the prediction performance.

3.4 Experimental Results and Analysis of Load Forecasting Coverage

In the experiment, in the experimental environment, different load forecasting methods are used to forecast short-term power load, the coverage area of load data processed by each forecasting method is calculated, and the experimental results are output through the third-party software. The specific content of load forecasting coverage area experiment results is shown in Fig. 3.

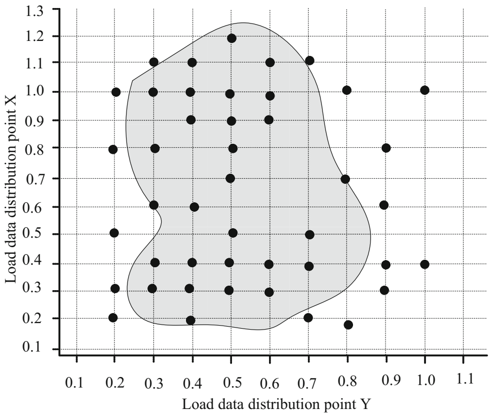


(a) Experimental results of traditional load forecasting method 1

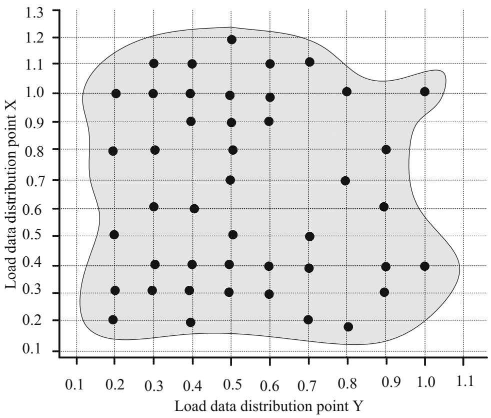


(b) Experimental results of traditional load forecasting method 2

Fig. 3. Experimental results of different load forecasting methods



(c) Experimental results of traditional load forecasting method 3



(d) Experimental results of the proposed load forecasting method

Fig. 3. continued

Through the observation of Fig. 3, it can be seen that the three traditional load forecasting methods can not cover all load data when processing load data. There are some missing data points outside the distribution point. Compared with the above, the proposed load forecasting method covers a larger area and covers all load data. According to the experimental results of prediction accuracy, the load forecasting method based on full convolution depth learning has high prediction accuracy and high application efficiency, which is better than the traditional load forecasting method.

4 Conclusion

With the rapid development of the economy and the advent of the electrical age, the demand for power is also increasing. The research on power load forecasting model has become a hot topic. The accuracy of power load forecasting is directly related to the supply and demand balance of power grid, and affects the operation cost and security and stability of power grid. Therefore, a new model of power load forecasting is explored, it is very important to improve the accuracy of power load forecasting. In order to improve the accuracy of power load forecasting, this paper proposes a short-term power load forecasting method based on full convolution deep learning.

Load data has dual characteristics of time and space. In terms of time, load data has the characteristics of long-term trend, periodicity and close association with its own historical data. Because the load data has these characteristics, the algorithm model can fit it well. In space, the operation of power system is affected by external factors, and the load data produced also has the characteristics of randomness. According to the characteristics of power load data, after the design of the short-term power load forecasting method based on full convolution deep learning is completed, the experimental environment is built, the load data is collected, and the prediction accuracy and prediction area are compared by using different prediction methods. According to the two groups of experimental results, it is proved that the prediction method based on full convolution deep learning has higher accuracy and wider application range. It is suitable for practical projects. However, there are still some deficiencies in the design process. In the follow-up study, some shortcomings of this method will be studied and analyzed one by one to further improve the load forecasting method.

Fund Projects. 2020 Jiangsu Province College and University Students' Innovation and Entrepreneurship Training Program (Key).

Power load peak prediction method based on time convolutional network (202011276015Z).

Reference

1. Xu, Y., Wu, Z., Zhu, H., et al.: Short-term load forecasting based on multi-scale convolutional neural network. J. Shenyang Univ. Technol. **42**(06), 618–623 (2020)
2. Zhao, B., Wang, Z., Ji, W., et al.: A short-term power load forecasting method based on attention mechanism of CNN-GRU. Power Syst. Technol. **43**(12), 4370–4376 (2019)
3. Chen, G., Teng, H.: Short-term load forecasting based on deep learning of hybrid neural networks. Water Resour. Power **38**(04), 193–196 (2020)
4. Chen, G., Teng, H.: Short-term load forecasting based on deep learning of hybrid neural networks. Water Resour. Power **56**(03), 91–96+102 (2019)
5. Zhao, H., Zhao, Y., Guo S.: Short-term load forecasting based on complementary ensemble empirical mode decomposition and long short-term memory. Electr. Power **53**(06), 48–55 (2020)
6. Zhao, W., Lin, R., Tang, W., et al.: Forecasting model of short-term PM2.5 concentration based on deep learning. J. Nanjing Norm. Univ. (Nat. Sci. Edn.) **42**(03), 32–41 (2019)
7. Li, H., Lin, J., Li, G., et al.: Short-term power load forecasting based on weighted cosine similarity and extreme learning machine. Sci. Technol. Eng. **20**(11), 4370–4374 (2020)
8. Xu, Y., Lu, Y., Zhu, B., et al.: Short-term load forecasting method based on FFT optimized ResNet model. Control Eng. China **26**(06), 1085–1090 (2019)
9. Luo, Y., Cai, Y., Qi, Y., et al.: Short-term power load forecasting algorithm based on maximum deviation similarity criterion BP neural network. Appl. Res. Comput. **36**(11), 3269–3273 (2019)

10. Li, Y.: Establishment and application of short-term power load forecasting. Comput. Simul. **28**(10), 316–319 (2011)
11. Liu, S., Liu, X., Wang, S., Muhammad, K.: Fuzzy-aided solution for out-of-view challenge in visual tracking under IoT assisted complex environment. Neural Comput. Appl. **33**(4), 1055–1065 (2021)
12. Gao, P., Li, J., Liu, S.: An introduction to key technology in artificial intelligence and big data driven e-learning and e-education. Mob. Netw. Appl. **26**, 2123–2126 (2021). https://doi.org/10.1007/s11036-021-01777-7
13. Liu, S., Liu, D., Srivastava, G., et al.: Overview and methods of correlation filter algorithms in object tracking. Complex Intell. Syst. **3**, 1–23 (2020)
14. Kalakova, A., Kumar, S., Jamwal, P.K., et al.: A novel genetic algorithm based dynamic economic dispatch with short-term load forecasting. IEEE Trans. Ind. Appl. **57**, 2972–2982 (2021)
15. Wang, Y., Chen, J., Chen, X., et al.: Short-term load forecasting for industrial customers based on TCN-LightGBM. IEEE Trans. Power Syst. **36**, 1984–1997 (2021)
16. Tavassoli-Hojati, Z., Ghaderi, S.F., Iranmanesh, H., et al.: A self-partitioning local neuro fuzzy model for short-term load forecasting in smart grids. Energy **199**, 117514 (2020)
17. Gilanifar, M., Wang, H., Sriram, L., et al.: Multitask Bayesian spatiotemporal gaussian processes for short-term load forecasting. IEEE Trans. Industr. Electron. **67**(6), 5132–5143 (2020)
18. Sheng, Z., Wang, H., Chen, G., et al.: Convolutional residual network to short-term load forecasting. Appl. Intell. **51**(4), 1–15 (2020)
19. Sharma, S., Majumdar, A., Arregui, V.E., et al.: Blind Kalman filtering for short-term load forecasting. IEEE Trans. Power Syst. **35**, 4916–4919 (2020)
20. Barman, M., Choudhury, N.: Season specific approach for short-term load forecasting based on hybrid FA-SVM and similarity concept. Energy **174**, 886–896 (2019)
21. Kassa, Y., Zhang, J., Zheng, D.: EMD-PSO-ANFIS based hybrid approach for short-term load forecasting in microgrids. IET Gener. Transm. Distrib. **14**(3), 470–475 (2019)
22. Yin, L., Xie, J.: Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems. Appl. Energy **283**(6), 116328 (2020)
23. Li, C.: Designing a short-term load forecasting model in the urban smart grid system. Appl. Energy **266**, 114850 (2020)
24. Sadaei, H.J., Candid, D., Guimaraes, F.G., et al.: Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. Energy **175**, 365–377 (2019)