Long-term prediction method of reactive load based on LSTM neural network

Xingwei Liu¹, Shixiong Fan¹, Jiaqi Qin², Yan Liu³,⁵ and Wei Wang⁴

¹ China Electric Power Research Institute, Beijing, China, 100192;
² College of Computer Science, Sichuan University, Chengdu, China, 610000;
³ College of Electrical Engineering, Sichuan University, Chengdu, China, 610000;
⁴ State Grid Gansu Electric Power Corporation, Lanzhou, China, 730050

Email: liuyan77@scu.edu.cn

Abstract. In the operation of the power system, the prediction of reactive load can provide accurate data foundation conditions for reactive power compensation planning. Aiming at the characteristics of time series, non-linearity and small fluctuation of power reactive load data, this paper proposes a reactive power load prediction method based on LSTM (long short-term memory) neural network. The proposed method is used to process and predict the power reactive load data of a certain area in Fujian Province. The experimental results show that the LSTM model with one layer and 64 neuron structures have the best result among the support vector regression (SVR), the random forest model (RF), Recurrent Neuron Network (RNN), LSTM-CNN hybrid neural network, the prediction of reactive load on long-term scale has higher accuracy and stability.

1. Introduction

Load forecasting plays an important role in the planning and operation of power system. In traditional researches, people focused on predicting active load instead of predicting reactive load. Recently, due to the development of large scale renewable energy and power generation technology, the changes of the operation model of power system, and the progress of ultra-high voltage direct current transmission, the uncertainty of reactive load has significantly increased in power system, so as to ask for more attentions. However, because of the characteristics of reactive load, such as locality and dispersivity, its power flow direction is not as clear as that of the active load. What’s more, the regularity of the reactive load change is very poor [1]. Due to the mentioned reasons, it is hard to predict reactive load as accurate as expected, and there is no quantitative assessment of the current and future reactive power status of the region. So far, only a few literatures have investigated reactive load forecast, some of them calculated the reactive power by directly converting the active power and active power factor. However, the uncertainty and chaotic characteristics of the reactive load are far stronger than those of the active load. The time-varying situation of reactive load tends to be different from that of the conventional load curve, which may cause huge errors in reactive load forecast. In [2], a multi-node load forecasting system could simultaneously predicted the multi-node active load and reactive load. It made full use of the regularity of the total change, and the relationship between the active and reactive power. In [3], the combined model of time series and support vector machine (SVM) was used to predict the reactive load, but it did not directly use the reactive load data to train the model. Based on the Sage-Husa adaptive filtering algorithm with time-varying noise estimator, the
paper [4] proposed an improved Sage-Husa adaptive filtering reactive load prediction method. However the method does not perform well in long term reactive load forecast.

This paper proposed a well-designed long short-term memory (LSTM) model to forecast the reactive load in long term. Instead of calculating the reactive power through active power and power factor, we used the real node reactive power data to make the prediction more realistic. We presented the entire network architecture and the parameter adjust method. In order to illustrate the advanced performance on reactive load forecasting of the proposed method, we compared it with other popular methods on traditional load forecasting. For the sake of fairness, these methods [5-10] were directly used to perform the reactive load prediction. Validations showed that the proposed model surpassed the rest methods.

2. The Forecasting framework based on LSTM

2.1. The LSTM model

Different from the traditional neural network, the hidden layer of recurrent neural networks (RNNs) contains memory cells. The memory cells can build the temporal correlations between previous information and the current circumstances. In the process of training RNN with backpropagation through time (BPTT)[11], the gradient will vanishing when the time interval of input data increases[12]. To solve the issues, LSTM architecture was introduced which include a memory cell and a forget gate.

2.1.1. The structure of the LSTM unit. LSTM increases the memory cell state \( c \) relative to a single hidden state \( h \) in the RNN. As shown in Figure 1, \( c \) is passed in the hidden layer. At time \( t \), the memory cell state \( c_{t-1} \), the hidden layer state \( h_{t-1} \) and the current excitation \( x_t \) jointly predict the system response \( o_t \). At the same time, generate a new memory cell state \( c_t \) and a hidden layer state \( h_t \) to perform timing calculations for the system. The LSTM unit consists of three gated structures, as shown in Figure 2.

![Figure 1. LSTM externa unit structure.](image1)

![Figure 2. LSTM internal unit structure.](image2)

The memory cell state \( c_{t-1} \) interacts with the ‘hidden layer’ output of \( h_{t-1} \) and the input \( x_t \) to determine which elements of the state vector should be updated. The structure of LSTM unit defines input node \( g_t \), input gate \( i_t \), forget gate \( f_t \) and output gate \( o_t \). The formula of all nodes in a LSTM unit as follows.

\[
\begin{align*}
    f_t &= \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \\
    i_t &= \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \\
    g_t &= \varphi(W_{gx}x_t + W_{gh}h_{t-1} + b_g) \\
    c_t &= c_{t-1} \odot f_t + g_t \odot i_t \\
    o_t &= \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \\
    h_t &= \varphi(c_t) \odot o_t
\end{align*}
\]
Where $W_{fx}, W_{fh}, W_{ix}, W_{ih}, W_{gx}, W_{gh}, W_{ox}$ and $W_{oh}$ are weight matrices of activation function; $\odot$ means an element-wise multiplication; $\sigma$ and $\varphi$ represent the sigmoid activation and the tanh function.

2.1.2. Building LSTM neural network structure. The LSTM based reactive load forecasting framework is given as Figure 3. It includes input layer, hidden layer, fully connected layer and output layer. When the number of hidden layers is greater than 1, the Dropout layer is added to prevent model over fitting.

![Figure 3. Structure of the LSTM network.](image)

3. Example analysis

All the experiments were performed on a workstation (Intel Xeon E5 2650 CPU and 256 GB RAM) equipped with a graphics processing unit card (GTX 1080Ti). The software architecture was developed using the Keras library with Tensorflow backend. The flow of reactive load forecasting is given in Figure 4.

![Figure 4. Flow chart of reactive load forecasting.](image)

3.1. Data preparation

| Feature name          | Feature number |
|-----------------------|----------------|
| **Weather feature**   |                |
| Temperature           | 1              |
| Weather type          | 2              |
| Wind speed            | 3              |
| Humidity              | 4              |
| **Time feature**      |                |
| Week                  | 5              |
| Holiday               | 6              |
| Hour                  | 7              |
| **Economic feature**  |                |
| Industrial growth     |                |
| value growth rate     | 8              |
| **Historical data**   |                |
| Active load           | 9              |
| Reactive load         | 10             |
3.1.1. Data introduction. The reactive load in the actual operation of the substation is affected by factors such as electricity habits, weather and local economic development. All features are shown in Table 1.

3.1.2. Input data structure. We use $x_t$ to represent the data at time $t$: $x_t = \{temperature, weather type, wind speed, humidity, week, holidays, hour, industrial growth value growth, active load, reactive load\}$. After the data entry into the network, the LSTM use window select four consecutive time points as the sequence to load. Then LSTM uses the data of the first three time points to forecast the fourth time point. After the forecasting, the window move to the next one.

![Graph of LSTM loop read data.](image)

3.1.3. Data cleaning. During the data collection process, there may be human error or equipment failure, which will result in data missing and data exceptions. For the missing values at individual time points, we use Lagrangian interpolation to make up. For the missing values in a period of time, it is directly discarded. Then use the box plot method to observe and the outliers. In addition, we normalize the original data to [0,1] for unify the dimensions. The formula is as follows,

$$x^* = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

(9)

where $x^*$ is the normalized value, $x_{\text{min}}$ and $x_{\text{max}}$ are the minimum and maximum values in the data.

3.1.4. Build datasets. The reactive load after cleaning is still sequential, so we divide the data set accordingly. The data from July to October is divided into training data set, and the November data is used as validation data set. The December data is used as the monthly test data set. And further take the four weeks of December as the weekly test data sets.

3.2. Evaluation metrics and loss function

In this paper, we choose the root mean square error (RMSE) and the mean absolute percentage error (MAPE) as error evaluation metrics. And we use the mean squared error (MSE) as the loss function. The formulas are given as follows.

$$e_{\text{MAPE}} = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - y^*_i|}{y_i} \times 100\%$$

(10)

$$e_{\text{RMSE}} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - y^*_i)^2}$$

(11)

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - y^*_i)^2$$

(12)

Where $n$ is the number of predicted points, $y_i$ is the actual load value of the $i$-th point, and $y^*_i$ is the predicted load value of the $i$-th point.

3.3. Analysis of Long-term reactive load prediction results

3.3.1. Results of LSTM models with different hidden layer structures. The number of layers and the number of neurons in each layer would affect the forecast results of the LSTM. Generally, in order to avoid over-fitting in the training process, the number of layers was less than 4. This paper established.
three LSTM models with different hidden layers: LSTM_1(layer=1), LSTM_2(layer=2), LSTM_3(layer=3). The number of iterations was 100, and tanh was selected as the activation function. After initial tuning, the number of neurons in the cell and the parameters of each layer (the number of neuron weights) were shown in Table 2. Three different layers of LSTM models were used to predict the test set, and the $e_{\text{MAPE}}$ and $e_{\text{RMSE}}$ values were shown in the Table 3.

Table 2. The number of neurons and params in each layer in different LSTM models.

|     | LSTM_1 | LSTM_2 | LSTM_3 |
|-----|--------|--------|--------|
| cell | param  | cell   | param  |
| Layer1 | 64     | 19712  | 32     |
| Layer2 | 64     | 24832  | 64     |
| Layer3 | 64     | 33024  |        |

Table 3. Error comparison of each LSTM model on month test set.

|      | $e_{\text{RMSE}}$ | $e_{\text{MAPE}}$ |
|------|-------------------|-------------------|
| LSTM_1 | 0.537             | 4.417             |
| LSTM_2 | 0.647             | 5.111             |
| LSTM_3 | 1.340             | 13.883            |

According to Table 3, The $e_{\text{MAPE}}$ and $e_{\text{RMSE}}$ values of the LSTM_1 model were obviously the smallest, especially compared to the LSTM_3 model. We speculated that it may be related to the total number of parameters. Therefore, we build three LSTM models with one layer and a LSTM model with two layers: LSTM_1_16 (layer=1, param=16), LSTM_1_32 (layer=1, param=32); LSTM_1_96 (layer=1, param=96), LSTM_2_32_32 (layer=2, param=14080). Table 4 showed the comparison of the average MAPE and RMSE on the weekly test data sets.

Table 4. Error comparison of each LSTM model on week test sets.

|      | Week1 | Week2 | Week3 | Week4 |
|------|-------|-------|-------|-------|
| $e_{\text{RMSE}}$ | $e_{\text{MAPE}}$ | $e_{\text{RMSE}}$ | $e_{\text{MAPE}}$ | $e_{\text{RMSE}}$ | $e_{\text{MAPE}}$ | $e_{\text{RMSE}}$ | $e_{\text{MAPE}}$ |
| LSTM_1_16 | 0.738 | 5.042 | 0.638 | 5.765 | 1.052 | 10.696 | 1.002 | 14.484 |
| LSTM_1_32 | 0.708 | 4.408 | 0.665 | 5.077 | 0.678 | 6.078 | 0.617 | 7.961 |
| LSTM_1_64 | 0.688 | 4.041 | **0.551** | **4.581** | **0.630** | 4.913 | **0.488** | **5.894** |
| LSTM_1_96 | 0.767 | 4.191 | 0.849 | 7.011 | 1.13 | 8.227 | 0.648 | 6.782 |
| LSTM_2_32_32 | **0.626** | **3.33** | 0.653 | 4.803 | 0.637 | **4.88** | 0.793 | 8.124 |

In Table 4, the average value of $e_{\text{MAPE}}$ in the LSTM_1_64 model was 0.589%, which was 0.2683%, 0.0777%, and 0.2593% lower than LSTM_1_16, LSTM_1_32, and LSTM_1_96, respectively. The average value of $e_{\text{RMSE}}$ in the LSTM_1_64 model was 4.8573, which was 4.1394, 1.0237, and 1.6947 lower than LSTM_1_16, LSTM_1_32, and LSTM_1_96, respectively. It showed that the LSTM model with one layer and 64 neurons worked better. Next, the number of LSTM layers was set to 2, the number of neurons in each layer was 32, and the total param was 14080, which was reduced by 5662 compared with the LSTM_1_64. Compared with the LSTM_2_32_32, the average value of $e_{\text{MAPE}}$ and $e_{\text{RMSE}}$ of LSTM_1_64 were reduced by 0.589% and 4.857 respectively. Through analysing, the main factor affecting the reactive load forecasting was the number of param in model. This paper finally chose the LSTM_1_64 neural network model.

3.3.2. Results of LSTM compared with other models. So far, the grid reactive load prediction accuracy were not ideal in long term. There were few research related to it. To verify the effect of the
LSTM_1_64 neural network model, we chose SVR, RNN, RF, and LSTM-CNN to have the contrastive experiments. The SVR and RF models were two traditional methods commonly used for regression prediction. The RNN model included a hidden layer and 64 neuron. LSTM-CNN had two-layer CNN convolution layers and one LSTM hidden layer. The comparison results were shown in Figures 6 and 7. It could be seen from the figures that LSTM_1_64 has a large advantage over the other three models in predicting long-term reactive load.

Figure 6. RMSE histogram of different models on each data set.

Figure 7. MAPE histogram of different models on each data set.

As shown in Figure 7, the $e_{MAPE}$ trend in test data sets was gradually increasing. This trend showed that the reactive load forecasting accuracy was decreasing with the time increases. It mean that the model had better forecasting accuracy in short term, which conform to the basic logic of prediction. The fourth week's reactive load forecasting curve and partial enlarged view of the curve were shown as Figure 8 and Figure 9.

Figure 8. Reactive load prediction curve.

Figure 9. Partial map of reactive load prediction curve.
Where the black, red, blue, green, yellow and pink lines represented the actual value, the results of LSTM_1_64, LSTM-CNN, RNN, RF, SVR, respectively.

In Figure 8 the curve exhibits periodic changed over time. It could be seen from Figure 9 that LSTM_1_64 fit the real curve best. LSTM-CNN had the largest deviation from the true value. The results further verified that the LSTM_1_64 had a better effect on reactive load forecasting in long term.

4. Conclusions
Reactive load forecasting has attracted the attention of many researchers. In this paper, we first deeply mines historical reactive power load data, meteorological features, time features, economic characteristics and other data information. Then we study in detail the effects of LSTM models with different numbers of layers and neurons. Finally, compared with the SVR, RF, RNN and LSTM-CNN models in long term, LSTM_1_64 also can achieve better forecasting. Subsequent research work will optimize the neural network model to improve the training convergence speed and the real-time performance of reactive load forecasting.

Acknowledgments
This work was supported by the Science and Technology Research Foundation of SGCC “Research and application of reactive power and voltage characteristic analysis and control technology of power system based on Data Mining”(5442DZ180016).

References
[1] Ekonomou L, Christodoulou C, Mladenov 2016 A short term load forecasting method using artificial neural networks and wavelet analysis International Journal of Power Systems 1 64
[2] Han x s, Han L, Gooi H B 2012 Ultra short term multi node load forecasting – a composite approach IET generation, transmission &amp; distribution 6(5) 436
[3] Dai Xinbo, Cui Yong, Ji Sen 2015 Research on the prediction method of the load - load of the power grid Hydropower and energy science 33(4) 181
[4] Chen y, Luo a, Shuai Z 2013 Robot predictive dual loop control strategy with reactive power compensation for single phase grid connected distributed generation system Iet Power Electronics 6(7) 1320
[5] Tai Zhang, Li Zhang, Jiahong Peng 2019 Prediction of tobacco yield in Shaoyang County Based on principal component and support vector machine China Agricultural bulletin 35(13) 152
[6] Lu Longmei, Zhao Mingsong, Lu Hongliang 2019 Application comparison of artistic natural network and random forest in expression problem Science and technology innovation and application 10 31
[7] Mikolov T, karafi á t m, Burger L, et al 2010 Recurrent network based language model 2010 seventh annual conference of the International Speech Communication Association 1045
[8] D'informatique D E, Ese N, Esent P 2001 Long short term memory in recent neural networks EPFL 9(8) 1735
[9] Sak H, Senior A, Beaufays F 2014 Long short term memory recurrent network architectures for large scale acoustic modeling Fifth Annual Conference of the International Speech Communication Association
[10] Jixiang Lu, Qipei Zhang, Zhihong Yang 2019 Short term load forecasting method based on CNN LSTM hybrid network model Power system automation 43(8) 131
[11] Werbos P J 1990 Backpropagation through time: what it does and how to do it Proceedings of the IEEE 78(10) 1550
[12] Bengio Y, Simard P, Frasconi P 1994 Learning long-term dependencies with gradient descent is difficult IEEE transactions on neural networks 5(2) 157