Application of residual self-fitting ensemble neural network based on LSTM in short-term power load forecasting

Baowen Zhang\textsuperscript{1,}a, Dejia Lao\textsuperscript{2,}b, Longxiang Sun\textsuperscript{3,}c

\textsuperscript{1}College of Electrical Engineering Zhejiang University
Hangzhou, China
\textsuperscript{2}Maintenance Branch of State Grid Zhejiang Electric Power CO., LTD
Hangzhou, China
\textsuperscript{3}State Grid Huzhou Power Supply Company
Huzhou, China
\textsuperscript{a}ezhangbw@163.com \textsuperscript{b}ldjstc@126.com \textsuperscript{c}zhydh@sohu.com

Abstract. Accurate load forecasting helps to improve the safety and stability of power systems. LSTM neural networks based on deep learning has lower predicted error than traditional machine learning and time series models. After calculating the residuals of the predicted values and observed values of the LSTM and traditional machine learning model, it can be found that the distribution of residuals also possessed with the regularity of time series data. Therefore, the residual time series can be fitted by another model based on this regularity. To avoid the limitations of a single model, a variety of neural network models have been used to compose ensemble model. In this model, the residual can be generated and fitted by itself. In other words, it implements self-fitting. After analysing the residual distribution and the ensemble method of predictive model, this paper presents a new ensemble neural network model used for short-term power load forecasting. After the experiment, it’s obvious that the predicted results of the new ensemble neural networks is more accurate than single LSTM.

1. Introduction

Power system load forecasting refers to forecasting power system load at a future time based on historical data. Load forecasting is an important part of power system planning. According to the predicted time range, the power load forecasting method can be divided into long-term, medium-term, short-term and ultra-short-term forecasts. Short-term load forecasting is usually a forecast within 1 year, and hours, which are mainly used to regulate and guide the daily operation of the power sector.

There are two types of commonly used short-term load forecasting models. One is the traditional time series and machine learning model. The typical time series model is the autoregressive integrated moving average model ARIMA [1]. Typical machine learning models are BPNN artificial neural networks [2], XGBoost integrated tree model [3], SVM support vector regression model [4]. Among the above models, BPNN is used most widely because of its excellent linear and nonlinear fitting ability. However, the common problem of the above models is the lack of consideration of temporal correlation of time series data, and it is necessary to add time characteristics to help predicting results. The second type is deep learning model such as RNN[5] and LSTM [6] neural networks that have been widely used as serial models in recent years. The features of RNN and LSTM [7][8] are that they can effectively use the historical data of time series to find their key information automatically. Their complex neural
network structure has a strong linear/non-linear fitting ability. Parameters are adjusted through BPTT (Back Propagation Through Time) method automatically. But the disadvantage of RNN is that when the time interval increases, the gradient may “vanish” [9] which means the gradient stay in zero and parameters can not be adjusted. LSTM has improved the RNN on the structure design [10][11][12], which effectively avoids this problem and become more excellent and powerful. Therefore, the application scenario of LSTM is more extensive.

2. Structure of LSTM cell

The cell unit structure of LSTM is shown as follows[13]:

![Figure 1. Structure of LSTM cell](image)

For the version of LSTM used in this paper is implemented by the following composite function:

\[ i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \]  
\[ f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \]  
\[ c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \]  
\[ o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \]  
\[ h_t = o_t \tanh(c_t) \]

where \( \sigma \) is the sigmoid function, and \( i \) is the input gate, \( f \) is the forget gate, \( o \) is the output gate, and \( c \) is the cell input activation vectors, \( h \) is the hidden vector. The weight matrix \( W \) subscripts have the obvious meaning, for example \( W_{hi} \) is the hidden-input gate matrix, \( W_{xo} \) is the input-output gate matrix etc.

The output of the activation function \( \sigma \) ranges in \([0, 1]\). If the output of \( \sigma \) is 0, all the information of the previous state would be discarded. If it is 1, all the information of the previous state would be retained.

3. Predicted results of LSTM and BPNN

Taking 283 samples from power load data of one city in 2017 as test data and 1827 samples from 2012 to 2016 are used as training data. The LSTM and BPNN models are trained separately, then compare the predicted error of their results, and calculating the residual error between the predicted and the real results.

MAPE (Mean Absolute Percentage Error) was used as the evaluation criterion. The lower MAPE means the lower error.

\[ MAPE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \]  

The residual of time \( i \) is \( r_i \):

\[ r_i = y_i - \hat{y}_i \]

\( y_i \) represents the observed load value of time \( i \), \( \hat{y}_i \) represents the predicted value of time \( i \) 

BPNN and LSTM model are built with TensorFlow which is a powerful open source deep learning architecture created by Google.
BPNN consists of 1 hidden layer, 1 input layer, and 1 output layer. Since it is necessary to add historical information to the BPNN, the load values of the past \( w(= 6,7,8, \text{etc.}) \) moments values are added to the data as time characteristics, so the number of neural units in the input layer is \( w \). The final output is a single predicted value so the output layer consists of 1 neural unit, the hidden layer consists of 50 neural units, and the activation function is Relu. The training times is 500.

LSTM consists of 2 hidden layer, 1 input layer, and 1 output layer. There is 1 neural unit in the input layer, because the final output is a single prediction value. The output layer consists of 1 neural unit. Two hidden layers consist of 100 neural units, the activation function is Relu, the optimization function is RMSProp, The training times is 1500.

After predicting on testing data set, we can get the predicted result of LSTM and BPNN. Therefore we can calculate the residual and MAPE of LSTM and BPNN.

The MAPE of LSTM is 5.19%, the MAPE of BPNN is 6.15%. It’s obvious the predicted result of LSTM is more accurate than BPNN.

4. Analysis of Residuals
After predicting on testing data set, we can get predicted results. Therefore the residual between predicted values and observed values can be calculated.

Based on the analysis method of time series data, the time distribution of residuals can be analyzed in terms of stationarity and correlation.

The stationarity of time series is a prerequisite for time series analysis. In order to test the stationarity of the residual time series, ADF Unit Root Test[14] is being used.

ADF Unit Root Test method can be used to calculate five quantitative values[15] which are ADF Test statistic value, p value, 1% Critical Value, 5% Critical Value, 10% Critical Value. By comparing ADF Test statistic value with three Critical Value and comparing p value with 0.05, the stationarity of the distribution of residuals time series will be verified.

The ADF test results are listed in Table 1. Residual-LSTM means the residual of LSTM:

| Residuals-LSTM | Residuals-BPNN |
|----------------|----------------|
| Test statistic Value | -9.67285        | -15.8005        |
| p-value           | 1.26335e-16     | 1.07537e-28     |
| 1% Critical Value | -3.47533        | -3.45427        |
| 5% Critical Value | -2.88127        | -2.87207        |
| 10% Critical Value| -2.57729        | -2.57238        |
As shown in Table 1, for the residuals of LSTM and BPNN, the p value is less than 0.05, and the ADF Test statistic value is less than 1% Critical Value and 5% Critical Value and 10% Critical Value. So both of their residuals time series are stable.

The correlation of residuals itself over the distribution of time can be found by analyzing the autocorrelation coefficient—ACF and the partial correlation coefficient—PACF[16][17] of residuals-LSTM and residuals-BPNN.

![ACF and PACF of residual-LSTM](image1)

**Figure 3. ACF and PACF of residual-LSTM**

In the ACF and PACF graphs, the blue area is 2 times the standard deviation of the correlation coefficient and the horizontal axis and the value $x$ on the horizontal axis represents the historical moment of $x$ time units from the current time, it is also named lags. The vertical axis represents the correlation coefficient, which ranges in $[-1,1]$.

Analysing the correlation coefficient within 30 lags, for the residuals of LSTM, the highest autocorrelation of lags are 7, 14, 17, 21, 28 lags. The highest partial autocorrelation of lags are 7, 14, 21, 25, 28. Their intersection set of lags is 7, 14, 21, 28. There are obvious periods of 7 time units.

For the residual of BPNN, the highest autocorrelation of lags are 1, 2, 7, 14, 17, 21, 26, 28 lags. The highest partial autocorrelation of lags are 1, 2, 7, 14, 17, 21, 26 lags. Their intersection set of lags is 1, 2, 7, 14, 21, 26. There are periods of about 7 time units which is not as obvious as LSTM residuals.

From the above verification, the residual distribution of BPNN and LSTM is not disorderly. It can be observed that there are obvious time series characteristics. So we can use the residual as another set of time series data to predict.

5. **Ensemble Neural Networks**

The building of ensemble neural network is based on the method of Bagging and Stacking in Ensemble Learning. Bagging method is to train multiple learning models in parallel and calculate the average after summarizing the results. Stacking method is also to train multiple models. The first model is trained based on original training data. The first learning model is also called the Base layer model. The second model is trained based on the results of the first model. The second learning model is called Meta Layer model. Ensemble learning model improves to be a stronger learner by integrating multiple learning models with Bagging and Stacking methods.

The structure of ensemble neural network is shown in Figure 4.

![Structure of Ensemble Model](image2)

**Figure 5. Structure of Ensemble Model**
First, the base layer neural network is used to fit the data, then calculating the residuals between the results and the observed values. Second, the meta layer neural network is used to fit these residuals. Finally, adding the results of the meta layer and the output of the base layer as the final result. In this process, the self-fitting of the residuals in the ensemble model is realized. After obtaining the results of two stacking models and calculating the geometric average and arithmetic average of them, then choosing the best average method.

Figure 6. Process of predicting current value

The training and testing process of Ensemble neural networks is shown as follows.

Figure 7. Training and testing process

Figure 8. LSTM predicting on residual-BPNN

Figure 9. LSTM predicting on residual-LSTM

The blue line represents the real residual value between the predicted values of Base layer model and observed values of 283 test samples. The red line represents the residual value predicted by Meta layer model based on historical residuals which is generated by Base layer model.

In Figure8, the Base layer model is LSTM. The Meta Layer model is also LSTM. And the MAPE is 15.2%. In Figure9, the Base layer model is BPNN. The Meta Layer model is LSTM. And the MAPE is 17.8%.

Comparing with real residuals in these two graphs, the distribution of predicted residuals is closer to the horizontal axis, and their absolute value is less than absolute value of real residual. As predicted residuals, it is used to compensate the error between predictive values and observed values. So it is proper to be not too large in case of over compensating.
After getting the predicted results of LSTM+LSTM and BPNN+LSTM stacking model, then calculating the arithmetic average and geometric average.

![Figure 10. Results of average](image)

The MAPE of arithmetic average is 3.87% and of geometric average is 4.46%. So Arithmetic average method is better than geometric average method in this experiment.

| Model                        | MAPE   |
|------------------------------|--------|
| LSTM                         | 5.19%  |
| BPNN                         | 6.15%  |
| LSTM+LSTM                    | 4.97%  |
| BPNN+LSTM                    | 5.01%  |
| Ensemble model(Geometric average) | 4.46%  |
| Ensemble model(Arithmetic average) | 3.87%  |

### 6. Conclusion

The LSTM neural network performs better than traditional models in power system load forecasting. To improve the LSTM network, LSTM+LSTM and BPNN+LSTM stacking models are designed with the ensemble method. The residuals predicted by Meta layer LSTM can compensate error between the Base layer predicted values and observed values by self-fitting based on historical residuals time series. More accurate results were obtained by calculating the average of the two stacked models using the Bagging method. Finally, the MAPE of ensemble neural network is 3.87%, which is 1.32% lower than the MAPE of LSTM.

### References

[1] Juberias G, Yunta R, Garcia M J and Mendivil C 1999 Proc. Int. Conf. on Transmission and Distribution (New Orleans) vol 1 (New York: IEEE) pp 314-319

[2] Sinha A K 2009 Proc. Int. Conf. on Industrial Technology (Victoria) vol 1 (New York: IEEE) pp 548-553

[3] Guangye L, Wei Li, Xiaolei Tian and Yifeng Che 2017 Smart Grid 7 274.

[4] Türkay B E and Demren D 2012 Electrical load forecasting using support vector machines (New York: IEEE) pp 49-53

[5] Barbounis T G, Theocharis J B and Alexiadis M C 2006 IEEE Trans. Power Appar. Syst. 21 273-84.
[6] Marino D L, Amarasinghe K and Manic M 2016 Proc. Int. Conf. on Industrial Electronics Society (Florence) vol 1 (Washington DC: IEEE Computer Society) pp 7046-7051

[7] Zheng J, Xu C, Zhang Z and Li X 2017 Proc. Int. Conf. on Information Sciences and Systems (Washington DC) vol 1 (New York: IEEE) p 55

[8] Kong W, Dong Z Y and Jia Y 2017 IEEE T. SMART GRID 1 1.

[9] Kolen J F and Kremer S C 2001 Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies (New York: Wiley-IEEE Press) pp 237-243.

[10] Hochreiter S and Schmidhuber J 1997 Proc. Neural, Parallel, Sci. Comput., [Int. Conf.], 2nd. pp 1735-1780.

[11] Jozefowicz R, Zaremba W and Sutskever I 2015 Proc. Int. Conf. on Machine Learning (Lille) vol 3 (New York: IMLS) pp 2342-2350.

[12] D’Informatique D E, Ese N and Esent P 2001 Long Short-Term Memory in Recurrent Neural Networks (Switzerland:Epfl) 9 pp 1735 - 1780.

[13] Graves A 2013 Generating Sequences With Recurrent Neural Networks (New York: Computer Science)

[14] Cheung Y W and Lai K S 1995 J. Bus. Econ. Stat. 13 277-80.

[15] Mushtaq R 2011 Augmented Dickey Fuller Test Social Science Electronic Publishing 11

[16] Hagan M T and Behr S M 1987 Proc. Int. Conf. on Power Engineering Soc(New Orleans) vol 2 (New York: IEEE) pp 785-791

[17] Schwab H 1996 Introduction to Time Series and Forecasting. 2nd ed. (Berlin:Springer)