Research on Power Load Forecasting and Visualization Method Based on Deep Neural Network

Yifei Wang\textsuperscript{1}, Xian Li\textsuperscript{1}, Shuang Mo\textsuperscript{2,*}

\textsuperscript{1}State Grid Jibei Information and Telecommunication Company, Beijing, China.
\textsuperscript{2}School of Computer, Beijing University of Posts and Telecommunications, Beijing, China.

*Corresponding author e-mail: moshuang@bupt.edu.cn

Abstract. Short-term load forecasting (STLF) has attracted much attention in the last decade, which uses data from historical power grid operation monitor to forecast future power demand for a few days. Previous works on STLF mainly focus on features, which are widely used to capturing external factors in power system. There are also approaches based on physics, which use linear combination of active and reactive power consumption to model dynamics of power demand. However, in many real-world applications they don’t work well. In the self-service power grid, STLF is a very important part in power scheduling and planning. But STLF is a challenging task for its complex power demand and the effects of weather. In this paper, we study the problem of combination of complex and seasonality of power demand, as well as effect of external weather. Therefore, we come up with an approach, called DNNCast, it’s a model based on deep neural network, which can achieve a good accuracy and fast training speed. In the structure it uses a deep neural network to model various power loads and external weather conditions. Finally, with visualization tools we can wisely make decision about power scheduling and planning. Compared with the current STLF methods, our model has a better prediction effect, and faster in the experiments of the extended experiment on 133 real power datasets.

1. Introduction

In the self-service power grid, when considering changes not only in weather conditions, such as extreme weather including a sharp rise in temperature or precipitation, but also a large number of electricals’ loads to estimate the power consumption of a single station or multiple stations over the next few days. The situation is namely called short-term load forecasting (STLF). It’s a key step in future power load scheduling and planning, and a good predictive performance can improve the stability, overhead and emissions of the self-service power grid. However, power time series forecasting is more challenging than general time series:

\begin{itemize}
  \item \textbf{Complex demand of electrical loads}. Due to various power-factor in the real situation, the power demand is very complex, such as active power and inductive power.
  \item \textbf{Seasonality of power demand}. There are seasonal patterns of power consumption during the time of a day, so we can use seasonal smoothing method to capture them.
  \item \textbf{Effect of external conditions}. Such as the changes of temperature, and precipitation have a strong correlation of power demand, so they can be features to feed into the model.
\end{itemize}
Figure 1. (a) DNNCast forecasts more accurately compared to competitors in different length. (b) The visualization of DNNCast.

Current power load prediction methods are mainly feature-based methods such as support vector regression (SVR), which captures external factors in the power system or physics-based models, which models the dynamics of the power system. However, their predictions’ effects are not as good as expected. Therefore, we proposed a deep neural network-based forecasting method, DNNCast, which can model the complex requirements of active and reactive power from different levels of load and the impact of the external environment to achieve short-term load forecasting. Our method firstly models time series from power system and external weather conditions by deep neural network to obtain a primary model, and then further use seasonal smoothing method to extend the time series. Finally, the visualization tool is used to show the predicted results to assist stuffs in scheduling decisions. In summary, the following are our contributions:

• **A deep neural structure**: We come up with a valid deep neural network structure to deal with power grid data.

• **Accuracy & Speed**: We almost has the best performance between state-of-art STLF methods, and in the same accuracy range we are the fastest, as we can see from Figure 1(a).

• **Visualization**: We combine STLF task and visualization tool, with them stuffs of the power system can make power planning and scheduling wisely, like Figure 1(b).

2. Related Work

Short-time load forecasting (STLF) has been well studied in past years, and STLF is more about time series forecasting. In order to make use of seasonality of time series, some prediction methods have been raised, for example, seasonal ARIMA[1], Holt-Winters[2], TBATS[3] and RNN, including LSTM and GRU, has recently been used as a black-box method for STLF tasks[4]. However, when RNNs are dealing with long series[5] there are some defects, whose length is up to 10,000 in the power series. In addition to standard prediction methods, which are based on feature are commonly used to capturing factors outside power systems, such as linear regression (LR)[6], artificial neural networks (ANNs)[7], and support vector regression (SVR)[8, 9]. Soft computing techniques such as the fuzzy logic method[10] constructs a logic rules set to capture complicated mathematical relationships between outputs and features. Models which are based on physical[11, 12, 13] use a linear combination of active power and reactive power to simulate power demand dynamics. Besides, PowerCast[14] and NeuCast[15] use potential factor models to capture seasonality, while NeuCast achieves the better performance. However, NeuCast is slow in training, which matters when take real-time demand into consideration. Compared those methods, our model additionally takes external factors into consideration, and uses a deep neural network to model the complicated power consumption and external weather conditions. When consider the accuracy and speed trade-off problem, our model does the best job.
3. Proposed Model

Short-term load forecasting pays attention to the changes of the future load during short period, which may be from half an hour, up to 7 days. The definition of short-term load forecasting question shows as following:

- **Given:** historical monitoring data of power grid, i.e. time series \( X(t) = [x_1(t), x_2(t)] \). \( x_1(t), x_2(t) \) are respectively active and reactive power \( (P(t), Q(t)) \) of previous \( t \) time points \( (t = 1, \ldots, N) \).
- **Forecast:** the power demand for \( N_f \) steps in the future (i.e., predict \( X(t) \), for \( t = N + 1, \ldots, N + N_f \)).

We will represent time series including active and reactive power as \( X(t) \), so as to forecast power demand in the future.

![An illustration of the structure of DNNCast.](image)

**Figure 2.** An illustration of the structure of DNNCast.

3.1. Framework Overview

The framework of model is shown as Figure 2, which has three main components: deep neural network modeling, seasonal smoothing, forecasting and visualization. Variables \( i \) and \( j \) represent the index of day and hour in total days and hours. With embedding operation which is a single-layer neural network, we embed index value to learn feature matrices \( D \in \mathbb{R}^{N_d \times R} \) and \( H \in \mathbb{R}^{N_h \times R} \), where \( N_d \) is the length of days, \( N_h \) is length of hours \( (N_d = 24) \), \( R \) is the dimension of embedded latent space. And then we concatenate the two embedded output vectors as input of following DNN (i.e. a deep neural network, which is multi-layer perceptron) layer. For the external factors (i.e. temperature and rainfall), they are real value in specific day and hour, so we handle them into a tensor \( E \in \mathbb{R}^{N_d \times N_h \times N_e} \), where \( N_e \) is the number of external conditions. And then we directly feed them into another DNN layer. After that, we concatenate the two DNN layers’ output as input of a new DNN layer. The final DNN layer outputs time series \( \mathcal{X} \in \mathbb{R}^{N_d \times N_h \times N_p} \), which contains the target values (i.e. active and reactive power), where \( N_p \) is the types of target value. However, we cannot predict the future days’ power for the day embedding of the future days are only initialized without learning. To address this, we use the seasonal smoothing to predict the future \( N_{fd} \) days’ weights, so as to get the extended day
embedding matrix $D' \in \mathbb{R}^{(N_d+N_{fd}) \times R}$. Finally, we can forecast the future days accurately and make a good decision of power planning and scheduling by visualization.

As shown as Algorithm 1, in order to capture external factors and complex demand of power load, deep neural network modeling factorizes various power load into variables in the hidden space. Moreover, we also integrate external factors with a DNN layer. After that we extend day embedding weights by seasonal smoothing and then make a prediction.

**Algorithm 1.**

| Algorithm1 |
|---|
| **Input:** Power grid time series X(1 : N); External factors E(1 : N + $N_f$) |
| **Output:** X(N : N + $N_{fd}$) |
| 1: Construct $\mathcal{X}$ with X(1 : N) |
| 2: Construct $\mathcal{E}$ with E(1 : N + $N_f$) |
| 3: random initialize $\mathbf{D}$, $\mathbf{H}$, $\mathbf{W}$, $\mathbf{b}$ |
| 4: $\mathbf{D}' = \mathbf{D}$ |
| 5: repeat |
| 6: $\mathbf{D} = \mathbf{D}'$ |
| 7: $\mathbf{D}$, $\mathbf{H}$, $\mathbf{W}$, $\mathbf{b}$ = DNN modeling by optimizing Eq.(5) |
| 8: $\mathbf{D}' = [\mathbf{D}, \mathbf{D'}]^T$, seasonal smoothing to extend D |
| 9: until maximum epochs K or $\mathbf{D}' \equiv \mathbf{D}'$ |
| 10: Forecasting $\mathcal{X}$ using Eq.(6) and Visualizing |

### 3.2. Deep Neural Network Modeling

In the Deep Neural Network Modeling step, firstly we use a single-layer full connected neural network to embed the day and the hour into vectors and learn matrices $\mathbf{D}$ and $\mathbf{H}$. After that we concatenate the two vectors to be input of a MLP (i.e. Multi-Layer Perceptron), at the same time external conditions are feeding into another MLP. Afterwards, we concatenate outputs of the two MLPs and feed into a new MLP. Finally, we predict the result $\hat{x}_{ij}$ which is the output of the new MLP.

The followings are the formulas in the deep neural network modeling.

$$\hat{x}_{ij} = f(\mathbf{D}^T o_i^D, \mathbf{H}^T o_j^H, e_{ij}|\mathbf{D}, \mathbf{H}, \mathbf{W}, \mathbf{b})$$  \hspace{1cm} (1)

where $o_i^D$ and $o_j^H$ are the one-hot encoding of $i$-th day and $j$-th hour. $\hat{x}$ is the output vector which contains target power value. $e_{ij}$ is external condition vector in date $i$ and hour $j$. For the MLP, we define functions $\phi$ and $\varphi$ for two types of MLP.

$$\phi(u, v) = a_1(W_1^T(u + v) + b_1)\ldots + b_L$$  \hspace{1cm} (2)

$$\varphi(u) = a_1(W_1^T(u) + b_1)\ldots + b_L$$  \hspace{1cm} (3)

where $W_L$, $b_L$, and $a_L$ are respectively weight matrix, bias vector and activation function of the $l$-th layer’s perceptron, $\oplus$ means the concatenation of two vectors.

Finally, the model can be summarized as

$$\hat{x}_{ij} = \phi^{FU}(\phi^{MF}(\mathbf{D}^T o_i^D, \mathbf{H}^T o_j^H), \varphi^{EXT}(e_{ij}))$$  \hspace{1cm} (4)

where $\phi^{FU}$, $\phi^{MF}$ and $\varphi^{EXT}$ are respectively the functions of DNN layers.

The model is trained by minimizing mean squared error between $\hat{x}_{ij}$ and $x_{ij}$.

$$L_{sqe} = \sum_{i=1}^{N_d} \sum_{j=1}^{N_{fd}} ||x_{ij} - \hat{x}_{ij}||^2 + \lambda(||\mathbf{D}||^2 + ||\mathbf{H}||^2) + \omega(||\mathbf{W}||^2 + ||\mathbf{b}||^2)$$  \hspace{1cm} (5)

### 3.3. Seasonal Smoothing

In this step, we use seasonal smoothing method to extend time series D for $N_{fd}$ days longer. For the smoothing method, we only use Holt-Winters for its effectiveness. And then we get the extended time series $D' = [D, \tilde{D}]^T$, where $\tilde{D}$ is the extended $N_{fd}$ rows.
3.4. Forecasting & Visualization

In this part, we make predictions and then visualize them.

In forecasting step, we feed extended $\mathbf{D}$ into the neural network to forecast.

$$\hat{x}_{ij} = \phi_{FU}(\phi_{MF}(\mathbf{d}_i, \mathbf{h}_j), \phi_{EXT}(\mathbf{e}_{ij}))$$  \hspace{1cm} (6)

where $\mathbf{d}_i$ is the $i$-th row of $\mathbf{D}$, and $\mathbf{h}_j$ is the $j$-th row of $\mathbf{H}$.

In the visualization step, we use visualization tool (i.e. Echarts) to realize the visualization task. To enrich the charts, we also compute apparent power with active and reactive power of predicted results by the following formula:

$$S = VI = \sqrt{P^2 + Q^2}$$  \hspace{1cm} (7)

where $S$, $P$, $Q$ respectively denotes apparent, active and reactive power. $V$, $I$ is voltage and current.

4. Experiments

We conduct experiments to answer the following research question:

- **RQ1**: How about the accuracy and training speed of our model predicting on real data from power grid?
- **RQ2**: How does our visualization component help to make a power scheduling decision?

4.1. RQ1

4.1.1. Dataset

We use data extracting from the databases of power system, whose fields contains date, hour, location id, active and reactive. Besides, we also integrate external weather such as temperature and rainfall. It contains 133 location and 328 day-long.

4.1.2. Evaluation Protocols

We preserve the last 5-day as test set, for the remaining randomly split them into training (90%) and validation (10%) sets. To evaluate the performance, we use the following evaluation metrics (the less value means the best):

- **RMSE**: $\text{RMSE}(x, \hat{x}) = \sqrt{\frac{\sum_{i=1}^{N} |x - \hat{x}|^2}{\sum_{i=1}^{N} |x|^2}}$
- **SMAPE**: $\text{SMAPE}(x, \hat{x}) = \frac{2}{N} \sum_{i=1}^{N} \frac{|x - \hat{x}|}{|x| + |\hat{x}|}$
- **MAE**: $\text{MAE}(x, \hat{x}) = \frac{1}{N} \sum_{i=1}^{N} |x - \hat{x}|$
- **DTW**: Dynamic Time Warping, which is a widely used distance metric in time series.

4.1.3. Baselines

We choose following models as our baselines:

- **AR**: ARIMA is a auto-regression method, whose parameters choosing with Akaike information criterion.
- **SAR**: Seasonal ARIMA is also a auto-regression method and parameters choosing with Akaike information criterion.
- **SVR**: Support Vector Regression is a feature-based regression method, which uses RBF kernel and the error penalty term is set to 1.
- **GBRT**: Gradient Boosted Regression is a feature-based regression method, which uses 100 trees and the learning rate is set to 0.1.
- **PowerCast**: We choose the parameter settings suggested by the paper, whose seasonal smoothing method is seasonal ARIMA.
- **NeuCast**: The parameters are same as in the paper, and the seasonal smoothing method is also Holt-Winters.
4.1.4. Parameter Settings

To fairly compare models’ capability, we learned all models by optimizing mean square loss error (Equation 5). For the deep neural model, learning rate is 0.001, the batch size is 128, the epoch is 100, DNN layers are both with one hidden layer of 16 neurons. For seasonal smoothing method we only use Holt-Winters.

4.1.5. Results

As shown as Table 1, although our accuracy is a bit less than Neucast, but our training speed is faster within 16% improvement when forecasts 5 days, for the more we keep faster in different day-length of a location as Figure 3. This is a trade-off problem, our model is best when considering more data and real time conditions.

| Methods     | RMSE  | SMAPE | MAE   | DTW |
|-------------|-------|-------|-------|-----|
| AR          | 0.4166| 0.4803| 9.321 | 7.5 |
| SAR         | 0.3954| 0.5175| 8.659 | 6.509 |
| SVR         | 0.3114| 0.4203| 7.16  | 5.116 |
| GBRT        | 0.3044| 0.377 | 6.82  | 4.868 |
| PowerCast   | 0.2590| 0.3632| 5.686 | 4.070 |
| NeuCast     | 0.2120| 0.2706| 4.170 | 2.972 |
| DNNCast     | 0.2176| 0.2813| 4.354 | 3.076 |
| Impv1.      | 15.9% | 22.5% | 23.4% | 24.4% |
| Impv2.      | -2.6% | -3.9% | -4.4% | -3.4% |

Table 1. The average performance of different methods on 133 locations real data from power grid. Impv1. means compared with PowerCast, Impv2. means compared with Neucast.

Figure 3. The time cost of DNNCast and NeuCast in predicting 25 days.
4.2. \textit{RQ2}

As we can see from Figure 4, it is the predicted results of power grid location 26, from which we can see the trend of power in the future. Besides, we can directly get which hour of a day the max apparent power will be in the future, it will be 170.82 KW in 81th hours of the future. For the more, we can change from different views (active power, reactive power and apparent power) by clicking the button on the top. Finally, we can also get the average power, which means it is easy to compute the total demanded power by average value times hours. And then with this information we can draw a conclusion about power planning and scheduling more wisely.

![Figure 4. The application of DNNCast. (a)Active power. (b)Reactive Power. (c)Apparent Power.](image)

5. Conclusion

We propose a deep neural network forecast method to realize STLF task, and use visualization tool to show the result so as to help make decision in power planning and scheduling. According extended experiments, our model achieve a quite accurate prediction and a fast training speed. For the future work, we will try to apply our model to real application and even embedding into power grid system.

6. Acknowledgments

This paper is funded by the State Grid Jibei Power Co. Ltd. Science and Technology Project “Research on Data Fusion and Visualization Key Technologies Based on Full-service Unified Data Center” (52018E18006N).

References

[1] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. Time series analysis: Forecasting and control, 5th edition. Journal of the Operational Research Society, 22(2):199-201, 2015.

[2] G Sudheer and A Susee-latha. Short term load forecasting using wavelet transform combined with holt-winters and weighted nearest neighbor models. International Journal of Electrical Power & Energy Systems, 64:340-346, 2015.

[3] Alysha M De Livera, Rob J Hyndman, and Ralph D Snyder. Forecasting time series with complex seasonal patterns using exponential smoothing. Journal of the American Statistical Association, 106(496):1513-1527, 2011.

[4] Jian Zheng, Cencen Xu, Ziang Zhang, and Xiaohua Li. Electric load forecasting in smart grids using longshort-term-memory based recurrent neural network. In Information Sciences and Systems (CISS), 2017 51st Annual Conference on, pages 1-6. IEEE, 2017.

[5] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm.1999.

[6] Grzegorz Dudek. Pattern-based local linear regression models for short-term load forecasting.130:139-147,2016.

[7] Arjun Baliyan, Kumar Gaurav, and Sudhansu Kumar Mishra. A review of short-term load forecasting using artificial neural network models. Procedia Computer Science, 48:121-125, 2015.

[8] A Selakov, D Cvijetinovi´ c, L Milovi´ c, SMellon, and DBekut. Hybridpso-svm method for short-term load forecasting during periods with significant temperature variations in city of Burbank. Applied Soft Computing, 16:80-88, 2014.
[9] Piyali Ganguly, Akhtar Kalam, and Aladin Zayegh. Short term load forecasting using fuzzy logic. In International Conference on Research in Education and Science, 2017.

[10] Guo-Feng Fan, Li-Ling Peng, Wei-Chiang Hong, and Fan Sun. Electric load forecasting by the SVR model with differential empirical mode decomposition and auto regression. 173:958-970, 2016.

[11] M Jereminov, A Pandey, HA Song, B Hooi, C Faloutsos, and L Pileggi. Linear load model for robust power system analysis. IEEE PES Innovative Smart Grid Technologies (2017, p.(submitted)) Google Scholar, 2017.

[12] Hyun Ah Song, Bryan Hooi, Marko Jereminov, Amritanshu Pandey, Larry Pileggi, and Christos Faloutsos. PowerCast: Mining and forecasting power grid sequences. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 606-621. Springer, 2017.

[13] Amritanshu Pandey Marko Jereminov Larry Pileggi Bryan Hooi, Hyun Ah Song and Christos Faloutsos. Streamcast: Fast and online mining of power grid time sequences. In The 2018 SIAM International Conference on Data Mining, 2018.

[14] Hyun Ah Song, Bryan Hooi, Marko Jereminov, Amritanshu Pandey, Larry Pileggi, and Christos Faloutsos. PowerCast: Mining and forecasting power grid sequences. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 606-621. Springer, 2017.

[15] Pudi Chen, Shenghua Liu, Chuan Shi, Bryan Hooi, Bai Wang and Xueqi Cheng. NeuCast: Seasonal Neural Forecast of Power Grid Time Series. In the International Joint Conference on Artificial Intelligence, 2018.