Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 122 (2017) 308–314
Procedia Computer Science
www.elsevier.com/locate/procedia

Information Technology and Quantitative Management (ITQM 2017)

Load Forecasting via Deep Neural Networks

Wan He[a,1]

[a] State Grid Energy Research Institute, State Grid Corporation of China, Beijing, 102209, China

Abstract

Nowadays, electricity plays a vital role in national economic and social development. Accurate load forecasting can help power companies to secure electricity supply and scheduling and reduce wastes since electricity is difficult to store. In this paper, we propose a novel Deep Neural Network architecture for short term load forecasting. We integrate multiple types of input features by using appropriate neural network components to process each of them. We use Convolutional Neural Network components to extract rich features from historical load sequence and use Recurrent Components to model the implicit dynamics. In addition, we use Dense layers to transform other types of features. Experimental results on a large data set containing hourly loads of a North China city show the superiority of our method. Moreover, the proposed method is quite flexible and can be applied to other time series prediction tasks.

Peer-review under responsibility of the scientific committee of the 5th International Conference on Information Technology and Quantitative Management, ITQM 2017.

Keywords: Load Forecasting; Convolutional Neural Network; Recurrent Neural Network; Deep Learning

1. Introduction

At present, electricity plays an important role in economic and social development electricity plays an increasingly important role in economic development, industrial production and everyday lives of ordinary people. One of the major characteristics of electricity is that it is difficult to store once it has been produced. Moreover, short term power load demand varies greatly and is subject to a number of factors. Therefore, accurate load forecasting is critical for power companies, so as to reduce electricity waste, improve revenue and maintain the stable operation of the power grid system.

A wide variety of approaches have been proposed to study load forecasting. Mbamalu and El-Hawary [1] considered load forecasting as an autoregressive process and used iteratively re-weighted least-squares (IRWLS) procedures to estimate model parameters. And Haida and Muto [2] also presented a regression based method with a transformation technique to predict daily peak load. Stochastic time series models have also been employed, since the list of power load data is actually a time series. Chen et al. [3] developed an adaptive autoregressive moving-average (ARMA) model to conduct 24-hours- and one-week-ahead load forecasts. While Juberias [4] used the Autoregressive Integrated Moving-Average (ARIMA) model to build a practical system for real time short term load forecasts in Spain.

*Wan He. Tel.: +86-10-6660-3713 ; fax: +86-10-6660-3573 .
E-mail address: streamlethe@126.com.

1877-0509
Peer-review under responsibility of the scientific committee of the 5th International Conference on Information Technology and Quantitative Management, ITQM 2017.
10.1016/j.procs.2017.11.374

Ever since the 1990s, statistical machine learning technology has developed quickly, and different machine learning methods have been applied in load forecasting as well as other economic prediction tasks. Guo et al. [5] proposed to use support vector machine for modeling the nonlinear influencing factors in load forecasts. Lahouar and Slamab [6] used the random forest to integrate various features such as customer behaviors, load profile and special holidays in one day ahead load prediction. And Gaussian Process based approaches [7] [8] have been introduced in energy and load forecasting.

Several competitions about load forecasting have been organized and have attracted a lot of attention. Fifty-six research groups participated in the 2001 EUNITE load forecast competition [7]. Global Energy Forecasting Competition 2012 (GEFCom2012) attracted hundreds of participants from worldwide universities and companies [9]. And participants were required to backcast and forecast hourly loads for a US utility with 20 zones at both the zonal and system levels. And in the year 2014 a second Global Energy Forecasting Competition (GEFCom2014) which required competitors to forecast the quantiles of hourly loads for a US utility on a rolling basis was held [10].

Neural Network (NN) technology is an important branch of statistical machine learning and has been frequently used in various kinds of forecasting tasks. NNs are extremely good at modeling the non-linearities in data of many fields and have theoretically provable capability to approximate any complex functions with arbitrary precision. Hippert et al. [11] critically examined a collection of papers that tried out NNs in short term load forecasting. And Jetcheva et al. [12] presented an NN based ensemble model for day-ahead building level load prediction. In addition, since the remarkable success in the ILSVRC2012 challenge, the research and application of Deep Neural Networks (DNN) and Deep Learning [13] have been very hot in a number of domains, including computer vision, natural language processing, speech recognition and signal processing, etc. Then He [14] introduced Deep Feed-forward Networks to improve the performance of load forecasting by focusing on pre-training and parameter optimization. Din and Marnerides [15] utilized the time-frequency feature selection procedure and compared the accuracy of both Deep Feed-forward and Deep Recurrent Networks. And Marino et al. [16] showed the superiority of Long Short Term Memory (LSTM) [17] based Sequence to Sequence model to standard LSTM based Recurrent Networks in predicting one-minute step load data.

In this paper we propose a novel Deep Neural Network based architecture to perform one day ahead hourly load forecasting. Our idea is to use different types of neural network components (or modules, or layers) to model different types of factors that may impact load consumption. We borrow the approach in modern image recognition [18] and use multiple Convolutional Neural Network (CNN) components to learn rich feature representation from historical load series. Then we model the variability and dynamics in historical loading using LSTM based Recurrent Neural component. As for other features such as temperature and holiday, we use dense (Feed-forward) component to project them into vector representations. And finally, we concatenate all learned features through dense layers to predict load value. We evaluate our method on a data set consists of about 3 years of hourly load data in a north China city. And experiment results demonstrated the advantage of our method.

The rest of this paper is organized as follows. In Section 2, we analyze the characteristics of power load series and various factors that may impact load forecasting. Then we elaborate our methods in Section 3. After that, we conduct experiments and show evaluation results in Section 4. Finally, we conclude our work in Section 5.

2. Feature Analysis

Electricity is consumed in people's production and everyday life. So in essence, anything related to the production and living pattern of people may influence power load demand. In this section, we analyze several important types of factors that have significant impacts on power consumption, so that we can obtain some intuition of how each of them influence load forecasting. Moreover, we decide how to include each type of feature into our Deep Neural Network model based on these analyses.

The short term load data of a specific region have significant approximate periodicity. And this is due to the regular work and life mode of people. Figure 2(a) illustrates how hourly load value varies periodically from June 15 to June 21 in our year 2011 data. And in figure 2(b) we can see that the curves of load data from three adjacent weeks (July 2 to July 22, 2012 in our data set) change and match each other accordingly. So historical load usage data are strong indicators for load prediction. Thus in this paper, we use historical loads of up to one week to predict the day ahead hourly load.

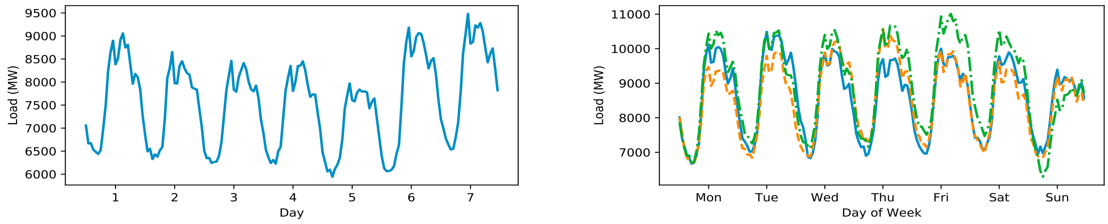

Fig. 1. (a) Daily Periodicity; (b) Weekly Periodicity.

In addition to periodicity, the historical load sequence also exhibits variability and has dynamic properties. To get rich features from the sequence, we use Convolutional Neural Network modules to learn multiple feature maps. Additionally, we use Recurrent Neural Network to model the temporal dynamics in historical load series.

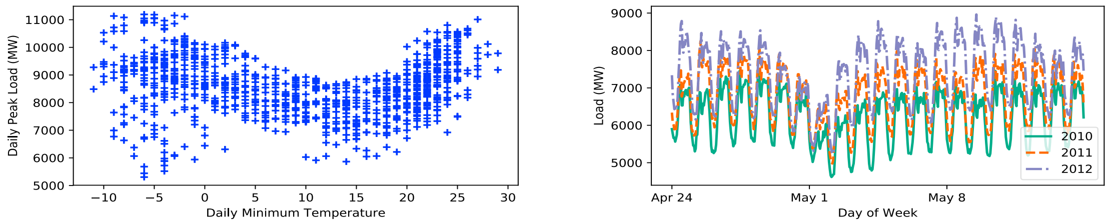

Fig. 2. (a) Influence of Temperature; (b) Holiday Effect.

Weather can largely affect power usage since it has great influence on people's work. The effect of temperature is clear that in uncomfortable temperature people tend to use more electricity to heat or cool, or have more in-house activities. From figure 2(a) we can see that the correlation between daily minimum temperature and daily peak load is quite strong. Other weather factors such as humidity and wind speed may also be very helpful for load forecasting, but we have no such information in our data set.

On holidays, people's behavior patterns are generally quite different from usual. So public holidays, especially long holidays, have an impact on power demand. We show the power usage curves near May 1 (the International Workers' Day) in figure 2(b). And we can find that every year, the load usage in days near May 1 is notably lower than the adjacent weeks.

Furthermore, to predict hourly load, the hour itself and the day of week of the targeting hour are obviously useful indicators. So we include these 2 types of features in our model. The maximum and minimum temperatures, whether a holiday, the hour of day and the day of week features are employed as inputs to the Deep Feed-forward component of our model.

There are potentially other kinds of features for load forecasting, for example, electricity price and energy policies. However, currently we do not pay attention to such factors in this paper.

3. Deep Neural Network for Load Forecasting

In this section, we elaborate our sophisticated method that combines multiple types of Deep Neural Network (DNN) components. There are different types of input features to our load forecasting task. So we use appropriate neural network components to learn the deep representation and extract rich features from the specific input. To do this, we benefit from the fast development of Deep Learning and the flexible and easy to use toolkit Keras [19]. Figure 3 demonstrates the basic architecture of our model.

In our opinion, the historical load sequence is the most important input which contains rich information for predicting future load demand. As shown in figure 3, we use multiple parallel Convolution Neural Network (CNN) components to process the historical load data. Traditionally, people tend to manually extract various

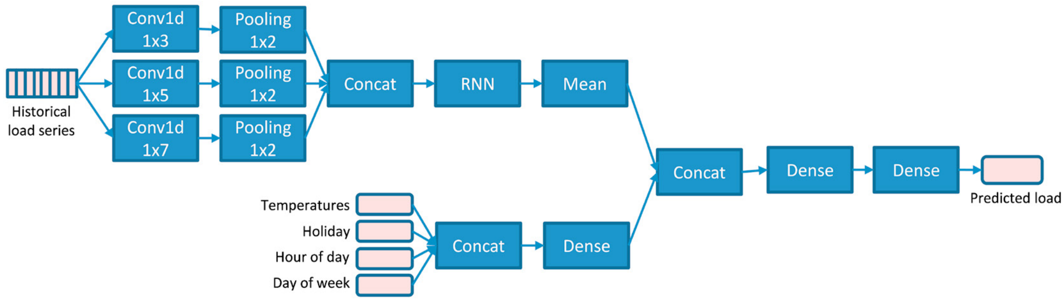

Fig. 3. Model Architecture

features from the load sequence. The extracted features are usually statistics of the raw data, such as maximal load on the previous day or load at the same hour of the previous day, etc. However, in current Deep Learning applications, we use the DNNs to automatically learn the feature representation from the raw data. Typically, CNNs are used in the first layers of a DNN model to perform feature learning and feature extraction. And using kernels with locally connected receptive fields, they usually act as filters to transform the input signals. Thus they can learn various features from the raw input. For example in computer vision tasks, lower layer CNNs usually identify features such as edges, corners, and patches, while higher layers CNNs can extract more abstract features like dogs in a scene and arms of a human, etc [18]. CNNs have also been successfully applied in extracting features from 1-dimensional signals such as human speech and electroencephalogram data [13]. So here we employ CNNs to transform the historical load series to get a variety of features for the subsequent load prediction.

By using multiple parallel CNN components with different filter size we introduce parallel structure into the DNN model, instead of just stacking DNN layers in a cascade manner. In this way, the DNN model can learn even richer features from the input historical load sequence. The Google Inception model [18] demonstrated the effectiveness of parallel CNNs in image recognition. And this approach is also widely used in Natural Language Processing applications where parallel CNNs structures extract features from input sentences.

The historical load sequence, as well as the features extracted by the CNN components are time series which have rich temporal dynamics. Recurrent Neural Networks (RNN) are very effective in modeling dynamics in sequential data, since they have the ability to remember the dynamics in variable length of previous inputs in its memory. However, when modeling long sequences, simple RNN structure suffers from the gradient vanishing and explosion problem in its back propagation based model training [17][13]. So RNNs with the gated recurrent unit called Long Short Term Memory (LSTM) [17] are applied in practical applications. LSTM can avoid the vanishing and exploding gradient problems and has become very successful in modeling long term dependencies in temporal data.

The Recurrent Neural Network models its input sequence $\{x_1, x_2, ..., x_n\}$ using the recurrence:

$$h_t = f(h_{t-1}, x_t), \tag{1}$$

where x_t is the input at time t, and h_t is the hidden state which can be considered as a vector representation of all inputs seen up to time t. In order to solve the gradient vanishing or explosion problem, LSTM introduces gates into the recurrence function f and computes the hidden state as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t * \tanh(C_t), \tag{7}$$

where i_t , f_t , and o_t are the input gate, forget gate, and output gate respectively; C_t is cell state and $\tilde{C}_t$ represents candidate cell state; and the W's and B's are parameters of the LSTM unit. By feeding the input sequence $\{x_1, x_2, ..., x_n\}$ into the RNN, we get a list of hidden corresponding states $\{h_1, h_2, ..., h_n\}$ as the outputs. And the mean of these hidden states will be used as the feature representation of the entire input sequence.

Besides the historical load series, we also incorporate other types of features into our forecasting model, including weather information, holidays and the specific hour and day of the time for the prediction. We collect the daily maximum and minimum temperatures for our load data set, and directly use the temperatures as two dimensions of numerical features. We can also obtain the public holidays of each year. And we use them as a binary feature where value 1 indicates that the day is a holiday while 0 means not a holiday. As for the hour itself and the day of week of the hour for prediction, we believe that they should be considered as categorical features and be represented using the one-hot method as vectors of 24 and 7 dimensions, respectively. For example, if we want to predict the hourly load from 13:00 to 14:00 on some Friday, the length of the vector for hour of day should be 24 and the value at index 12 (starting from zero) will be 1 while other values are all zeros; and similarly the vector for day of week is a vector of length 7 which has its only non-zero value at index 5. As shown in figure 3, we concatenate all these features as input to the DNN part of our model. And let the model itself learn even more abstract features from these raw inputs.

4. Experimental Results

In this section, we present the evaluation results of the proposed method. We first briefly introduce our data set. Then we provide some implementation details of our deep learning based architecture. And finally we show the performance comparison between our method and several baselines.

The data set consists of about 3 years' (from February 10th, 2000 to December 31st, 2012) hourly load values of a city in North China. For weather information, hourly temperature and humidity are important indicators of load forecasting. But unfortunately, we can only obtain daily highest and lowest temperatures on the Internet. Since in holidays people show different working and living patterns from usual, we also collect the data of public holidays of each year.

Our goal is to provide one day (24 hours) ahead prediction of hourly loads. More precisely, we are going to forecast the load at a specific hour using weather, timing and holiday information of the targeting hour and historical load information of 24 hours before. As an example, we can only use historical loads up to 5:00 on March 9th, in order to forecast the hourly load at 5:00 on March 10th.

In our final forecasting model, we use historical loads of an entire week. So the length of the historical load sequence is 168 (7 days of 24 hours each). And then the model uses 3 parallel CNN components to extract features from these raw historical loads. The filter sizes of these CNN modules are 3, 5 and 7 respectively, while each of them has 64 filters. After the max-pooling with size 2 and stride 2, the outputs are combined and fed into the RNN part. As described in section 3, the RNN component employs LSTM units and the size of its hidden state is 64. Features other than historical loads are represented as vectors of length 34, where 2 components are for daily highest and lowest temperatures, 1 for holiday information, 24 for hour of day and 7 for day of week. And these are inputs to the dense module. In our model, the number of units for all dense components is 64. The model is implemented using the Keras [19] toolkit and trained use the RMSProp optimizer.

For evaluation, we randomly split the data set which contains hourly loads of totally 1017 days into training, validation and testing sets. The validation and testing sets have 100 days' data, respectively. And finally we get 19,608 training, 2400 validation and 2400 testing samples. Some research papers [7][14] used data during a continuous period of time (eg. one month or 2 months) as validation and testing data. However, we believe that this may not be quite suitable when limited data are available, since there are inherently different power consumption patterns in each month. So in this paper, we randomly select validation and testing samples.

Typical performance metrics for evaluating short term load forecasting methods are mean absolute percentage error (MAPE) and mean average error (MAE), which can be computed using the following two formulas:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{8}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{9}$$

where y_i is the true load value of a specific hour, $\hat{y}_i$ is the predicted load value, and N is the number of testing samples.

Table 1. Performance Comparison

	Dev MAPE (%)	Test MAPE (%)	Dev MAE (MW)	Test MAE (MW)
Linear Reg.	2.761	2.939	203.48	216.68
SVR	1.650	1.720	121.22	126.71
DNN	1.665	1.664	122.68	123.11
CNN_RNN	1.456	1.475	107.53	109.57
Parallel_CNN_RNN	**1.349**	**1.405**	**99.41**	**104.24**

In table 1 we compare the proposed method with several strong baselines, including linear regression, Support Vector Regression (SVR), a DNN with 3 hidden layers and the CNN_RNN baseline which is similar to the proposed method but has only 1 CNN layer with filter size 3 followed by the RNN part. We use the python scikit-learn [20] package to implement the linear regression and SVR. And for SVR, we exploited the RBF kernel and used the grid search to find the best parameter values. In addition, each hidden layer of the DNN baseline has 64 units.

We can see from table 1 that our method outperforms all baselines in both MAPE and MAE. Comparing with the linear regression approach, the testing MPAE reduced by 52.2% while MAE reduced by 51.9% relatively. And even with the CNN_RNN method which is rather powerful, our method achieves about 5% relative reduction of both MAPE and MAE. And this demonstrates the ability of parallel CNN structure for feature extraction. The DNN baseline outperforms the SVR, which show the effectiveness of deep learning. Moreover, when using the DNN baseline together with manually extracted features as described in [14], the testing MAPE and MAE are 2.110 and 15.587, respectively. And this may indicate that deep learning methods can learn better feature representations from raw data.

5. Conclusion

With the development of smart grid, load forecasting has become increasingly important. In this paper we study one day ahead forecasting of hourly loads based on Deep Learning. We present a novel flexible architecture that incorporates multiple types of input features, where the inputs are processed using different types of neural network components according to their specific characteristics. We conduct the evaluation on a data set consisting of hourly loads of about 3 years and experimental results show the effectiveness of our method.

Power load consumption is influenced by a large number of factors, and thus load forecasting is a rather complicated problem. From the data perspective, more training data are usually required to achieve better performance. And we also need more relevant features, for example, the hourly temperature and humidity. As for methods, Deep Learning has been developing quickly, and we will keep on paying attention to new models. The neural attention mechanism has achieved a lot of success in different domains, and in the future, we will try to apply it to load forecasting.

References

[1] G. A. N. Mbamalu, M. E. Elhawary, Load forecasting via suboptimal seasonal autoregressive models and iteratively reweighted least squares estimation, IEEE Transactions on Power Systems 8 (1) (1993) 343–348.

[2] T. Haida, S. Muto, Regression based peak load forecasting using a transformation technique, IEEE Transactions on Power Systems 9 (4) (1994) 1788–1794.

[3] J.-F. Chen, W.-M. Wang, C.-M. Huang, Analysis of an adaptive time-series autoregressive moving-average (arma) model for short-term load forecasting, Electric Power Systems Research 34 (3) (1995) 187–196.

[4] G. Juberias, R. Yunta, J. G. Moreno, C. Mendivil, A new arima model for hourly load forecasting, in: Transmission and Distribution Conference, 1999 IEEE, Vol. 1, IEEE, 1999, pp. 314–319.

[5] Y.-C. Guo, D.-X. Niu, Y.-X. Chen, Support vector machine model in electricity load forecasting, in: Machine Learning and Cybernetics, 2006 International Conference on, IEEE, 2006, pp. 2892–2896.

[6] A. Lahouar, J. B. H. Slama, Day-ahead load forecast using random forest and expert input selection, Energy Conversion and Management 103 (2015) 1040–1051.
[7] M.-W. Chang, B.-J. Chen, C.-J. Lin, Eunite network competition: Electricity load forecasting, in: National Taiwan University, Citeseer, 2001.
[8] J. R. Lloyd, Gefcom2012 hierarchical load forecasting: Gradient boosting machines and gaussian processes, International Journal of Forecasting 30 (2) (2014) 369–374.
[9] T. Hong, P. Pinson, S. Fan, Global energy forecasting competition 2012, International Journal of Forecasting 30 (7) (2014) 357–363.
[10] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, R. J. Hyndman, Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond, International Journal of Forecasting 32 (3) (2016) 896–913.
[11] H. S. Hippert, C. E. Pedreira, R. C. Souza, Neural networks for short-term load forecasting: A review and evaluation, IEEE Transactions on power systems 16 (1) (2001) 44–55.
[12] J. G. Jetcheva, M. Majidpour, W.-P. Chen, Neural network model ensembles for building-level electricity load forecasts, Energy and Buildings 84 (2014) 214–223.
[13] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, `http://www.deeplearningbook.org`.
[14] W. He, Deep neural network based load forecast, Comput. Model. New Technol 18 (3) (2014) 258–262.
[15] G. M. U. Din, A. K. Marnerides, Short term power load forecasting using deep neural networks, in: Computing, Networking and Communications (ICNC), 2017 International Conference on, IEEE, 2017, pp. 594–598.
[16] D. L. Marino, K. Amarasinghe, M. Manic, Building energy load forecasting using deep neural networks, in: Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE, IEEE, 2016, pp. 7046–7051.
[17] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.
[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
[19] F. Chollet, et al., Keras, `https://github.com/fchollet/keras` (2015).
[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.