Efficient Self-attention with Relative Position Encoding for Electric Power Load Forecasting

Jing Cui\textsuperscript{1,a}, Yang Li\textsuperscript{1,b}, Jiaolong Liu\textsuperscript{1,c}, Jie Li\textsuperscript{1,d}, Zheng Yang\textsuperscript{1,e}, Chunlin Yin\textsuperscript{1,f}

\textsuperscript{1}Electric Power Research Institute, Yunnan Power Grid, Co. Ltd., Kunming, China  
\textsuperscript{a}e-mail: 917659783@qq.com, \textsuperscript{b}e-mail: 281914244@qq.com  
\textsuperscript{c}e-mail: jiaolong0818@163.com, \textsuperscript{d}e-mail: 1226645407@qq.com  
\textsuperscript{e}e-mail: 279039301@qq.com, \textsuperscript{f}e-mail: 1078968225@qq.com

Abstract—To effectively mine historical data information and improve the accuracy of short-term load prediction, this paper aims at the characteristics of time series and nonlinear power load. Deep learning for load forecasting has received a lot of attention in recent years, and it has become popular in the analysis of electricity load forecasting. Long short-term memory (LSTM) and gated recurrent unit (GRU) are specifically designed for time-series data. However, due to the gradient disappearing and exploding problem, recurrent neural networks (RNNs) cannot capture long-term dependence. The Transformer, a self-attention-based sequence model, has produced impressive results in a variety of generating tasks that demand long-range coherence. This shows that self-attention could be useful in power load forecasting modeling. In this paper, to effectively and efficiently model the large-scale load forecasting, we further design the transform encoder with relative position encoding, which consists of four main components: single-layer neural network, relative positional encoding module, encoder module, and feed-forward network. Experimental results on real-world datasets demonstrate that our method outperforms the GRU, LSTM, and original Transformer encoder.

1. INTRODUCTION
To anticipate future loads, electric power load forecasting typically examines and mines historical load data. It employs a variety of innovative technologies to precisely estimate power demand, which is a critical component of achieving energy-efficient, cost-effective, and efficient operation.

In the past few decades, because machine learning algorithms have parametric self-learning and can deal with nonlinear data, many methods have been proposed by researchers and scholars to load, including linear regression (LR) [1], artificial neural networks (ANN) [2], support vector machine (SVM) [3], XGBoost [4] and so on. Machine learning algorithms provide various advantages in terms of nonlinear problem robustness [5]. However, as data types diversify and the amount and types of data increase, the above methods have some disadvantages when processing large-scale load data.

Various types of deep neural network models have been introduced into power data analysis in recent years, with the strong growth of the idea of deep learning, and have achieved significant success, including Convolutional Neural Networks (CNNs) [6], Recurrent Neural Networks (RNNs) [7], etc. It demonstrates great adaptive and nonlinear mapping capabilities in the face of large high-dimensional input [8]. However, due to the gradient disappearing and exploding problem [9], RNNs cannot capture long-term dependence. Despite the development of other variations, such as long short-term memory...
units (LSTM) and gated recurrent units (GRU), the problems remain unsolved. CNN needs to be captured long-distance features by increasing the network depth.

The well-known Transformer model [10] is a novel type of neural network that swiftly rose to the top of the heap in a variety of fields such as image processing [11], and so on. The transformer model consists of encoder and decoder blocks. Transformer models have a lot of tricks: self-attention, position encoding, multi-head attention, masked attention, residual connections, layer normalization, feed-forward layer. The whole architecture transforms the input sequence into a new output sequence that incorporates data from all other input parts. The Transformer model is better than RNNs. The Transformer model, unlike recurrent networks, can access any point in the past regardless of the distance between words, making it better suited to loop modes with long-term dependencies.

Therefore, in this paper, we explore the use of the self-attention neural network and relative position encoding technique to enhance electric power load forecasting accuracy and efficiency. We showed that in the real world load forecasting case our model achieves state-of-art forecasting results.

2. METHODOLOGY

Our model is based on the original Transformer design, but with a few tweaks to deal with short-term historical load inputs: relative position encoding, multi-head attention, we just look at the encoder side of the transformer in this study. The overall architecture of the short-term historical load forecasting model, as shown in Figure 1, consists of four main components: single-layer neural network, relative positional encoding module, encoder module, and feed-forward network. Finally, the implementation details and training loss are introduced.

![Figure 1. The overall methodology of the forecasting model](image)

2.1 Self-attention in Transformer

Researchers at Google Research and Google Brain proposed the concept of self-attention. It was proposed in response to the difficulties encoder-decoders experience when working with extended sequences. The Transformer comprises two sub-models: an encoder and a decoder, with the self-attention mechanism at the heart of the encoder and decoder. Self-attention can learn relevant information in various representation subspaces. The basic concept of attention is to learn a scoring system that gives each piece of data a distinct weight.
In the transformer encoder part, the input temporal data vectors \( x_i \in \mathbb{R}^{L \times D} \) of length \( L \) and dimension \( D \) into query vector \( Q = XW^Q \), key vector \( K = XW^K \), and value vector \( V = XW^V \), where \( W^Q, W^K, W^V \) are each \( D \times D \) square matrices. Then, computes a dot product for each query vector with all of the key vectors, subsequently, divides each result by \( \sqrt{d_k} \) as in (1)

\[
e_{ij} = \frac{(x_iW^Q)(x_jW^K)^T}{\sqrt{d_k}} \tag{1}
\]

Where \( e_{ij} \) is the attention weight from position \( j \) to \( i \), calculation using scaled dotted product attention. Each weight coefficient \( \alpha_{ij} \) computed by applying a softmax function, as in (2)

\[
\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \tag{2}
\]

The output temporal data \( z_i = (z_1, \ldots, z_n) \) is computed as a weighted sum of input elements, as in (3)

\[
z_i = \sum_{j=1}^{n} \alpha_{ij} (x_jW^V) \tag{3}
\]

### 2.2 Relative Position Encoding for Transformer

Position encodings based on sinusoids of varied frequency are added to the encoder and decoder in the vanilla transformer to capture the token's location information. Because the added position embedding depends on the absolute positions of tokens in a sequence, it is called absolute position encoding. We’ll use relative position encoding [12], which can directly encode the distance between tokens. To increase the accuracy of the short-term load forecast, we modified the similarity functions (Equation (1) and Equation (5)) as in

\[
z_i = \sum_{j=1}^{n} \alpha_{ij} (x_iW^V + R_{ij}^V) \tag{4}
\]

\[
e_{ij} = \frac{\exp((x_iW^Q)(x_jW^K + R_{ij}^K)^T)}{\sqrt{d_k}} \tag{5}
\]

This approach encodes the relative position between the input elements \( x_i \) and \( x_j \) into vectors \( R_{ij}^V \) and \( R_{ij}^K \).
3. EXPERIMENTS RESULTS AND ANALYSES
A real-world dataset from UCI Electricity was used to verify the proposed model's forecasting accuracy. We compared GRU, LSTM, encoder with self-head attention and the proposed model in this paper.

3.1 Data Preprocessing
The proposed mechanism's performance in daily short-term load forecasting is tested using the UCI Electricity Load Diagrams Dataset (ELDD), which contains 106 days of load values and the electricity consumption of 370 users [13]. The load datasets were sampled every 15 min from December 30, 2013, to April 14, 2014. The dataset is split into training and test sets, with an 80%-20% ratio. We only used the data of ten users in this work, from “MT_200” to “MT_209”.

By modifying all the input variables within a close range, normalization is frequently used to prevent large-scale features from having too much influence. For this, we used Equation (6) to apply the min-max normalization to all input variables. The minimum and maximum values of the input variable could be -1 and 1.

\[ X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \] (6)

Where the \( X_{\text{max}} \) and \( X_{\text{min}} \) are the respective minimum and maximum of the variable of the scenario I in the entire time length.

3.2 The Detailed Experimental Setting
The learning rate was 0.0001 and the batch size was 64. The hidden neuron of the LSTM and GRU module was set as 256. The active function was used the Tanh function. The Adam Optimizer was used to optimize the parameters of the model by performing mini-batch stochastic gradient descent (SGD). We implemented load forecasting and deep learning method in the PyTorch framework and ran all the experiments with one GTX 1080Ti GPU. We used more than 300 thousand parameters for our proposed model.
3.3 Model Evaluation Indexes

In this paper, we used mean absolute percentage error (MAPE) and Coefficient of Determination (R2 Score) to evaluate the performance of the proposed models. The error measures are defined as in (7) and (8)

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

(7)

The MAPE value lower indicate the model is better. Let \( y_i \) be actual values, \( \hat{y}_i \) forecasted values, and the \( N \) be the number of values.

\[
R^2 = 1 - \frac{SS_{res}}{SS_{tot}}
\]  

(8)

The \( R^2 \) value of 1 is the representation of perfect fit. Let \( SS_{res} \) be the sum of squares of residuals and \( SS_{out} \) be the total sum of squares.

The predicting accuracy and efficiency of self-attention with a relative model for load forecasting have high accuracy.

3.4 Experimental Results and Analysis

We compare our approach to load forecasting to a number of important baselines, which are LSTM and GRU models, the performance as we will see in Figure 3, our model has outperformed the other models.

![Figure 3. Experimental results](image)

The results of the experiments suggest that the Transformer network can get good results and has upsides over LSTM and GRU.

The results of the forecasting can objectively and precisely represent the changing load regulations. The suggested approach may be utilized to increase the accuracy of short-term load forecasting and decrease anticipated value variations during the forecasting process.

4. CONCLUSION

In this paper, a novel mechanism, self-attention with relative position is proposed to enhance load forecasting. The attention network’s mechanism allows it to focus more on input variables that have a stronger association with history load. Unlike previous forecasting deep learning systems, our methodology models time series data via self-attention processes, allowing it to learn complicated relationships of varying durations from time-series data. Relative position encoding is currently more applicable to natural language processing, we extend it to load forecasting models and achieve good performance.
ACKNOWLEDGMENT
This work has been supported by the Major Science and Technology Project of Yunnan Province No. 202102AA100021, Natural Science Foundation of Yunnan Province No. 202101AT070167, Scientific Research Fund Project of Yunnan Education Department No. 2012Y250, Yunnan Software Engineering Key Laboratory Open Fund Project No. 2020SE311.

REFERENCES
[1] Dudek, G. (2016). Pattern-based local linear regression models for short-term load forecasting. Electric power systems research, 130, 139-147.
[2] Charytoniuk, W., & Chen, M. S. (2000). Very short-term load forecasting using artificial neural networks. IEEE transactions on Power Systems, 15(1), 263-268.
[3] Chen, B. J., & Chang, M. W. (2004). Load forecasting using support vector machines: A study on EUNITE competition 2001. IEEE transactions on power systems, 19(4), 1821-1830.
[4] Li, G., Li, W., Tian, X., & Che, Y. F. (2017). Short-term electricity load forecasting based on the xgboost algorithm. Smart Grid, 7(4), 274-285.
[5] Yildiz, B., Bilbao, J. I., & Sproul, A. B. (2017). A review and analysis of regression and machine learning models on commercial building electricity load forecasting. Renewable and Sustainable Energy Reviews, 73, 1104-1122.
[6] Li, L., Ota, K., & Dong, M. (2017, June). Everything is the image: CNN-based short-term electrical load forecasting for smart grid. In 2017 14th International Symposium on Pervasive Systems, Algorithms and Networks & 2017 11th International Conference on Frontier of Computer Science and Technology & 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC) (pp. 344-351). IEEE.
[7] Shi, H., Xu, M., & Li, R. (2017). Deep learning for household load forecasting—A novel pooling deep RNN. IEEE Transactions on Smart Grid, 9(5), 5271-5280.
[8] Kang, T., Lim, D. Y., Tayara, H., & Chong, K. T. (2020). Forecasting of power demands using deep learning. Applied Sciences, 10(20), 7241.
[9] Pascanu, R., Mikolov, T., & Bengio, Y. (2013, May). On the difficulty of training recurrent neural networks. In International conference on machine learning (pp. 1310-1318). PMLR.
[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
[12] Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-attention with relative position representations. arXiv preprint arXiv:1803.02155.
[13] Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting.