Simplifying Long Short-Term Memory for Fast Training and Time Series Prediction

Yuyan Zhang¹, Xin Hao¹,a, Yong Liu¹

¹Beijing University of Posts and Telecommunications, Beijing, 100876, China

a Corresponding author: hx_sherry@163.com

Abstract. Long short Term Memory(LSTM) has been widely used in sequential problems. However, for the time series prediction problems, its complex structure limits its running speed and performance. In order to solve this problem, this paper simplified the standard LSTM model by reducing the number of gates and the parameters involved in gates computation. Experiments on univariate data set and multivariate data set show that the proposed simplified model not only has better accuracy, but also has higher running speed.

1. Introduction

Recurrent neural networks(RNNs) are powerful learning models by introducing recurrent connections on the hidden layers allowing temporal information to be passed through many time steps. Thus they can model sequential series that elements are time dependent. Theoretically, RNN can definitely handle any length of sequence data. However, with the number of network layer increases, RNN suffers from the gradient vanishing problem, making the network difficult to be trained[1].

Gated recurrent neural networks(GRNNs), such as long short-term memory(LSTM) and gated recurrent unit(GRU), use carefully designed recurrent units with fixed units weight to solve the vanishing gradient problem[2]. Long short-term memory(LSTM) is the most popular and successful GRNN model proposed by Hochreiter and Schmidhuber[3]. It introduces the memory cell, a unit of computation that replaces traditional nodes in the hidden layer of a network, which helps to overcome difficulties with training encountered by earlier recurrent networks[2]. The architecture of typical LSTM is very complex, with 3 gates and a hidden state. Many researchers are dedicated to designing effective simpler GRNN structures for better performance, such as Gated Recurrent Unit(GRU)[4], the recurrent additive networks(RAN)[5], simplified LSTM(S-LSTM)[6] and minimal gated unit(MGU)[7]. Complex GRNN models not only are difficult to interpret what function they have learned, but also mean more parameters to be learned and well tuned.

This paper proposed a new variant of LSTM by simplifying the recurrent unit of the standard LSTM model. The proposed model reduces the number of gates and the parameters involved in gates’ computation. Such a simple structure is very useful to time series prediction problems, which gets better prediction performance and improves the running speed.

The rest of the paper is organized as follows. The related works about RNN algorithm, LSTM model and its variants are described in Section2. A simplified LSTM model for improving time series forecasting speed is proposed in section 3. In section4, the proposed LSTM model is evaluated using Airline Passenger data set and TAIEX data sets. The conclusion is discussed in Section 5.
2. Related works
A briefly review of RNN algorithm and various GRNN models are introduced in this section.

2.1. RNN algorithm
Recurrent neural network (RNN) is an extension of conventional feedforward neural network with dynamic input[8]. And it is characterized by introducing recurrent connections on the hidden layers, which makes modeling sequence data possible. RNN uses index $t$ to represent the different positions in the input sequence and suppose a hidden state $h_t$ to represent the state of the system at time $t$. At time $t$, RNN accepts input $x_t$, and the recurrent hidden state $h_t$ is activated by the former hidden state $h_{t-1}$ and current input $x_t$, and updated in real time by a nonlinear activation function:

$$h_t = \tanh (W [x_t, h_{t-1}] + b) = \tanh (W_x x_t + W_h h_{t-1} + b)$$  \hspace{1cm} (1)

where $W$ is the weight matrix for the input $x_t$, $W_h$ is the weight matrix for the recurrent input $h_{t-1}$, and the $b$ term is the bias vector. By fitting the parameters $W$ and $b$, the sequence data could be well learned.

2.2. Typical LSTM model
RNN can be considered as a neural network that passes on time, which means the length of time sequence is the depth of RNN network. Once length of time sequence is very long, the problems of vanishing and exploding gradients occur when back propagating errors across many time steps[2].

The LSTM model is proposed to overcome the problem of vanishing gradient from RNN by introducing gating mechanism in recurrent units to control how information flows. In addition to three multiplicative gates (forget gate $f_t$, input gate $i_t$, output gate $o_t$) and hidden state $h_t$, the recurrent unit includes a memory state $S_t$, in which $S_t$ is used to help maintaining long-term memory. The gates are sigmoid units that take activation from the current input $x_t$ as well as from the hidden layer at the previous time step. At time $t$, new memory state $S_t$ is formed by its self-connected recurrent edge $S_{t-1}$ and $G_t$ with corresponding parameters $W$ and $b$. The associated equations are given as follow:

$$f_t = \sigma (W_f \cdot [x_t, h_{t-1}] + b_f)$$ \hspace{1cm} (2)
$$i_t = \sigma (W_i \cdot [x_t, h_{t-1}] + b_i)$$ \hspace{1cm} (3)
$$o_t = \sigma (W_o \cdot [x_t, h_{t-1}] + b_o)$$ \hspace{1cm} (4)
$$G_t = \sigma (W_g \cdot [x_t, h_{t-1}] + b_g)$$ \hspace{1cm} (5)
$$S_t = f_t \cdot S_{t-1} + i_t \cdot G_t$$ \hspace{1cm} (6)
$$h_t = o_t \cdot \tanh (S_t)$$ \hspace{1cm} (7)

where $\cdot$ denotes matrix multiplication and $*$ is pointwise multiplication, $\sigma$ is a sigmoid function, $W_f$, $W_i$, $W_o$, $W_g$ are weight matrices of $f_t$, $i_t$, $o_t$, $G_t$, and $b_f$, $b_i$, $b_o$, $b_g$ are the corresponding bias terms.

2.3. Variant models of LSTM
After the appearance of LSTM, many variations have been proposed to improve the performance.

RAN simplifies the content layer and removes the output gate[5]; S-LSTM derives the input gate from the forget gate and simplifies the output gate[6]; GRU incorporates the functions of memory content and hidden state to simplify the structure and it only contains two gates[4]; MRU simplified the GRU model by using one gate[7]; SRU simplified memory state computation by dropping the dependence on hidden state in a recurrence step[9].

The design idea of above LSTM variants improves the training speed by simplifying the hidden block structure under the fairly accurate conditions. And the popular ways to simplify the LSTM structure are:
3. The proposed model

In this section, a novel LSTM model is proposed to speed up time series forecasting by simplifying the standard LSTM model’s inner structure to improve the prediction performance.

3.1. Model Definition

In our proposed model, the recurrent unit includes two gates, one is forget gate $f_t$ which is considered to be critical for partially forgetting the existing memory[10], and the other is an update gate $z_t$, which controls how much information will be updated lossless to its content.

Existing GRNN implementations use the previous output state $h_{t-1}$ in the recurrence computation, which limits its running speed and parallel computing power[9]. The simplified model replaced the connection between $h_{t-1}$ and the neural gates of step $t$ to the connection between $S_{t-1}$ and the neural gates.

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

where $W_f$ and $b_f$ are weight matrix and the bias term for $f_t$ respectively.

For time series prediction problems, the input information structure is relatively simple and its redundancy is low, so the input gate is removed and all the input information is retained. In order to increase the flexibility of input, the weight matrix $W_x$ is used to adjust its amplitude.

$$\tilde{x}_t = tanh (W_x x_t + b_x)$$

Next, the memory $S_t$ is updated by the output of forget gate and the joint input.

$$S_t = \tilde{x}_t + S_{t-1} \cdot f_t$$

Finally, highway connections are exploited, it proved to be a very effective method for training very deep network[11]. Update gate $z_t$ is used for controlling the proportion of memory cell $S_t$ and current input $x_t$, and it creates a highway for $x_t$, which makes it transmit to $h_t$ lossless as possible.

$$h_t = z_t \cdot tanh (S_t) + (1 - z_t) \cdot x_t$$

where update gate $z_t = \sigma (W_z \cdot [x_t, S_{t-1}] + b_z)$, $W_z$ and $b_z$ are weight matrix and the bias term for $z_t$ respectively.

4. Simulations and result analysis

In this section, to evaluate the performance of the proposed sequence forecasting model, the simulations are conducted on two data sets. One is the monthly volume of international airline passengers during Jan 1949 to Dec 1960[12]. The other is TAIEX index with two factors in year 2003 and 2004[13].

To ensure the fairness and reliability of the experiment, LSTM is used as a baseline architecture, and compares it with the proposed model. If there's no specific instructions, these two models are trained with the same loss function, same number of hidden units and same number of layers. And the experiments are implemented with the tensorflow package on platform Anaconda3.
4.1. Airline passengers data set

The airline passengers data set provided by Box and Jenkins(1976)[12] contains data on the total number of passengers in 144 months from Jan. 1949 to Dec. 1960. The number of passengers in each month is one record, a total of 144 records. In this task, first 100 samples are used for training, and last 44 samples are used for prediction. Both standard LSTM and the proposed model use 20 hidden units, batch size is set to 30, and the learning rate is set to 0.001. And two models are trained in 1000 epochs.

The prediction accuracy of each forecasting model is measured by root-mean-square error(RMSE), defined as follows,

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(F(t) - Y(t))^2}{n}}$$

(12)

where \(F(t)\) denotes the final forecasting value of the models at time \(t\); \(Y(t)\) denotes the actual value of time series at time \(t\); \(n\) represents the number of forecasting data.

Table 1. Comparison of RMSE and running time for proposed model and standard LSTM on airline passengers data set.

| Model            | RMSE  | Time(seconds) |
|------------------|-------|---------------|
| Standard LSTM    | 45.55 | 23.74         |
| Proposed model   | 42.72 | 19.13         |

Here, the relative error and running time are used as criteria to evaluate the performance, and the results with respect to the two models are shown in Table 1.

Further, in order to observe the forecasting performance of the model, testing comparison between two models based on monthly totals airline passengers from Jan. 1949 to Dec. 1960 is shown in Fig.1.

![Figure 1. Comparison of the prediction values of the standard LSTM and the proposed LSTM based on monthly totals airline passengers from Jan. 1949 to Dec. 1960.](image)

As Table 1 shows, the proposed LSTM gets better prediction accuracy than standard(relative error is 45.55 vs. 42.72). At the same time, the running time of the proposed LSTM is shorter than that of standard LSTM(running time is 19.13 seconds vs. 23.74 seconds). As can be seen from Figure 1, the proposed LSTM fitted the test data better than standard LSTM. Thus, the proposed LSTM model provides better performance to the univariate time series prediction.

4.2. TAIEX data set

In this section, the Taiwan stock exchange's stock index(TAIEX) data set is used to verify the performance of the proposed LSTM model[13]. The TAIEX is seen as one of the main indicators of Taiwan's economic trend. And TAIEX is adopted being the main factor and Dow Jones Index being the secondary factor to forecast TAIEX. The historical data of the TAIEX and Dow Jones of year 2003
and year 2004 are used to build the models, separately. Main factor input TAIEX and secondary factor input Dow Jones are constructed as an input pair, which is used for sequential input of a single node. Since it is a two-variables time series modeling, the main series and the secondary series are normalized before modeling. And the predicted values are obtained by weight fitting output pairs TAIEX - Dow Jones, where PSO algorithm is used for weight fitting. The training data of each year is from January to October, and the forecasting data of each year is November and December. In this task, both the number of hidden unit and step length are adjustable. The learning rate is set to 0.0005, the size of mini-batch is set to 20 and trained the models 300 times. The prediction accuracy of each forecasting model is measured by RMSE.

Table 2 compares the proposed model with standard LSTM on RMSE and running time on TAIEX data sets in year 2003 and year 2004. The forecast results of the TAIEX in year 2003 and 2004 are shown in Figure 2 and Figure 3, respectively.

From Table 2, the RMSE of the proposed LSTM is slightly lower than standard LSTM, with 57.05 vs. 59.03 in year 2003, and 55.30 vs. 56.72 in year 2004. However, the running time of the proposed model is significantly faster than standard LSTM. In the year 2004, the running time of the proposed model (19.27 seconds) is about one third less than standard LSTM (29.83 seconds).

Table 2. Comparison of RMSE and running time for the proposed model and standard LSTM on TAIEX data set.

| Year |          |          |
|------|----------|----------|
|      | RMSE     | Time(secs)|
| 2003 | Standard LSTM | 59.03    | 19.28    |
|      | Proposed model | 57.05    | 17.69    |
| 2004 | Standard LSTM | 56.72    | 29.83    |
|      | Proposed model | 55.30    | 19.27    |

Figure 2. Comparison of the prediction values of the standard LSTM and the proposed LSTM based on TAIEX in year 2003.
By analyzing the forecasting curve of Figure 2 and 3, it can be obtained that: in year 2003, the performance of the proposed model is slightly better than standard LSTM, meanwhile, in year 2004, both the two models fit excellent.

5. Conclusion
In this paper, a simplified variant of LSTM is presented by reducing the number of gates and the parameters involved in gates’ computation. Such a simple structure is very useful to time series prediction problems, which gets better prediction performance and improves the running speed. The proposed LSTM model is compared with standard LSTM model on univariate data set (airline passengers forecasting) and multivariate data set (TAIEX). Experiment results show that the proposed LSTM model has better performance than standard LSTM model.

Acknowledgments
This work is supported by National Natural Science Foundation of China (No. 61501041).

References
[1] Pearlmutter B. A., Gradient calculations for dynamic recurrent neural networks: a survey. IEEE Transactions on Neural Networks, 6(5), 1212-1228 (1995)
[2] Lipton Zachary C., J. Berkowitz and C. Elkan, A Critical Review of Recurrent Neural Networks for Sequence Learning. Computer Science (2015).
[3] Hochreiter Sepp, Schmidhuber Jürgen, Long Short-Term Memory. Neural Computation, 9(8), 1735-1780 (1997).
[4] Cho Kyunghyun, Merrienboer Bart van, Gulcehre Caglar, Bahdanau Dzmitry, Bougares Fethi, Schwenk Holger, Bengio Yoshua, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of EMNLP, pages 1724-1734 (2014).
[5] Lee Kenton, Levy Omer, Zettlemoyer Luke, Recurrent additive networks. CoRR abs/1705.07393 (2017).
[6] Yajie Miao, Jinyu Li, Yongqiang Wang, Shi-Xiong Zhang, Yifan Gong, Simplifying long short-term memory acoustic models for fast training and decoding. IEEE International Conference on Acoustics (2016).
[7] Guo-Bing Zhou, Jianxin Wu, Chen-Lin Zhang, Zhi-Hua Zhou, Minimal Gated Unit for Recurrent Neural Networks. International Journal of Automation and Computing, 13(3):226-234 (2016).
[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR abs/1412.3555 (2014).
[9] Tao Lei, Yu Zhang, Training RNNs as Fast as CNNs. CoRR abs/1709.02755 (2017).
[10] Greff Klaus, Srivastava Rupesh K., Koutník Jan, Steunbrink Bas R., Schmidhuber Jürgen, LSTM: A Search Space Odyssey[J]. IEEE Transactions on Neural Networks & Learning Systems, 28(10):2222-2232 (2015).

[11] Srivastava Rupesh Kumar, Greff Klaus, Schmidhuber Jürgen, Training Very Deep Networks. NIPS, pp. 2377-2385 (2015).

[12] International airline passengers: monthly totals in thousands. Jan 1949- Dec. 1960.
[13] Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX).