Can Tho University Journal of Science
website: sj.ctu.edu.vn

DOI: 10.22144/ctu.jen.2020.016

Forecasting time series with long short-term memory networks

Nguyen Quoc Dung[1,3*], Phan Nguyet Minh[2] and Ivan Zelinka[3]

[1]Van Lang University, Vietnam

[2]University of Information Technology, Vietnam

[3]Technical University of Ostrava, Czech Republic

[*] Correspondence: Nguyen Quoc Dung (email: nqdung@vanlanguni.edu.vn)

Article info.

Received 04 Mar 2020
Revised 25 Apr 2020
Accepted 31 Jul 2020

Keywords

Long short-term memory, recurrent neural network, sequence prediction, time series

ABSTRACT

Deep learning methods such as recurrent neural network and long short-term memory have attracted a great amount of attentions recently in many fields including computer vision, natural language processing and finance. Long short-term memory is a special type of recurrent neural network capable of predicting future values of sequential data by taking the past information into account. In this paper, the architectures of various long short-term memory networks are presented and the description of how they are used in sequence prediction is given. The models are evaluated based on the benchmark time series dataset. It is shown that the bidirectional architecture obtains the better results than the single and stacked architectures in both the experiments of different time series data categories and forecasting horizons. The three architectures perform well on the macro and demographic categories, and achieve average mean absolute percentage errors less than 18%. The long short-term memory models also show the better performance than most of the baseline models.

Cited as: Dung, N.Q., Minh, P.N. and Zelinka, I., 2020. Forecasting time series with long short-term memory networks. Can Tho University Journal of Science. 12(2): 53-59.

1 INTRODUCTION

Time series is a sequence of data points indexed along the time they are collected. Most often, the data is taken at regular time intervals. Forecasting future values of time series data is a common problem in many practical fields such as economics, finance, weather forecasting, as well as applied science and engineering. Predicting the weather for the next days, the closing price of a stock each day, product sales in units sold during summer for a shop and future heart failure are well-known examples.

Time series data introduces a dependent relationship among collected observations. Time series forecasting makes use of a forecasting model to predict future values based on previously observed values. A time series forecasting model is also known as a sequence prediction model as shown in Fig. 1.

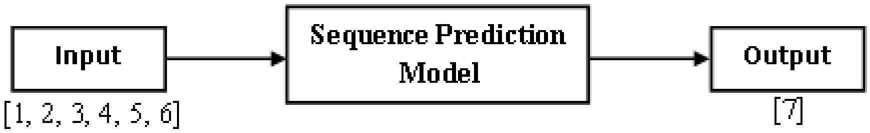

Fig. 1: Example of Sequence Prediction Problem. The prediction model takes the input sequence of observed values 1-6 and generates the predicted value 7 at the output

Some interesting properties of time series are stationarity, seasonality, and autocorrelation. A time series is called stationary when the mean and variance are constant over time, while a time series has a trend if the mean is changing over time. Seasonality refers to the phenomenon of variations over an observed period of time, for example, tourist numbers increase every summer. Time series with trend or with seasonality are non-stationary. A common approach to making the time series stationary is to use some transformation such as differencing by subtracting the time series data in the current time from the previous one. Autocorrelation refers to the correlation between the time series with a copy of itself from a previous time.

Classical methods like autoregressive integrated moving average (ARIMA) models (Box and Jenkins, 1970) require stationary time series data. Eliminating a trend or seasonality component to have the time series stationary is done in the data preprocessing step of the forecasting model.

In this paper, a deep learning method is introduced, named as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), which applies a sequence of observed values as input to predict the next values without the data preprocessing step for stationary time series.

LSTM is an improved version of recurrent neural network (RNN) (Rumelhart *et al.*, 1986; Karpathy, 2015) designed for processing sequential data by learning patterns over time. The LSTM-based methods can be found in many applications of voice, text, image, and video processing such as machine translation, speech recognition, image captioning, and action detection in video streams (Sutskever *et al.*, 2014; Li and Wu, 2015; Vinyals *et al.*, 2015; Ullah *et al.*, 2017). Since LSTM network is capable of handling sequence dependence among observed inputs, it is well-suited to sequence prediction problems, especially for non-linear and complex time series data (Malhotra *et al.,* 2015; Guo *et al.,* 2016; Hsu, 2017).

2 LONG SHORT-TERM MEMORY ARCHITECTURES FOR TIME SERIES PREDICTION

2.1 Long short-term memory

LSTM is a type of RNN, which is widely used on a large variety of problems in the field of deep learning such as computer vision, machine translation and speech recognition. It is capable of learning long-term dependencies, as well as dealing with the exploding and vanishing gradient problems that are encountered in traditional RNNs. The LSTM network was introduced by Hochreiter and Schmidhuber (1997), and was continually refined in the following works such as Gers *et al.*, 1999 and 2000; Cho *et al.*, 2014.

LSTM extends the memory capability of RNN by introducing three gates (input gate, output gate and forget gate) to regulate the flow of information inside the LSTM unit. The memory part of the LSTM unit is known as the cell. The cell takes care of keeping track of the dependencies between the elements in the input sequence. The input gate regulates how much information from the current input flows into the memory cell, the forget gate regulates how much information from the previous cell will be retained (or discarded) into the current cell, and the output gate scales the value in the current cell used to compute the output activation of the LSTM unit.

As similar to RNN, LSTM network can be unrolled in time as a chain of repeating modules of neural network. Each repeating module comprises four interacting layers as shown in Fig. 2.

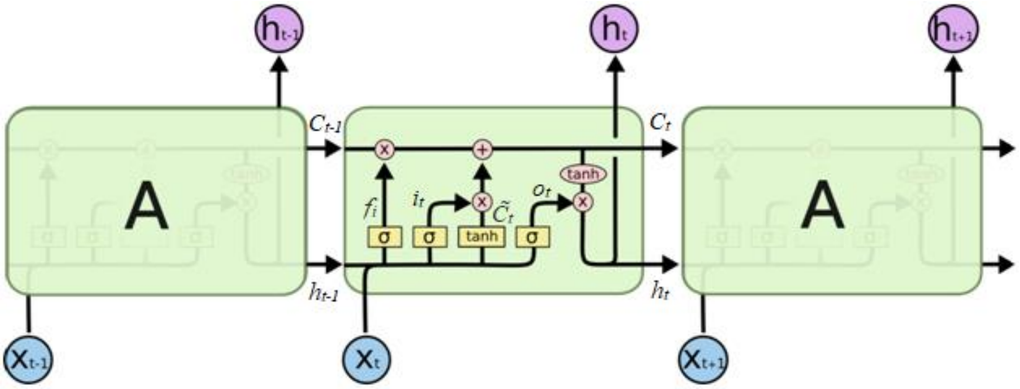

Fig. 2: The repeating module in an LSTM network (Olah, 2015). The LSTM network can be viewed as a chain of repeating modules, each including four interacting layers (input gate, forget gate, cell update and output gate)

The four layers in the LSTM unit are formulated as follows:

Input gate layer:

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i).$$

Forget gate layer:

$$f_t = \sigma\left(W_f[h_{t-1}, X_t] + b_f\right).$$

Cell update layer:

$$\tilde{C}_t = tanh(W_C[h_{t-1}, X_t] + b_C).$$

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

Output gate layer:

$$o_t = \sigma(W_o[h_{t-1}, X_t] + b_o).$$

$$h_t = o_t * tan\,h(C_t).$$

where:

σ is the logistic sigmoid function, *tanh* is the hyperbolic tangent function.

X_t is the input at time step *t*.

i_t, f_t and o_t are the input gate state, the forget gate state and the output gate state at time step *t* respectively.

C_t is the cell state at time step *t*.

h_t is the hidden state at time step *t*, also known as the output of the LSTM unit.

2.2 Long short-term memory architectures

In this paper, three types of LSTM architectures are used for the time series forecasting problem. They are vanilla LSTM, bidirectional LSTM and stacked LSTM which present the way the LSTM network is used as layers in network architectures (Jurafsky and Martin, 2019).

2.2.1 Vanilla LSTM

The vanilla LSTM is a simple LSTM architecture as shown in Fig. 3, where memory cells of a single LSTM layer are used in a simple network structure. The input layer contains inputs from time steps *1* to *n*, input for each time step is fed to the LSTM layer. The output layer with a single element is used to make prediction at next time step, which is an interpretation from the end of output sequence of LSTM units.

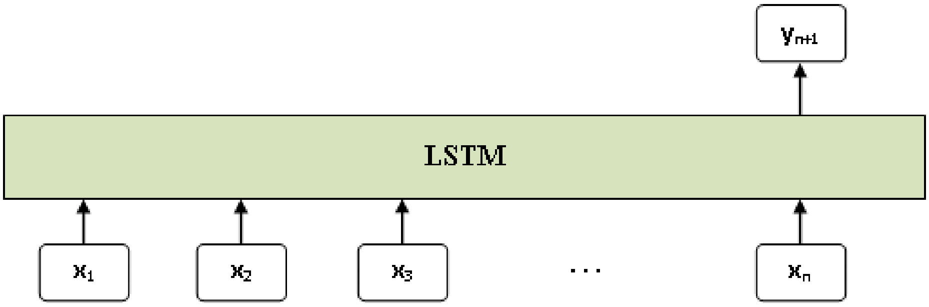

Fig. 3: Structure of a vanilla LSTM. The vanilla model takes the input sequence *x_1, x_2,…, x_n* and generates the next value *y_{n+1}*, which is an interpretation of the output from the last LSTM unit

2.2.2 Stacked LSTM

In the stacked LSTM, LSTM layers are stacked one on top of another into deep recurrent neural networks as shown in Fig. 4. The output is taken from the last LSTM layer.

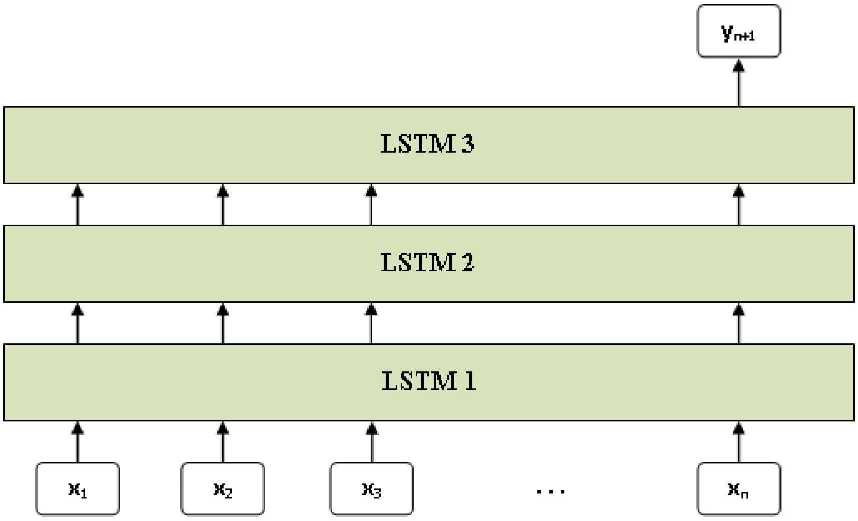

Fig. 4: Structure of a stacked LSTM. The stacked model takes the input sequence *x_1, x_2,…, x_n* and generates the next value *y_{n+1}*, which is an interpretation of the output from the last unit of the last LSTM layer

2.2.3 Bidirectional LSTM

The bidirectional LSTM model consists of two independent LSTM networks, one where the input sequence is processed from left to right and the other from right to left. This kind of LSTM architecture allows the model to learn the input sequence in both forward and backward directions and combine both interpretations at the output as shown in Fig. 5.

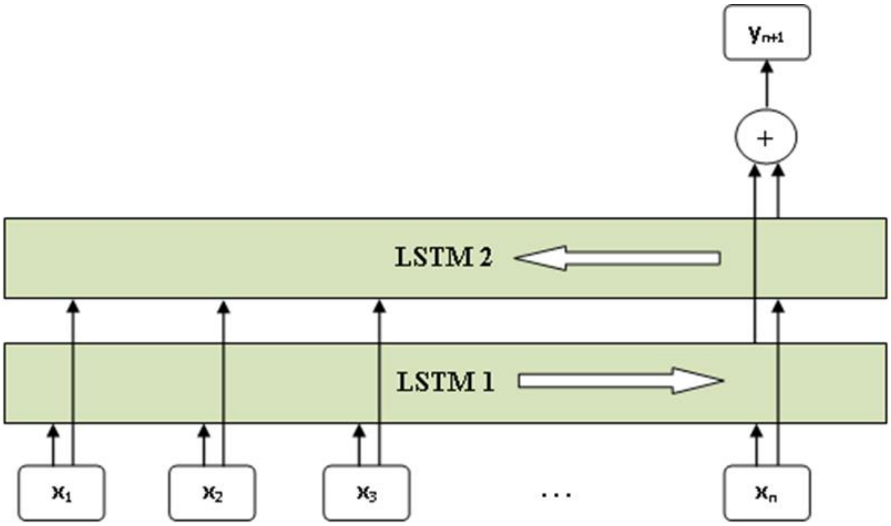

Fig. 5: Structure of a bidirectional LSTM. The bidirectional model takes the input sequence *x_1, x_2,…, x_n* and generates the next value *y_{n+1}*, which is combined from the interpretations of the outputs of the forward and backward LSTM networks

2.3 Dataset

The M-Competitions (Makridakis and Hibon, 2000; Makridakis *et al.*, 2018) have been organized for empirical studies in the field of forecasting. Various methods have been proposed and compared to each other by their forecasting performance on the benchmark datasets.

In the experiments of this paper, the M3-Competition data (M3-Competition, 2000) is used. This data consists of 3003 time series, mainly in business and economic domains. Only the yearly dataset is employed in evaluating the LSTM models to reduce the training time. The yearly dataset is subdivided into six categories (micro, industry, macro, finance, demographic and other) and includes 645 time series with different numbers of observations as shown in Table 1.

Table 1: The categories of 645 yearly time series used in the M3-Competition

Types of time series	Number of time series	Minimum observations	Maximum observations
Micro	146	20	20
Industry	102	21	47
Macro	83	22	23
Finance	58	20	47
Demographic	245	20	47
Other	11	36	38

As in the M3-Competition (Makridakis and Hibon, 2000), the number of forecasts is chosen as six for the yearly time series. In other words, the last six observations of each time series are reserved for evaluating the forecasting performance of the LSTM models, while the preceding observations are used in developing the forecasting models. The forecasted values are subsequently compared with the actual values to measure forecasting accuracy of the models.

The symmetric mean absolute percentage error (sMAPE) metric is used as the forecasting accuracy measure for the model performance evaluation, defined as:

$$\frac{100}{N}\sum_{i=1}^{N}\frac{2*|a_i - f_i|}{(a_i + f_i)}.$$

where a_i is the actual value, f_i is the forecasted value and N is the number of forecasts. The sMAPE metric is averaged across the horizon of all the forecasts. This metric is often used as an accuracy measure in forecasting competitions because it avoids the problem of large errors when the actual values a_i are close to zero, and the asymmetry in absolute percentage errors when the values a_i and f_i are different.

2.4 Experimental results and discussion

All the experiments have been run on a system Intel(R) 2-core Xeon CPU2.30GHz, 13GB RAM. The system is installed with the library packages including Tensorflow version 1.15 and Keras version 2.2 for developing and evaluating the LSTM models.

Table 2 shows the sMAPE values of the three LSTM architectures on the different categories of time series. It can be seen that the three LSTM models show the good results on the macro and demographic categories with the average sMAPE around 8% and 11.5% respectively, while their performances on the other four types of the time series data are worse with the average sMAPE close to or more than 20%. Besides, the overall average sMAPE of each LSTM model is less than 18%, particularly 17.9%, 17.3% and 17.1% for the single LSTM, the stacked LSTM and the bidirectional LSTM, respectively. In general, the bidirectional LSTM has the better performance than the two remaining LSTM models.

Table 2: The sMAPE values of the three LSTM architectures on the different categories

LSTM Model	Category of time series						Average
	Micro	Industry	Macro	Finance	Demographic	Other	
Single LSTM	27.7	19.7	8.0	27.9	11.7	29.6	17.9
Stacked LSTM	25.8	19.5	7.7	28.4	11.4	28.8	17.3
Bidirectional LSTM	25.8	19.2	7.5	27.6	11.3	29.1	**17.1**

Table 3 shows the sMAPE values of the LSTM models on the different forecasting horizons. The LSTM models achieve low absolute percentage errors at the first time steps, particularly close to 7.5% and 11.5% at the time steps *1* and *2*. The errors become larger when the time steps increase. Overall, the bidirectional LSTM model obtains the lower average sMAPE compared to the single and stacked LSTM models on the next four and six forecasts. The main reason might come from the fact that the bidirectional LSTM model can learn the time series trends in both the forward and backward directions.

Table 3: The sMAPE values of the LSTM models on the different forecasting horizons

LSTM Model	Forecasting Horizon						Average	
	1	2	3	4	5	6	1 to 4	1 to 6
Single LSTM	7.7[*]	11.7	16.8	20.0	24.1	26.9	14.07[**]	17.88
Stacked LSTM	7.7	11.6	16.5	19.3	23.1	25.4	13.79	17.28
Bidirectional LSTM	7.6	11.5	16.5	19.2	22.8	24.9	**13.69**	**17.08**

[*] *The sMAPE values on the different forecasting horizons are rounded to one decimal place.*

[**] *The average sMAPE values are rounded to two decimal places.*

Table 4 shows the sMAPE values on the same yearly dataset of several baseline models proposed by the competitors participating in the M3-Competition. It is seen that the LSTM models show the better results than most of the proposed models in Table 4 regarding the average sMAPE on the next four and six forecasts. Exceptionally, the LSTM models have the lower performance than the Autobox2 model. In particular, the average sMAPE of the bidirectional LSTM is 13.69% higher than that of the Autobox2 (13.65%) on the next four forecasts, and is 17.08% higher than 16.52% of the Autobox2 on the next six forecasts. However, the bidirectional LSTM obtains the lower sMAPE values than the Autobox2 at the very first forecasting horizons; for example, 7.6% and 11.5% of the bidirectional LSTM lower than 8% and 12.2% of the Autobox2 at the first and second horizons respectively. In the M3-Competition, the Autobox2 model is shown to be the best performer on the next four forecasts, and one of the best performers on the next six forecasts when it is evaluated on the yearly dataset with the sMAPE accuracy measure.

Table 4: The sMAPE values of several baseline methods in the M3-Competition

LSTM Model	Forecasting Horizon						Average	
	1	2	3	4	5	6	1 to 4	1 to 6
Holt[a]	8.3	13.7	19	22	25.2	27.3	15.77	19.27
Winter[b]	8.3	13.7	19	22	25.2	27.3	15.77	19.27
Dampen[c]	8	12.4	17	19.3	22.3	24	14.19	17.18
B–J automatic[d]	8.6	13	17.5	20	22.8	24.5	14.78	17.73
Autobox1[e]	10.1	15.2	20.8	24.1	28.1	31.2	17.57	21.59
Autobox2[e]	8	12.2	16.2	18.2	21.2	23.3	**13.65**	**16.52**
Autobox3[e]	10.7	15.1	20	22.5	25.7	28.1	17.09	20.36
ARARMA[f]	9	13.4	17.9	20.4	23.8	25.7	15.17	18.36
Automat ANN[g]	9.2	13.2	17.5	20.3	23.2	25.4	15.04	18.13

[a] *Automatic Holt's Linear Exponential Smoothing (two parameter model).*

[b] *Holt–Winter's linear and seasonal exponential smoothing (two or three parameter model).*

[c] *Dampen Trend Exponential Smoothing.*

[d] *Box–Jenkins methodology of 'Business Forecast System'.*

[e] *Robust ARIMA univariate Box–Jenkins with/without Intervention Detection.*

[f] *Automated Parzen's methodology with Auto regressive filter.*

[g] *Automated Artificial Neural Networks for forecasting purposes.*

3 CONCLUSIONS

In this paper, three different LSTM architectures are introduced for time series forecasting problem. They include the vanilla LSTM, the stacked LSTM, and the bidirectional LSTM. They perform well on the macro and demographic categories of the benchmark time series dataset. The bidirectional LSTM shows the best results among the three LSTM models in both the experiments of different time series data categories and forecasting horizons. In comparison with the baseline models, the LSTM models achieve the better performance than most of them except for the Autobox2 model. In future work, ensemble learning models combined with LSTM will be used to forecast time series data, as well as these models will be evaluated on various benchmark time series datasets.

REFERENCES

Box, G. and Jenkins, G., 1970. Time Series Analysis: Forecasting and Control. San Francisco: Holden-Day.

Cho, K., Merrienboer, B., Gulcehre, C., *et al.*, 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv preprint. https://arxiv.org/abs/1406.1078v3

Gers, F.A., Schmidhuber, J. and Cummins, F., 1999. Learning to forget: continual prediction with LSTM. 9th International Conference on Artificial Neural Networks: ICANN '99, Edinburgh, UK, 850-855. https://doi.org/10.1049/cp:19991218

Gers, F.A., Schmidhuber, J. and Cummins, F., 2000. Learning to Forget: Continual Prediction with LSTM. Neural Computation, 12(10): 2451-2471. https://doi.org/10.1162/089976600300015015

Guo, T., Xu, Z., Yao, X., Chen, H., Aberer, K. and Funaya, K., 2016. Robust online time series prediction with recurrent neural networks. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Montreal, QC, Canada, pp. 816-825.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural Computation, 9(8): 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hsu, D., 2017. Time series forecasting based on augmented long short-term memory. arXiv preprint https://arxiv.org/abs/1707.00666

Jurafsky, J. and Martin, J.H., 2019. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (third edition draft), accessed on 27 February 2020. Available from https://web.stanford.edu/~jurafsky/slp3/edbook_oct162019.pdf

Karpathy, A., 2015. The Unreasonable Effectiveness of Recurrent Neural Networks, accessed on 27 February 2020. Available from http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Li, X. and Wu, X., 2015. Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. *In*: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, pp. 4520-4524.

M3-Competition, 2000. The 3003 Time Series of The M3-Competition, accessed on 01 February 2020. Available from https://forecasters.org/resources/time-series-data/m3-competition/

Makridakis, S. and Hibon, M., 2000. The M3-Competition: results, conclusions and implications. International Journal of Forecasting, 16(4): 451-476. https://doi.org/10.1016/S0169-2070(00)00057-1

Makridakis, S., Spiliotis, E. and Assimakopoulos, V., 2018. The M4 Competition: Results, findings, conclusion and way forward. International Journal of Forecasting, 34 (4): 802-808. https://doi.org/10.1016/j.ijforecast.2018.06.001

Malhotra, P., Vig, L., Shroff, G. and Agarwal, P., 2015. Long short-term memory networks for anomaly detection in time series. In: ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), pp. 89-94.

Olah, C., 2015. Understanding LSTM Networks, accessed on 27 February 2020. Available from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Rumelhart, D., Hinton, G. and Williams, R., 1986. Learning representations by back-propagating errors. Nature, 323: 533–536. https://doi.org/10.1038/323533a0

Sutskever, I. and Vinyals, O. and Le, Q. V., 2014. Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems, 27: 3104-3112.

Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M. and Baik, S.W., 2017. Action Recognition in Video Sequences using Deep Bi-Directional LSTM with CNN Features. IEEE Access, 6: 1155-1166.

Vinyals, O., Toshev, A., Bengio, S. and Erhan, D., 2015. Show and Tell: A Neural Image Caption Generator. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3156-3164.