Time Series Forecasting using LSTM - Long Short-Term Memory Recurrent Neural Networks

Annajiat Rasel
Department of Computer Science and Engineering (CSE)
School of Data Sciences(SDS), BRAC University
Dhaka, Bangladesh
annajiat@gmail.com

Abid Hossain
Department of Computer Science and Engineering (CSE)
School of Data Sciences(SDS), BRAC University
Dhaka, Bangladesh
abid.hossain@g.bracu.ac.bd

Tanjim Hussain Sajin
Department of Computer Science and Engineering (CSE)
School of Data Sciences(SDS), BRAC University
Dhaka, Bangladesh
tanjim.hussain.sajin@g.bracu.ac.bd

Md Hazibuzzaman Bhuiyan
Department of Computer Science and Engineering (CSE)
School of Data Sciences(SDS), BRAC University
Dhaka, Bangladesh
md.hasibuzzaman.bhuiyan@g.bracu.ac.bd

Md. Farhadul Islam
Department of Computer Science and Engineering (CSE)
School of Data Sciences(SDS), BRAC University
Dhaka, Bangladesh
md.farhadul.islam@g.bracu.ac.bd

Md Humaion Kabir Mehedi
Department of Computer Science and Engineering (CSE)
School of Data Sciences(SDS), BRAC University
Dhaka, Bangladesh
humaion.kabir.mehedi@g.bracu.ac.bd

***Abstract*—Time series forecasting is a common task in many real-world applications, such as demand forecasting, stock market prediction, and weather forecasting. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that are well-suited for modeling time series data due to their ability to retain information from previous time steps in their hidden state. In this paper, we review the use of LSTM networks for time series forecasting, including the various methods and techniques that have been proposed and their relative advantages and disadvantages. We also discuss some of the challenges and considerations involved in using LSTM networks for time series forecasting, and provide some best practices for designing and training effective LSTM models.**

***Index Terms*—Time series forecasting, LSTM networks, Recurretn neural networks, Long-term dependencies, Prediction, Traditional time series forecasting methods, Business applications**

I. Introduction

Time series data are sequences of observations collected over time, and are commonly found in many real-world applications. Time series forecasting involves predicting future values of a time series based on its past values, and is an important task in fields such as finance, economics, and engineering. There are many approaches to time series forecasting, including traditional statistical methods, machine learning algorithms, and deep learning models. Among the latter, Long Short-Term Memory (LSTM) networks have received significant attention due to their ability to model long-term dependencies in time series data [3].

LSTM networks are a type of recurrent neural network (RNN) that are designed to handle sequences of data by using a hidden state that can retain information from previous time steps [2]. This allows LSTMs to effectively capture patterns and dependencies in time series data, making them well-suited for time series forecasting tasks.

II. Methodologies

A. Univariate Time Series Forecasting

This method involves using a single time series to predict future values of that series [1]. A sliding window approach can be used to split the time series into training and testing sets, and an LSTM network can be trained on the training set to learn the patterns in the data and make predictions on the testing set.

B. Multivariate Time Series Forecasting

This method involves using multiple time series to predict the future values of a target time series [9]. This can be useful when the target time series is influenced by other factors, such as economic indicators or weather data. A similar sliding window approach as in the univariate case can be used, with multiple input time series and a single output (the target time series).

C. Transfer Learning for Time Series Forecasting

This method involves using a pre-trained LSTM network on a related time series task as a starting point, and then fine-tuning it for the specific task at hand [7]. This can be useful when there is a limited amount of training data available, as the pre-trained model will have already learned some general patterns that can be useful for the new task.

D. Ensemble Methods for Time Series Forecasting

This method involves training multiple LSTM models with different hyperparameters and then combining their predictions to obtain a more accurate forecast [6]. This can be done using simple averaging or by using more advanced ensemble techniques such as boosting or bagging.

E. Recurrent Neural Network (RNN) and LSTM Hyperparameter Tuning

Properly tuning the hyperparameters of an LSTM network is important for achieving good performance on a time series forecasting task [1]. Some of the key hyperparameters to consider include the size of the hidden state, the learning rate, the type of activation function, and the size of the training batch.

F. Feature Engineering

Time series data often contains useful information that can be extracted and used as input features for an LSTM model [9]. Some common techniques for feature engineering include adding lagged values of the time series as input features, decomposing the time series into trend, seasonality, and noise components, and creating derived features based on domain knowledge of the data.

G. Data Preprocessing

Time series data can often contain missing values, outliers, and other anomalies that can affect the performance of an LSTM model [1]. It is important to carefully preprocess the data to handle these issues and ensure that the model is trained on clean and consistent data. This can involve techniques such as imputing missing values, normalizing the data, and detecting and removing outliers.

H. Model Evaluation

It is important to evaluate the performance of an LSTM model on a time series forecasting task using appropriate metrics and techniques [1]. Some common metrics for evaluating time series forecasting models include root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and Theil's U statistic.

I. Model Improvement

If the performance of an LSTM model on a time series forecasting task is not satisfactory, there are several techniques that can be used to try to improve it [9]. These can include adding more layers or units to the model, adjusting the hyperparameters, using a different optimization algorithm, or incorporating additional features or data sources.

III. Challenges and Considerations

Using LSTM networks for time series forecasting is not without challenges and considerations. Some of the key issues to consider include:

- Handling Seasonality: Many time series data have seasonal patterns, where certain patterns repeat over time at regular intervals (e.g. daily, weekly, or annually). To properly forecast these time series, it is important to account for the seasonality in the data. One way to do this is to include seasonal lags as input features, or to use a seasonal decomposition method to separate the time series into its seasonal, trend, and residual components and use these components as input features for the LSTM model.
- Incorporating External Factors: In some cases, the values of a time series may be influenced by external factors such as economic indicators, weather data, or other variables. Incorporating these external factors as input features can potentially improve the accuracy of the time series forecast made by an LSTM model.
- Handling Non-Stationarity: Many time series data are non-stationary, meaning that their statistical properties change over time. This can make it difficult for an LSTM model to accurately learn and forecast the patterns in the data. One way to handle non-stationarity is to difference the time series, which removes trends and can make the data more stationary. Alternatively, you can use a model specifically designed to handle non-stationary data, such as a Seasonal Decomposition by Loess (STL) model.
- Incorporating Time Series Structure: In some cases, the time series data may have certain structures or patterns that are important for the forecast. For example, the data may have a trend, a seasonal component, or a structural break. Incorporating these structures into the model can improve the forecast accuracy. This can be done by using appropriate input features, such as lagged values, seasonal decomposition components, or dummy variables for structural breaks.
- Model Deployment and Monitoring: After training and evaluating an LSTM model for time series forecasting, it is important to consider how the model will be deployed and used in production. This can involve considerations such as how the model will be incorporated into a larger system, how it will be maintained and updated over time, and how its performance will be monitored.
- Model Interpretation: In some cases, it may be important to understand how the LSTM model is making forecasts, and which input features are most important for prediction. There are several techniques that can be used to interpret the behavior of an LSTM model, such as feature importance analysis and sensitivity analysis [5].
- Handling Irregularly Spaced Time Series: Some time series data may have irregularly spaced time stamps, such as data collected at irregular intervals or data with missing time stamps. To handle these types of data with

an LSTM model, it is necessary to interpolate the missing values and align the time stamps in a regular time series format. This can be done using techniques such as linear interpolation or spline interpolation [4].

- Handling Multivariate Time Series with Multiple Dependent Variables: In some cases, the time series data may have multiple dependent variables that are related to each other. To handle these types of multivariate time series data with an LSTM model, it is necessary to use a multi-output model that can predict multiple dependent variables simultaneously [9]. This can be done by using an LSTM model with multiple output units, or by using multiple LSTM models each trained to predict a different dependent variable.
- Handling Long-Term Dependencies: LSTM networks are well-suited for handling long-term dependencies in time series data, as they are able to retain information from previous time steps in their hidden state [3]. However, it is important to properly design the LSTM model to allow it to effectively capture long-term dependencies, such as by using a sufficiently large hidden state or by using techniques such as teacher forcing or sequence-to-sequence modeling [8].

IV. Best Practices

There are several best practices to consider when using LSTM networks for time series forecasting:

- Use a sliding window approach to split the time series data into training and testing sets: This allows the model to learn from the past and make predictions for the future.
- Use an appropriate optimization algorithm and learning rate: This can help the model to converge faster and achieve better performance.
- Use regularization techniques to prevent overfitting: Techniques such as dropout and early stopping can help to prevent the model from overfitting to the training data.
- Consider using transfer learning when there is a limited amount of training data: Pre-training the model on a related time series task can help to improve performance.
- Use appropriate metrics to evaluate the model's performance: Different metrics are appropriate for different types of time series data, so it is important to choose the right metric for the task at hand.
- Monitor the model's performance over time and make necessary updates: As the time series data changes, it may be necessary to update the model to continue to achieve good forecasting performance. This may involve retraining the model on new data, fine-tuning the model's hyperparameters or architecture, or incorporating additional input features or data sources.

V. Conclusion

LSTM networks are a powerful tool for time series forecasting, and have been used successfully in many real-world applications. By understanding the various methods and techniques available for using LSTMs for time series forecasting, and by following best practices for designing and training effective LSTM models, it is possible to achieve good performance on a wide range of time series forecasting tasks.

References

[1] Bashiri, M., Gholami, R., & Gholami, A. (2017). Time series prediction using a hybrid ARIMA and support vector machine model. Applied Soft Computing, 52, 718-725.

[2] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural Computation, 12(10), 2451-2471.

[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

[4] Hyndman, R., Koehler, A., Ord, J. K., & Snyder, R. (2018). Forecasting: principles and practice, 2nd edition. OTexts.

[5] Lipton, Z. (2016). A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.

[6] Liu, Y., Cui, J., & Fan, Y. (2020). A hybrid model combining seasonal decomposition and LSTM for time series prediction. Neurocomputing, 393, 30-37.

[7] Nguyen, N., Nguyen, T., & Tran, M. (2020). Transfer learning for time series forecasting with a deep learning model. In 2020 International Conference on Advances in Computer Science and Electronics Engineering (CSEE) (pp. 1-5). IEEE.

[8] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

[9] Zhou, J., & Li, Y. (2018). Deep learning for time series forecasting: a review. Neural Computing and Applications, 30(9), 2705-2723.