International Journal of Academic Research and Development
www.academicjournal.in
ISSN: 2455-541X
Received: 05-02-2022, Accepted: 20-02-2022, Published: 07-03-2022
Volume 7, Issue 2, 2022, Page No. 6-10

Behavioral Analysis of LSTM and BiLSTM Models on prediction and forecasting

Yasir Ahmad Itoo[1], Ikhlas Ahmad Sheikh[2]
[1] MSc Information Technology, University of Kashmi, Jammu and Kashmir, India
[2] Master in Computer Applications, I. K. Gujral Punjab Technical University, Punjab, India

Abstract

A time series is a series of data points ordered in timely manner. Time Series can be regular or irregular based on the intervals over which it has been collected. Usually, the time series data has been collected over a period of time at regular intervals and same is used to observe patterns among it thereby predicting or forecasting future events. Apart from the conventional regression-based [1] modeling, deep learning-based algorithms [2] are the well-meaning approaches in addressing prediction/forecasting problems in time series and where the latter technique have been shown to produce more accurate results than former. A Lot of Researches suggest that Recurrent Neural Networks (RNN) with memory, such as Long Short-Term Memory (LSTM), are superior compared to Autoregressive Integrated Moving Average (ARIMA) or Seasonal Autoregressive Integrated Moving Average (SARIMA) with a huge margin. In order to memorize large sequences, the neurons in LSTM-based models comes with three different gates. Are these gates alone enough for the purpose of memorizing? Will more training on data improve the prediction or forecasting? Bidirectional LSTMs provide further training on the input data twice. i.e., It first traverses the data from the left to the right and then traverses back from right to left. Whether BiLSTM, with additional training capability, outperforms regular unidirectional LSTM? Here we report the behavioral analysis as well as comparison of LSTM and BiLSTM models. The objective is to explore to what extend additional layers of training of data would be beneficial to tune the involved parameters. The results show that additional training of data and thus

The results show that BiLSTM based modeling offers better predictions than LSTM based models.

Even the BiLSTM based models provide a lot fine predictions when compared to the ARIMA as was observed practically. Another important thing to mention is that BiLSTM models take much time to reach the equilibrium than the regular LSTM-based models.

Keywords: LSTM and BiLSTM, prediction and forecasting

Introduction

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. A time series is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Time series data are simply measurements or events that are tracked, monitored, down sampled, and aggregated over time. Time-Series Data are frequently encountered these days, for example in economic, stock price, weather data, etc. The most challenging yet interesting aspect for the time series data analysis is Forecasting. The important factors effecting the performance and accuracy of time series data analysis and forecasting techniques are the type of time series data alongside the underlying context. Seasonality, unexpected events, wars, workforce imbalance, internal changes of an organization etc. are some of the domain-based factors that affect prediction.

“Auto-Regressive Integrated Moving Average”, also known as ARIMA which is a linear regression-based approach has many variations like SARIMA, ARIMAX can do justice with the time series problems for long-term predictions. These approaches are model-driven. Machine learning especially deep learning has enhanced the data analytical processes to a large extent as the models built are data-driven rather than model-driven. Based on the domain, a suitable model can be used training like RNNs are used for time series while as CNNs are used for image-based data processing.

To find out that what role does incorporating of additional layers for training into the architecture of an LSTM, this paper brings into use Bidirectional LSTM (BiLSTM) where the given input data is utilized twice for training. For this paper we are going to analyze the behavior of both LSTM & BiLSTM given a particular problem. For the same this paper reports the performance and behavior of both these models based on the experimental results. For the record we will get answers of the following research-based questions:

1. Which architecture performs well at the end?
2. How does both differentiate at the architectural level?
3. How does both the architecture’s treat the incoming data?

4. Which one of the following is time and resource efficient?

In order to answer these queries, this paper conducts a series of experiments on the available time series data thereby puts forward the results. This paper gives deep understanding of the both architectures and provides sufficient evidence in support of the best one. Also, it helps one to understand what is the impact of the number of layers onto the results generated. It also provides useful insights regarding what and how the data should be used.

Background

A). Long Short-Term Memory (LSTM) Models

As RNNs have difficulties in learning long-term dependencies. The LSTM-based models which are an extension for RNNs, are able to address the vanishing gradient problem in a very clean way. LSTMs have an edge over conventional feed forward neural networks and RNN in many ways. This is because of their property of selectively remembering patterns for long durations of time. LSTMs on the other hand, make small modifications to the information by multiplications and additions. With LSTMs, the information flows through a mechanism known as cell states. By this LSTMs can selectively remember or forget things based upon the gated values. This memory extension has the ability of remembering information over a longer period of time and thus enables reading, writing, or deleting information from these cell memories. The LSTM have the ability to make the decision of preserving or ignoring the memory information. The LSTM model is able to identify and capture important features from the inputs at each neuron/cell over a long period of time. Weights are assigned to the processing information during training and thereby decision regarding deleting or keeping of information is made. Therefore, LSTM model is able to learn which information is to be kept or removed. Generally, an LSTM model consists of three gates: output, input, and forget gates.

The forget gate makes the decision of preserving/removing the existing information, the input gate specifies the extent to which the new information will be added into the memory, and the output gate controls whether the existing value in the cell contributes to the output.

I) Forget Gate.

A forget gate is responsible for removing information from the cell state. The information that is no longer required for the LSTM to understand things or the information that is of less importance is removed via multiplication of a filter. This is required for optimizing the performance of the LSTM network. This gate takes in two inputs; h_t-1 and x_t, where h_t-1 is the hidden state or output of the previous cell and x_t is the input at that particular time step. The sigmoid function applied outputs a vector, with values ranging from 0 to 1, corresponding to each number in the cell state. Basically, the sigmoid function is responsible for deciding which values to keep and which to discard. If a ‘0’ is output for a particular value in the cell state, it means that the forget gate wants the cell state to forget that piece of information completely. Similarly, a ‘1’ means that the forget gate wants to remember that entire piece of information. This vector output from the sigmoid function is multiplied to the cell state. . This output is computed as:

$$f(t) = \sigma(\mathrm{Wfh}\ [\mathrm{ht} - 1], \mathrm{Wfx}\ [\mathrm{xt}], \mathrm{bf}\ )$$

II) Input Gate.

The input gate is responsible for the addition of new information to the cell state. This gate consists of two layers: 1) a sigmoid layer, and 2) a tanh layer. This addition of information is basically three-step process.

a). Regulating what values need to be added to the cell state by involving a sigmoid function. This is basically very similar to the forget gate and acts as a filter for all the information from h_t-1 and x_t.

b). Creating a vector containing all possible values that can be added (as perceived from h_t-1 and x_t) to the cell state. This is done using the tanh function, which outputs values from -1 to +1.

c). Multiplying the value of the regulatory filter (the sigmoid gate) to the created vector (the tanh function) and then adding this useful information to the cell state via addition operation. Once this three-step process is done with, it ensures that only that information is added to the cell state that is important and is not redundant. The following equation represents its mathematical equation:

$$i(t) = \sigma(\mathrm{Wih}\ [\mathrm{ht} - 1], \mathrm{Wix}\ [\mathrm{xt}], \mathrm{bi}$$

$$\mathrm{c}\tilde{}\ \mathrm{t} = \tanh(\mathrm{Wch}\ [\mathrm{ht} - 1], \mathrm{Wcx}\ [\mathrm{xt}], \mathrm{bc}$$

where *I(t)* represent whether the value needs to be updated or not, and c˜ t indicates a vector of new candidate values that will be added into the LSTM memory.

III) Output Gate.

This job of selecting useful information from the current cell state and showing it out as an output is done via the output gate. Here is its structure: The functioning of an output gate can again be broken down to three steps:

1. Creating a vector after applying tanh function to the cell state, thereby scaling the values to the range -1 to +1.
2. Making a filter using the values of h_t-1 and x_t, such that it can regulate the values that need to be output from the vector created above.
3. Multiplying the value of this regulatory filter to the vector created in step 1, and sending it out as an output and also to the hidden state of the next cell. This job of selecting useful information from the current cell state and showing it out as an output is done via the output gate. The following equation represents the formulas to compute the output:

$$o(t) = \sigma(\mathrm{Woh}\,[\mathrm{ht} - 1], \mathrm{Wox}\,[\mathrm{xt}], \mathrm{bo})$$

$$\mathrm{h(t)} = \mathrm{o(t)} * \tanh(\mathrm{ct})$$

Where *o(t)* is the output value, and *h(t)* is its representation as a value between −1 and 1.

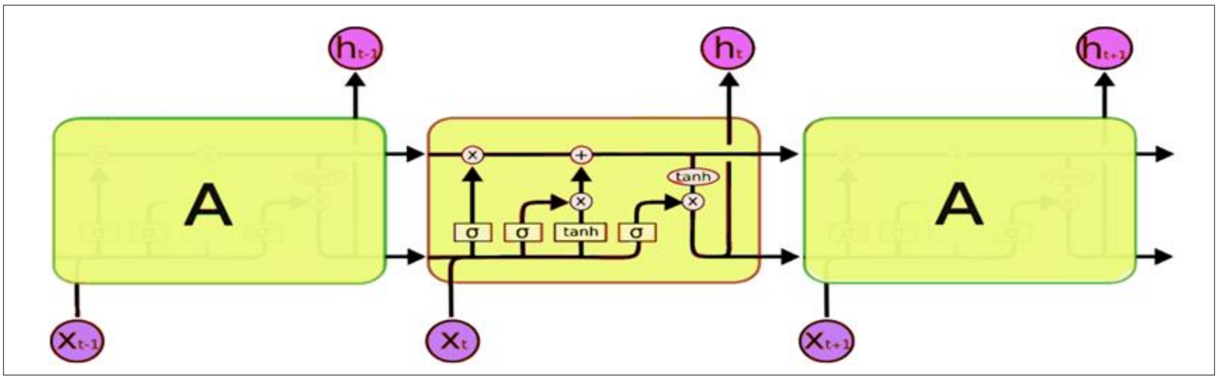


Fig 1

B). Bidirectional LSTMs (BiLSTM)

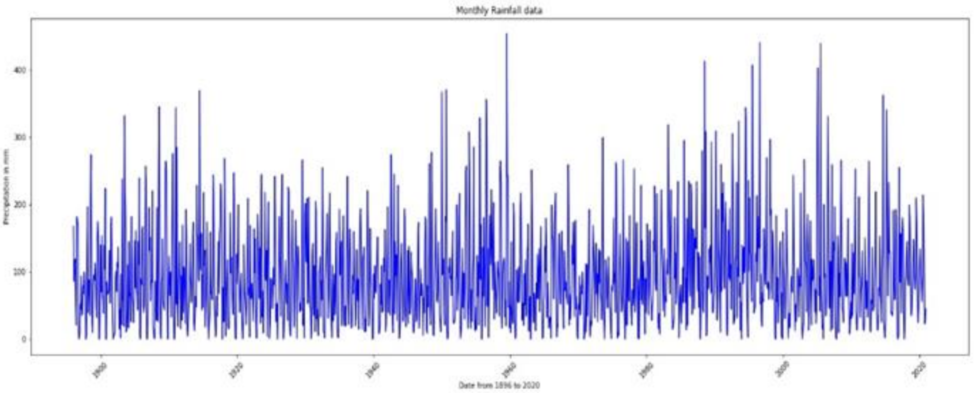
The bidirectional LSTMs which is an extension of the already described LSTM models applies two LSTMs to the input data. Firstly, an LSTM is applied on the input sequence in forward manner (i.e., forward layer) then reverse form of the input sequence is fed into the LSTM model in backward manner (i.e., backward layer). Applying the LSTM twice in forward as well as backward manner leads to improved learning long-term dependencies and thereby improves the results of the model.

An experimental study using LSTM and BiLSTM

This paper compares the performance of LSTM, and BiLSTM in the context of predicting and forecasting weather time series.

A. Data Set

The data which is available open source on one of the Government website of India over the internet (i.e., open Government Data (OGD) Platform India) [3]. The data in this data set is available for almost all the states of India and the data is available in monthly manner for about 100 years. Our prime focus of observation was the data available for the state of Jammu and Kashmir.



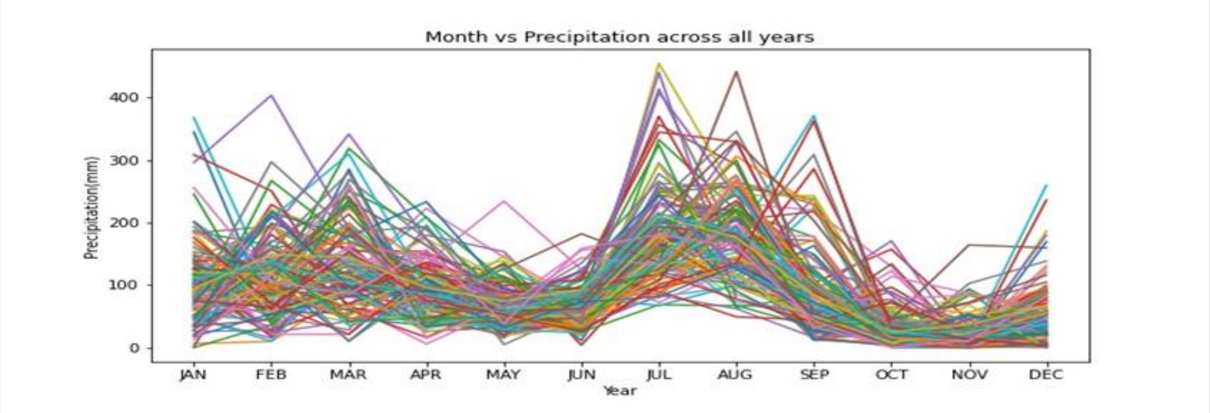


Fig 2

B. Training and Test Data

The “Rainfall (in mm)” variable was chosen as the only feature of weather time series to be fed into the LSTMs [4] and BiLSTM [5] models. The data set was divided into training and test where 80% of each data set were used for training and 20% of each data set was used for testing the accuracy of models.

C. Assessment Metrics

The purpose behind evaluating a model is to bring out a systematic approach that can measure the success, effectiveness and efficiency of the training exercise. Mean Absolute Error, Mean Squared Error and Root Mean Square Error [6] are the three metrics describing the performance of a predictive Regression Model. The main benefit of using RMSE is that it penalizes large errors. It also scales the scores in the same units as the forecast values. RMSE measures the differences between actual and predicated values. The formula for computing RMSE is:

$$\boldsymbol{RMSE} = \sqrt{\sum_{i=1}^{N}}\frac{(\mathbf{yi}\ -\ \mathbf{y\hat{}i})\mathbf{\hat{}2}}{N}$$

Where N is the total number of observations, yi is the actual value; whereas, yˆi is the predicated value.

Algorithms

Artificial Neural Networks (ANNs) allow training the model in a feed-forward manner i.e., traveling in one direction only without any mechanism of feedback from the past data. As a result, the output of any layer does not affect the training process performed on the same layer (i.e., no memory). Much or less these models perform the same as regression-based modelling. Such models are mainly used in pattern recognition. RNNs differ from ANNs as there exist a mechanism called feedback by which it can remember the past data (in parts). RNNs are dynamic as their state keeps on changing continuously until they reach the equilibrium status and are thus optimized. RNNs cannot preserve and thus does not remember long inputs which is their biggest disadvantage. To dealt with it Long Short-Term Memory (LSTM) was introduced which is an extension of RNNs. LSTMs remember long sequence of data through the utilization of several gates at neuron level which are as:

1) input gate, 2) forget gate, and 3) output gate. Furthermore, BiLSTMs are enhanced LSTMs, in which the model is trained in both directions (i.e., inputs to outputs, and outputs to inputs).

Two algorithms were developed each using LSTM & BiLSTM. Both the algorithms were fed the same data in-order to check the resulting behavior and comparison of the results. It is important to mention that there are many other factors that one should keep a good eye on like no. of epochs, batch size, activation function which for our case would be same for both the algorithms.

Results

While evaluating our models we used different metrics to compute the results like Mean Absolute Error, Mean Square Error and Rooted Mean Squared Error (RMSE). The observed results are shown in the table given below:

Table 1

Model	***MAE***	***MSE***	***RMSE***
LSTM	1.01257	0.39856	0.6313
BiLSTM	0.09857	0.02104	0.14508

Clearly, a significant reduction in the magnitude of the MAE, MSE and RMSE values is observed. As reported in the table, there is a huge difference in the values of the different metrics used which we have used for the evaluation of the performance of our model. We can totally see that from the data above, it is apparent that BiLSTM models perform well with respect to the LSTM models significantly. The results thereby can be used to conclude that modeling using BiLSTM instead of LSTM and any other conventional models (like SARIMA, ARIMA etc.,) improves the prediction accuracy with a huge margin.

Conclusion and future work

This paper reported the results of a machine learning project which was to predict the weather for a state given the historic time series data, thereby analyzing the performance and accuracy of the models as well as behavioral aspects of the model during the training. Vanilla LSTM (LSTM), and bidirectional LSTM (BiLSTM) models were analyzed and compared. The research question that we wanted to get answers which were mentioned at the start of this page can now be concluded. Training of data from an both directions (i.e., from left to right and right to left), compared to the regular form of training of data (i.e., left to right only) have a significant impact on improving the precision of time series forecasting. From the results we can conclude that with additional layer of training (BiLSTM) it helps in improving the accuracy of forecast by a huge percentage, thus is beneficial for modeling purposes of time series problems. It was also observed that during the behavioral analysis of Vanilla LSTM and BiLSTM models, that training based on BiLSTM is slower and it further involves fetching of additional batches of data to reach the equilibrium. We can conclude that with BiLSTM taking more time to reach equilibrium it helps it to understand the relation between the available data more minutely. Also, we can conclude that LSTM lacks on this capability of tracking minute details during training thereby leading to less efficiency than BiLSTM. As a result, this paper recommends using BiLSTM instead of LSTM for forecasting problem in time series analysis. This research can be further expanded to forecasting problems for time series problems with more than on feature and seasonal time series.

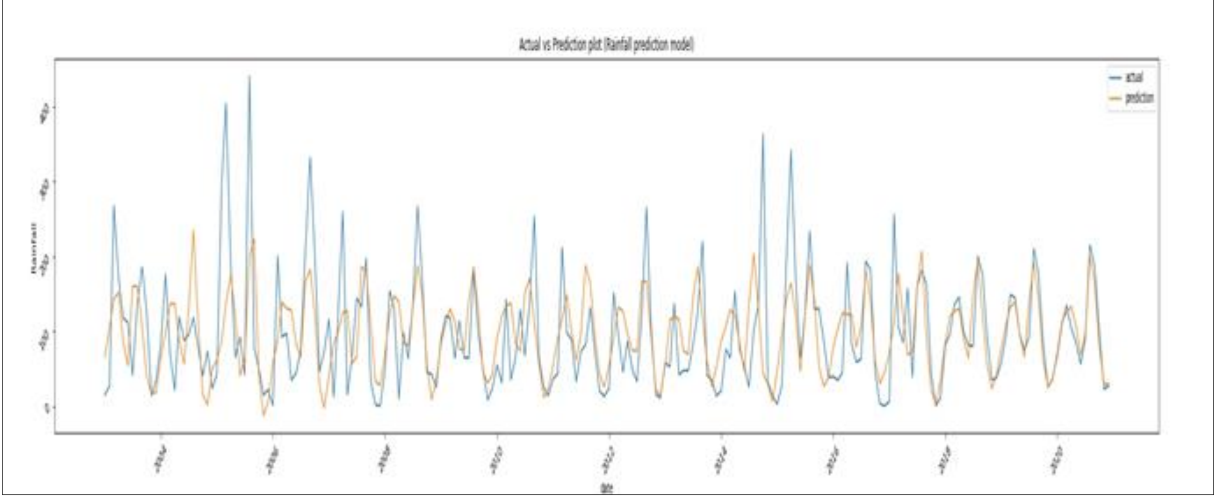


Fig 3

References

1. An Application of Time Series Analysis in Modeling Monthly Rainfall Data for Maiduguri, North Eastern Nigeria by Emmanuel Sambo Uba Department of Statistics, Ramat Polytechnic, Maiduguri H R Bakari Department of Mathematics & Statistics, University of Maiduguri.
2. An easy approach to Neural Networks. https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9?gi=94295b6a8720
3. The data set used was acquired from the open Government Data (OGD) Platform India: https://data.gov.in/resources/rainfall-all-india-and-its-departure-normal-during-monsoon-session-june-sept-1901-2019
4. rainfall prediction using machine learning Syeda Roshni Ahmed, Aathira Das, Kavya K Naik, Arunashree
5. BiLSTM model based on multivariate time series data in multiple field for forecasting trading area by Jinah Kim and Nammee Moon.
6. Essentials of deep learning: LSTM MODEL EVALUATION https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/