Course Project for CPTS 534:

Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN) for Forecasting Temperature and Comparison with Classical Time Series Model Results

Md Mahedi Hasan[1]

Abstract

With the help of efficient algorithm and high computational power machine, now a day, the deep learning methods are widely used to forecast the climatic parameters such as temperature, rainfall, humidity etc. However, the existing methods such as time domain approaches for forecasting the sequential data are also widely used. In this study, using the monthly temperature data a Recurrent Neural Network (RNN) has been designed that has the Long Short-Term Memory (LSTM). The model has learnt from the past time dependent data and based on that learning the model made the forecast for the future data. Besides, this study also adopted some classical time series forecasting methods such as Auto Regressive Moving Average (ARMA) model to forecast the same data. The accuracy of forecast between the classical time domain model and the deep learning model has been compared to see if there is any clear winner in better forecast.

Keywords: Recurrent Neural Network (RNN), Long Short-Term memory (LSTM), Auto Regressive Integrated Moving Average Model (ARIMA), Temperature, Forecast

1. Problem Background

There is no denying fact that the machine learning or the deep learning methods are often outperforming many traditional techniques in terms of the prediction performances and that is one of the main reasons for which the machine learning or the deep learning techniques are getting more and more popularity across the disciplines. The other benefits of the using the machine learning techniques is that they can deal with wide varieties of data and also can deal with large volume of data. However, there are researches going on where the size of the sample in the study or the varieties of the data are not diverse because of the nature of those studies. One such study could the forecasting a single climatic parameter which has data for few decades. So, by the definition of big data or large data set, this is actually not a very large data. Moreover, there are standard methods available for forecasting this type of the variable in the literatures. And these methods are basically dependent of different assumptions and follows the classical approaches to get the results. However, it is of interest to see that if we apply the machine learning or the deep learning techniques with the small-scale data set, can we expect to have better result than

[1] WSU ID: 011682905, Graduate Student (PhD in Statistical Science) in the Dept. of Mathematics and Statistics

the conventional methods in terms of getting better accuracy? This study is basically designed focusing this question in mind. I have been working with time series data with the conventional statistical models and methods and after doing this course I was thinking about doing a small scale study to see if the machine learning methods are still outperforming the traditional statistical assumptions based method in terms of prediction.

Besides, the reason for choosing climatic data is that the precise forecast for the climatic variables is very important for a country like Bangladesh where its economy is largely dependent on the agricultural production. And there is no denying fact that the agricultural production is immensely dependent on the climatic factors such as temperature, rainfall etc. From cultivation to harvest, the agro products are dependent on the climatic conditions, so, it is of paramount importance that to provide precise forecast to the farmers.

2. Previous Studies

One such study is conducted in China by (Zhang and Dong, 2020) where they also have used the similar approach to forecast the daily temperature data. They have used past temperature data map series as input data and build a CRNN model with convolutional layers and LSTM payers. With showed that their developed CRNN model forecasted better than other benchmark models. In another study by (Chu, et. al., 2018) also used the RCNN model to predict the temperature data using image data as the training data. They also have used two steps modelling approach where in the first step they extracted the visual features by the CNN model and then in the second step they imparted the RNN model to make the forecast using the features extracted in the previous step. They also have found that the deep learning model outperform compared to the state-of-the-art models.

3. Data and Data Sources

The monthly temperature data for Bangladesh from 1901 to 2015 has been collected from the secondary source. For this study, the average temperature has been used.

4. Methodology

In order to forecast the sequential data, there are different methods available. Some of the methods are following traditional time dependency in the data and build the model based on some distributional assumptions. However, with the advent of modern techniques now a day, the forecast for the time series data is often made based on the model that study the past behaviour of the data and learn from the data to make forecast. One such method is recurrent neural network (RNN) which has long short-term memory structure to study the time dependency of the data. In this study, the forecast for monthly temperature data is made by a recurrent neural network (RNN). The result of the two types of models will be compared to see if there is a significant difference in forecast accuracy for the sequential arrays.

4.1 Time Domain Approach

As time domain method, this study has incorporated the widely used Autoregressive Integrated Moving Average (ARIMA) model. This method can decompose the time dependent data into several components, namely, trend, seasonality, and randomness and by extracting the features from these components this creates a model that is used to estimate the trend of the sequence and to make forecast in the short term. This model has three parameters, namely, auto regressive part, moving average part and this adopt integration for the purpose of dealing with abrupt changes in the trend of the data. Moreover, it is also a common situation that the time series data often comes with seasonal variation in it. So, in order to address this seasonality, this model also used some parameters in its formulation. Therefore, the model can be specified as ARIMA(p, d, q)[P, D, Q] where, (p, d, q) are the non-seasonal parameters specifying, autoregression (p), integration (d), and moving average (q). On the other hand, [P, D, Q] are representing, seasonal autoregression (P), seasonal integration (D), and seasonal Moving average (Q). The mathematical model is as follows:

$$W_t = \varphi_1 W_{t-1} + \varphi_2 W_{t-2} + \dots\dots\dots\dots + \varphi_p W_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots\dots\dots\dots - \theta_q e_{t-q}$$

4.2 Deep Learning Method

The deep learning method that is used in this study is the Recurrent Neural Network (RNN) which has a Long Short-Term Memory (LSTM) network in the recurrent layers. A LSTM model is a powerful type of recurrent neural network, so, the application of this deep learning method might give some idea to compare the performances of two different types of methodologies.

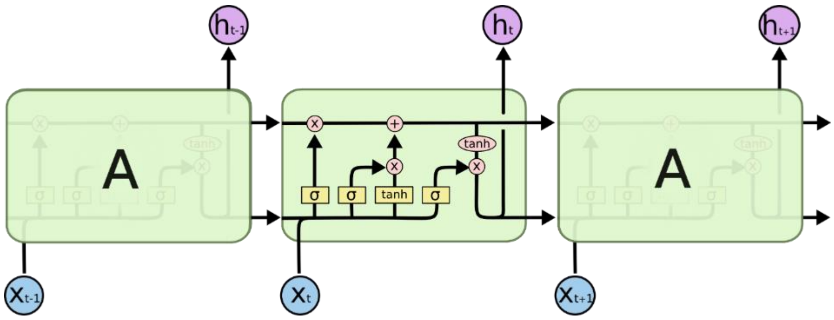

Figure 1: Long Short-Term Memory (LSTM) model Architecture

Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

In the LSTM model, the tensor format has the predictors (x) in a 3D array where the first one is the sample which represents the length of the values, the second one is the number of time steps (lags) and the third one is the number of predictors. The outcome variable is a 2D arrays where the first dimension is the length of the values and the second one is the number of time steps (lags). This model also requires having a training and test data which are evenly divisible. Moreover, the batch size is also need to be evenly divisible into both training and test data.

5. Results

5.1 Time Series Decomposition

The following chart tells about the different component of the time series data. The topmost graph is representing the temperature with respect to time and we if look at this chart we don’t see much information from there. However, the next three graphs are the decomposed information from the time series data. The trend is showing some kind of increasing pattern of temperature over the period of time. On the other hand the seasonal decomposition is showing the strong presence of seasonality in the data. And finally, the random noise graph is showing that the data has the presence of noise over the period of time. So, this time series decomposition plot gives idea about the necessary things to do for getting a better forecast. One things is to study the past behaviour of the data well so that the trend can be identified well, then the seasonality needs to be addressed as well and finally, the most important thing is to study the random noise term well.

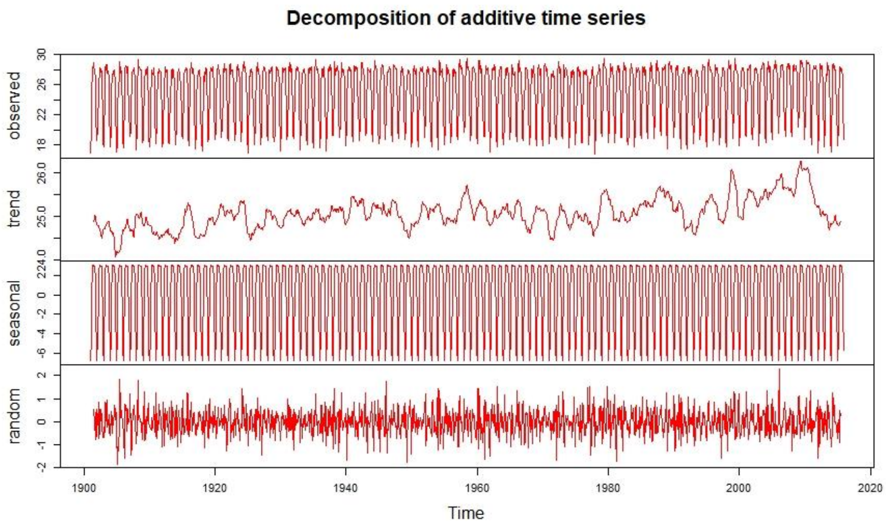

Figure 2: Time series decomposition plot

So, in order to study the time series data well which otherwise means to incorporate the components of the time series data, the conventional techniques are making some assumptions on the random error terms and estimating the model. Before estimating the model, it is important to see the lag step of the time series data. An autocorrelation plot can show whether a time series data has any lag dependency or not.

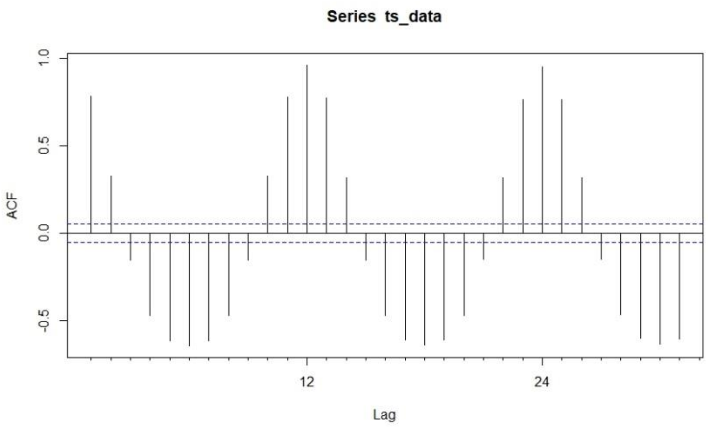

Figure 3: Autocorrelation plot for the time series data

From this auto correlation plot, we can see that out time series data has autocorrelation and the seasonal pattern of almost 12 lag steps. This information is vital for modelling the time series data. Now, incorporating all the input that we have got from the time series decomposition and the autocorrelation, we have fitted an Seasonal ARIMA model such as ARIMA(1, 0, 2) (2, 1, 0)[12]. This indicates the model has lag span of 12 months and it has the seasonal and non-seasonal parameters with specified order. In the following table the output of the model estimation is shown:

5.2 Model estimation:

After trying different model (Appendix Table 1), the following model is selected as the candidate model for forecasting the temperature data.

Table 1: Estimates for the fitted ARIMA model parameters

Parameters	**Coef.**	**Std. Error**	**Z value**	**P-value**
AR(1)	0.8757	0.0407	21.49	<0.001
MA(1)	-0.6803	0.0492	-13.81	<0.001
MA(2)	-0.091	0.0288	-3.15	0.002
SAR(1)	-0.666	0.0255	-26.08	<0.001
SAR(2)	-0.365	0.0253	-1439	<0.001
RMSE		**0.682**		
AIC		**2864.04**		
BIC		**2895.37**		

This model gives the RMSE of 0.682 to forecast the monthly temperature data.

5.3 Forecast based on the fitted model

This is the 100 steps ahead forecast based on the fitted ARIMA model.

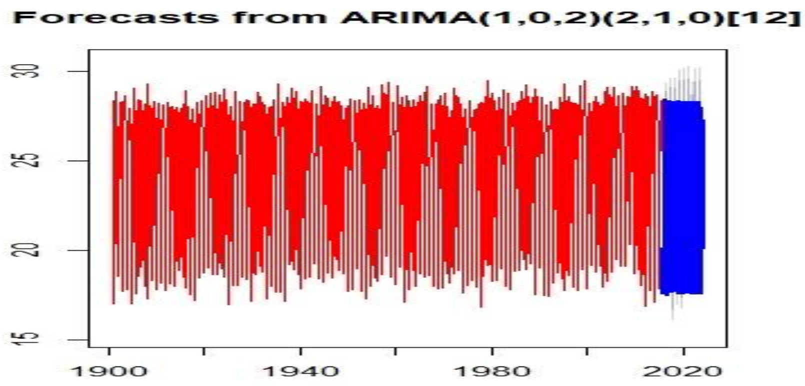

Figure 4: Forecast for the ARIMA time series model

Now, it is of our interest to see, if we build a model using the deep learning method and calculate the accuracy of the estimate, does it work significantly better or not that the ARIMA model or not.

5.4 Deep Learning Model

In order to apply the RNN model with LSTM, a model is set up with lag 12, batch size of 10, time steps of 1 with the adam optimizer and mean squared error loss function. Sufficient number of epochs are used to update the model parameters. After fitting the model and running the model with the time series sequential data, the final model is obtained and forecast is made. The following chart is showing the forecast of the time series data using RNN based LSTM model.

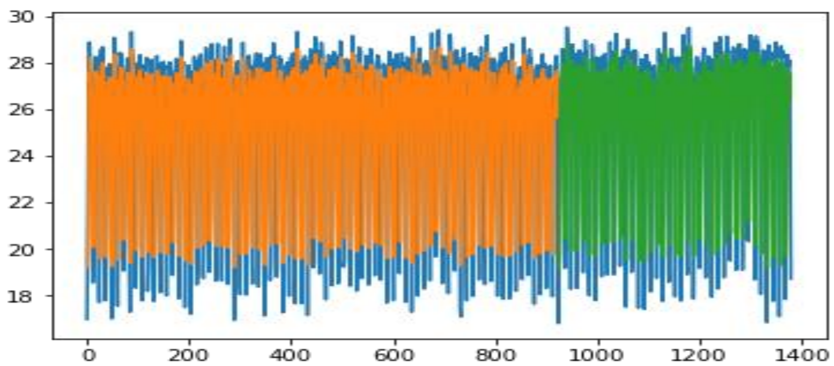

Figure 5: Time series data forecast using RNN based LSTM model

The RMSE for the estimate model is obtained for the training and the test data. These are as follows:

Indices	RMSE
Training	2.22
Test	2.26

So, apparently based on the RMSE comparison, it seems that the assumptions-based time series model has given better results that the deep learning model. However, a further exploration has been done for the deep learning method using the concept of backtesting.

5.5 Backtesting LSTM Model

Here we have divided the data into sub samples and fit an RNN based LSTM model for each of the samples. The division of the sample is made based on the pattern of the data, so, it is expected that these subsamples are more homogenous in terms of model fitting. We have then calculated the RMSE for each of the sub samples and take the average of the RMSE for the subsamples. After doing the adjustment for the whole sample we have made the forecast for the dataset. The RMSE of the adjusted full model for the backtested LSTM is 0.483 which is lower than the previous two estimates of stateful LSTM and the time domain ARIMA model. So, it is seen that if we can fit the deep learning model in a proper way then this can still outperform any existing method.

6. Conclusion

This small scale study is planned to compare the forecasting accuracy of two methodological approaches, namely, the conventional time series ARIMA model and the recurrent neural network based long short-term memory model. The results showed that the ARIMA model performs well and also outperform the RNN based LSTM model when the RNN model was build for the whole sample at a time. However, with a backtested subsampling approach it is found that the RNN model outperform the classical ARIMA model in terms of forecasting accuracy. So, this indicate the proper application of the deep learning method works better for the relatively small size sample and also compare to the assumptions based conventional methods.

Appendix Table: All fitted ARIMA models and the best candidate model for forecast.

```
Fitting models using approximations to speed things up...

 ARIMA(2,0,2)(1,1,1)[12] with drift         : Inf
 ARIMA(0,0,0)(0,1,0)[12] with drift         : 3464
 ARIMA(1,0,0)(1,1,0)[12] with drift         : 3064
 ARIMA(0,0,1)(0,1,1)[12] with drift         : Inf
 ARIMA(0,0,0)(0,1,0)[12]                    : 3462
 ARIMA(1,0,0)(0,1,0)[12] with drift         : 3420
```

```
ARIMA(1,0,0)(2,1,0)[12] with drift         : 2882
ARIMA(1,0,0)(2,1,1)[12] with drift         : Inf
ARIMA(1,0,0)(1,1,1)[12] with drift         : Inf
ARIMA(0,0,0)(2,1,0)[12] with drift         : 2948
ARIMA(2,0,0)(2,1,0)[12] with drift         : 2880
ARIMA(2,0,0)(1,1,0)[12] with drift         : 3063
ARIMA(2,0,0)(2,1,1)[12] with drift         : Inf
ARIMA(2,0,0)(1,1,1)[12] with drift         : Inf
ARIMA(3,0,0)(2,1,0)[12] with drift         : 2882
ARIMA(2,0,1)(2,1,0)[12] with drift         : 2869
ARIMA(2,0,1)(1,1,0)[12] with drift         : 3058
ARIMA(2,0,1)(2,1,1)[12] with drift         : Inf
ARIMA(2,0,1)(1,1,1)[12] with drift         : Inf
ARIMA(1,0,1)(2,1,0)[12] with drift         : 2876
ARIMA(3,0,1)(2,1,0)[12] with drift         : 2873
ARIMA(2,0,2)(2,1,0)[12] with drift         : 2871
ARIMA(1,0,2)(2,1,0)[12] with drift         : 2868
ARIMA(1,0,2)(1,1,0)[12] with drift         : 3056
ARIMA(1,0,2)(2,1,1)[12] with drift         : Inf
ARIMA(1,0,2)(1,1,1)[12] with drift         : Inf
ARIMA(0,0,2)(2,1,0)[12] with drift         : 2880
ARIMA(1,0,3)(2,1,0)[12] with drift         : 2870
ARIMA(0,0,1)(2,1,0)[12] with drift         : 2890
ARIMA(0,0,3)(2,1,0)[12] with drift         : 2882
ARIMA(2,0,3)(2,1,0)[12] with drift         : 2872
ARIMA(1,0,2)(2,1,0)[12]                    : 2866
ARIMA(1,0,2)(1,1,0)[12]                    : 3054
ARIMA(1,0,2)(2,1,1)[12]                    : Inf
ARIMA(1,0,2)(1,1,1)[12]                    : Inf
ARIMA(0,0,2)(2,1,0)[12]                    : 2878
ARIMA(1,0,1)(2,1,0)[12]                    : 2874
ARIMA(2,0,2)(2,1,0)[12]                    : 2869
ARIMA(1,0,3)(2,1,0)[12]                    : 2868
ARIMA(0,0,1)(2,1,0)[12]                    : 2888
ARIMA(0,0,3)(2,1,0)[12]                    : 2880
ARIMA(2,0,1)(2,1,0)[12]                    : 2867
ARIMA(2,0,3)(2,1,0)[12]                    : 2870

Now re-fitting the best model(s) without approximations...

ARIMA(1,0,2)(2,1,0)[12]                    : 2864

Best model: ARIMA(1,0,2)(2,1,0)[12]
```

References

Chu, W. T., Ho, K. C., & Borji, A. (2018, March). Visual weather temperature prediction. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 234-241). IEEE.

Zhang, Z., & Dong, Y. (2020). Temperature forecasting via convolutional recurrent neural networks based on time-series data. Complexity, 2020.