Time Series Forecasting with LSTM
Machine Learning - UE18EC338

Project Report

Session : January-May 2021

Team Composition

Surya Dutta - PES1201800674
Hemanth R - PES1201802038

Guide

Prof. Vanamala HR
Asst. professor, Dept. of ECE
PES University

Abstract

Most machine learning models use as input features some observations (samples or examples) but there is no time dimension in the data. Time series adds an explicit order dependence between observations: a time dimension. This additional dimension is a structure that provides a source of additional information.A time series is a sequence of observations taken sequentially at a time.Time-series forecasting models are the models that are capable to predict future values based on previously observed values, and it is widely used for non-stationary data.It has many applications and is used in areas such as financial analysis, weather analysis, network data analysis, healthcare analysis etc. This project concentrates on using time series forecasting with an LSTM network to model and predict the number of air passengers per month of a certain airline.

Motivation

Time series analysis and forecasting has been long explored because of its applicability in various domains such as economic forecasting, sales forecasting, budgetary analysis, stock market analysis, yield projections, process and quality control, inventory studies, workload projections, utility studies, census analysis etc. Using RNN models with time series analysis can be used to accurately predict and model many machine learning problems which can have a great impact on real world scenarios.

Introduction

A time series is a series of data points ordered in time. Time series adds an explicit order dependence between observations: a time dimension. In a normal machine learning dataset, the dataset is a collection of observations that are treated equally when the future is being predicted. In time series the order of observations provides a source of additional information that should be analyzed and used in the prediction process. Time series can have one or more variables that change over time. If there is only one variable varying over time, we call it univariate time series. If there is more than one variable it is called multivariate time series**.**

Time series analysis extracts meaningful statistics and other characteristics of the dataset in order to understand it. Time series analysis can help to make better predictions. Time series forecasting involves taking models fit on historical data (the training set) and using them to predict future observations (the test set). At the first step past observations are collected and analyzed to develop a suitable mathematical model which captures the underlying data generating process for the series. In the second step the future events are predicted using the model. This approach is particularly useful when there is a lack of a satisfactory explanatory model.Making predictions about the future is called extrapolation in the classical statistical handling of time series data. More modern fields focus on the topic and refer to it as time series forecasting. The skill of a time series forecasting model is determined by its future prediction performance. Time series forecasting has important applications in various fields.Over the past several decades many efforts have been made by researchers for the development and improvement of suitable time series forecasting models.

While more traditional methods such as centred average algorithms, moving average models such as ARIMA, and SVMs are helpful when trying to forecast future time series, find anomalies, and classify data, they tend to rely on more recent data, or focus on only data points that aren’t outliers (as in SVMs).

More recently, the deep learning model called the Long Short-Term Memory model or LSTM, has been used in order to take advantage of its ability to recall older information in the data to create a more accurate forecast or recognize anomalies that might have gone unnoticed by more traditional methods. STMs are a variation of a Recurrent Neural Network or RNN which is a great deal more complex than one of the more simple deep learning models, feedforward networks. Unlike a feedforward neural net which does classification or prediction based on a single forward pass of the input through the nodes in the network, an RNN relies on two inputs. One input is the new data the network hasn't seen before, and the other input is the information retained from previous data that has already passed through the network. In other words, RNNs consume their own output as input and use this maintained awareness to inform their perception of new data. This is great for time series data since many times patterns are formed from previous lags that the neural net can learn from, due to the persistence of recently learned information. However, when it comes to older information learned, say 5 lags ago, then RNNs aren't as powerful since they don't retain any long term dependencies. This is crucial for important tasks such as detecting network attacks or fraudulent activities for banks. Luckily, LSTMs recall older information from previous passes in the network.

Thus, LSTM models are robust and have mutifold applications, and they can also be used with multivariate time series data.

Problem Statement

To model and predict the number of air passengers per month of a certain airline using time series forecasting with an LSTM network.

Methodology/Working Principle

1) All the necessary packages required are downloaded. A random number generator is used and it is seeded (using numpy) to make the code reproducible.

2) The dataset is loaded on to the model and the dataset is normalised. The dataset is split into testing and training data.

3) The dataset also undergoes some pre-processing with the help of a helper function where the array of values are converted into a dataset matrix. The input is reshaped to be [samples ,time steps, features]. As all the tasks are completed concerning data preparation to fit into the LSTM model, the next step involves creation of the LSTM network.

4) Our LSTM network consists of 4 LSTM layers and 1 dense layer. Since each layer has only 1 input tensor and 1 output tensor, tf.keras.sequential() class is used. The LSTM layers are easily added and customized using the built-in function in the keras RNN API.

5) The loss function used is the standard mean square error function and it returns one scalar value per unit sample. Thus the model is configured with losses and metrics with model.compile() and the model is trained with the model.fit() functions.

6) Predictions are made and the root mean square error is calculated for both the training and testing data.

7) Finally, the predictions are plotted using the matplotlib package. The plot includes the predicted plot for the training data, the predicted plot for the testing data, and the plot of the original dataset.

8) The number of epochs are set to 100 and it can also be observed that the loss I.e. the mean square error reduces during each epoch and thus, the training process converges to an optimal model.

Software and Dataset Details

Software Used –

Colab – Python development environment that runs in the browser using google cloud.

Keras and Scikit-learn – Deep learning frameworks based on python.

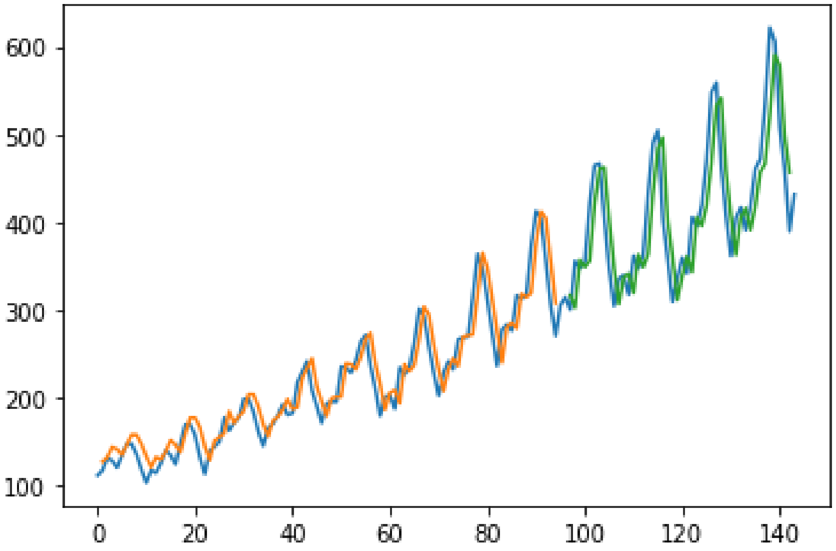
Python libraries such as pandas, matplotlib and numpy.

Dataset Used –

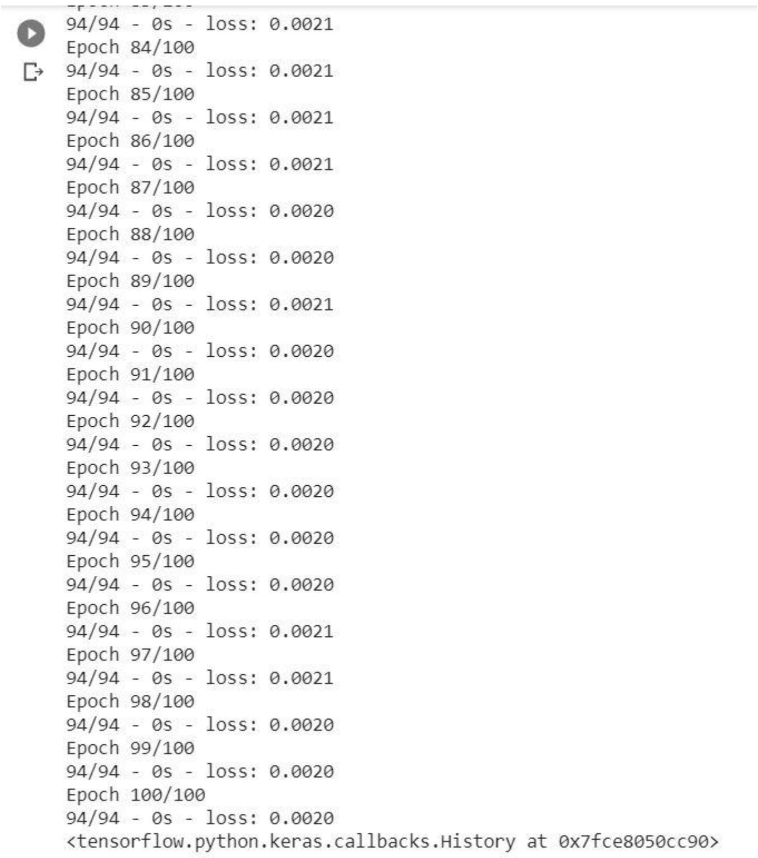
Air passengers dataset from Kaggle (international community for ML and data science) which gives the number of air passengers per month of a certain airline as a csv file.

Results and Analysis

The predicted plot for the training data, predicted plot for the test data, and the original dataset are plotted in the same graph. It can be observed that the prediction for the test data almost follows the same trend as that of the original dataset. This implies that the model is trained to reach or converge to the optimal solution.



A screenshot of the mean square error shows that the error decreases during every epoch and thus the error reaches minimum value at the end of the training/prediction process. If the number of epochs are increased, the error further reduces.



```
Epoch 83/100
94/94 - 0s - loss: 0.0021
Epoch 84/100
94/94 - 0s - loss: 0.0021
Epoch 85/100
94/94 - 0s - loss: 0.0021
Epoch 86/100
94/94 - 0s - loss: 0.0021
Epoch 87/100
94/94 - 0s - loss: 0.0020
Epoch 88/100
94/94 - 0s - loss: 0.0020
Epoch 89/100
94/94 - 0s - loss: 0.0021
Epoch 90/100
94/94 - 0s - loss: 0.0020
Epoch 91/100
94/94 - 0s - loss: 0.0020
Epoch 92/100
94/94 - 0s - loss: 0.0020
Epoch 93/100
94/94 - 0s - loss: 0.0020
Epoch 94/100
94/94 - 0s - loss: 0.0020
Epoch 95/100
94/94 - 0s - loss: 0.0020
Epoch 96/100
94/94 - 0s - loss: 0.0021
Epoch 97/100
94/94 - 0s - loss: 0.0021
Epoch 98/100
94/94 - 0s - loss: 0.0020
Epoch 99/100
94/94 - 0s - loss: 0.0020
Epoch 100/100
94/94 - 0s - loss: 0.0020
<tensorflow.python.keras.callbacks.History at 0x7fce8050cc90>
```

Conclusions and Future Scope

Our LSTM network was able to model and predict the number of air passengers per month of a certain airline in an exemplary manner. The model converges to the optimal solution when the number of epochs is high, and the error between the original and predicted values decrease.

Major percentages of ML problems are based on non-time series data such as numerical data, text data (NLP) , image data and speech data. Thus, for non-stationary data, time series analysis is the de facto method since time plays a crucial role in it. Time series forecasting is used in a wide area of applications and is very important because there are so many prediction problems which involve a time component. Time series analysis and forecasting is used in financial analysis, weather analysis, network data analysis, healthcare analysis etc.

Time series analysis and forecasting include many robust architectures and models like prophet model, autoregressive moving average (ARIMA) models etc with different error metrics along with the LSTM model and these models are being used in many applications too. We believe the efficiency of prediction could be increased if such models are used for our project. Applications of LSTM include robot control, speech recognition, rhythm/grammar learning, protein homology detection, anomaly detection, semantic parsing, drug design, market prediction etc.

References

https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

https://www.tutorialspoint.com/time_series/index.htm

https://en.wikipedia.org/wiki/Long_short-term_memory#Applications

https://towardsdatascience.com/lstm-time-series-forecasting-predicting-stock-prices-using-an-lstm-model-6223e9644a2f

https://www.tensorflow.org/tutorials/structured_data/time_series

https://www.analyticsvidhya.com/blog/2020/10/multivariate-multi-step-time-series-forecasting-using-stacked-lstm-sequence-to-sequence-autoencoder-in-tensorflow-2-0-keras/

https://medium.com/coders-camp/10-machine-learning-projects-on-time-series-forecasting-ee0368420ccd

https://www.kaggle.com/rakannimer/air-passengers

Github repository - https://github.com/surya-dutta/Time-series-forecasting-LSTM