Stock Prediction Using Time Series Data

Anup Shakya
ashakya@memphis.edu

Bikram Karki
bkarki@memphis.edu

1. Introduction

The stock market is known as a place where people can make a fortune if they can crack the mantra to successfully predict stock prices. As a stock broker or an investor we would want reasonably decide when to buy stocks and when to sell them to make a huge profit, Though it's impossible to predict a stock price correctly most of the time, the question arises if we can estimate and consider all the factors to predict a movement or a future value of a stock. Machine learning has found its applications in many interesting fields over these years. Taming the stock market is one of them.

In this project we will predict stock price using a time-series model known as Long Short-Term Memory (LSTM). LSTM models are powerful especially for retaining a long-term memory. It is an artificial recurrent neural network architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections [2]. It cannot only process single data points such as images but also entire sequences of data. This characteristic is extremely useful when we deal with Time-Series or Sequential Data. When using an LSTM model, we are free and able to decide what information will be stored and what discarded.

A common LSTM architecture is composed of a cell and three regulators usually called gates, of the flow of information inside the LSTM unit: an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. The cell is responsible for keeping track of the dependencies between the elements in the input sequence. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. There are connections into and out of the LSTM gates, a few of which are recurrent. The weights of these connections, which need to be learned during training, determine how the gate operates.

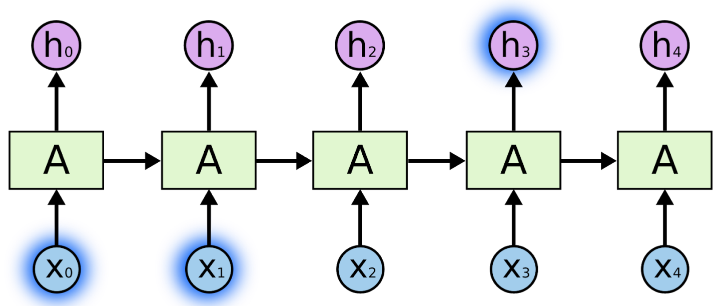

Fig 1. LSTM architecture

2. Dataset Collection and Processing

Stock data is regarded as one of the most important assets in today's commercial world. Analysts analyze stock data to the precision of every single second Stock data is however freely available on the internet. There are many APIs that provide stock data for free. One such online API is [Alpha Vantage](). With this API, we can simply need an API key and make a get request with required parameters. The output is a well-formatted JSON data. [3] Following are the rules for making the parameterized query API.

2.1. API Parameters and Query

- **function**: required
  The time series of your choice. For ex: function = TIME_SERIES_INTRADAY, function= TIME_SERIES_DAILY
- **symbol**: required
  The name of the equity (ticker name) of your choice. For example: symbol=IBM, symbol=TSLA
- **interval**: required
  This indicates the time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min

Example for a GET query for IBM stocks:

https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=IBM&apikey=demo

2.2. Dataset Management

The API call to the above GET URL returns JSON data. The next task is to manage this data into something that is usable. For this, we parse the JSON data and generate a CSV file to store and reuse this data. Once we create the CSV data, we use functionality provided by the library pandas to manipulate this data. We then visualize the data imported into the project. The data contains daily prices for a certain company on a certain date. For this project, we decided to choose Apple Inc. for our experiment purposes. The data contains four fields of our interest: Date, High, Low and Close. High represents the highest stock price on a given date. Similarly, Low represents the lowest stock price on that day. Close represents the closing stock price on that day. Fig 2 shows the visualization of the imported data.

Then, the next step is to preprocess the data to generate training and test data instances. For pre-processing, we used a MinMaxScaler with feature value from 0 to 1 for normalizing the price instances. Now, the next step is to generate the input and output instances for these normalized price data. For this, we used a custom window with size WINDOW_SIZE to select the x-instances or the input instances. If we suppose WINDOW_SIZE=60, then we take the first 60 price data in the series as input and the 61st price data as the output. Then, we move the window one step and take 2nd to 61st price data as input and the 62nd price data as output and so on. We generate the training data instances with this procedure on the normalized data. For this project, we split 80% of the data as training data and 20% as test data. The training data is then fit to our learning model which follows.

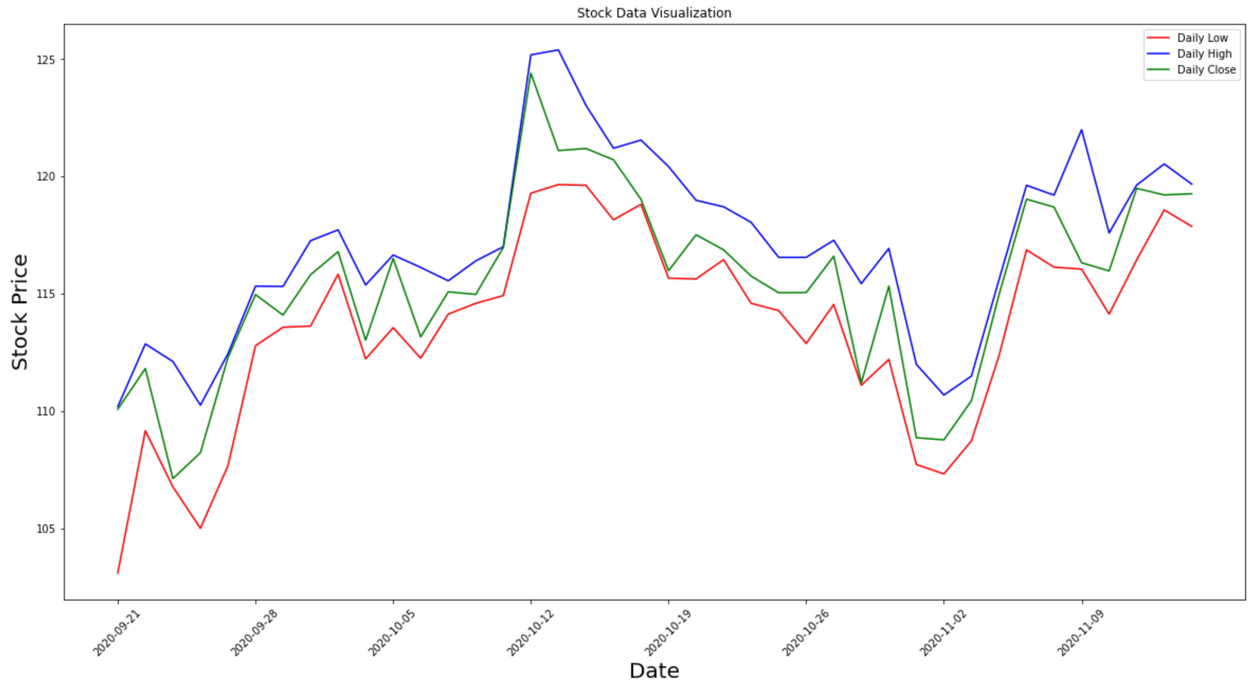

Fig 2. Plot for Stock Price Vs Date presenting Daily High, Low and Closing stock prices

3. The Learning Model

Our model consists of two LSTM layers with 100 neurons, a dense layer with 50 neurons and a final dense layer with one neuron as shown in the figure below. The model is compiled with Mean Square Error(MSE) error and Adam optimiser. For optimal training of the dataset we set the batch size as 1 and a varied number of epochs to get the best result.

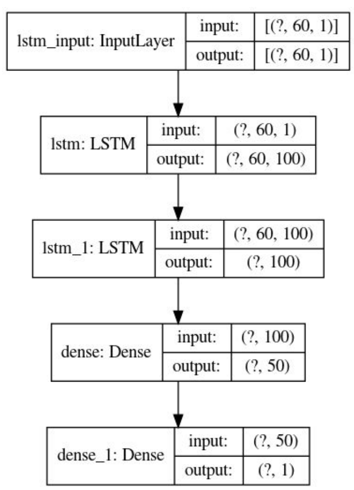

Fig 3. The Learning Model

4. Results

The predicted value of stock along with the correct stock price for Apple Inc. is shown in the diagram below. The root mean square error achieved for the predicted values is 16.38.

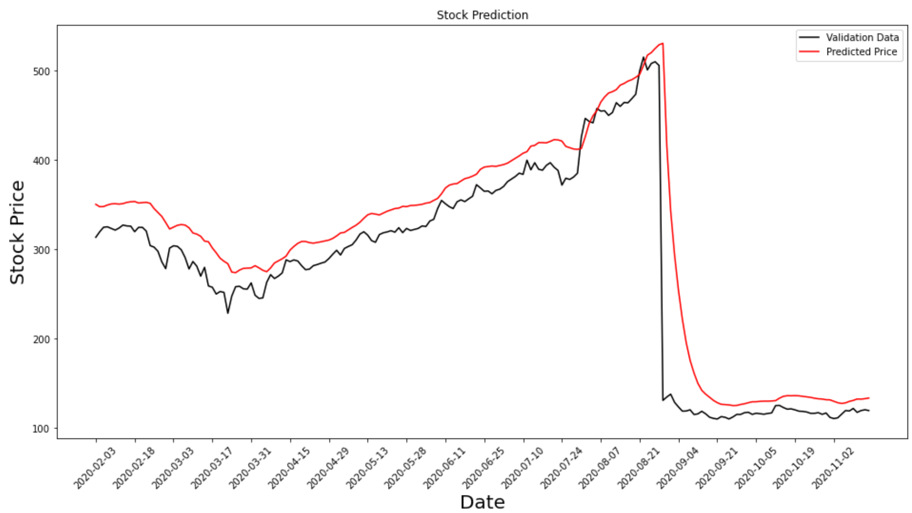

Fig 4. Predicted stock price wrt. validation price

We evaluated the performance of our model with respect to the number of epochs and loss.

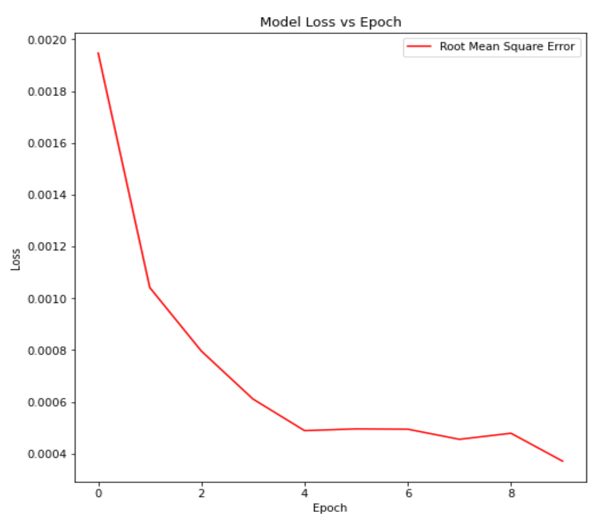

Fig 5. Loss vs Number of epochs

5. Conclusion

From this experiment, we can conclude that recurrent neural networks are very useful and perform with great accuracy in predicting time series data. LSTM networks specifically worked very well in this case, because LSTMs are very powerful in learning long term dependencies and co-relations in the observed data. Even though stock prices are very unpredictable, there are still some hidden relationships in the data. With proper tweak to the LSTM network, we can make the model accurately mimic the actual hidden relations in the data. We tried predicting stock prices for Apple Inc. for future dates. We found that the predictions almost represented linear growth. This might have been a result of over-fitting in the training dataset. Further works for this project might be addressing the overfitting issue by implementing a dropout layer.

6. References

[1] Adil Moghar and Mhamed Hamiche (2020). Stock Market Prediction Using LSTM Recurrent Neural Network. Procedia Computer Science 2020.

[2] Wikipedia, "Long Short Term Memory", https://en.wikipedia.org/wiki/Long_short-term_memory

[3] Alpha Vantage, "Aplha Vantage API Documentation", https://www.alphavantage.co/documentation/