Project Report

Title: Stock Price Detection Using LSTM

Pattern Recognition Lab Project

<u>Group: PR06</u>

Name: Abue Ammar ID: 011162134

Name: Md. Shihab Hossain ID: 011173066

Name: Md. Imtiaz Hossain Shovon ID: 011143129

Name: Alif Nomani Rahman ID: 011142107

Section : B

Course Code: CSI 416

Problem Definition:

Prediction of stock price and stock price movement patterns has always been a critical area of research. While the well-known efficient market hypothesis rules out any possibility of accurate prediction of stock prices, there are formal propositions in the literature demonstrating accurate modeling of the predictive systems can enable us to predict stock prices with a very high level of accuracy. In this project, we present a suite of deep learning-based regression models that yields a very high level of accuracy in stock price prediction.

Dataset Description:

Dataset name: NSE-TATA

- No. of instances=1236
- No. of attributes=8 (Date, Open, High, Low, Last, Close, Total trade, Turnover)
- No. of null value = 0
- No. of missing value = 0
- Normalization was required
- For training 987 instances
- For testing 249 instances

Approach:

1. Read the dataset.

2. Analyze the closing prices from dataframe.

3. Sort the dataset on date time and filter "Date" and "Close" columns.

4. In data preprocessing we have normalized to the new filtered dataset. We used minmaxscaler to scale data with range 0 to 1.

5. Build sequential LSTM model with 20% layer dropate and 50 units in each layer.

6. Then training the LSTM model. Mean squared error were calculated for loss and Adam optimizer was used.

7. Take a sample of a dataset to make stock price predictions using the LSTM model. Then used inverse transformation to get predicted cost.

8. Then saving the model.

9. Visualize the predicted stock costs with actual stock costs

using matplotlib and also print the predicted cost and actual cost side by side.

Problems Faced:

1. LSTMs solve two technical problems:
    a. disappearing gradients
    b. exploding gradients

both of which are connected to how the network is taught.

Result/Comparison:

Figure: Predicted cost vs Actual cost

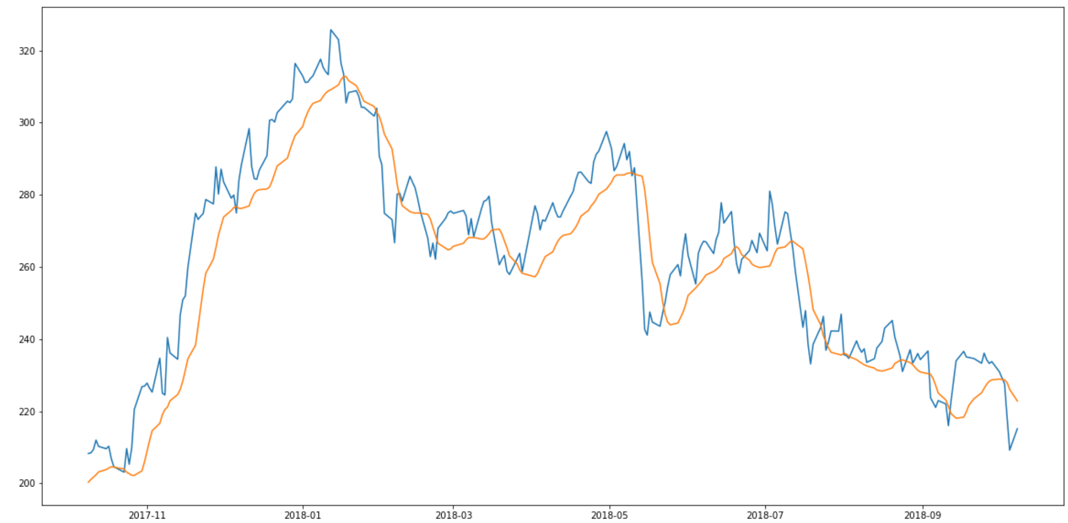

Figure: the remaining valid data only

Conclusion:

Learning rates, input and output biases, and learning rates are only a few of the parameters provided by LSTMs. As a result, no precise modifications are required. With LSTMs, the difficulty of updating each weight is decreased to O(1), comparable to Backpropagation Through Time (BPTT), which is a benefit. Given time delays of uncertain duration, LSTM is well-suited to identify, analyze, and forecast time series. So, with a timestep-based dataset, LSTM performs better. The minmax scaler works with continuous data, as we discovered. Adam optimizer uses hill-climbing and the gradient descent technique, among other things. These are the things that we have learned.