Predictive Analytics on Stock Market Data using Machine Learning Algorithms.

We are using **S&P 500 index** companies historical prices with fundamental data of NYSE to apply predictive analytics.The dataset contains the following files:

Dataset consists of following files:

- **prices.csv**: raw, as-is daily prices. Most of data spans from 2010 to the end 2016, for companies new on stock market date range is shorter. There have been approx. 140 stock splits in that time, this set doesn't account for that.
- **prices-split-adjusted.csv**: same as prices, but there have been added adjustments for splits.
- **securities.csv**: general description of each company with division on sectors
- **fundamentals.csv**: metrics extracted from annual SEC 10K fillings (2012-2016), should be enough to derive most of popular fundamental indicators.

We have a total number of different stocks and we have the variable names along with it

number of different stocks: 501

['VAR', 'CMCSA', 'TRIP', 'O', 'FAST', 'PRGO', 'APA', 'NFX', 'ADI', 'FSLR']

Now Stock Price and Stock volume is plotted against time to observe it's behavior according to time.

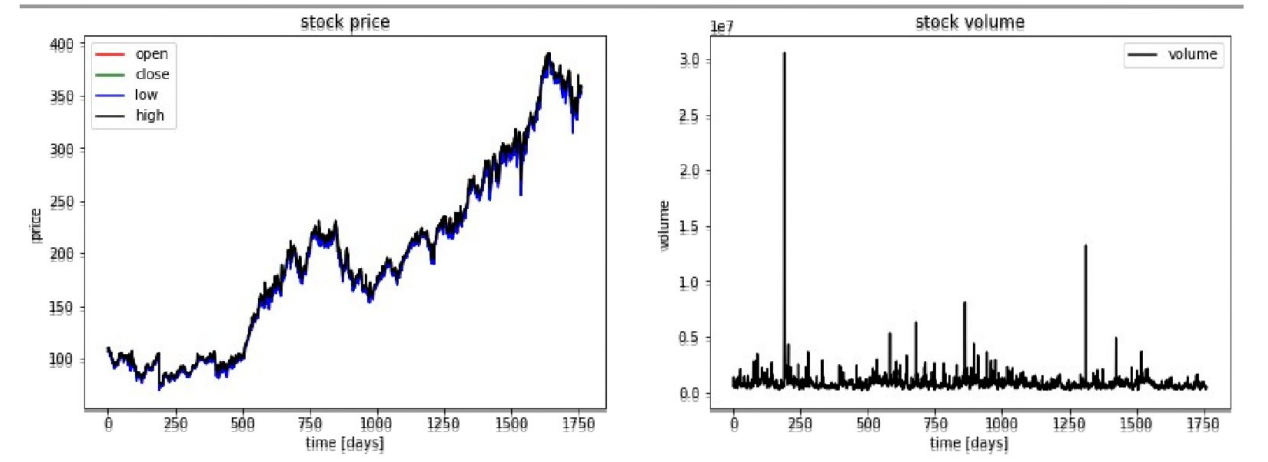


Fig. 14

We can see the variables **open,close,low and high sets an upward trend** with respect to time and they take nearly same values at a particular time point.

Stock volume data takes abnormally high value at particular data point making noise in the data while it keeps a grounded trend during the time period.

We need to drop the Stock volume feature in order to normalize the data that will eliminate data redundancy.

After normalizing our data the new datset structure we get is as follows:

df_stock.columns.values = ['open', 'close', 'low', 'high']

x_train.shape = (1394, 19, 4)

y_train.shape = (1394, 4)

x_valid.shape = (174, 19, 4)

y_valid.shape = (174, 4)

x_test.shape = (174, 19, 4)

y_test.shape = (174, 4)

After normalization we again plot the variables ‘open’,’close’,’low’,’high’ and we get a upward trend of these variables plotted against time with less gradiend descent,i.e the graph is now less steeper.

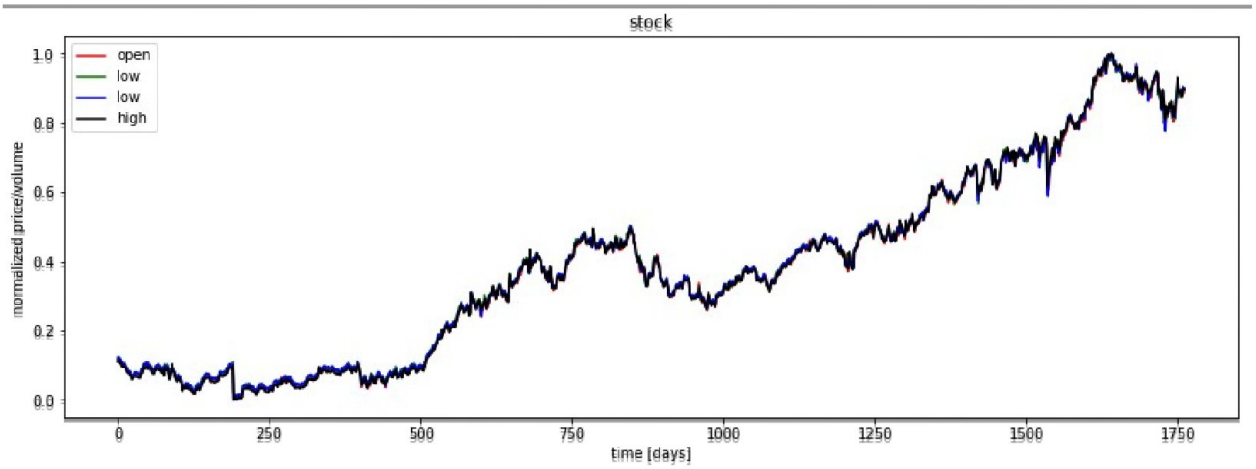
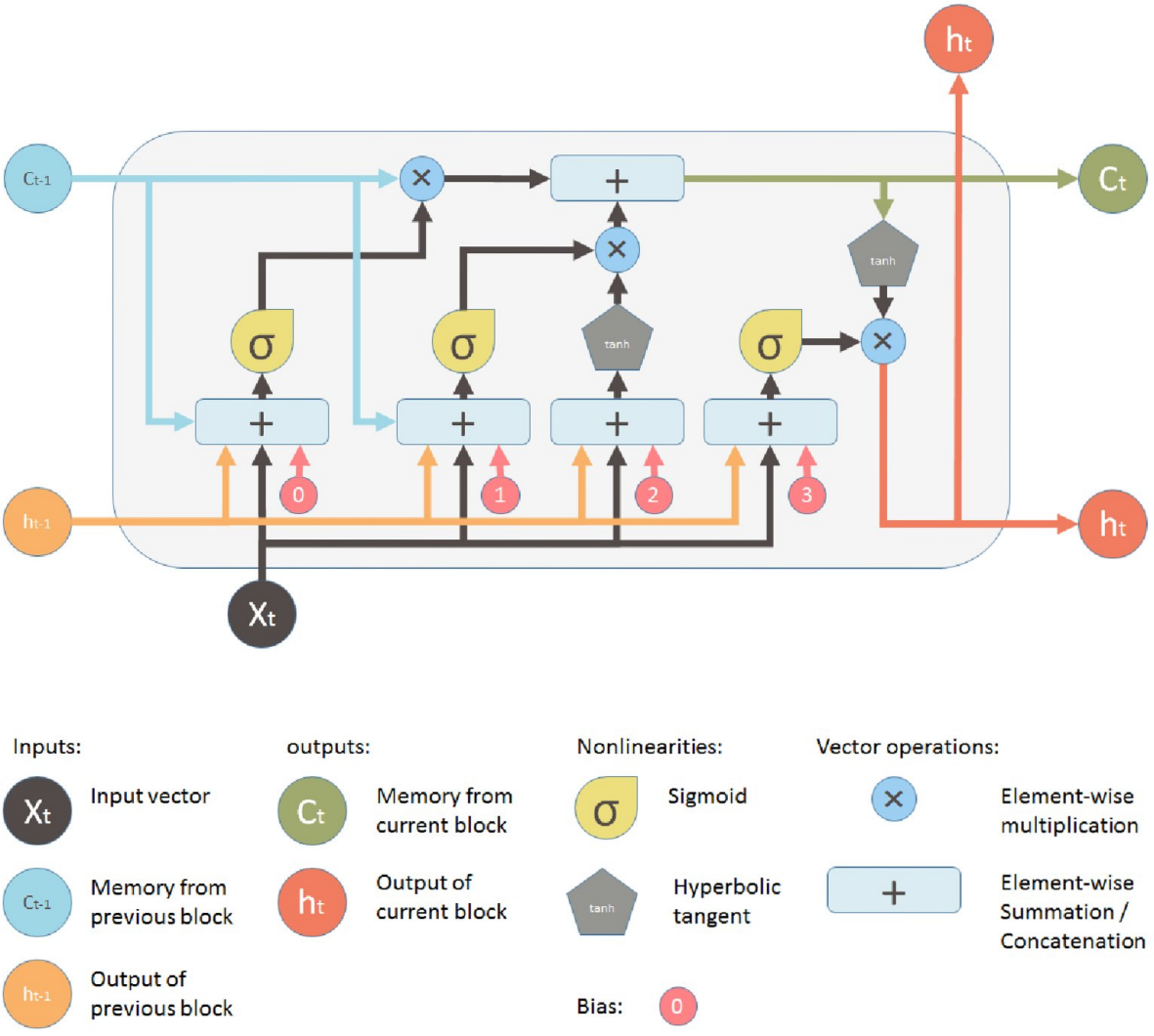


Fig. 15

Application of LSTM to perform predictive analytics

We will now split Train and Test dataset and apply LSTM to the train dataset to learn the prediction model the previous values of the variables.LSTM works on the datset as follows:It is the naive way to let neural network accept a time series data is connecting several neural networks together. Each of the neural networks handles one time step. Instead of feeding the data at each individual time step, LSTM provides data at all time steps within a window, or a context, to the neural network. The following diagram explains LSTM.



We are applying LSTM on training dataset and we get accuracy measure Mean Square Error with different epochs:

```
0.00 epochs: MSE train/valid = 0.113780/0.150386
4.99 epochs: MSE train/valid = 0.000265/0.001396
9.97 epochs: MSE train/valid = 0.000175/0.000878
14.96 epochs: MSE train/valid = 0.000154/0.000795
19.94 epochs: MSE train/valid = 0.000133/0.000673
24.93 epochs: MSE train/valid = 0.000123/0.000438
29.91 epochs: MSE train/valid = 0.000104/0.000417
34.90 epochs: MSE train/valid = 0.000100/0.000435
39.89 epochs: MSE train/valid = 0.000092/0.000322
44.87 epochs: MSE train/valid = 0.000106/0.000296
```

```
49.86 epochs: MSE train/valid = 0.000079/0.000271
54.84 epochs: MSE train/valid = 0.000099/0.000265
59.83 epochs: MSE train/valid = 0.000111/0.000580
64.81 epochs: MSE train/valid = 0.000075/0.000239
69.80 epochs: MSE train/valid = 0.000078/0.000315
74.78 epochs: MSE train/valid = 0.000074/0.000308
79.77 epochs: MSE train/valid = 0.000081/0.000371
84.76 epochs: MSE train/valid = 0.000068/0.000224
```

The train Mean square error and the Test mean square error are given.Both are approximately zero.The model seem to be a good predictor model.

```
Train Score: 0.00108 MSE (0.03 RMSE)
Test Score: 0.00812 MSE (0.09 RMSE)
```

The first figure shows the train variable and valid target plotted against time.The valid target dominates the end of time axis.Train target and train prediction can be seen to have almost same values over period of time.Test target and test prediction seen to have the same values at the end of time axis.We can take a different set of train data and test data set to validate the answer. The figure in the right side shows explicitly the behavior f test target data (including all variables).It is seen to have a seasonal fluctuation along with time but the test target and prediction curve converge almost everywhere making this model a good predictor model.

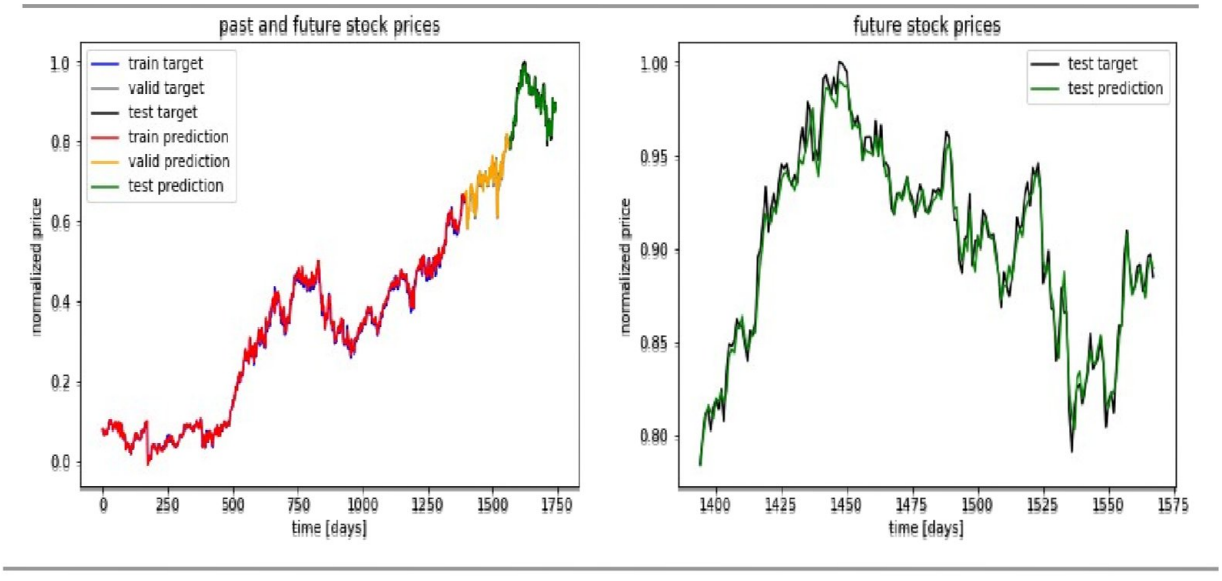


Fig. 16