

Stock Price Prediction

Team Members:

1.MANIKANDAN B

2.KARTHICK M

3. MATHAN S

4.SHIBIN G

5.PRASANNA KM

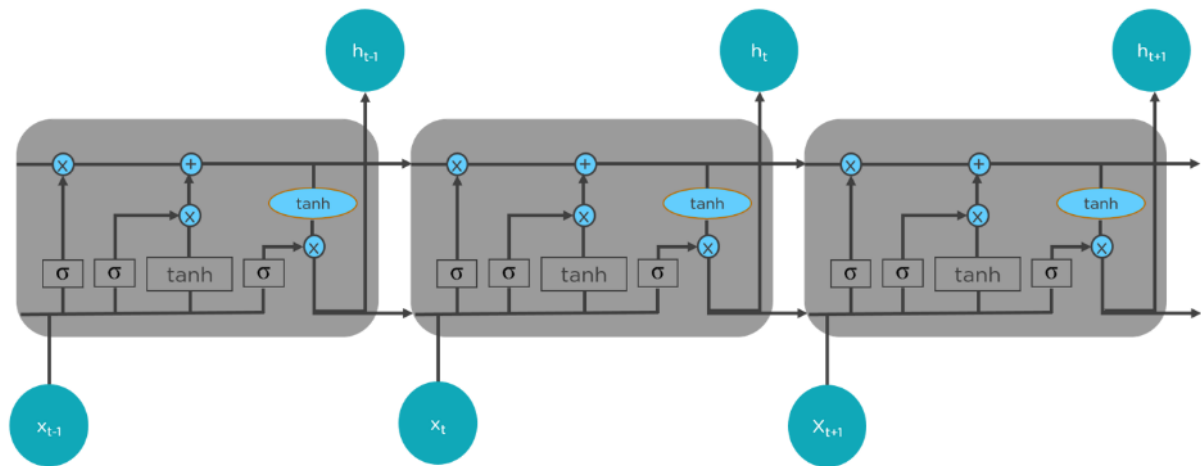
INTRODUCTION

Stock Price Prediction using machine learning helps you discover the future value of company stock and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits. Predicting how the stock market will perform is a hard task to do. There are other factors involved in the prediction, such as physical and psychological factors, rational and irrational behavior, and so on. All these factors combine to make share prices dynamic and volatile. This makes it very difficult to predict stock prices with high accuracy.

Understanding Long Short Term Memory Network

Here, you will use a Long Short Term Memory Network (LSTM) for building your model to predict the stock prices of Google.

LSTMs are a type of [Recurrent Neural Network](#) for learning long-term dependencies. It is commonly used for processing and predicting time-series data



From the image on the top, you can see LSTMs have a chain-like structure. General RNNs have a single [neural network](#) layer. LSTMs, on the other hand, have four interacting layers communicating extraordinarily.

LSTMs work in a three-step process.

- The first step in LSTM is to decide which information to be omitted from the cell in that particular time step. It is decided with the help of a sigmoid function. It looks at the previous state (h_{t-1}) and the current input x_t and computes the function.
- There are two functions in the second layer. The first is the sigmoid function, and the second is the tanh function. The sigmoid function decides which values to let through (0 or 1). The tanh function gives the weightage to the values passed, deciding their level of importance from -1 to 1.
- The third step is to decide what will be the final output. First, you need to run a sigmoid layer which determines what parts of the cell state make it to the output. Then, you must put the cell state through the tanh function to push the values between -1 and 1 and multiply it by the output of the sigmoid gate.

Google Stock Price Prediction Using LSTM

1.Import the Libraries.

```
#Import libraries
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

2. Load the Training Dataset.

The Google training data has information from 3 Jan 2012 to 30 Dec 2016. There are five columns. The Open column tells the price at which a stock started trading when the market opened on a particular day. The Close column refers to the price of an individual stock when the stock exchange closed the market for the day. The High column depicts the highest price at which a stock traded during a period. The Low column tells the lowest price of the period. Volume is the total amount of trading activity during a period of time.

```
dataset_train = pd.read_csv("Google_Stock_Price_Train.csv")
dataset_train.head()
```

	Date	Open	High	Low	Close	Volume
0	1/3/2012	325.25	332.83	324.97	663.59	7,380,500
1	1/4/2012	331.27	333.87	329.08	666.45	5,749,400
2	1/5/2012	329.83	330.75	326.89	657.21	6,590,300
3	1/6/2012	328.34	328.77	323.68	648.24	5,405,900
4	1/9/2012	322.04	322.29	309.46	620.76	11,688,800

3. Use the Open Stock Price Column to Train Your Model.

```
training_set = dataset_train.iloc[:,1:2].values  
  
print(training_set)  
print(training_set.shape)
```

```
[[325.25]  
 [331.27]  
 [329.83]  
 ...  
 [793.7 ]  
 [783.33]  
 [782.75]]  
(1258, 1)
```

4. Normalizing the Dataset.

```
from sklearn.preprocessing import MinMaxScaler  
  
scaler = MinMaxScaler(feature_range = (0,1))  
scaled_training_set = scaler.fit_transform(training_set)
```

```
scaled_training_set
```

```
array([[0.08581368],  
       [0.09701243],  
       [0.09433366],  
       ...,  
       [0.95725128],  
       [0.93796041],  
       [0.93688146]])
```

5. Creating X_train and y_train Data Structures.

```
X_train = []
y_train = []
for i in range(60,1258):
    X_train.append(scaled_training_set[i-60:i, 0])
    y_train.append(scaled_training_set[i, 0])
X_train = np.array(X_train)
y_train = np.array(y_train)
```

```
print(X_train.shape)
print(y_train.shape)
```

```
(1198, 60)
(1198,)
```

6. Extracting the Actual Stock Prices of Jan-2017.

```
dataset_test = pd.read_csv("Google_Stock_Price_Test.csv")
actual_stock_price = dataset_test.iloc[:,1:2].values
```

7. Preparing the Input for the Model.

```
dataset_total = pd.concat((dataset_train['Open'], dataset_test['Open']), axis = 0)
inputs = dataset_total[len(dataset_total)- len(dataset_test)-60:].values

inputs = inputs.reshape(-1,1)
inputs = scaler.transform(inputs)

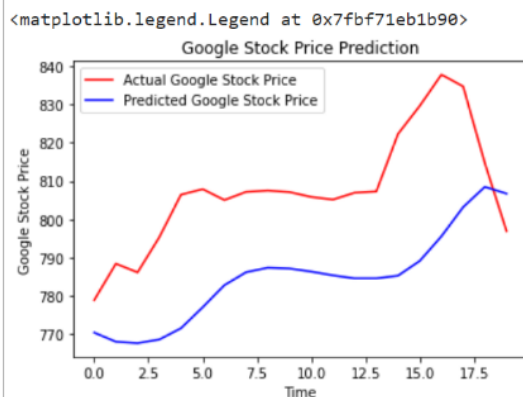
X_test = []
for i in range(60,80):
    X_test.append(inputs[i-60:i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test,(X_test.shape[0], X_test.shape[1], 1))
```

8. Predicting the Values for Jan 2017 Stock Prices.

```
predicted_stock_price = regressor.predict(X_test)
predicted_stock_price = scaler.inverse_transform(predicted_stock_price)
```

9. Plotting the Actual and Predicted Prices for Google Stocks.

```
plt.plot(actual_stock_price, color = 'red', label = 'Actual Google Stock Price')
plt.plot(predicted_stock_price, color = 'blue', label = 'Predicted Google Stock Price')
plt.title('Google Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Google Stock Price')
plt.legend()
```



As you can see above, the model can predict the trend of the actual stock prices very closely. The accuracy of the model can be enhanced by training with more data and increasing the LSTM layers.

Conclusion

The stock market plays a remarkable role in our daily lives. It is a significant factor in a country's GDP growth. In this tutorial, you learned the basics of the stock market and how to perform stock price prediction using machine learning.

Do you have any questions related to this tutorial on stock prediction using machine learning? In case you do, then please put them in the comments section. Our team of experts will help you answer your questions.