Stock Price Prediction Using Stacked LSTM

Name: Tanmoy Paul Date:1st Sept 2023

Objective:

- Training a stacked LSTM model for predicting stock price.
- Predicting Stock price for the next 30 days in the future.
- Experiment with different timesteps (input sequence length)=100, 150.

LSTM vs Stacked LSTM:

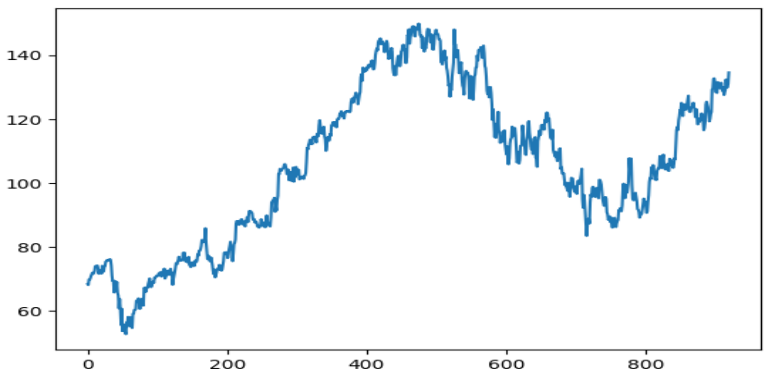
The original LSTM model is comprised of a single hidden LSTM layer followed by a standard feedforward output layer. The stacked LSTM is an extension to this model that has multiple hidden LSTM layers where each layer contains multiple memory cells.

Advantages of Stacked LSTM over Simple LSTM:

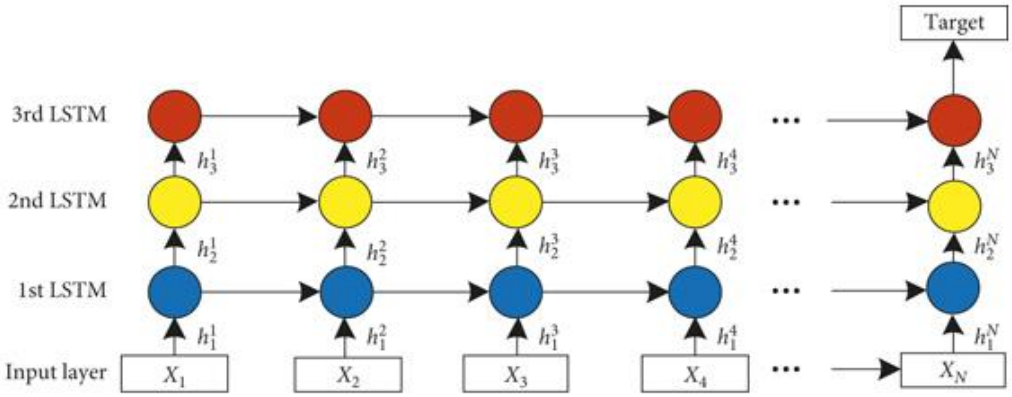
1. Increased Model Capacity: to capture complex patterns.
2. Hierarchical Feature Learning: The lower layers can capture short-term dependencies, while the higher layers can focus on capturing longer-term dependencies in the data.
3. Increased Expressiveness: Stacked LSTMs offer greater expressiveness in modeling sequential data. They can capture and retain information over longer sequences
4. Parallelism in Training: The LSTM units within a layer are processed sequentially at each time step, different layers can be trained in parallel, which can result in faster convergence during training.

Procedure:

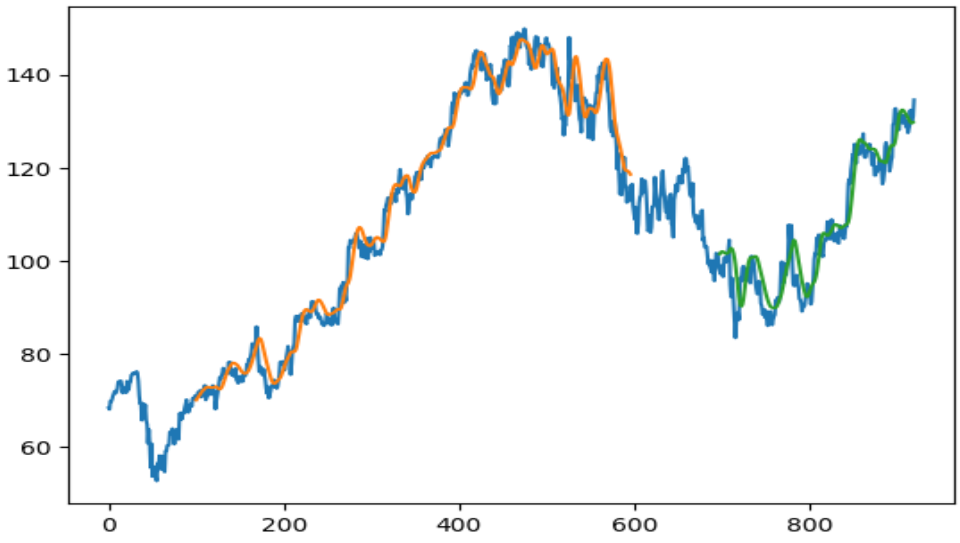
1. Collected the Stock data-GOOGL from 2020-01-01 to 2023-08-30 using yfinance package



2. Here we only deal with the close price of the stock. Using MinMaxScaler we scaled the closing price.
3. Splitted dataset into Train set and Test set into 65% -35% ratio without disturbing the sequential order.
4. Reshaped the Train and Test set into a sequential data set with time step = 100.
5. Reshaped input to be [samples, time steps, features] which is required for LSTM
6. Created the Stacked LSTM model and Trained the Model.



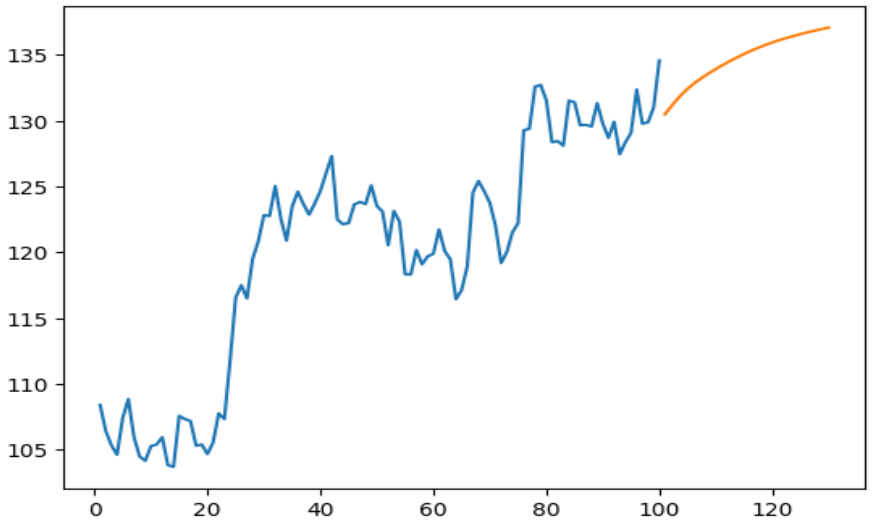
7. Calculated Train and Test Root Mean square error to check if the model is generalized well or not



Orange=Prediction based on Train set, Green=Prediction based on Test Set, Blue=Original data

8. Predicting Stock Price for the next 30 days in the future.

   → *Here we have taken the previous 100 days' stock close price data from the test set as starting input to predict the stock price of day 1 in the future. After getting the output from day 1, we shift the time window by 1. By including the future day 1 price in the time window and excluding the day 1 price of the previous time window, we predict the price for day 2 in the future. Likewise, we do for 30 days in the future.*



Orange=Next 30 days Prediction in Future, Blue= Short snip of the Original data

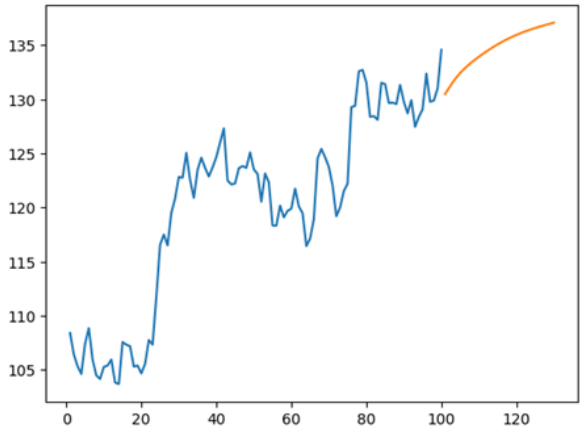
Evaluation:

```
[ ]  ### Calculate RMSE performance metrics
     import math
     from sklearn.metrics import mean_squared_error
     math.sqrt(mean_squared_error(y_train,train_predict))

     114.49907496148978

[ ]  ### Test Data RMSE
     math.sqrt(mean_squared_error(ytest,test_predict))

     107.88225588941687
```

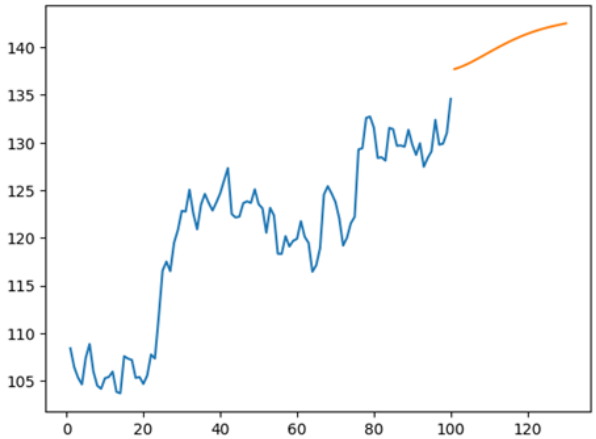
- Difference between the Train and Test data RMSE is quite small.Hence the model is generalized well.

Experiment:

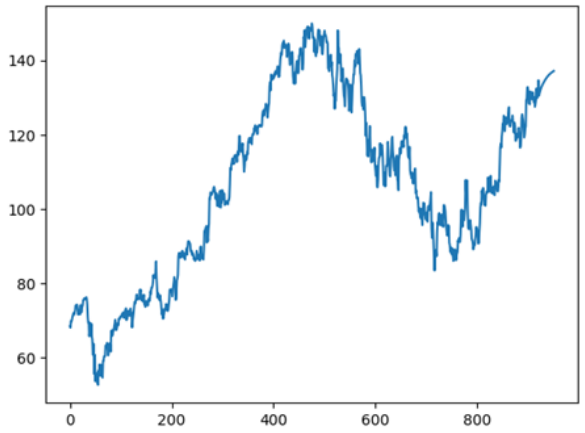
We have experimented with different time step =100, 150.

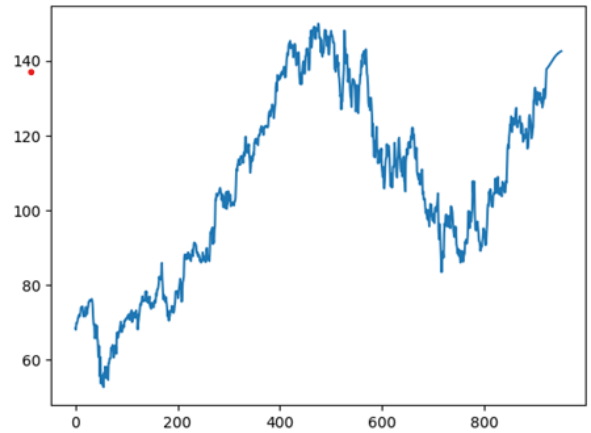


Time Step = 100



Time Step = 150





Conclusion:

- Stacked LSTM works quite well for predicting the stock Price
- Difference between the Train and Test data RMSE is quite small. Hence Stacked LSTM model demonstrates good generalization.
- While experimenting we observed that for time step=150, the model expects to have a higher closing price for the next 30 days with respect to the model with time step =100.

  Possible reason: ***The model with the smaller time step (100) might be overfitting to short-term noise or random fluctuations in the data***

  My Opinion: ***We should take time step =150 because when we take a small time step models can be less reliable for long-term predictions.***