Stock Price Prediction Using LSTM

By -
Srishti Rathi

Abstract

1.In the finance world stock trading is one of the most important activities. Stock market prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. This paper explains the prediction of a stock using LSTM.

2.In this paper we propose a Deep Learning approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction. In this context this study uses a machine learning technique called Long Short Term Memory(LSTM) to predict stock prices .

Introduction

1.The prediction of the market value is of great importance to help in maximizing the profit of stock option purchase while keeping the risk low.

2.There are various factors that affect the future stock value ,but combination of various factors can be used to predict market directions more accurately.

Different data points we can use for predicting stock prices :

1. Twitter posts about the company
2. Reddit posts about the company.
3. News headlines about the company.
4. Past prices.
5. Combination of the above data points.

Apart from these there can be many other factors.

How to gather company's past stock price details ?

Using financial websites :

1. Yahoo finance
2. Google finance

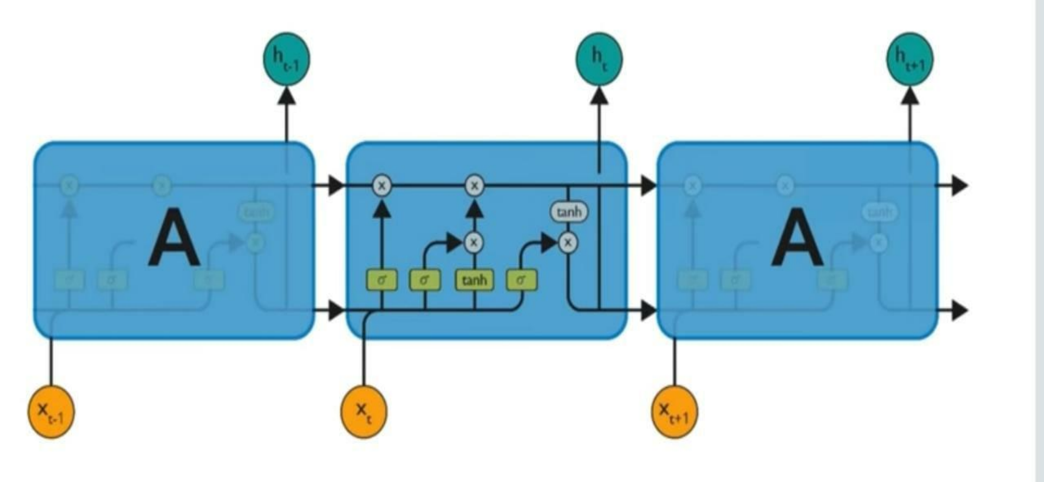
Let’s have a look on sample dataset.

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05


LSTM

LSTM(Long short term memory)

- Special kind of RNN used in the field of deep learning
- Has feedback connections
- Chain like structure



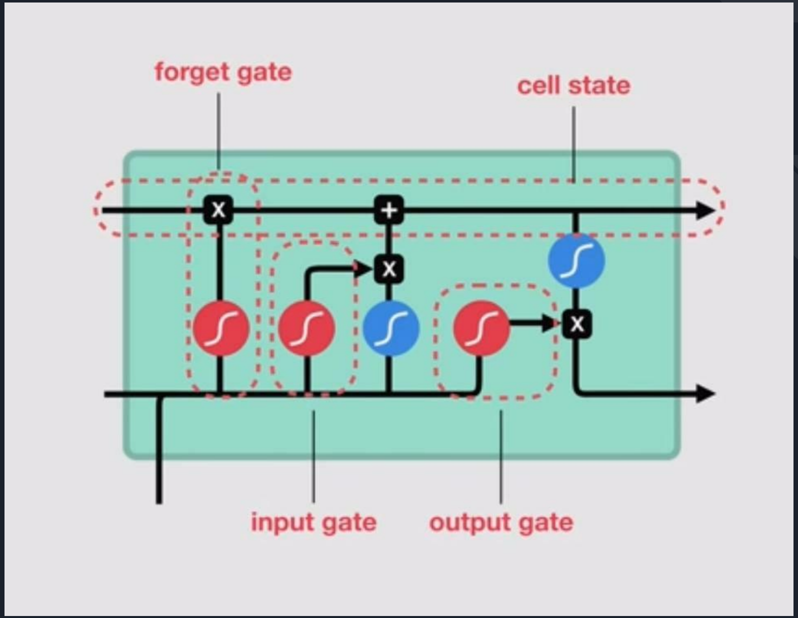
Structure Of LSTM

1.Cell state:The cell state act as a transport highway that transfers relative information all the way down the sequence chain.

2.Forget gate:Decides what information to keep and what to get rid of.

3.Input gate:Decides what values to update

4.Output gate:Decides what values to output based on input and the memory of the block



Steps to build LSTM Model

- Define Network
- Compile Network
- Fit Network
- Make Predictions

1.Define Network

The first step is to create an instance of the Sequential class. Then you can create your layers and add them in the order that they should be connected.

```
model=Sequential()

model.add(LSTM(50,return_sequences=True,input_shape=(x_train.shape[1],1)))

model.add(LSTM(50,return_sequences=False))

model.add(Dense(25))

model.add(Dense(1))
```

Input must be three-dimensional, comprised of samples, timesteps, and features.

- **Samples**. These are the rows in your data.
- **Timesteps**. These are the number of past observations for a feature which we are using to learn.
- **Features**. These are columns in your data.

2.Compile Network

This is basically Precompute step for your network.

Compilation requires a number of parameters to be specified:

1.The optimization algorithm is used to modify weights.

2.The loss function is used to compute error.

```
model.compile(optimizer='adam',loss='mean_squared_error')
```

3.Fit Network

Once the network is compiled, it can be fit, which means adapt the weights on a training dataset.

The network is trained using the backpropagation algorithm.

```
model.fit(x_train,y_train,batch_size=1,epochs=200)
```

batch_size:total number of training examples in a single batch

epochs:defines the number of times that the learning algorithm will work through the entire training dataset.

4.Predict Network

Once we are satisfied with the performance of our fit model, we can use it to make predictions on new data.

This is as easy as calling the predict() function on the model with an array of new input patterns.

```
predictions = model.predict(x_test)
```

Results

Parameters	No. of Epochs	Training RMSE	Testing RMSE
Open/ Close	250	0.01491	0.01358
Open/ Close	500	0.01027	0.00918
High/Low/Close	250	0.01511	0.014
High/Low/Close	500	0.01133	0.01059
High/Low/Open/ Close	250	0.0133	0.01236
High/Low/Open/ Close	500	0.00983	0.00859

Conclusion
As the number of epochs increases RMSE decreases

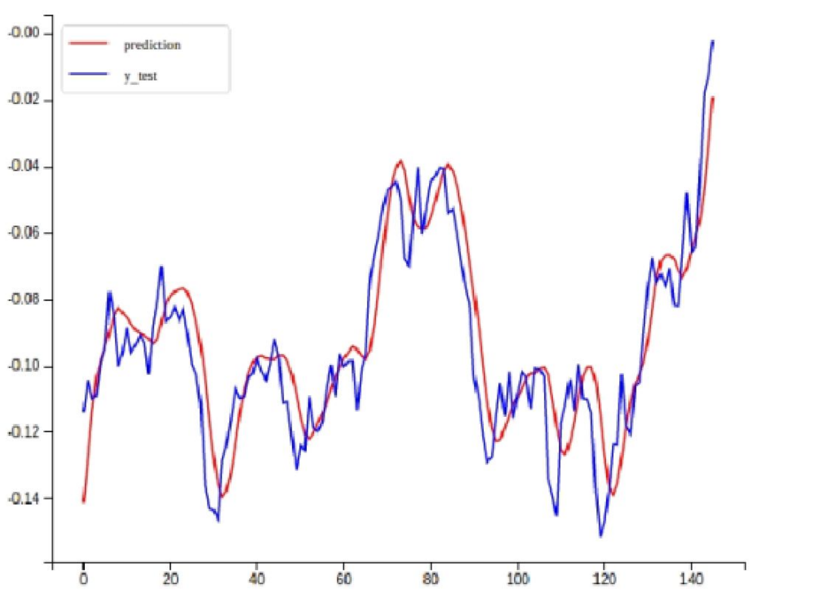


Figure 2: Open/close with 250 epochs.

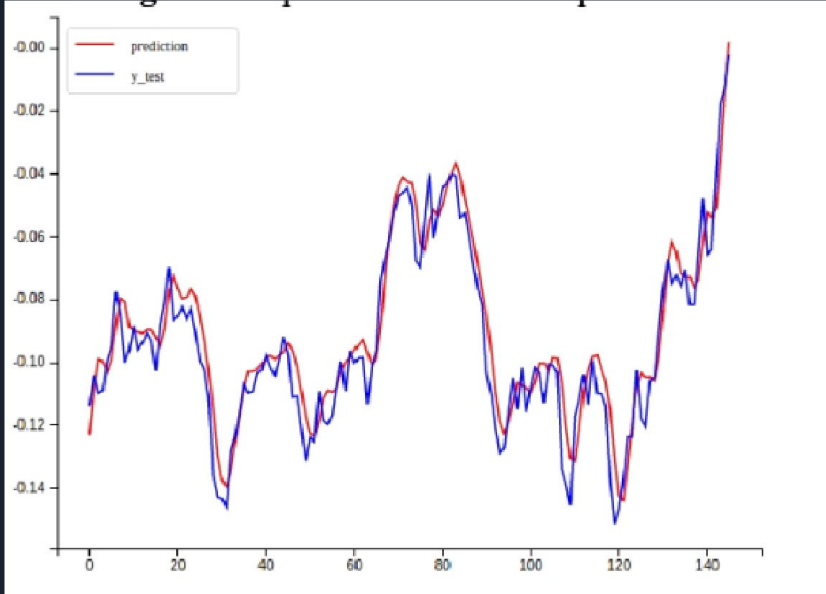


Figure 3: Open/close with 500 epochs.

THANK YOU !