Stock Price Prediction using Deep Learning Model

M. Hima Varshith (22125020), K. Ramakoti (22125014)
DA-202 ML Course Project
Data Science and Artificial Intelligence.

Abstract

This Paper represents a Deep Learning model which is used for the prediction of the stock price of a Particular company using LSTM. Since stock's price on a particular day depends more on it's lagged values, we use LSTM so that the information from the previous time steps is used to predict the price. Two layers of LSTM and two dense layers are used in the model. The model is trained over the price on a particular day and on its lagged values.

Introduction.

As we are humans we are very anxious and curious about what happens further in the time.

When you come to stock market, what is Stock market exactly ? Stock market is actually like a platform where you can buy and sell shares of the companies.These are very much useful in increasing the company s capital.For example if a company wants to increase its capital it can issue stocks which represent the ownership of the company.As said before stocks can be sold and bought in stock market .If many investors are there for a stock to invest on it than to sell it the stock price increases on the demand of investors.Conversly,if the company is not performing good then many people would try to sell it than investing in it so then stock price decreases.There are many factors which can make a stock price higher and lower.The stock prices vary from day to day based on investor needs,company's requirement and many other factors.Stock price prediction and its need:

The stock market sales generally show the economic condition of the company .So to be in good economic situation we must predict good about stocks tommorow to be in a profitable position and to expand

and improve the company.Now we have come to stock market prediction,if we can able to predict stock prices efficciently then we can be unbiased to our favourite stocks and other stocks,because we have some predicted values we can choose good ones.And it is obvious that it minimises our losses and another important use of stock market prediction is that it helps us to be consistent in predicting the prices.

Methods to predict stock prices:

There are many methods top predict the stock prices starting from statistical old classic models to modern deep learning techniques

Like ,In time series analysis moving average,exponential smooting,ARIMA process are used.In machine learning linear regression,CNN,SVM,random forests,LSTM are used.

And many other methods are used.Yes there are many methods but what method are we choosing and what is the efficient one

We are using Long term short memory(LSTM) because of the following reasonsLSTM can store the long term information by maintaining an internal state

This makes them to learn and predict trends in stock market prediction time series data

Traditional RNNs suffer from the vanishing gradient problem, which affects their ability to learn long-term dependencies.

LSTMs address this issue by introducing gating mechanisms (such as input, forget, and output gates) that allow them to retain relevant information over longer sequences.We generally use RNN's than Traditional NN's because stock price data is sequential which mean that it depends on past obsevartions where as traditional NN's beacuse they process each input independently not in temporal order .So we use LSTM which is A special case where gradient is vanishing for RNN case

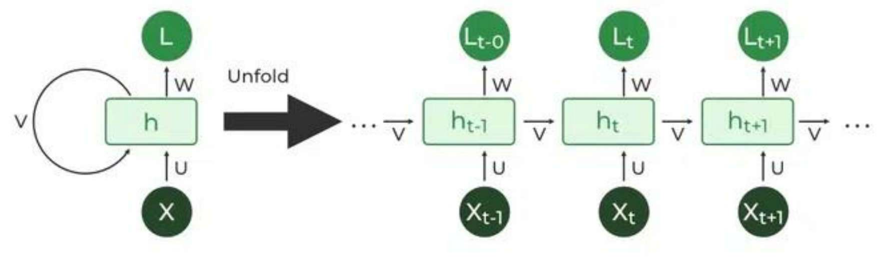
LSTM-

Long short-term memory (LSTM) is a type of Recurrent Neural network(RNN) used to deal with the problem of Vanishing Gradients.

RNN-
Recurrent Neural Networks (RNNs) differ from traditional neural networks by incorporating a mechanism where the output from the previous step is utilized as input for the current step. Unlike conventional neural networks,where inputs and outputs are independent, RNNs are particularly useful in tasks like predicting the next word in a sentence, where contextual information from previous words is essential. Used in case of sequential data like time series etc..The key component of RNNs is the hidden layer, which maintains a 'memory state' of previous input allowing the network to retain information about the sequence it has processed. This memory state is crucial for tasks

requiring temporal dependencies.Unlike some other neural network architectures, RNNs share the same set of parameters across all inputs and hidden layers, enabling them to perform consistent operations on the entire input sequence. This parameter sharing reduces themodel's complexity and enhances its ability to generalize.The ability of the RNN to maintain info about its previous states help it to make better performance in case of sequential models. In our case as the stock price depends upon its lagged values we use it for predictions.

Unfold of a RNN-



h_t is the hidden state of the time step t which is used along with x_{t+1} to find L_{t+1} (output)

$h_t=f(h_{t-1},x_t)$

h_t->current state

h_{t-1}->previous state

x_t->input

$h_t=\tanh(W_{hh}h_{t-1}+W_{xh}x_t)$

W_{hh} ->Weight at recurrent neuron

W_{xh}->weight at input neuron

$Y_t=W_{hy}h_t$

W_{hy}->weight at output neuron

The RNN is trained in the following way

1)It is initialized with random values for the h_0 and weights.

2)It is processed over $x_0,x_1,x_2,\ldots x_t$

3)It is backpropagated over Loss function which depends on h_t for 1 iteration.

4)Update the values

5)Repeat 2-4 steps for no of iterations provided (epoch)

The problem arises in the backpropagation part for long term cases (t is big) while backpropagating it is done through time too this is because h_t depends on h_{t-1} ....h_0 which are functions of w and since the activation function is tanh the values converge to 0 when t is big this is known as Vanishing gradients.To overcome this LSTM is used for cases where time steps are more.

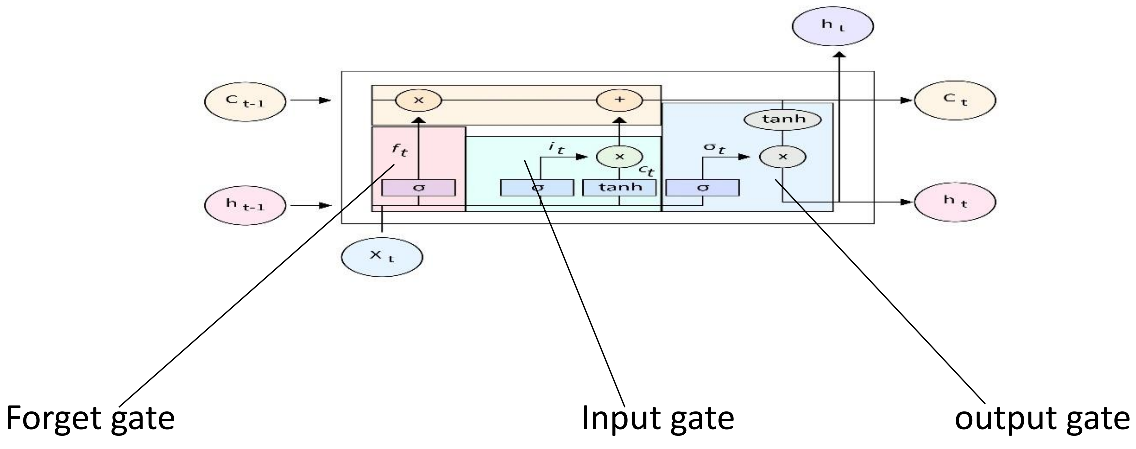
$$\frac{\partial \mathbf{L}(\theta)}{\partial W} = \sum_{t=1}^{T} \frac{\partial \mathbf{L}(\theta)}{\partial W}$$

For t=3,

$$\frac{\partial L(\theta)}{\partial W} = \frac{\partial L(\theta)}{\partial h_3}\frac{\partial h_3}{\partial W}$$

$$\begin{aligned}
\frac{\partial h_3}{\partial W} &= \frac{\partial h_3^+}{\partial W} + \frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial W} \\
&= \frac{\partial h_3^+}{\partial W} + \frac{\partial h_3}{\partial h_2}\left[\frac{\partial h_2^+}{\partial W} + \frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial W}\right] \\
&= \frac{\partial h_3^+}{\partial W} + \frac{\partial h_3}{\partial h_2}\frac{\partial h_2^+}{\partial W} + \frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial h_1}\left[\frac{\partial h_1^+}{\partial W}\right]
\end{aligned}$$

In this way it is backpropagated. To avoid this LSTM is introduced where the memory cell has extra gates(input gate, forget gate and the output gate) and along with hidden state we have another parameter which is cell states which is transferred along with hidden states of the time steps



C_t is the cell state, h_t is the hidden state

$f_t=\sigma(W_f.[h_{t-1},x_t]+b_f)$

$i_t= \sigma(W_i.[h_{t-1},x_t]+b_i)$

$S_t=\tanh(W_s.[h_{t-1},x_t]+b_f)$

$o_t =\sigma(W_o.[h_{t-1},x_t]+b_0)$

$C_t= f_{t*} C_{t-1}+ i_{t*} S_t$

$h_t= o_t*\tanh(C_t)$Here since the C_t is linearly dependent on C_{t-1} it escapes from vanishing even over long t whereas in traditional RNN h_t is depended on h_{t-1} using non linear activation function (tanh)which has values bounded between -1 and 1 so over huge t it converges to 0.

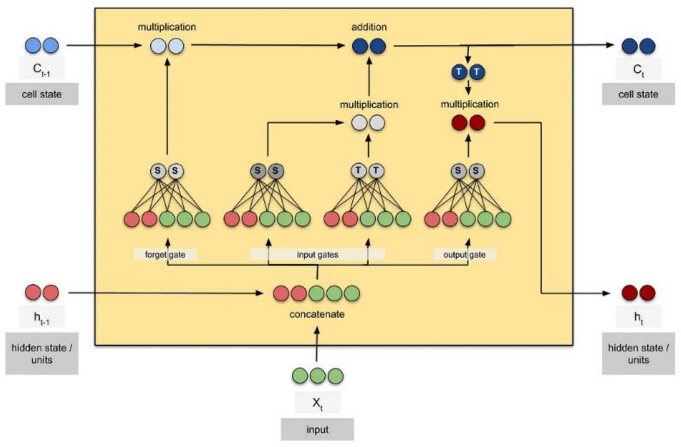
So LSTM is solution to the issue of RNN long-term dependency.

Data used and Data Preprocessing-

The data we used is from Yahoo finance which is the history of Apple company between certain dates. It consists of the Opening,high,low,close values of the stock of every day. Our aim is to predict the close values (the price of the stock and the end of the day).We only use the values of close not all the open,high and low. So we separate close from the others.The data is not used directly, it is preprocessed before used. In preprocessing first the missing values and NAN values in the data are assigned with the mean values. After that the data is scaled to avoid saturation(for huge values the gradients tends to zero).Now the data is split into training and testing data.Now the data is changed into input,output let X be the close values we divide it into sub parts like in our we divided each into a group of 60(xt,xt+1,xt+2,…..xt+59) which serves as input of 60 features and xt+60 acts as output. In simple terms we predict the stock price on a day based the 60 lagged values of its closing price on past 60 days.So if we need to find stock price X_t then the input we provide is its close values over past 60 days.

Model –

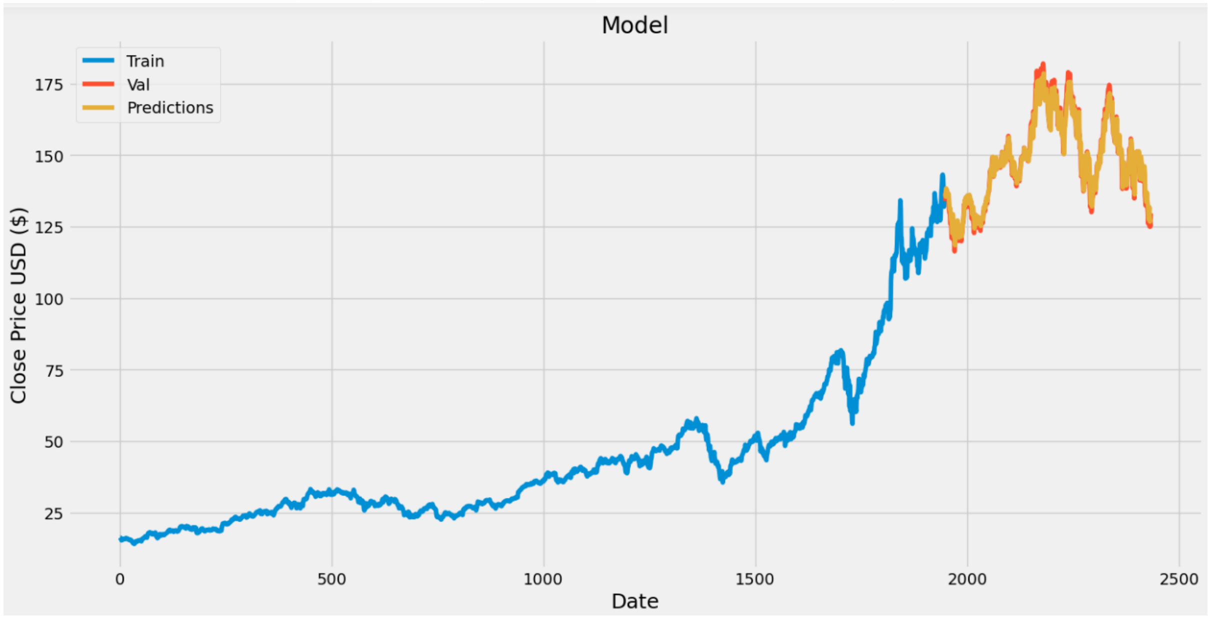
The model we used for the stock prediction consists of two LSTM layers of each 50 hidden units and two dense layers one with 25 neurons and the other with 1 neuron. The first LSTM layer takes the input (which has 60 features) and the output of this layer is served as input for the 2nd layer.



A LSTM layer with n hidden units is nothing but the dimension of the hidden states and cell states are n. In our case it is 50 so the output of the first layer has 50 features which serves as input for the second and the second layer is connected to dense layer of 25 neurons it is further connected with a dense layer of 1 neuron which gives a single featured output. This value is unscaled to get the price on that particular day.

Results-

The performance of the model is checked over the testing data by calculating the root mean squared error(rmse). The performance of the model can be imp roved by changing the no. of epochs or you can try to use more lagged values i nstead of 60 and by using more LSTM layers.The rmse we got is 0.0485323872 08035904 which is pretty enough to say that out model is good.



Conclusion-

This paper consists of deep learning model for stock price prediction using LST
M. LSTM serves as pretty good model for stock prediction given it is trained wit
h good data and the hyperparameters are tuned well.The increase in the LSTM
layers can make it to train for more complex cases.The epochs should be given
accurately so that the model gets the RSME minimum. Since Stock market play
s a important role in our country economy so discovering different methods in
predicting the prices is important and deep learning methods are the best.We
can use ensemble learning along with the LSTM which further improves the
performance.

References –

- https://intrinio.com/blog/data-science-predict-the-stock-market-a-guide#:~:text=Now%20that%20you%20know%20data%20science%27s%20role%20in,Deep%20Learning%3A%204%204.%20Natural%20Language%20Processing%20%28NLP%29%3A
- https://www.stockpathshala.com/advantages-of-stock-market-prediction/#:~:text=Why%20is%20Stock%20Market%20Prediction%20Important%3F%201%201.,6.%20Allows%20the%20Smart%20Way%20of%20Making%20Money
-https://www.patreon.com/computerscience
- https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/
- https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/
- https://ai.stackexchange.com/questions/43378/why-is-the-vanishing-gradient-problem-especially-relevant-for-a-rnn-and-not-a-ml#:~:text=In%20the%20most%20simple%20case,propagate%20through%20the%20numerous%20layers.