Stock Market Analysis/Prediction using Recurrent Neural Networks(RNN)

*Note: Sub-titles are not captured in Xplore and should not be used

K Kalyan Kumar Singamsetty
Computer Science
The University of Texas at Dallas
Dallas, USA
kxs200019@utdallas.edu

Nikitha Ramisetty
Computer Science
The University of Texas at Dallas
Dallas, USA
nxr210008@utdallas.edu

Saideep Veerabomma
Computer Science
The University of Texas at Dallas
Dallas, USA
sxv210020@utdallas.edu

Abhilash Vadla
Computer Science
The University of Texas at Dallas
Dallas, USA
axv200018@utdallas.edu

***Abstract*—In this document we study the analysis of stock market prediction/forecast using Machine Learning(ML)[1] and Artificial Intelligence(AI)[2] Concepts like Recurrent Neural Networks(RNN)[3] with Long Short-Term Memory(LSTM)[4]. We try testing the LSTM model using different parameters and configurations to observe which has a better performance over the others. This includes changing the number of neurons, adding multiple hidden layers or increasing the size of the dataset.**

We collect the daily stock data from the New York Stock Exchange(NYSE) which has columns like Date, Open, close, high, low, volume and train this data using RNN and LSTM. The dataset of the Stock is available on kaggle[5], In this document we use the stock data of GE

Keywords—Recurrent Neural Networks(RNN), Long Short-Term Memory(LSTM), Hidden Layers

I. Introduction and background work

Forecasting the stock market has been one of the most trending and attractive topics. Major reason for this attracting huge attention is because it's been determined that the stock market is unpredictable and volatile due to many factors. Even just predicting whether the stock will go up or down is also pretty hard to do. Stock market is very volatile and sensitive to the political and macroeconomic situations surrounding the particular company. There are a number of factors that influence the stock markets such as demand and supply, technical indicators of the particular stock, global economy, financial/Quarterly results of the company, public sentiments, good or bad news regarding the companies, a pandemic, war. All of these can change the mood of investors who can turn into net buyers or sellers.

We will build Recurrent Neural Networks(RNN) with Long Short-Term Memory(LSTM) to forecast the stock market trend. LSTM is generally used in the field of Deep Learning(DL)[6]. LSTM has feedback connections which is very different from Normal Neural Networks which have feedforward connections. LSTM became popular because it is extensively used in time series data, Handwriting recognition and speech recognition. A LSTM unit consists of a cell, input gate, output gate and a forget gate.

The dataset consists of daily stock data including date, high, low, open, close and volume of shares traded in a day. All these features are feeded into our RNN and LSTM which give us a prediction in the end.

II. Theoretical and conceptual study of RNN and LSTM

The Recurrent Neural Network is a type of artificial neural network which is Supervised Deep Learning[7] used in speech recognition, NLP[8], and many more. It can recognize sequential characteristics of data and use patterns to predict the next likely scenario. RNN performs extraordinarily well where the scenarios to be predicted are critical, and they are different from artificial neural networks as they use feedback loops to analyze the final output in a sequence of data.

The bulk of RNN applications is tied to language models in which the previous data determines the next letter in a word or the next word in a phrase.

LSTM is an advanced version of RNN. RNN is a very good neural network however it has a huge drawback, which is that it sometimes repeatedly predicts the same thing over and over. It basically has a Short-term memory that means it only remembers upto 1 or 2 previous iteration predictions. To solve this problem we introduce the concept of Long Short-Term Memory concept. Since stock markets are very volatile we need to have as much data as possible to predict the trend of the stock after a series of increments/decrements in the stock. Hence we need to memorize all the critical points of data. LSTM helps us remember these and at the same time also decide which all things to forget. This is achieved by introducing a number of activation and squashing functions like exponential, sigmoid, sigmoid derivative and tanh function.

With every iteration and prediction the neural networks learn what all data is to be memorized, what data is to be forgotten and updates the weight matrix/vectors accordingly. The weight matrices are generally multiplied with the input data to predict the output, if the weight is 0 then it means that the algorithm chose to forget the particular piece of information.

Now that we have understood the theoretical concept and working of the algorithms, these algorithms do better if the data we give is better, refined and reliable. Unreliable data can lead to very poor prediction and since forecasting stock prices is a very difficult task we need to be even more careful. So we will refine the dataset[5] using different statistical methods, big data technologies like MapReduce[9], spark[10], etc.

We need to do the following in order to forecast stock prices, filter data, create neural network and place functions and gates:

1. Filter Dataset – The original dataset includes 30 different stocks present in the NYSE from 1990 to 2018. Forecasting based on this huge data will require a lot of computation power so we'll filter out 1 stock(GE) using spark filter function and from 2006 to 2018. We won't look at the data before 2006 because that data is unreliable as there was a smaller amount of population involved in the stock markets globally.

2. Preprocess data – For this document we will not consider the stock prices of the after hours on NYSE, since those are not reliable. Data available is always not complete, so in order to fill the missing data we will use basic statistics like taking mean or median. Suppose a day's data is missing then we'll fill the data by taking the mean of previous and the next day.

3. Build Neural Networks – Our Recurrent Neural Networks consists of different parameters like input data, output data, hidden layer size, batch size, learning rate and dimensions of input and output data. There is a dense weight matrix which connects lstm to the dense layer. Along with these the RNN also has different vectors which store the values of input gate, output gate, forget gate and activation gate. These values of the gate vectors are updated after each data point

   Now we create our LSTM neural network which has to be initialized with parameters like input data, output data, batch size, learning rate, cell state. It will also have parameters like input gate, output gate, forget gate, activation gate matrix. The LSTM class will have different functions like sigmoid function, sigmoid derivative and tanh derivative functions which are used during forward and backward pass to calculate the gate vectors and then these gate vectors are then passed through the update weight matrix function to get updated weights with the help of a learning rate then these new weights are used on the next data point.

4. Train the dataset – We trained the Neural Network of RNN with LSTM by passing the filtered and preprocessed data from year 2008 to 2017. Since the data was really huge we had to use a significant number of epochs the training time took more than 5 hours to complete.

5. Test on Unseen data – Now on the trained neural network we pass the unseen data to predict/forecast the future stock prices that is from the year 2017 to 2018. We have split the data into 90:10.

6. Loss, RMSE and Accuracy – on the last Epoch our complex neural network predicted a RMSE of 0.47 and accuracy of 59%. By changing the number of epochs and putting different learning rate values, we'll get different loss and accuracies.

III. RESULTS AND ANALYSIS

We worked on a variety of concepts and learned how to build a complex neural network of RNN with LSTM from scratch. We also tuned a lot of parameters like learning rate, number of epochs to deduce which set of parameters works best for our dataset.

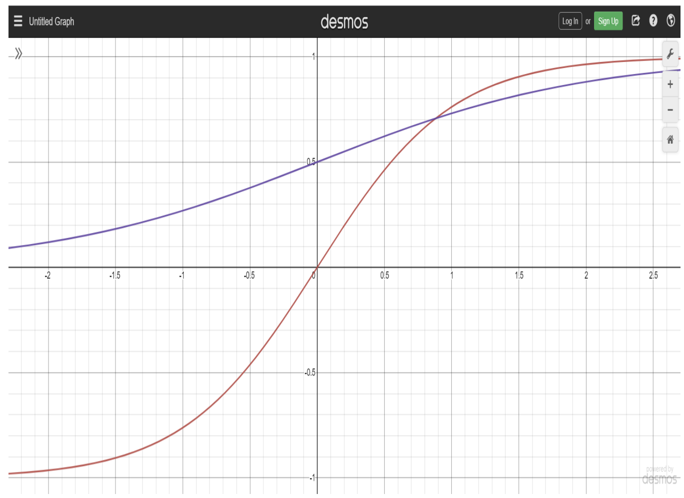
Again, predicting the stock market price is very difficult and we can only predict to a certain extent because human behavior is very unpredictable.

A. Abbreviations and Acronyms

- RNN - Recurrent Neural Networks
- LSTM - Long Short-Term Memory
- DL - Deep Learning
- NYSE - New York Stock Exchange
- NLP - Natural Language Processing
- GE - General Electric Co.
- RMSE - Root Mean Square Error

B. Equations

We have used a number of functions to calculate the weight matrices of input gate, output gate, forget gate and activation gate. The functions are tanh function, sigmoid function(also known as logistic function or squashing function), derivative of sigmoid and derivative of tanh function. We use squashing function after every data point iteration or after updating the weight matrices to ensure that our output values don't go beyond 0 and 1. However the activation function uses the tanh functions which gives an output between values -1 and 1. Below is a comparison of the sigmoid and the tanh function and other important equations are also listed.



The blue line above is a sigmoid function(0 and 1 as lower bound and upper bound respectively) and the red line is a tanh function(-1 and 1 as lower bound and upper bound respectively). We generally want our sigmoid function output to be between 0 and 1 as negative values can generate some bad predictions. Below are some of the equations that have been used in our neural network.

1. Sigmoid Function[11] –

$$S(x) = 1 / 1 + e^{-x}$$

2. Tanh Function[12] –

$$f(x) = e^{x} - e^{-x} / e^{x} + e^{-x}$$

3. Sigmoid Derivative –

$$sd(x) = S(x) * (1 - S(x))$$

Results and Analysis in a tabular format

Epochs	Hidden Layers	Learning Rate	Batch Size	RMSE Train	RMSE Test
1	1	0.01	1	0.5469	0.592
1	1	0.01	4	0.617	0.6562
2	1	0.01	1	0.4927	0.513
1	1	0.05	1	0.61	0.62
2	1	0.01	1	0.45	0.47

Our results show that with increase in batch size the error has increased but the computation time is better since it can process more data at same time. Then we see that with an increase in the number of epochs the error has decreased, which means we conclude that the neural network becomes more intelligent when we increase the number of epochs. With increase in learning rate hyperparameter[13], the error has increased.

IV. CONCLUSION AND FUTURE WORK

In this paper, we built an RNN with LSTM to predict stock prices. If this was done using simple regression techniques the accuracy would've been very low and unreliable. LSTM makes this prediction much more reliable since LSTM is intelligent and somewhat performs very similar to our human brain since it uses the history of stock price movement and takes those considerations to predict the future stock price. We also saw the different types of functions that are being used in order to predict stock prices.

We used a different set of hyperparameters every time we trained our neural network to check which set of parameters worked the best on the network and led to better accuracy. However 59% is not that reliable either but very good since we are predicting stock prices which are highly volatile.

In the future we can do a lot better, our research showed with a lot more data and resources, the accuracy can be brought up. Stock price charts have a lot of patterns and we can use these unique patterns along with the history of stock data to forecast stock prices. To study these patterns daily stock data won't create good patterns, so for this we'll need hourly stock data(10 min stock data would be even better).

This data can be used to create a Candle bar chart and search for candlestick patterns[14] like Morning star, Evening star, white spinning top, black spinning top, doji candle, white soldiers, black soldiers, hammer, inverted hammer, hanging man, white and black marubozu. When these patterns are learnt by RNN with LSTM, they can make even better predictions as it's more information and knowledge to the network. But since we have to use hourly stock data for this, the computation power required would be very high or could take days to train the algorithm.

Many financial and trading companies have already started doing this who call these things as algorithmic trading[15]. In this when we use an algorithm to trade on its own the algorithm buys and sells when certain conditions(triggers) are met and are very effective. However these come with huge conditions where they don't work when major events like pandemic and war happen when it entirely disrupts the global economy.

REFERENCES

[1] https://en.wikipedia.org/wiki/Machine_learning

[2] https://en.wikipedia.org/wiki/Artificial_intelligence

[3] https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

[4] https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9

[5] https://www.kaggle.com/code/charel/learn-by-example

[6] https://machinelearningmastery.com/what-is-deep-learning/

[7] https://www.analyticsvidhya.com/blog/2021/05/introduction-to-supervised-deep-learning-algorithms/

[8] https://en.wikipedia.org/wiki/Natural_language_processing

[9] https://www.ibm.com/topics/mapreduce
[10] https://spark.apache.org/
[11] https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function/
[12] https://www.cuemath.com/trigonometry/tanh/
[13] https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a
[14] https://www.elearnmarkets.com/blog/35-candlestick-patterns-in-stock-market/
[15] https://en.wikipedia.org/wiki/Algorithmic_trading#:~:text=Algorithmic%20trading%20is%20a%20method,computers%20relative%20to%20human%20traders.