Stock Market Predictions Using LSTM

Ashwin Ramananda
Eric Johnsson School of Engineering and Computer Science
University of Texas at Dallas
Richardson, USA
axr190005@utdallas.edu

Chaitanya Atluri
Eric Johnsson School of Engineering and Computer Science
University of Texas at Dallas
Richardson, USA
vxa190013@utdallas.edu

Anuja Gore
Eric Johnsson School of Engineering and Computer Science
University of Texas at Dallas
Richardson, USA
axg1900882@utdallas.edu

Umaymah Sultana
Eric Johnsson School of Engineering and Computer Science
University of Texas at Dallas
Richardson, USA
uss170030@utdallas.edu

***Abstract*--The stock market can have a huge impact on people and the nation's economy as a whole. Investors wish to minimize the risk of loss and maximize the profits. This gives rise to the need of a good and effective prediction system which would help traders, investors and analysts by providing supportive information like the future trend of the stock market. In this report, we present the usage of Recurrent Neural Network(RNN) with Long Short-Term Memory(LSTM) approach that uses previous 6 days stock market information to predict future trend of stock prices. LSTM avoids long-term dependence issues because of its unique storage unit structure, and it helps to forecast financial time series. The network is trained with various combinations of parameters and the accuracies are tabulated in terms of Mean Squared Error(MSE), Root Mean Squared Error(RMSE) and Mean Absolute Percentage Error(MAPE). In addition, the accuracies are visualised through a plot of training and testing values prediction of stock prices.**

I. INTRODUCTION

A stock market is a public market for trading the company's stocks and derivative at an approved stock price. History has shown that the price of shares and other assets is one of the fundamental factors affecting a country's economic strength and development. Stock prediction, which aims to predict the future trend and price of stocks, is one of the most popular techniques to make profitable stock investment[1]. Determining more effective ways of stock market index prediction is important for stock market investors in order to make more informed and accurate investment decisions.

The purpose of this report is to build a forecasting system which gives better predictions by learning well from the historical data. A time series data can be defined as a historical sequence of observations for a selected variable. In our case the variable is stock price. Recurrent neural networks have been proved to be one of the most powerful models for processing sequential data. Long Short-Term memory is one of the most successful RNNs architectures. LSTM consists of the memory cell,which is a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. With these memory cells, networks are able to associate memories and input remote in time. This property makes them suitable to operate over the structure of data dynamically over time with high prediction capacity.[3]

The remaining report is organized as follows: Section II describes the background work in the field of machine learning in regards to building time-series prediction systems. Section III provides details about the theoretical and conceptual study of the proposed algorithm. Furthermore, Section IV describes the experimental setup required to implement the architecture including the libraries used, the dataset details and required preprocessing of the data. Section V includes the test results and analysis of the application. Lastly, Section VI and Section VII conclude the performance of this application and present research areas that can be addressed.

II. BACKGROUND WORK

A lot of research work has been done on financial time series predictions. In some works, data from a single time series were used as input[4]. Certain works considered the inclusion of heterogeneous market information and macroeconomic variables. In [5], an amalgamation of financial time series analysis and natural language processing(NLP) have been introduced. Support vector machine was applied to build a regression model of historical stock data and to predict the trend of stocks[2]. In recent years, with the significant advancement in deep learning techniques, RNN have emerged as a assuring model for handling sequential data in various tasks such as NLP, speech recognition, and computer vision.

LSTM were introduced by Hochreiter and Schmidhuber and it aimed for a better performance by addressing the vanishing gradient issue that recurrent networks would suffer when dealing with long data sequences. It does so by keeping the error flow constant through special units called ”gates”. Our work aims to use this novel approach to build a promising prediction system.

III. THEORETICAL AND CONCEPTUAL STUDY

The algorithm we have chosen to implement is a Recurrent Neural Network for time series prediction. A generic time series model is based on the idea that future behavior can be predicted based on past behavior. In this case, we chose an RNN to learn how to predict future stock prices based on the stock prices in the past.

A common misconception is that if we take a singular Neural Network, and call it repeatedly, then this serves the same purpose as an RNN. This, however, is inaccurate, as we realized with further research, since the inputs and outputs in a regular NN are independent, as can be seen in the image below[8].

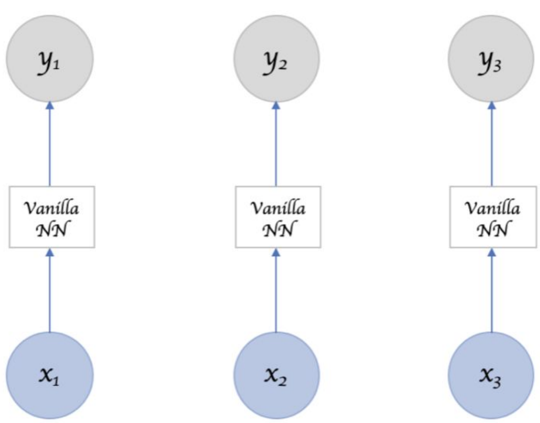


Figure 2: Can I simply not call a vanilla network repeatedly for a 'series' input?

The inputs are independent in the sense that once one iteration has completed, calling the neural net a second time will simply repeat the same process again, with no influence from the previous call.

A key component of RNN, however, is that it can take in more than one input vector, and also output more than one output vector. This feature of the RNN is what allows the network to retain its “memory”, and make predictions not simply based off of what it has learned during training, but from the prior inputs and outputs as well, which can be seen in the following image[8].

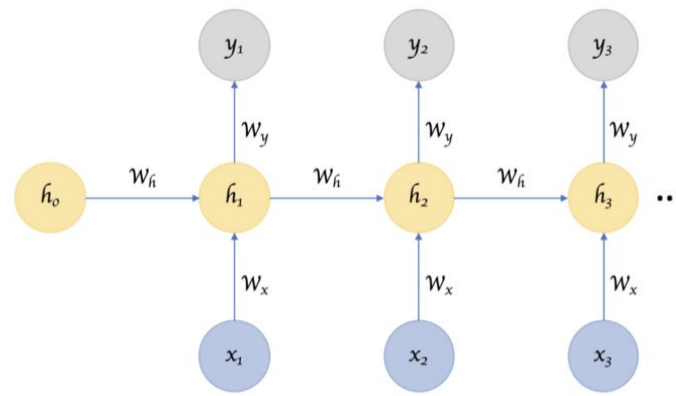


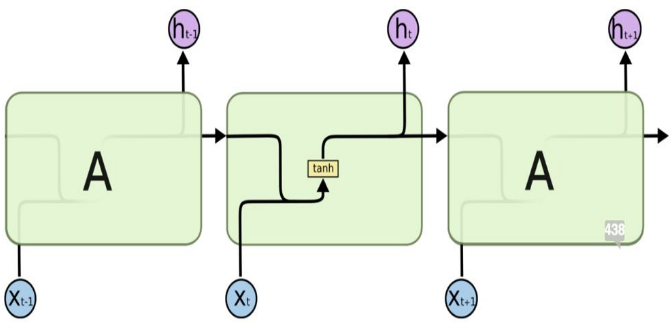
Figure 3: A Recurrent Neural Network, with a hidden state that is meant to carry pertinent information from one input item in the series to others.

Therefore, in the case of an RNN, we would be able to input a data into the network, receive an output, and then have the network store that in its memory, so that if we were to input the same data again into the RNN, it would then use what it has previously learned to more accurately predict the future value.

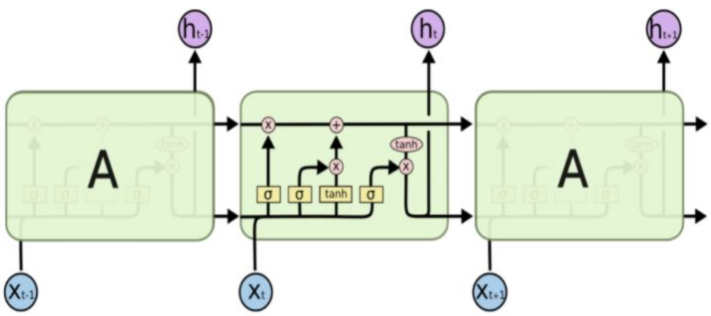
While RNNs typically have far more advantages than disadvantages when it comes to running prediction models, we did find that RNNs are more suitable for running “short term” predictions. To run more data through a regular neural network, more layers can be added to create a deep neural net. With an RNN, on the other hand, since keeping track of the previous inputs in memory is necessary, if the prediction is simply based on more recently stored information, then the RNN will be capable, but if the information is stored further back in memory, then the network may struggle[8].

Through our research, we have chosen to go forward with the LSTM (Long Short Term Memory) model, which is a special type of RNN model that does allow for the network to learn based off of long term dependencies.

To gauge a better idea of the difference between a regular RNN and the LSTM model, we can take a look at the following images[7].



The repeating module in a standard RNN contains a single layer.



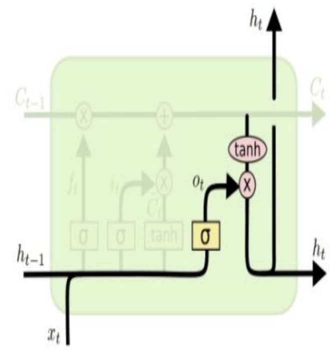
The repeating module in an LSTM contains four interacting layers.

As we can see, whereas the RNN on the top only has one layer, the LSTM model on the bottom has 4 components working together : the forget gate, the input gate, the Cell itself, and the output gate.

The forget gate is the first layer in the model, and this gate determines which data elements to keep and pass on to the cell, and which elements to get rid of entirely[7]. The Sigmoid function is used in this layer, and a 0 or 1 is output to indicate whether the data will be used or not. Data disregarded at this layer cannot be retrieved for future use.

The next part of the model is the input gate. The first part of this layer uses another Sigmoid function, which allows the relevant information which we will update to be passed on to the next part of the layer, which then uses the Tanh function to create the vector of values to be updated[7]. This vector of values is then used to update the cell.

Finally, the last component of LSTM is the output layer as shown below[7].



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

In this layer, the output from the cell is filtered to represent the necessary information in order to complete the desired prediction[7].

IV. DATASET DETAILS & PREPROCESSING

The dataset we are using in this project is the National Stock Exchange prices for the State Bank of India[9]. This dataset includes the historical prices for the SBI stocks, and our predicted values are based off of that previous data.

From the dataset, we extracted only the adjusted close prices of the selected stock, and based our model off of that column of data. Furthermore, we split the data into 4 sections. The first section is data from the first two days of the week, the second section is the third and fourth, and the third section is the fifth and sixth day. The data from the seventh day of the week is the fourth section, and this data is used for testing. In essence, our test/train split was split into four sections to better represent a RNN, where there are multiple training inputs.

V. RESULTS & ANALYSIS

An issue we came across while testing our algorithm, was that when considering large amounts of data, going back years, the accuracy for predictions decreases significantly. This is because not only does the LSTM model suffer, but the data regarding the stock market trends changes as well. Due to this, for the final analysis of our model, we chose to run the algorithm on the most recent two years of the stock data, since that would be most relevant data.

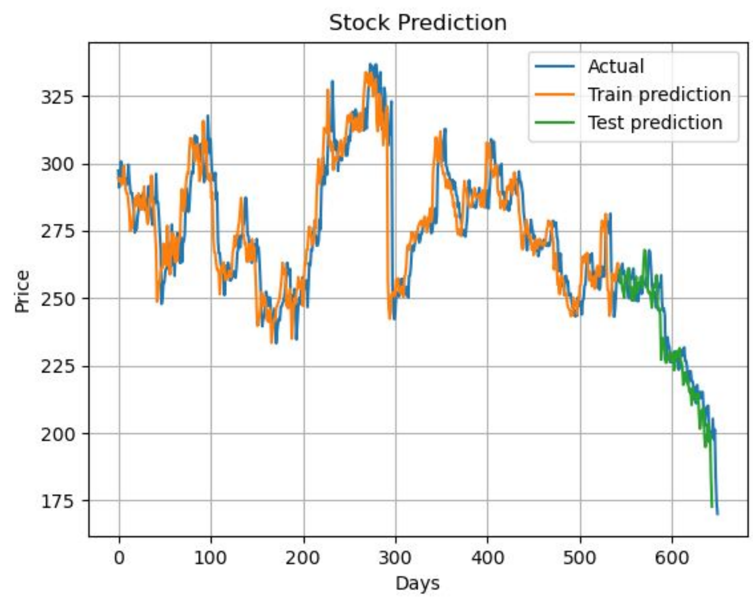
For our final results, we have tried many approaches to get the highest accuracy possible. To normalize the data, we originally used the min/max normalization technique, which allowed us to somewhat accurately predict the future stock values, but as it is stock prices, the error we were getting was too high.

As a result of the differences between the actual data and our predicted data, we decided to try different approaches and methods in terms of normalizing the data. Finally, we settled on using 100 iterations on the training data with a learning rate of 0.005. The following two tables show the various parameters we experimented with, and the corresponding training and test accuracies as well.

		TRAINING	
# of Training Cycles	Learning Rate	MSE	RMSE
100	0.1	0.0009	0.0301
100	0.3	0.0019	0.0313
100	0.4	0.0010	0.0315
100	0.5	0.0010	0.0314
100	0.05	0.0009	0.0300
100	0.005	**0.0009**	**0.0294**
200	0.005	0.0009	0.0256

		TESTING	
# of Training Cycles	Learning Rate	MSE	RMSE
100	0.1	0.0009	0.0298
100	0.3	0.0010	0.0309
100	0.4	0.0010	0.0315
100	0.5	0.0009	0.0308
100	0.05	0.0008	0.0291
100	0.005	**0.0008**	**0.0283**
200	0.005	0.0011	0.0326

In both the previous tables, the most accurate combinations of the number of training cycles and the learning rate are bolded. In both testing and training, the learning rate with the highest accuracy remained 0.5. On both train and test we also checked to see if with that same learning rate, if changing the number of iterations would help to improve the accuracy. As can be seen from the tables, changing the number of training cycles did not significantly impact the accuracy. Furthermore, as is apparent in the tables above, changing the learning rate did not change the error values too much, yet since this is the prediction of stock market prices, accuracy is very important, and therefore it was imperative we tune the parameters as much as possible. The following graph shows the final plot of our predicted values for test and train against the actual values of the stock prices.



VI. CONCLUSION & FUTURE WORK

By creating this time series model using a recurrent neural network, we have allowed for a method in which we can more accurately predict the stock prices in the future. As for real world applications, this algorithm can be used to decide which stocks to invest in, and potentially at which times as well. At the same time, however, it must be taken into consideration that while this model predicts future stock prices simply based on historic prices, there are many other factors that do go into the changes in stock values, and as a future work, those additional factors can be implemented as a part of this algorithm. By implementing those additional factors, we can better predict stock prices, especially as they may fluctuate due to economic downturns or economic growth. Furthermore, as future work, this model can also be expanded to better fit stock exchange price trends in various countries, as trends may change based on the area of the world.

REFERENCES

[1] G Preethi and B Santhi. 2012. STOCK MARKET FORECASTING TECHNIQUES: A SURVEY. Journal of Theoretical & Applied Information Technology 46, 1 (2012).

[2] Xia Y, Liu Y, Chen Z (2013) Support vector regression for prediction of stock trend. 6th Int Conf Inf Manag Innov Manag Ind Eng (ICIII) 2:123–126

[3]Stock Market Prediction by Recurrent Neural Network on LSTM Model (https://blog.usejournal.com/stock-market-prediction-by-recurrent-neural-network-on-lstm-model-56de700bff68)

[4] H. White, Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns, ser. Discussion paper- Department of Economics University of California San Diego. Department of Economics,University of California, 1988

[5] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction." in IJCAI, 2015, pp. 2327–2333.

[6] J. W. Smith, "Stock prediction using recurrent neural networks," *Medium*, 21-Aug-2019.Available:https://towardsdatascience.com/stock-prediction-using-recurrent-neural-networks-c03637437578.

[7] "Understanding LSTM Networks," *Understanding LSTM Networks -- colah'sblog*.Available:https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[8] M. Venkatachalam, "Recurrent Neural Networks," *Medium*, 22-Jun-2019. Available:https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce.

[9] "Quandl,"*quandl.com*.Available:https://www.quandl.com/data/NSE/SBIN-State-Bank-of-India.