RNN Time Series for Stock Market Prediction

Aishwarya Singh
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas
Richardson, USA
AXS200109@utdallas.edu

Sakshi Laddha
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas
Richardson, USA
SAL200000@utdallas.edu

Pooja Ramanlal Bhaiya
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas
Richardson, USA
PXR210008@utdallas.edu

Soumya Mohapatra
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas
Richardson, USA
SXM200273@utdallas.edu

***Abstract*— Presently the world's highest amount of investments are inclining towards stock markets. The stock markets deter- mines the accomplishments of nationwide financial systems. In today's economy everyone starting from a planned investor to a common man is interested in stock market. With such a huge interest in stock markets and willingness of masses to get involved in trading stocks there is a desire for forecasting systems which could provide accurate predictions of future stock prices by learning on the past stock price time series data. Our designed network aims to predict stock prices in the next couple of days for a certain institution.**

Keywords—Recurrent Neural Networks, Time Series Data, Stock

I. INTRODUCTION

A stock is a financial asset that represents ownership in a corporation or organization and provides a proportionate case for its resources (what it claims) and revenue (what it produces in benefits). Stocks are also known as shares or the value of a company. These are ownership rights in a company or organization that give investors voting rights as well as a long-term assurance of corporate profit as capital and profits rise. More explicitly, stock ownership implies that the investor owns a portion of the company equal to the number of shares held as a percentage of the company's total outstanding shares. Financial supporters gather on stock exchanges to buy and sell shares in a public setting. As buyers and sellers place orders, market interest determines share prices.

Share prices on a stock exchange can be set in a variety of ways. The most well-known method is through an auction, in which buyers and sellers submit bids and offers to buy or sell items. A bid is the price at which someone wants to buy something, whereas a deal (or ask) is the price at which someone wants to sell something. A trade is made when the bid and ask are equal. The securities exchange is shaped by millions of traders and investors who may have differing opinions about the value of a specific company and, as a result, the price at which they will buy or sell it. Thousands of transactions occur throughout a trading day when these financial backers and brokers transform their expectations into actions by purchasing or potentially selling a stock, causing minute-by-minute gyrations in it.

With millions of traders in the stock market it is desirable for systems with predictive capability of stock price values for various institutions. The stock prices are time series data where stock prices vary every business day. On some days it rises by a certain amount while on other days it falls by a certain amount. Thus, we can use any time series expectation model to foster a framework which can be utilized at forecast of stock costs for a future date.

The purpose of this report is to build a predictive system that has learnt well from the historical data in order to perform better forecasting of stock prices. A time series data can be defined as a historical sequence of observations for a chosen variable. In our case the variable is stock price. Quite possibly one of the most impressive and demonstrated models for handling sequential data has been Recurrent neural networks since the past decade. One of the most successful RNN's architectures is Long Short-Term memory(LSTM), which replaces the traditional artificial neurons with a memory cell as an unit of computation in the hidden layer of the network. LSTM is suitable to operate over data dynamically with high forecasting capacity because the memory cells help the network to associate memories and input remotely in time. [1]

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 future scopes that can be addressed.

II. BACKGROUND WORK

The analysis and modeling of finance time series is a vital task for directing investors' decisions and trades, and time series forecasting has been frequently utilized to determine future stock values. However, forecasting prices using a time series is not easy, and it necessitates a thorough examination of indexes, variables, and other data.[2] A time series is a collection of data collected over time to determine the state of a particular activity.[3] For stock market forecasting, linear models such as AR, ARMA, and ARIMA [4][5] have been utilized. The major problem with these models is that they only function for a given time series of data, thus a model that works well for one organization may not work well for another.

Over the past few years RNN and LSTM has been used a lot for problems involving time series data. A recurrent neural network (RNN) can be considered as multiple copies of the same NN, each copy sends the data to its successor using backpropagation through time. RNN is simple but it has a downside which is known by Curse of dimensionality where gradient can be either vanishing or exploding. RNN can't store long time memory, so the use of the Long Short-Term

Memory (LSTM) based on "memory line" proved to be very useful in estimating cases with long time data.

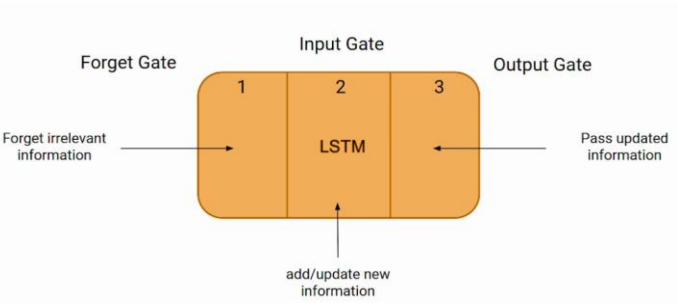


Fig. 1. Basic LSTM

III. Theoretical And Conceptual Study

Neural networks are a subset of machine learning that recognize patterns from the given data and generalize from it. This model is composed of a network of neurons, just as human brain that transmit data and information based on the input from neighbouring neurons. Neural networks computationally approximate a mapping between the inputs and outputs. They can be best considered as a function approximation algorithm to generalize and draw statistics from the input.

An Artificial Neural Network (ANN) is structured into layers of neurons and connections:

1. Input Layer
2. Hidden Layers
3. Output Layer [7]

Each neuron is connected to another neuron with an associated weight and threshold. Further, these association weights are adjusted to improve the performance of the network. An activation function enables the ANN to learn any nonlinear function. The output at each neuron is the activation of the weighted sum of its input.

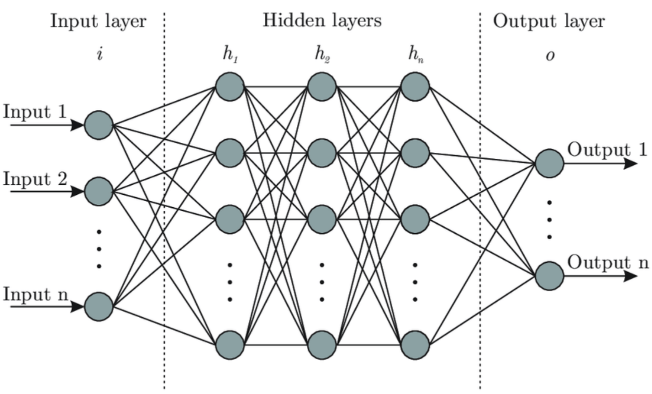


Fig. 2. Artificial Neural Network

However, ANNs cannot be used while dealing with sequence data as they do not capture the sequential information. Hence, we have used Recurrent Neural Networks. They are the class of Neural Networks that predict future value based on the sequence of operations. Recurrent Neural Networks forecast future trends and learn from the data based on prior stages. The previous phases of information need to be remembered for prediction. Here, hidden layer stocks sequential data for future calculations. The term recurrent is utilized to portray the method involved with utilizing components of prior successions to conjecture future information.

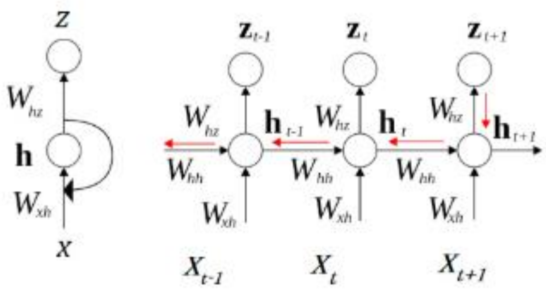


Fig. 3. Recurrent Neural Network

RNN takes two inputs, one is the present data and the other is from the past. The output for the new set of data is dependent on these two sources. Feedback loop enables us to send the output of each instant as the input to the next.[8] Each input sequence has abundant information, which is kept in the hidden state of recurrent networks. As the network sweeps forward to cope with a new input, this concealed information is recursively exploited.

In a recurrent neural network, the input layer receives the input and then activations are applied in each hidden layer successively to produce the output. Each hidden layer is characterized by its own weights and biases. The commonly used activation functions are:

a) Sigmoid
b) Tanh
c) ReLU

Backpropagation learning is used in recurrent networks. The primary distinction between backpropagation and recurrent networks is that the former is unaffected by the order in which the input is encountered, whereas the latter does. As a result, recurrent networks are a better choice for predicting stock market returns based on recent history. The ability of the recurrent network's nodes to execute a feedback mechanism is another key difference between a feed forward backpropagation network and a recurrent network. This mechanism speeds up the process of combining information from prior patterns with current inputs. [9]

The formula for the current state can be written as –

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

where,
h_t : New state
h_{t-1} : Previous state
i_t : Input

Applying the weights, we can update the above equation as –

$$h_t = f\,(w_{hh}h_{t-1}, w_{xh}i_t)$$

where,
w_{hh} : weight at the recurrent neuron

w_{xh} : weight at the input neuron

After multiple hidden states, the final output state can be written as –

$$y_t = w_{hy} h_t$$

$$\boldsymbol{h_t} = \sigma(\boldsymbol{W_{ih}} \cdot \boldsymbol{x_t} + \boldsymbol{U} \cdot \boldsymbol{h_{t-1}} + \boldsymbol{b})$$
$$\boldsymbol{\hat{y}_t} = \sigma(\boldsymbol{W_o} \cdot \boldsymbol{h_t})$$

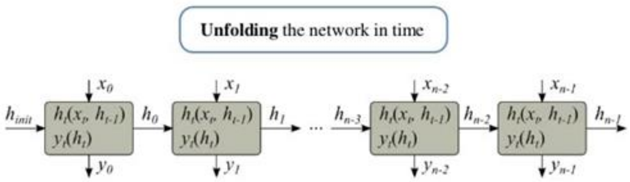


Fig. 4. Recurrent Neural Network - Calculations

IV. Dataset And Preprocessing

The dataset we are using in this project is the New York Stock Exchange [10]. This dataset contains historical stock prices with fundamental data of over 500 companies such as Yahoo Finance, fundamentals are from Nasdaq Financials, extended by some fields from EDGAR SEC databases.

We are using the prices-split-adjusted.csv file from this dataset which consists of raw daily prices where maximum data spans from 2010 to 2016 of companies new in the stock market. There have been 140 stock splits with some adjustments in this data.

For dataset pre-processing, we have dropped Null and NA values. We have also processed the data through min max scalar normalization for values to be in range. Furthermore, for better results and training our model we are focusing on 4 prime features – Open, Close, Low and High and dropped the remaining features. We have split our dataset in an 80:20 ratio for training and testing respectively.

V. Analysis And Results

While running our algorithm, we came across a major issue- on training the model with the dataset the Root Mean Square Error of the model kept increasing. As we increased the size of the dataset, the error increased exponentially. This is because of a fundamental issue of using RNN on timeseries data set. With the RNN model arises the exploding gradient issue which was causing the error to increase significantly.

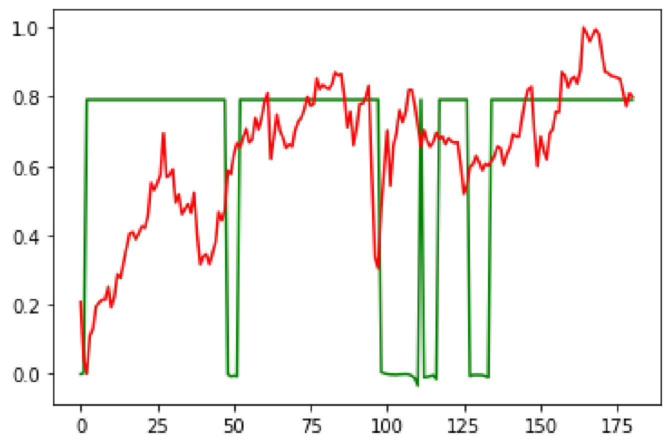
To overcome this problem, we have defined a range for updating the weights, so that the calculated weights do not go beyond range specified.

For our eventual outcomes, we have attempted many ways to deal with get the most elevated precision conceivable. To standardize the information, we initially utilized the min/max standardization strategy, which permitted us to fairly precisely anticipate the future stock qualities, yet as it is stock costs, the mistake we were getting was excessively high.

Because of the contrasts between the actual information and our predicted information, we chose to attempt various methodologies and techniques as far as normalizing the information. At last, we chose utilizing 10 cycles on the preparation information with a learning rate of 0.00002 on 200k data. The accompanying two tables show the different boundaries we explored different combination of various parameter values with their testing and training accuracies as well.

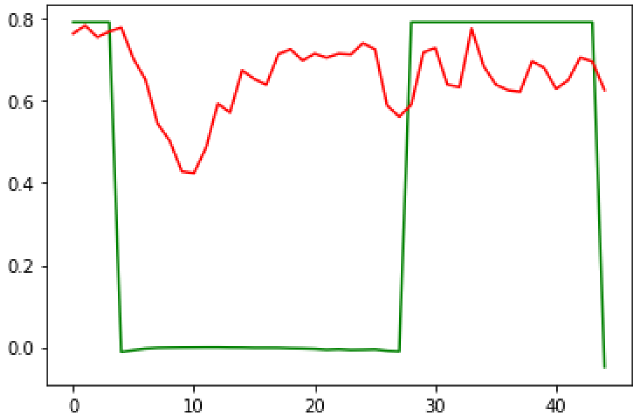
Training Results and Graph:

Dataset Size	**Learning Rate**	**RMSE Test**
200000	0.00005	0.4897110
200000	**0.00002**	**0.1855463**
500000	0.00002	0.6648556
500000	0.00005	0.6093905
650000	0.00005	0.6643707
650000	0.00002	0.6637472



Testing Results and Graph:

Dataset Size	Learning Rate	RMSE Train
200000	0.00005	0.3587356
200000	**0.00002**	**0.2764533**
500000	0.00002	0.6433556
500000	0.00005	0.4856990
650000	0.00005	0.6451522
650000	0.00002	0.6454222



VI. Conclusion And Future Work

By making this time series model utilizing a recurrent neural network, we have took into account a technique wherein we can all the more precisely anticipate the stock costs later on. With respect to true applications, this calculation can be utilized to conclude which stocks to put resources into, and possibly at which times too. Simultaneously, in any case, it should be thought about that while this model predicts future stock costs basically dependent on noteworthy costs, there are numerous different variables that do go into the progressions in stock qualities, and as a future work, those extra factors can be carried out as a piece of this calculation. By executing those extra factors, we can all the more likely foresee stock costs, particularly as they might change because of financial slumps or monetary development. Besides, as future work, this model can likewise be extended to more readily fit stock trade value patterns in different nations, as patterns might change dependent on the space of the world.

All RNNs have feedback loops in the recurrent layer. This lets them maintain information in 'memory' over time. But for use cases with long-term temporal dependencies, it is difficult to train standard RNNs. This is because the gradient of the loss function decays exponentially with time (called the vanishing gradient problem). Hence the use of the Long Short-Term Memory (LSTM) based on “memory line” proved to be very useful in forecasting cases with long time data.

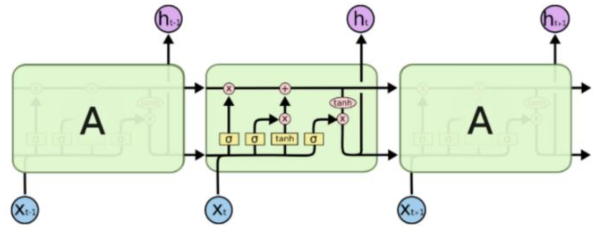


Fig. 4. Modules of a LSTM

As LSTM maintains a strong gradient over many time steps it can be trained with relatively long sequences.

References

[1] 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)

[2] https://www.sciencedirect.com/science/article/abs/pii/S156849461500352X

[3] Hamzaebi C., Akay D. and Kutay F. (2009). ”Comparison of direct and iterative artificial neural network forecast approachesin multi-periodic time series forecasting.” Expert Systems with Applications 36 (2): 3839- 3844.

[4] Zhang G. P. (2003). ”Time series forecasting using a hybrid ARIMA and neural network model.”Neurocomputing 50:159-175.

[5] Menon V. K., Vasireddy N. C., Jami S. A., Pedamallu V. T. N., Sureshku- mar V., and Soman K. P. (2016, June). ”Bulk Price ForecastingUsing Spark over NSE Data Set.”In International Conference on Data Mining and Big Data :137-146.

[6] Zineb Lanbouri, Said Achchab,Stock Market prediction on High frequency data using Long-Short Term Memory,Procedia Computer Science,Volume 175,2020,Pages 603-608,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2020.07.087.

[7] Adil Moghar, Mhamed Hamiche, Stock Market Prediction Using LSTM Recurrent Neural Network,Procedia Computer Science,Volume 170,2020,Pages 1168-1173,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2020.03.049.

[8] Hiransha M, Gopalakrishnan E.A., Vijay Krishna Menon, Soman K.P.,NSE Stock Market Prediction Using Deep-Learning Models,Procedia Computer Science,Volume 132,2018,Pages 1351-1362,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2018.05.050.

[9] J. Roman and A. Jameel, "Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns," Proceedings of HICSS-29: 29th Hawaii International Conference on System Sciences, 1996, pp. 454-460 vol.2, doi: 10.1109/HICSS.1996.495431.

[10] Dataset source - https://www.kaggle.com/dgawlik/nyse.