Recurrent Neural Networks for Stock Market Prediction

Navya Teja Gajula
Erik Johnson School of Engineering & Computer Science
University of Texas at Dallas
Richardson, Texas
nxg220002@utdallas.edu

Pooja Minna Ravindra
Erik Johnson School of Engineering & Computer Science
University of Texas at Dallas
Richardson, Texas
pxm210084@utdallas.edu

Sanjana Majeti
Erik Johnson School of Engineering & Computer Science
University of Texas at Dallas
Richardson, Texas
sxm220015@utdallas.edu

***Abstract*—The Stock market analysis and it's prediction is an immensely demanding and challenging topic due to some of its properties like complexity, inherent nonlinearity, and volatility. Recurrent neural networks (RNNs) have proved and recently demonstrated that there is possibility of very promising outcomes in the identification of all the sequential dependencies and the patterns in time series data. This paper will help to study the efficacy of the RNN-based models for the help of the stock market forecasting and to explore their capability to produce very precise, accurate and trustworthy predictions. In this paper, there is an employment of the RNN architectures that is used to show the temporal dynamics of the historical stock market data. This has been done by the inclusion and usage of the Long Short-Term Memory, (LSTM). To find out level of complex interactions of the market dynamics, in which the input characteristics will include a wide range of the financial indicators, the macroeconomic aspects, and the sentiment analysis data. The execution of the Recurrent Neural Networks (RNN) , models is thoroughly assessed by this paper by the utilization of a big historical stock market dataset that includes a variety of assets and markets. The examination will include a range of different forecasting timeframes, from short-term forecasts to long-term projections, in order to evaluate how well they can be applied to various investing strategies.Our results show that when compared to traditional time series prediction techniques, the RNNs do perform in a competing manner. The key elements that affect various RNN architectures's ability of prediction. We also have highlighted the pros and cons of the same. The study will also reveal many details on how these kinds of models react to various market situations and circumstances and their ability to adapt in those situations and to the changing trends. This paper offers useful insights into the practical use of RNNs for stock market forecasting, this could be an added advantage to many investors and financial scientists. RNNs does provide a strong framework for time series prediction and lays a strong foundation for advanced improvements in the predictive analytics field, within the trading domain apart from other financial prediction difficulties usually faced.**

Keywords— RNN, LSTM, Backward Propagation, Neural Networks, Vanishing gradient Problem, Forward Propagation, Forget gate, Root mean squared error, Output gate,Accuracy, Prediction, R-squared error, Input gate

Introduction

Stock market prediction usually requires a lot of components that should go together. It could be the study of economics, finance, machine learning, offering significant potential benefits to investors, traders, and financial analysis by enabling better investment decisions and higher profitability.

Recurrent Neural Networks are the most popular among the hierarchy of the techniques of machine learning. These are immensely useful with handling the sequential data, making them curated, as well as most relevant for analyzing the time-series data in the stock market. There is a hidden state that carries the information from past time to the present time. The RNNs constructively find out the temporal dependencies among the data. The conventional Recurrent Neural Networks have a limitation which is vanishing gradient problem. This is the reason why it can learn only the long range dependence

To overcome and to address this limitation, we adopt Long Short-Term Memory (LSTM) units, which will help and aid in successfully overcoming the vanishing gradient issue that is being occurred. LSTM units do exhibit an inherent property that is used to recognize and grasp long-term dependencies. This finds its applications in speech recognition, language translation, and image captioning. This model is implemented in Python, PySpark and visualization libraries like Matplotlib have been used to analyze the dataset and present the trained model's outcomes.

In the sections that follow, there is examination in the trends of the dataset and uses the trained model to present a summary of our findings. By successfully predicting the stock prices of a popular company, this method hopes to make the science of stock market forecasting, interesting and better.

Background work

Back into the history, there are different ways that have been followed for the purpose of stock Prediction. where the main task is to predict the price of stocks for the future.As the trends change, there has been a notable influence of social networking platforms on the stock price prediction. for example: tweets and news on social media. The techniques that have been widely used are:

Technical Analysis:

Technical analysis involves the assessment of financial assets like stocks, currencies, commodities, or cryptocurrencies through the examination of historical market data, with a main emphasis on price and volume patterns. The main objective is to predict the future price

movements and to pinpoint the potential trading prospects by analyzing the past market performance.

Fundamental Analysis:

Fundamental analysis will usually involve evaluating financial assets like stocks, currencies, or commodities by analyzing the underlying economic, financial, qualitative, and quantitative factors that can impact their true value. The main aim of this kind of analysis is to assess whether an asset is overvalued or undervalued in the market and to make well-informed investment choices accordingly.

Sentiment Analysis:

Sentimental Analysis is used to analyze the sentiment of a text. It is of the techniques of Natural Language Processing, which involves the analysis of a sentence or text. It could be a positive one, a negative one or a neutral one. This is nothing but filtering out the opinions of the public. It has many applications like Social Media Monitoring , Market Research, Customer Support, Brand Monitoring.

In recent times, Machine Learning and Artificial Intelligence have brought many advancemnts in stock price prediction techniques. Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM), are playing a prominent part for their effective modeling of stock market movements and the predictions. LSTM's ability to retain information over time series data makes it powerful for predicting stock prices, revolutionizing the investment landscape.

THEORETICAL AND CONCEPTUAL STUDY

A. LSTM: RNN is one of the categories of the neural networks that have been designed to handle the sequential data, emphasizing the significance of the data order. RNNs do offer the advantage of using the context-sensitive information, specifically the feedback, that is, to map the input and output sequences. This property will enable the RNNs to take into account all the previous inputs and as well as their connections with the current input to be able to produce the desired output. Traditional RNNs face many limitations in accessing extensive context information.

The main issue arises when the network processes data over time, this will cause the action of the initial input upon the state that is hidden to weaken, this will make it difficult for this network to retain all kind of long-term dependencies. This is nothing but called as the vanishing gradient problem, where in error gradient of the function becomes too small as it undergoes the backpropagation in time. This leads to slow learning or convergence to a suboptimal solution. To tackle this problem, the LSTM structure was being introduced. LSTMs are a specific type of RNN that was designed to address this issue by doing selective remembrance over time. This allows it to effectively handle the long-term dependencies in any sequential data.

LSTMs as given above, achieve their capability by utilizing the memory cells to store information over time and they employ three gating mechanisms to regulate the movement of information into and out of the cell.. Within the LSTM networks, each memory unit does contain three gates: the input gate, the forget gate, and the output gate, all of which are used for the selective management of the input gate manages the flow of information into and out of the memory cell by deciding which data from the current input should be stored within the cell. This is accomplished by passing the input through a sigmoid function, and this will result in an output vector consisting of the values between 0 and 1, for each of the elements of the input. The output vector is then elementwise multiplied with another vector derived from passing the input through a hyperbolic tangent function. The multiplication operation is responsible for identifying the specific elements from the input that will be incorporated into the memory cell through addition. When the sigmoid output for an element is close to 0, it will indicate that the corresponding input element won't be aggregated to the memory cell. On the other hand, when the sigmoid output approaches 1, the input element will be entirely integrated into the memory cell. Meanwhile, the forget gate governs the selection of information from the previous hidden state that should be omitted or discarded.

The forget gate operates in a similar manner to the input gate, but instead of taking the current input, it takes into account the previous hidden state. It employs a sigmoid function to produce values ranging from 0 to 1 for each element of the hidden state. This indicates the proportion of information to be discarded. When the sigmoid output is close to 0, all the corresponding information is fully discarded, while a value closer to 1 will imply that the information will be retained.

The output gate plays a crucial role in determining which information from the memory cell should be transferred to the next hidden state. Similar to the input and forget gates, this process involves passing the current input and all previous hidden states through sigmoid and hyperbolic tangent functions, respectively. Additionally, the output gate utilizes another sigmoid function to determine the proportion of the current memory cell state that will be output to the next hidden state. The key advantage of LSTMs over standard RNNs lies in their capability to access and selectively retain or discard context information across time. This ability allows LSTMs to effectively capture long-term dependencies in sequential data.

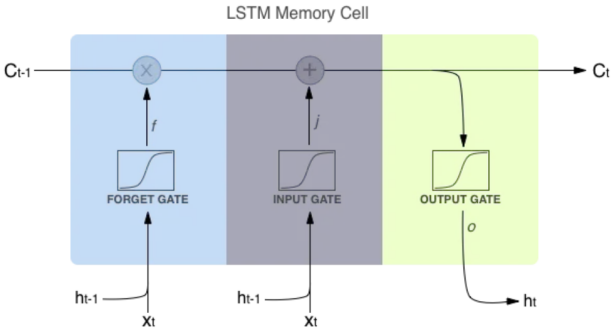

Fig-:1-LSTM Memory Cell [8]

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

Equation of Gates

Fig-:2-Equation of gates [8]

$i_t \rightarrow$ represents the input gate.

$f_t \rightarrow$ represents the forget gate.

o_t - represents output gate.

$\sigma \rightarrow$ represents sigmoid function.

$w_x \rightarrow$ weight for the respective gate(x) neurons.

$h_{t-1} \rightarrow$ output of the previous Istm block (at timestamp t-1).

$x_t \rightarrow$ input at current timestamp.

$b_x \rightarrow$ biases for the respective gates(x).

$$\tilde{c_t} = tanh(w_c[h_{t-1}, x_t] + b_c)$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{c_t}$$
$$h_t = o_t * tanh(c^t)$$

$c_t \rightarrow$ cell state(memory) at timestamp(t).

$\sim c_t \rightarrow$ represents a candidate for cell state at timestamp(t).

Fig-:3-Equation of LSTM Cell [8]

An LSTM model is basically trained in three steps. Firstly, using the forward calculation (Forward propagation) method, the output values of the LSTM cells are determined. Second step is the error term of each LSTM cell is reverse-calculated, considering two different reverse propagation directions based on time and network level. Thirdly, gradient-based optimization methods are used to update the weights after calculating each weight's gradient in accordance with the associated error term.

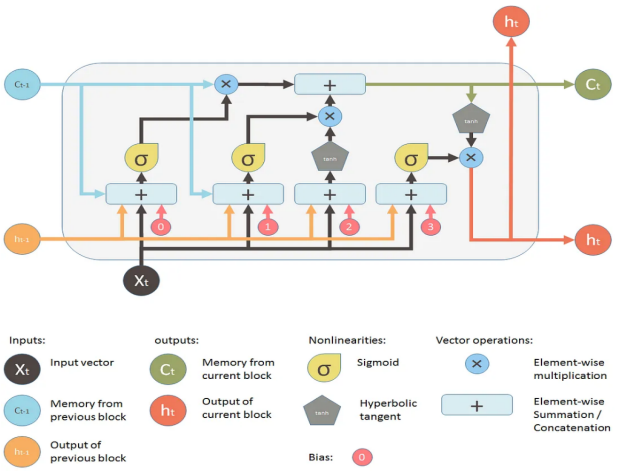

Fig-4:-LSTM building block from [7]

B. RNN: It is a neural network, a machine learning technique, which consists of three fundamental parts: the input layer, the output layer and the hidden layers. Each layer contains the nodes, or neurons, these are related and connected to the nodes that are present at neighboring layers through the weighted connections. The output layer will produce the network's final output after receiving the initial input data. The hidden layers, located between the input and output layers, will process and transform the input data to generate the final output. In neural networks, the temporal lobe is primarily employed for classification and regression tasks, thanks to its long-term memory function, this will allow it to retain information from earlier time steps. This makes it particularly useful and curated for tasks like language processing or speech recognition, where data order does matter. For computer vision tasks, neural networks primarily use the occipital lobe, where CNNs (convolutional neural networks) are most generally employed. The occipital lobe is responsible for recognizing patterns in a visual input, such as edges, forms, and textures.

The frontal lobe, in contrast, is mainly used for time series analysis, sequences, and lists in neural networks. In the frontal lobe, RNNs (recurrent neural networks) are frequently employed, which are especially useful for tasks involving sequential data like language processing or time series analysis. Each neuron in the frontal lobe does receive input from the previous time step as it progressively processes the input data, allowing the network to preserve and store the data across time and make predictions based on past inputs. Feedforward neural networks (FFNNs) with amnesia are different, RNNs can handle input sequences without losing the relevant information due to their recurrent nature with the temporal feedback loops. These loops create recurrent connections, where the output from the previous step becomes the input for the current step in the process.

In the training process, the RNN performs the following steps: computing prediction errors in the forward pass to determine the loss values on both the training dataset and validation set, calculating gradients at each particular layer, by back propagating errors across t timesteps, and then there is updation of the weights in a forward pass loop. The RNN is altered by mapping it in a way that reduces the disparity between its output and all the real observations in the training dataset.

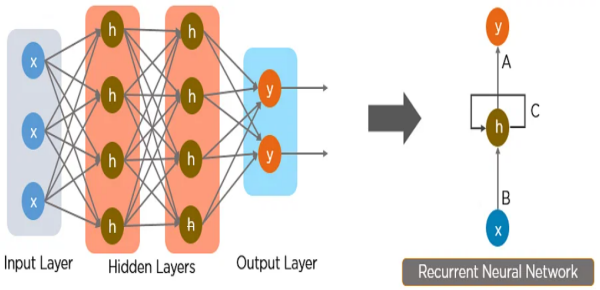

Fig-5: RNN architecture from [6]

C. Root Mean Squared Error: This is a widely used common statistic used to assess the accuracy of a regression model. This is computing by calculating the square root of the sum of squared differences between the expected value and actual value of the variable which is dependent. RMSE basically measures the square of the root of the variance of the residuals. Residuals are the discrepancies that are present between the actual value and the predicted value of the dependent variable in a regression model. They indicate how far each data point deviates from the line of best fit. The spread of these residuals around the regression line is usually quantified by the RMSE. A low RMSE indicates that the model can predict outcomes accurately, while a large RMSE suggests that the model is not performing well. RMSE can be used to compare the level of performance of different regression models as well as the performance of a single model over time. To perform time-series analysis using LSTM and RNN, several steps need to be followed:

Data Preparation: Formatting and transforming the data for time series analysis. This includes handling all the missing values, null values and splitting of the data into different datasets.

Data Preprocessing: Preprocessing of the data to make it suitable for the use of RNN or LSTM models, such as scaling, normalizing, or making other necessary modifications.

Model Selection: Choose an appropriate model architecture for the time series analysis. Evaluate different models' performance using metrics and methods like mean squared error, (MSE) and the root mean squared error, (RMSE) post execution of the selected model.

Model Training: Use Stochastic Gradient Descent (SGD) or other optimized algorithms to train the model on the training dataset. Overfitting should be avoided and stop training the model if that is necessary.

Model Evaluation: Assess the model's performance on the testing dataset. Calculate performance metrics like MSE, RMSE, or MAE by comparing predicted values versus actual values.

By following all these above mentioned steps, one can effectively perform time-series analysis using LSTM and RNN models.

PREPROCESSING DATASET

The dataset is in the .csv format about the stock pricing of google. The dataset is taken from https://github.com/poojaminna/bid_data_dataset.The dataset is loaded as a dataframe for the preprocessing. The dataset has the following features: Open, High, Low, Close and Volume. This study extracted the open feature for prediction.The MinMaxScaler, which is present in sklearn.preprocessing is used, in order to scale the open feature between 0 and 1 .The concept of map reduce was used for reshaping the data as per the requirements. Post the preprocessing of data, it is split into two sets : the train set and the test set using the train_test_split form the sklearn.model_selection and the train_size is 0.7.

RESULTS AND ANALYSIS

Experiment No.	Parameters	Results
1	Learning Rate: 0.05 No. of Epochs: 15	Train R2 score = 0.940 Train RMSE = 0.244 Test R2 score = 0.937 Test RMSE = 0.249
2	Learning Rate: 0.05 No. of Epochs: 10	Train R2 score = 0.932 Train RMSE = 0.260 Test R2 score = 0.929 Test RMSE = 0.265
3	Learning Rate: 0.05 No. of Epochs: 5	Train R2 score = 0.929 Train RMSE = 0.266 Test R2 score = 0.926 Test RMSE = 0.271

Repeating the analysis for the same parameters but different learning rate.

4	Learning Rate: 0.07 No. of Epochs: 15	Train R2 score = 0.948 Train RMSE = 0.243 Test R2 score = 0.938 Test RMSE = 0.248
5	Learning Rate: 0.07 No. of Epochs: 10	Train R2 score = 0.933 Train RMSE = 0.258 Test R2 score = 0.930 Test RMSE = 0.264
6	Learning Rate: 0.07 No. of Epochs: 5	Train R2 score = 0.931 Train RMSE = 0.262 Test R2 score = 0.928 Test RMSE = 0.268

Table-1:Result analysis for different values of learning rate and Epochs

Firstly, the train R2 scores are higher than the test R2 scores and similarly the RMSE is higher for test than train which makes it evident that there is more error in prediction of data in test dataset than in train dataset. The observations that are made by making learning rate constant and changing the epochs is that as the number of epochs decreases the R2 score is decreasing and thereby increasing the error.Increasing the learning rate often leads to a higher R2 score compared to using a lower learning rate. And the root mean squared error is also reduced in comparative to lower learning rate.

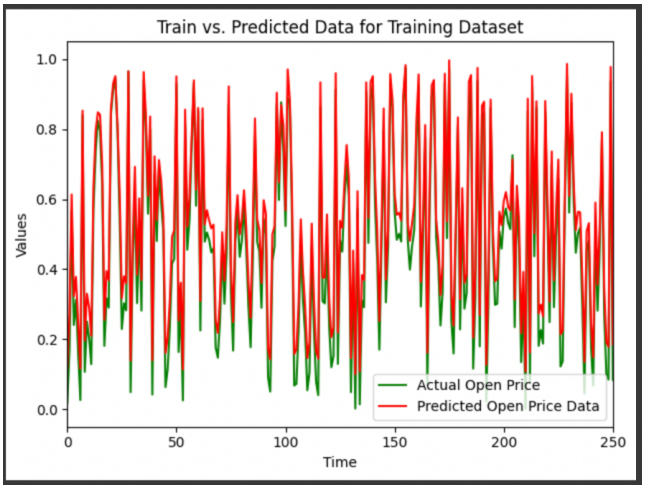

Fig-6: Plot of Actual Data vs Predicted Data for training dataset

The plot of predicted data and the actual data for google stocks are shown in Fig.4 and Fig.5. Fig.4 shows the plot for training dataset, whereas Fig.5 shows the plot for testing dataset.

The purpose of showing both Fig.4 and Fig.5 is to compare the model's performance on the training and testing datasets.The two plots should ideally resemble one another, showing that the model is neither overfitting nor underfitting. Underfitting happens when the model performs poorly on both datasets, as opposed to overfitting, which occurs when the model performs well on the training dataset but poorly on the testing dataset. We evaluate the model's

capacity for generalization and its potential to perform well on unknown data by contrasting the two figures.

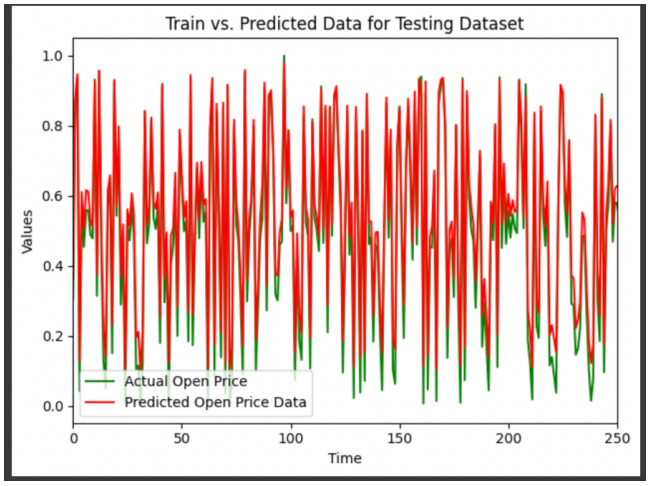

Fig-7: Plot of Actual Data vs Predicted Data for testing dataset

CONCLUSION AND THE FUTURE WORK

As the research paper is being concluded, the entire vision focus is primarily on prediction of stock market trends by utilizing LSTM units within a Recurrent Neural Network (RNN). LSTM is popular for its capability to properly analyze the time series data, making it the most suitable choice for the forecasting of stock prices. For checking the model's effectiveness, there are two evaluation metrics: Root Mean Square Error (RMSE) and the R2 score; these parameters allow us to measure the accuracy of the predictions that are made.

This study tried to achieve promising results through the successful training of the model, using the various activation functions and the hyperparameters. The model demonstrates high accuracy. This has been indicated by the uplifted RMSE and R2 values obtained for the stock market that has been examined. The level of accuracy that is obtained clearly suggests that this model could actually serve as a good tool for investors, traders, and financial analysts in making trust-worthy decisions in predicting the stock market, despite the results obtained, there are still areas that need improvement. There are certain significant factors that can influence stock market movements, like natural disasters, catastrophes, wars, geopolitical events, which have not been taken into account in the model's decision-making process. So, the model's predictions might not completely account for the impact of such unpredictable events on the future stock prices. One more aspect that requires further investigation is the model's generalizability to other stocks that could be predicted. It remains uncertain whether the same level of accuracy and effectiveness can be maintained when dealing with different kinds of stocks, different industries, or market conditions. To address these kinds of limitations and to enhance this model's performance, future research could be more on the use of ensemble models.

Future scope could be, Researchers might consider incorporating external factors and the real-time events into this model. By integration of relevant data on economic indicators, news sentiment, other external variables, geopolitical events, this model could have a possibility to gain a comprehensive understanding of the factors influencing stock prices. This will ultimately improve its predictive capabilities. In conclusion, while the LSTM-based model shows a valid promise for stock market prediction, there should be a continuation in efforts to refine and expand the approach. This is required to account for the external influences and improve its applicability and integrability to diverse market conditions. By addressing these kinds of challenges, the integration of ensemble models and external factors make it even more accurate and effective for the stock market predictions, benefiting investors and all the trade market professionals.

REFERENCES

[1] Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network by Alex Sherstinsky

[2] RNN and LSTM by Navin Kumar Manaswi

[3] Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU by Apeksha Shewalkar, Deepika Nyavanandi and Simone A. Ludwig

[4] Stock Market Price Prediction Using LSTM RNN by Kriti Pawar, Raj Srujan Jalem & Vivek Tiwari

[5] A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures In Special Collection by CogNet Yong Yu, Xiaosheng Si, Changhua Hu, Jianxun Zhang

[6] https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

[7] https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714

[8] https://medium.com/@divyanshu132/lstm-and-its-equations-5ee9246d04af