Predict stock prices using RNN and LSTM Models

Shivangi Bhave
A1908566
University of Adelaide

a1908566@adelaide.edu.au

Abstract

This paper explains the implementation of deep learning-based models precisely consider Recurrent Neural Network (RNN) and long short-term memory (LSTM) networks. The goal is to predict the stock prices using the past data. The data used for this experiment is Google Stock Price from Kaggle (https://www.kaggle.com/datasets/rahulsah06/google-stock-price). RNN and LSTM were both trained on the training set given withing this data and evaluating it on hidden data i.e. test set available with the data. The RNN model achieved better performance than LSTM, whereas LSTM showed the signs of overfitting, giving worse performance on test set. The evaluation is based on Mean Squared error (MSE) and metric Root mean squared error (RMSE) and Mean Absolute error. The implementation highlights the RNN models generalization capabilities and base LSTM model's overfitting if not tuned and the RNN being the better base model for real world application of stock predictions.

Introduction

Stock prediction is a very tedious task, it depends on various factors given the dynamic nature of stock market. This domain is one of the challenging when it comes to automating the stock predictions. Therefore, instead of Convolutional Neural networks the preferred architectures are those which remember the earlier values within the data. As this is the time series data, therefore the value at timestamp is dependent on past timestamp therefore the RNN and LSTM are preferred architecture. This project focuses on implementing these two models and analyzing whether they can become a good baseline for further development and more complex models.

Related Work

Techniques and various deep learning architectures have evolved, and their use has increased in various domains and one such domain is time series data where tasks like forecasting, stock price predictions come. The time series data is sequential where past values within the sequence drives the future value. The following are some of the works done in domain of time series data.

Recurrent Neural Network

Many studies have utilized deep learning techniques and concepts for stock price prediction, leveraging the sequential nature of stock data. Zhu (2020) proposed a research model employing a two-layer Recurrent Neural Network (RNN) for stock price prediction, using 10 years of Apple stock price data retrieved from Yahoo Finance (Zhu 2020, p. 4) .The focus of the experiment was to predict stock prices based on the previous 5–10 days of stock prices, demonstrating the RNN's ability to handle time-series data effectively (Zhu 2020, p. 5)The architecture consisted of 50 units in the first RNN layer and 100 units in the second (Zhu 2020, p. 5).The model achieved a loss close to 0.1% for predictions using smaller time steps (5 days), though its performance declined as the number of past days increased (10 days) (Zhu 2020, pp. 5-6). This highlights the RNN's strengths and limitations in stock price forecasting when handling varying time-step sizes.

Deep learning optimization

A recent research study conducted by Julian et al. (2023) explored the usage of finance technical indicators with deep learning concepts and techniques to predict the stock prices. The finance related technical indicators are moving averages and Relative strength index and moving average convergence Divergence (MACD). The implementation uses LSTM network to capture the temporal patterns of stock price data. The experiment also utilizes hyperparameter tuning techniques like dropout and learning rate tuning. The model achieved a very low MSE of about 68.49 on test dataset, which suggests that by utilizing technical indicators of stock data prediction capabilities an increase of the model (Julian et al. 2023, p. 7)

Predicting stock prices with transformers

Mozaffari and Zhang (2024) proposed a very novel implementation of transformers to predict the stock prices and mentions its comparison with LSTM models in the experiment. The dataset the researchers worked on is the stock prices of American Airlines Group Inc. and Atlantic American Life insurance. The transformer model in this experiment achieved a minimum MST of 0.0085 which is much better than the LSTM MSE which is 0.1972 (Mozaffari & Zhang, 2024, p. 4).

3. Method Description

3.1. Dataset

Google stock price data has two sets of data one is training data having total 1258 records whereas the test set has total 20 records. Each data has features like the date, open, close. high, low, close and volume of the google stock. The initial dataset does not possess the validation set therefore the preprocessing step involved creation of validation set as well as the dataset is unclean as features like volume, close price is string having comma so proper data conversion took place in preprocessing step for both the train and test set. The cleaned data was further scaled to bring all the price and quantity value between 0 to 1, by utilizing the min-max scaling.

3.2. Recurrent Neural Network (Base model)

The Recurrent Neural Network considers the last knowledge as they have feedback loops system which allow knowledge to carry forward to the next network thus maintaining memory (Colah, 2015). In our experiment we have developed a small RNN network which comprises of 3 layers of Simple RNN with each having 50 units and to handle the overfitting issue we introduce the dropout layers. The final layer is simple Dense layers whose task is to spit out the final value of stock open price which the model will detect after learning. The Adam Optimizer is utilized here and it's a regression type of problem therefore the loss function used here is mean squared error.

3.3. Long Short-Term Model

LSTM network is more advanced architecture which overcomes the issue with RNN where it maintains the long-term information instead of just the recent or last one. In our experiment the LSTM layers are used by default the activation function is tanh therefore no requirement of mentioning it while designing the architecture. In a similar way the drop out layer is used to control the overfitting by the model.

3.4. Optimized LSTM

In another experiment we tried to tune the very basic LSTM developed in earlier experiment by introducing methods like L2 regularization so as to minimize overfitting (Shevchenko, 2024) issue which was observed in the basic long-short term model earlier. Since the model showed signs of LSTM as it was performing exceptionally well on train and validation set but the performance that will be highlighted in analysis part performed very poorly on unseen data. This model also incorporated the method of bidirectional LSTM layer as the task is to predict the future price of stock which depends on earlier or past price and bidirectional layer is known for learning from both past and future values in sequence (Anishnama, 2023).This experiment also utilizes the power of altering learning rate by using the TensorFlow ReduceLRonPlateu to update learning rate if the metric in our case the validation loss stops showing improvement.

4. Analysis

4.1. Evaluation

The three models were evaluated using the metrics like mean squared error (MSE) and mean absolute error (MAE) and root mean squared error, which are one of the recommended metrics to evaluate the performance of models when the task is of predicting a value. Also, the table is designed to understand the deviation of predicted stock prices by each of the models from the actual value for the test stock data.

Model	Dataset	MSE	RMSE	MAE
RNN model	Train set	1897.84	43.56	39.77
	Validati on set	1924.04	43.86	39.61
	Test set	188.79	13.74	12.45
	Train set	368.49	19.20	14.42

LSTM model	Validation set	462.66	21.51	15.83
	Test set	17542.70	132.45	131.78
Optimized LSTM	Train set	182.52	13.51	9.42
	Validation set	271.40	16.47	11.13
	Test set	4104.93	64.07	62.76

Table 1. Evaluation of models on train, validation and test dataset

Below is the table where all the 3 models' predictions on unseen stock data and their actual prices are compared in this experiment. The last ten data is fed to all the three models and then next days predictions are predicted by these three models are done. The price is the opening price of the google stock is predicted.

Actual price	Predicted price
805.81	819.63
805.12	819.34
806.91	816.95
807.25	816.20
822.30	816.91
829.62	820.04
837.81	824.15
834.71	825.91
814.66	826.63
796.86	824.89

Table 2. Prediction vs actual price on test set by RNN model

Actual price	Predicted price
805.81	679.01
805.12	680.91
806.91	682.25
807.25	683.33
822.30	683.85
829.62	684.19
837.81	685.04
834.71	686.52
814.66	688.40
796.86	689.78

Table 3. Prediction vs actual price on test set by LSTM model

Actual price	Predicted price
805.81	749.47
805.12	749.95
806.91	750.37
807.25	750.43
822.30	751.03
829.62	752.76
837.81	755.29
834.71	757.40
814.66	758.95
796.86	757.75

Table 3. Prediction vs actual price on test set by Optimized LSTM model

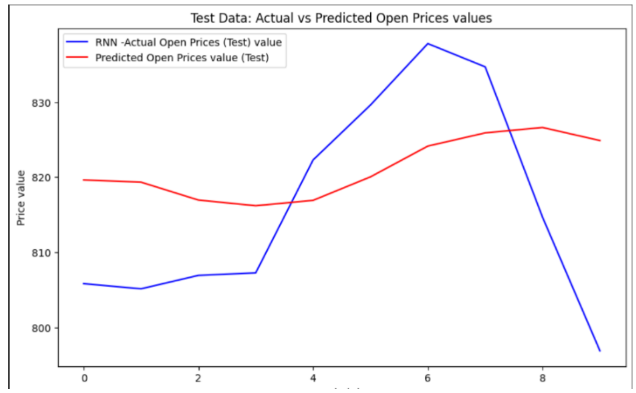

Fig 1. Test set performance by RNN model

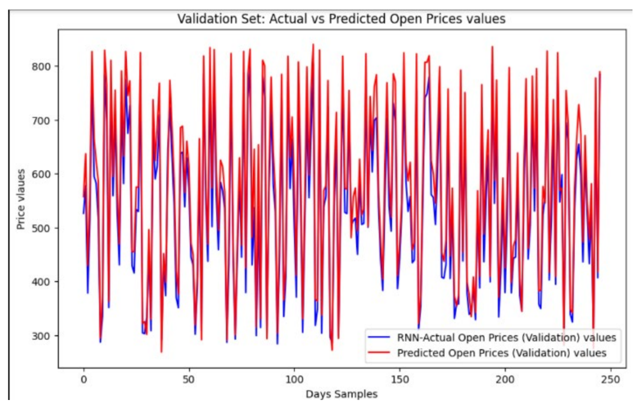

Fig 2. Validation set performance by RNN model

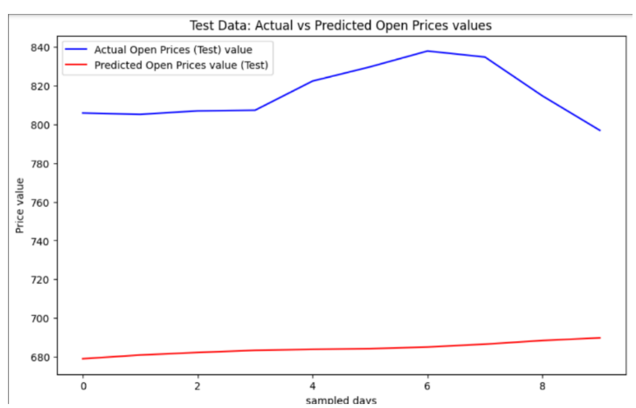

Fig 3. Test set performance by LSTM model

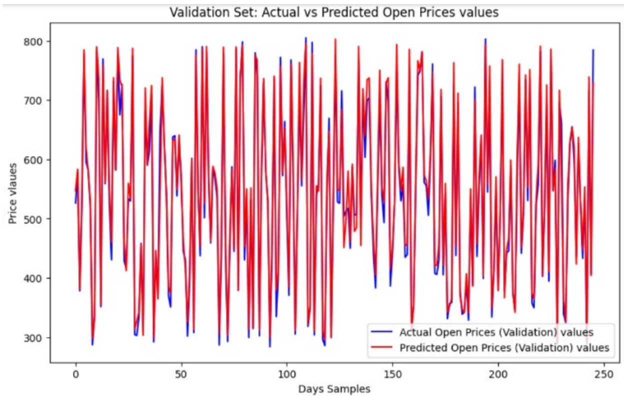

Fig 4. Validation set performance by LSTM model

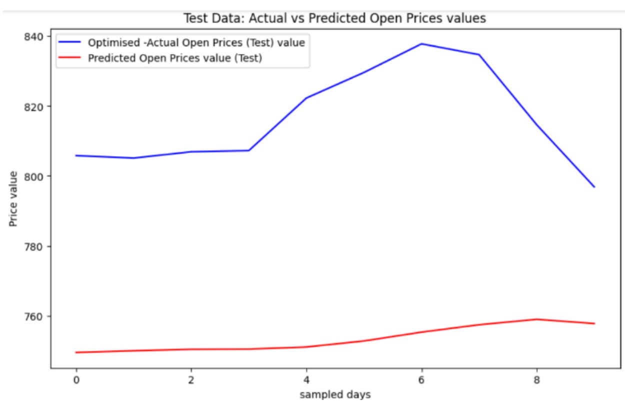

Fig 5. Test set performance by Optimized LSTM model

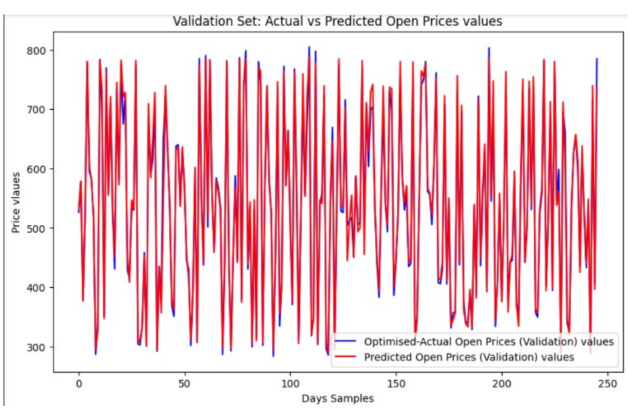

Fig 6. Validation set performance by Optimized LSTM model

4.1. Observation

While observing all the three models' performances on each of the train, validation and finally the test set is very visible through both the values written in table as well as the graphs that RNN model has outperformed even the basic optimized LSTM model which makes it a very candidate for a base model to further research in this domain of stock prediction. While the vanilla LSTM did not show any extraordinary performance on test set it made the path for the optimization which proved right as after introducing basic optimization in LSTM model the performance showed a good increase in its prediction also the MSE value of 17542.70 reduced to 4104.93 which is a tremendous decrease.

5. Reflection on project

5.1. Major Design options

Three deep learning models are taken into consideration which are basic RNN models model and an optimized version of LSTM model introducing bidirectional layer in it.The RNN model outperformed all the other two proving as a benchmark model for further research in this domain when it comes to timeseries data in our case stock prices, but this also suggests its importance in other time based task like forecasting of sales, revenue and many other temporal data.

5.2. Data processing steps

The raw data required a cleaning process as the features like volume and closing price column were string data. Then application of scaling where Min-max scaling is the opted as the features in consideration are values and mainly prices so to keep values in positive application of min-max scaling is preferred.

5.3. Tuning Hyperparameters

The major work in this experiment is to utilize the hyper parameter tuning techniques like early stopping to avoid unnecessary training when the model learning becomes stagnant, drop out layer to avoid overfitting is applied in each of these three models. In the optimized LSTM another method of hyper parameter is deployed where a method of learning rate scheduling and bidirectional layer is introduced in its architecture to make it generalize more.

5.4. Learning from research

The model complexity can introduce more overfitting if not proper optimization in consideration whereas very basic vanilla model can outperform such complex models and can be used as better benchmark, thus proving that the accuracy does not depend on the complexity of model architecture.

5.5. Future scope

The experiments done in this research give an ideal benchmark for the task of stock predictions, and in future the plan is to experiment with:

1. More financial indicators to predict the stock prices such as economic conditions, news sentiments.
2. Transfer learning
   The plan is to utilize the learning from large deep learning model for the task of stock predictions thus to improve performance in limited resources like GPU.
3. Experiment with more optimization techniques for LSTM model like batch normalization, deeper model
4. Real world implementation at this stage not a hundred percent possible but the RNN can be an ideal model to start with in actual world the data is very dynamic so to deploy the model we need good performing servers with latency as low as possible and utilize the real time streaming technical stack like Kafka, spark and Apache Flink.

6. Conclusion

This research experimented with neural architectures which considers the sequence data and carry forward the learning from last sequence thus for the task of stock value prediction fits perfectly well with this type of deep learning architectures. The research shows that the basic RNN model proves to be a very good benchmark model and with small optimization considered in LSTM model the model performance increased exponentially on hidden unseen data.

7. Github code

Link to GitHub repo for the code of this project
https://github.com/sbhaveuni/assignment3

References

Anishnama, A 2023, *Understanding Bidirectional LSTM for Sequential Data Processing*, Medium, viewed 1 December 2024, https://medium.com/@anishnama20/understanding-bidirectional-lstm-for-sequential-data-processing-b83d6283befc.

Colah, C 2015, *Understanding LSTM Networks*, Colah's blog, viewed 1 December 2024, https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Julian, T, Devrison, T, Anora, V & Suryaningrum, KM 2023, 'Stock price prediction model using deep learning optimization based on technical analysis indicators', *Procedia Computer Science*, vol. 227, pp. 939–947, viewed 6 December 2024, https://www.sciencedirect.com/science/article/pii/S1877050923017684.

Mozaffari, L & Zhang, J 2024, ‘Predictive modeling of stock prices using a transformer model’, Proceedings of the 2024 9th International Conference on Machine Learning Technologies (ICMLT), ACM Digital Library, Oslo, Norway, May 2024, viewed 5 December 2024, https://doi.org/10.1145/3674029.3674037.

Shevchenko, E 2024, *Regularization Methods for Recurrent Networks*, Medium, viewed 1 December 2024, https://medium.com/@eugenesh4work/regularization-methods-for-recurrent-networks-215e0147d922.

Zhu, Y 2020, 'Stock price prediction using the RNN model', *Journal of Physics: Conference Series*, vol. 1650, no. 3, pp. 1-7, viewed 5 December 2024, https://doi.org/10.1088/1742-6596/1650/3/032103.