Computer Engineering Department

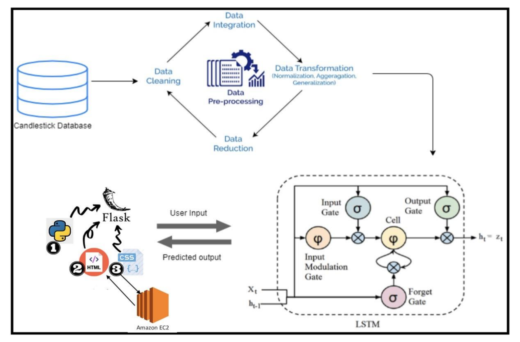
Stock prediction using neural networks

Project Advisor: Gokay Saldamli

Konatham, Mahesh Reddy (MS Software Engineering)
Hagalahalli Srinivas, Shreya (MS Computer Engineering)
Habibi, Mohdi (MS Software Engineering)
Sneha (MS Software Engineering)

Introduction

Stock market historically has been very volatile with many factors having direct impact on the fluctuating nature of it. Many stock trading techniques involve technical analysis and fundamental analysis of each individual company's stock to understand the future of its value. Not only does a company's health and earnings have weight in deriving its stock value, but also many other external factors such as sentiment of the market and public reaction toward events have direct effect on the volatility.

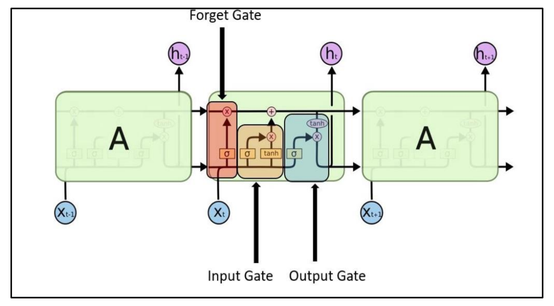


This research focuses on the previous movement of the stock market to predict the near future volatility without taking into consideration other external factors. The idea behind this approach is how to derive certain knowledge from stock market time series data using deep learning to understand near future positive or negative volatility. The automation of such a process will allow anyone to make quick decisions for making profit from the stock market volatility. Since LSTMs are specialized in keeping the historical context of inputs, it makes it a viable option for stock market prediction.

Methodology

Long Short-Term Memory (LSTM)

LSTM Neural Networks are a type of Recurrent Neural Networks that are designed to specifically handle long term dependencies. The RNN module is a simple design having a single tanh layer. LSTMs consist of four layers of tanh where each layer communicates in a unique way. The LSTM can remove or add information to a cell state through a structure called gates.



The architecture of LSTM neural network is as shown above. It consists of three gates - input gate, forget gate, and output gate. The design has 4 layers among which three of them are sigmoid layers connected to the three gates and the fourth layer is the tanh layer. Forget gate layer -The first step in the LSTM model is to remove the information in the cell state c(t-1) with the help of sigmoid layer.

Analysis and Results

Input gate layer – The next step in the model is to evaluate what new data in the cell state will be processed. The sigmoid layer is given x(t) and h(t-1) as input. The candidate values that can be added to the cell state is determined by the tanh layer.

Output layer – The final step is to determine the cell state output information. Sigmoid layer is part of this decision. Sigmoid layer will be used to set the information to be sent as output by taking x(t) and h(t-1) as input. This data is multiplied to only relay this part of the information as the output by the tanh layer. The formula is shown below to determine the output value.

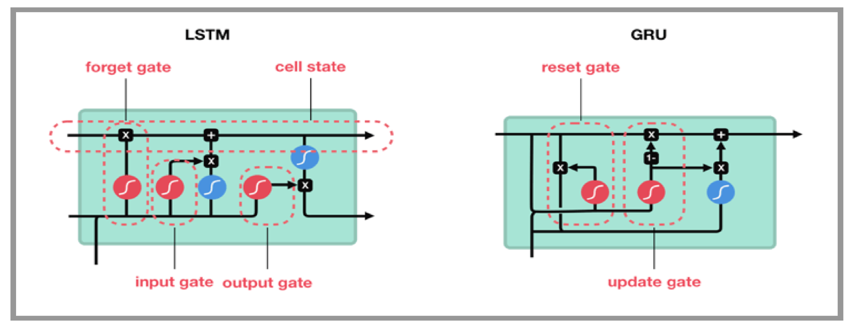
Date	High	Low	Open	Close	Volume	Adj Close	Ticker
2000-01-03	1.0044642686843900	0.9079241156578060	0.9363839030265810	0.9994419813156130	535796800.0	0.8551679253578190	AAPL
2000-01-04	0.9877232313156130	0.9034598469734190	0.9665178656578060	0.9151785969734190	512377600.0	0.7830682992935180	AAPL
2000-01-05	0.9871651530265810	0.9196428656578060	0.9263392686843870	0.9285714030265810	778321600.0	0.7945277094841000	AAPL
2000-01-06	0.9553571343421940	0.8482142686843870	0.9475446343421940	0.8482142686843870	767972800.0	0.7257705330848690	AAPL
2000-01-07	0.9017857313156130	0.8526785969734190	0.8616071343421940	0.8883928656578060	460734400.0	0.7601492404937740	AAPL
2000-01-10	0.9129464030265810	0.8459821343421940	0.9107142686843870	0.8727678656578060	505064000.0	0.74677973985672	AAPL
2000-01-11	0.8872767686843870	0.8080357313156130	0.8565848469734190	0.828125	441548800.0	0.708581268787384	AAPL
2000-01-12	0.8526785969734190	0.7723214030265810	0.8482142686843870	0.7784598469734190	976068800.0	0.6660856008529660	AAPL
2000-01-13	0.8816964030265810	0.8258928656578060	0.8436104655265810	0.8638392686843870	1032684800.0	0.7391401529312130	AAPL
2000-01-14	0.9129464030265810	0.8872767686843870	0.8928571343421940	0.8967633843421940	390376000.0	0.7673114538192750	AAPL
2000-01-18	0.9464285969734190	0.8967633843421940	0.9017857313156130	0.9280133843421940	459177600.0	0.7940503358840940	AAPL
2000-01-19	0.9709821343421940	0.9229910969734190	0.9430803656578060	0.9514508843421940	597643200.0	0.8141044378280640	AAPL
2000-01-20	1.0848214626312300	1.0133928060531600	1.03125	1.0133928060531600	1831132800.0	0.8671048879623410	AAPL
2000-01-21	1.0200892686843900	0.9838169813156130	1.0200892686843900	0.9938616156578060	495924800.0	0.8503931760787960	AAPL
2000-01-24	1.0066964626312300	0.9386160969734190	0.9681919813156130	0.9486607313156130	440876800.0	0.81171715259552	AAPL
2000-01-25	1.0100446939468400	0.9140625	0.9375	1.0022321939468400	497145600.0	0.8575553894042970	AAPL

The LSTM model has the ability to distinguish between recent and early samples via this architecture by assigning them various weights. This helps to boost stock price prediction results. The input sample here is considered to be independent by the LSTM model. In stock price prediction, this serves as an advantage. The current stock is linked to previous stock prices, so the model is capable of capturing this dependence to provide a better outcome.

Gated Recurrent Unit (GRU)

Gated Recurrent Unit is the improvised version of Recurrent Neural Network and has a lot in common with LSTM. While LSTM used cell state to maintain internal memory, GRU uses the hidden state to transfer information. It has two gates, a reset gate and an update gate.

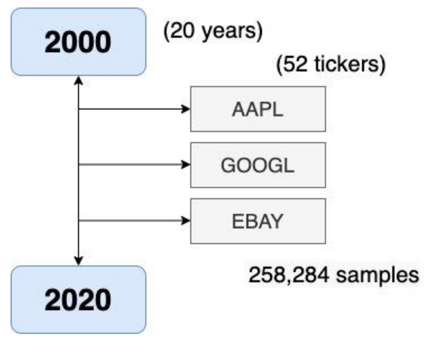
GRU is computationally more efficient considering fewer parameters and need less data to generalize. However, GRU does not have any mechanism to control the degree to which its state or memory content is exposed but exposes the whole state or memory content each time.

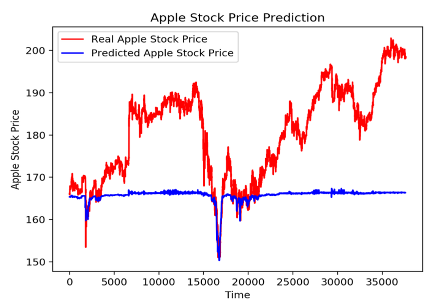


$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

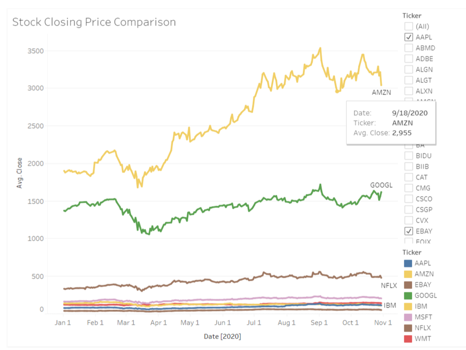
Since the stock market data is spread into different directories. Data processing involves combining the time series data together. The combined data is then cleaned to handling any missing values in the data set, handling any duplicate rows, to ensure right data format of the attributes .

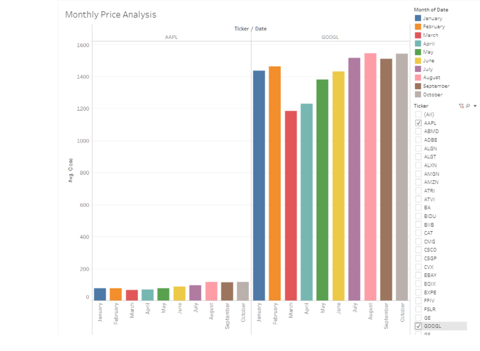
The processed dataset is divided into 75% training set and 25% test to be applied for the LSTM model. The visualization of the training set and test set for AAPL stock .





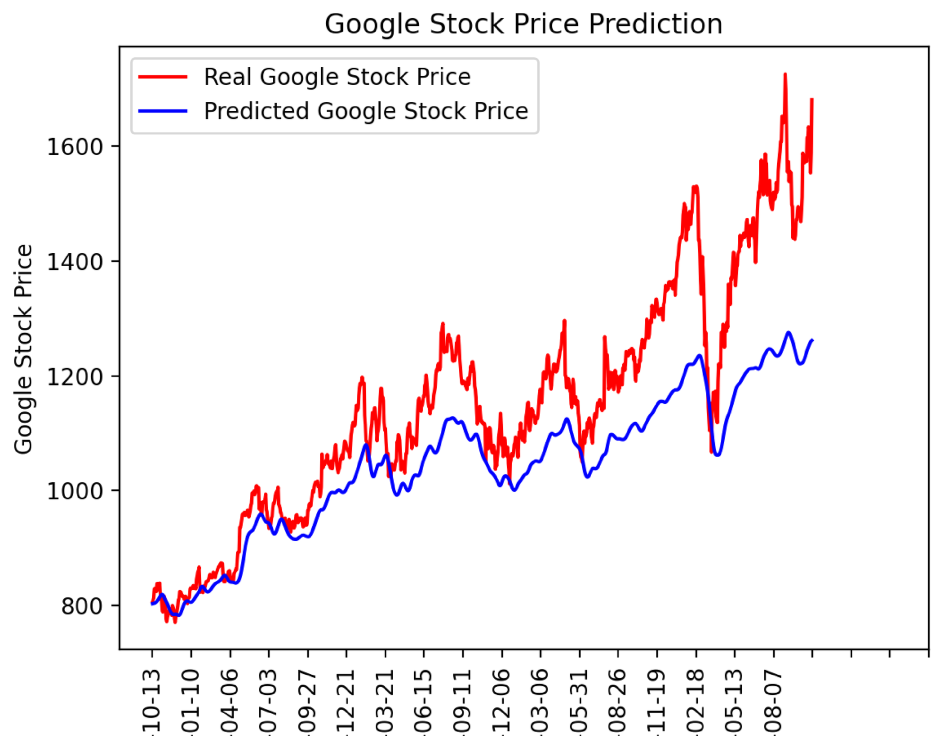
With Minimizing MSE, one can observe that the performance of the model increases. Usually in the training step, the MSE starts to be a higher number but through different epochs, there is a significant reduction in the error rate.





The processed data is given as input to the LSTM model. The LSTM neural network currently consists of four layers and a 20 % dropout is attained at each layer to prevent any overfitting. Utilized Adam optimizer with mean square error as loss function. The above figure shows the preliminary results obtained. As we can see the model is able to predict the negative movements in stock price clearly. This model will further be fine-tuned to be able to get the predictions for positive movements in stock price.

Mean Squared Error (MSE)is used is to calculate the square root of the mean of actual values minus predicted values divided by the number of observations. Mathematically it is defined by the following formula:



After data preprocessing for correlation analysis, it was clear that there was a significant impact of trend on the stock prices. With the Apple Stock price analysis the model was only able to trace the negative veracity in the stock prices. But after the preprocessing and hyper parameter tuning, the model was better at tracking the trend. Both the uptrend and downtrend were very closely matched by the model's predictions.

The model was then integrated with a web application running on a flask server deployed on AWS EC2 for remote availability.

Summary/Conclusions

The model currently tracks the real stock prices very closely; but there is still scope for improvement. The parameters can still be tuned for better predictions. Also, the latest developments in the field of deep learning for time series like Transformers can be used to obtain better results. The project uses static data. This can be improved to make predictions dynamically the streaming data.

Key References

[1] Kim, Kyoung-Jae, and Won Lee. "Stock Market Prediction Using Artificial Neural Networks with Optimal Feature Transformation." *Neural Computing & Applications* 13.3 (2004): 255-60.

[2] Tilakaratne, C., M. Mammadov, and S. Morris. "Modified Neural Network Algorithms for Predicting Trading Signals of Stock Market Indices." *Journal of Applied Mathematics & Decision Sciences* 2009 (2009): 1-22.

[3] L. Yu, H. Chen, S. Wang and K. K. Lai, "Evolving Least Squares Support Vector Machines for Stock Market Trend Mining," in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 87-102, Feb. 2009

[4] . Kwon and B. Moon, "A Hybrid Neurogenetic Approach for Stock Forecasting," in *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 851-864, May 2007.

Acknowledgements

We are very grateful to our project advisor Professor Gokay Saldamli for his invaluable assistance and guidance in our research.

We are also very thankful to Computer Engineering Department for providing us the facilities to conduct our research.