Recurrent Neural Networks and Stock Price Predicition

Cecilia Tipping
CS5393
Sotuhern Methodist University

***Abstract*—This report explores the application of two recurrent neural network architectures, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), for stock price prediction. Focussing on assessing how each model performs using historical stock data, and how additional input features such as Volume and OHLC (Open, High, Low, Close) prices influence results. Furthermore, various activation functions were tested within the GRU architecture to determine their effect on performance. Through these experiments, the project highlights how model design choices and input complexity impact predictive accuracy and training efficiency.**

I. Overview

Stock price prediction presents a significant challenge, driven by the complexity and volatility inherent to financial markets. Sequential models like LSTM and GRU are well-suited for this task due to their ability to capture temporal dependencies in time series data. This project investigated how these models handle stock price prediction, particularly when presented with increasingly rich datasets.

Initial experiments focused on predicting future prices using only historical Close prices. The scope was then expanded to include Volume and OHLC data to determine whether additional context would improve model performance. Additionally, alternative activation functions were tested in GRU models to explore how activation choices affect learning dynamics and outcomes.

II. Methods

A. Dataset and Preprocessing

Historical stock data for Apple Inc. (AAPL) was obtained via the yfinance API, covering the period from 2010 to late 2023. Data was normalized using MinMaxScaler to ensure stable training and to prevent issues arising from differing scales between features.

A custom sequence generator was implemented to create overlapping 60-day input sequences. The target variable for all experiments was the next day's Close price.

B. Model Architecture

Both LSTM and GRU models shared a consistent structure:

- Two recurrent layers (either LSTM or GRU) with 50 units each
- One dense layer with 25 units
- One final dense layer to output the predicted price

In later experiments, the GRU model was modified to test various activation functions, including ReLU, LeakyReLU, and ELU.

C. Feature Expansion

In addition to the baseline Close price input, Volume was introduced as a secondary feature, followed by Open, High, Low, Close, and Volume (OHLC + Volume) to form the most comprehensive feature set tested.

III. Results and Analysis

A. Close Price Only

In the simplest test using only Close price data, the two models behaved very differently. LSTM, which typically excels at modeling complex patterns, struggled to converge effectively in this low-information setting. The model output almost entirely flat predictions, failing to follow the underlying price movements in any meaningful way.

GRU, on the other hand, adapted better despite the limited input. While not highly accurate, it still produced predictions that tracked the general trend of the stock prices. This contrast highlighted the flexibility of GRU, particularly when working with simpler datasets where intricate long-term dependencies are not as important.

Observations:

- LSTM underfit the Close-only dataset, producing constant or near-constant predictions.
- GRU produced more meaningful outputs, demonstrating its advantage in handling small and less complex input spaces.
- This stage reinforced the importance of providing adequate contextual information for models like LSTM to perform well.

B. Close + Volume

Introducing Volume data into the models improved performance across the board. GRU, in particular, benefited from the additional input, reducing the performance gap with LSTM considerably.

Observations:

The inclusion of Volume provided useful market context, especially during periods of volatility. GRU adapted well, demonstrating that additional features could enhance its efficiency without adding significant complexity.

C. OHLC + Volume

With OHLC and Volume data included, GRU surpassed LSTM in predictive accuracy. The simpler gating mechanism and faster training process appeared to make GRU more effective in handling the expanded input space.

Observations:

This marked a turning point. GRU's ability to efficiently integrate multiple features became evident. In contrast, LSTM struggled slightly to leverage the richer dataset.

D. GRU Activation Function Experiments

The activation function experiments revealed clear differences in how GRU models responded to the task.

ReLU followed the overall price trends but consistently underpredicted, leaving a noticeable gap between actual and predicted values.

LeakyReLU provided modest improvements, tracking more closely than ReLU but still falling short.

ELU, however, produced the strongest results, with predictions that aligned closely with actual stock prices and minimized error.

While the differences were not drastic, ELU demonstrated superior performance and proved to be the most reliable activation function tested in this experiment.

IV. Discussion

The experiments demonstrated that while LSTM initially held an advantage in scenarios with limited input features, GRU proved more adaptable as complexity increased. When Volume and OHLC data were incorporated, GRU's efficiency and responsiveness enabled it to outperform LSTM both in terms of RMSE and training speed.

When comparing the architectures, GRU proved to be more adaptable. While LSTM struggled in the Close-only dataset and failed to converge meaningfully, GRU provided usable predictions even in this limited scenario. As features were added, both models improved, but GRU consistently handled complexity and limited data with greater flexibility.

Activation function experimentation within the GRU model revealed that minor architectural adjustments can have a measurable impact on performance. LeakyReLU emerged as the preferred choice, combining stability with improved learning dynamics.

Overall, GRU paired with ELU and multi-feature input (OHLC + Volume) represented the most effective configuration.

V. Challenges

Several challenges emerged during this project. Expanding feature sets required careful management of sequence shapes and data dimensions, which led to occasional errors during model training. Additionally, activation function tuning introduced instability in certain cases, particularly with ReLU. Addressing these issues provided valuable insights into model design and the importance of rigorous debugging and validation.

VI. Conclusion

This project highlighted the strengths and trade-offs associated with LSTM and GRU models for stock price prediction. While LSTM performed well with simple datasets, GRU demonstrated superior flexibility and efficiency when presented with richer feature sets. Adjusting activation functions further optimized GRU's performance, with LeakyReLU offering the best balance between responsiveness and stability.

Future work could expand on these findings by testing across additional stocks, integrating external market indicators, or exploring newer architectures such as Transformers for time series prediction. This study reinforced the value of thoughtful model design and experimentation in achieving meaningful predictive performance.

References

[1] YC-Coder-Chen. "GRU Stock Price Prediction." GitHub. https://github.com/YC-Coder-Chen/GRU-stock-price-prediction .

[2] Alameen, A., and A. Alsayat. "Stock Price Prediction Using LSTM and GRU Models." International Journal of Computer Science and Network Security (IJCSNS), vol. 19, no. 1, 2019. http://paper.ijcsns.org/07_book/201901/20190126.pdf .

[3] Sathyabama Institute of Science and Technology. "Recurrent Neural Networks and their Applications in NLP." https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/1922-b.sc-cs-batchno-24.pdf .

[4] [4] AiHints. "Stock Price Prediction Using LSTM — Tutorial with Code." YouTube. https://www.youtube.com/watch?v=WcJ6Ojfvla4 .

[5] Sunscrapers. "Deep Learning for NLP — An Overview." https://sunscrapers.com/blog/deep-learning-for-nlp-an-overview/.

[6] [6] LMU Munich. "Recurrent Neural Networks and their Applications in NLP." https://slds-lmu.github.io/seminar_nlp_ss20/recurrent-neural-networks-and-their-applications-in-nlp.html .

[7] IBM. "What is a Recurrent Neural Network (RNN)?" https://www.ibm.com/topics/recurrent-neural-networks.

[8] NVIDIA. "Recurrent Neural Network (RNN)." https://developer.nvidia.com/discover/recurrent-neural-network