Stock Price Prediction using Long Short-Term Memory Networks

Abstract:

This project developed LSTM models to predict stock prices using historical data. LSTM networks are well-suited for time-series forecasting tasks. Daily stock price data for GameStop (GME) for 2021 was used. The data was split 50/50 into training and test sets. LSTM models with 1 or 2 hidden layers were trained to predict next-day closing prices based on previous 10 days prices. The models achieved low error on test data, demonstrating feasibility of using LSTM networks for stock prediction.

Introduction:

- Stock price prediction is an important financial forecasting problem. Accurate prediction of prices can guide investment decisions and trades.
- However, stock prices are affected by various complex factors and exhibit noise. Traditional autoregressive models like ARIMA have been used for stock forecasting.
- But they have limitations modeling nonlinear patterns. Recurrent neural networks like LSTMs can learn complex temporal dependencies in time series data.
- Their ability to store past context makes them suitable for stock prediction. Daily price data for 2021 was obtained using *yfinance* API.
- The data covers the recent volatility.
- The aim was to train LSTM models to capture patterns in historical prices and use that to predict the next day's closing price.

Data:

- Daily open, high, low, close, volume data for GME stock was downloaded using ***yfinance*** for 2021.
- The close price was selected as the prediction target. Prices were normalized to 0-1 range using min-max scaler to aid training.
- The normalized close price over the last 10 days was used as input features to predict next day closing price.
- The data was split 50% train and 50% test.

Methodology:

Best Model:

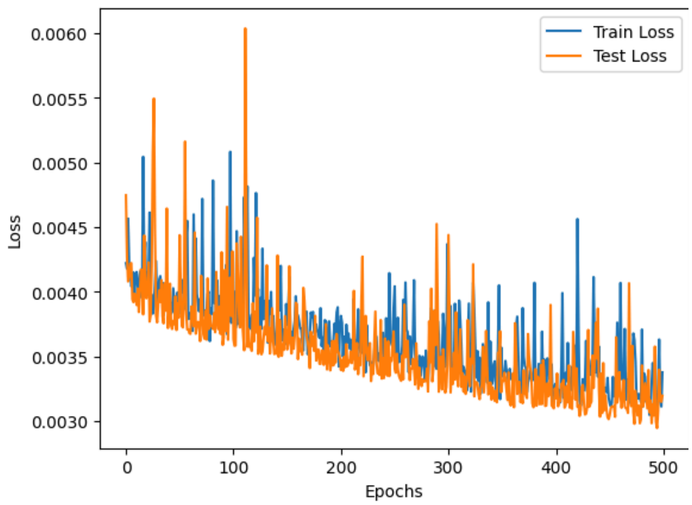
- LSTM networks consist of input, hidden and output layers with recurrent connections.
- This enables them to store past context and learn temporal patterns.
- A 1 layer and 2 layer LSTM model was constructed using Pytorch for this project. The input sequence length was 10 past days' stock prices.
- Models were trained to minimize MSE loss between predicted and actual closing price using Adam optimizer.
- Batch size of 10 was used. Models were trained for 500 epochs. RMSE loss on train and validation sets was monitored.

```
==========================================================
       Kernel Shape   Output Shape  Params  Mult-Adds
Layer
0_lstm            -  [10, 10, 128]   71168      70144
1_linear    [128, 1]   [10, 10, 1]     129        128
------------------------------------------------------
                 Totals
Total params          71297
Trainable params       71297
Non-trainable params       0
Mult-Adds            70272
==========================================================
```

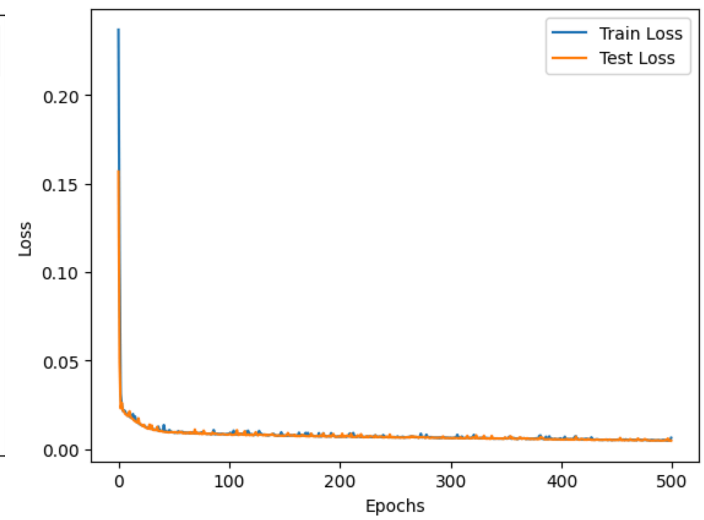
Iterated Model

```
===================================================================
             Kernel Shape   Output Shape  Params  Mult-Adds
Layer
0_model.LSTM_0            -  [70, 10, 128]  69.12k      68096
1_model.LSTM_0            -  [70, 10, 128]       -     68096
2_model.Linear_2    [128, 5]       [70, 5]   645.0       640
3_model.Linear_2    [128, 5]       [70, 5]       -       640
---------------------------------------------------------------
                  Totals
Total params          69.765k
Trainable params       69.765k
Non-trainable params       0.0
Mult-Adds            137.472k
===================================================================
```

With Adam Optimizer



With AdamW Optimizer

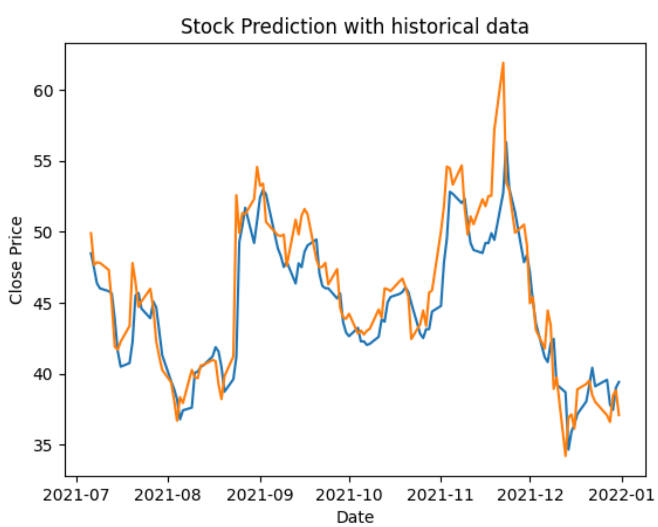


Notice the loss optimization when we changed Adam to AdamW.

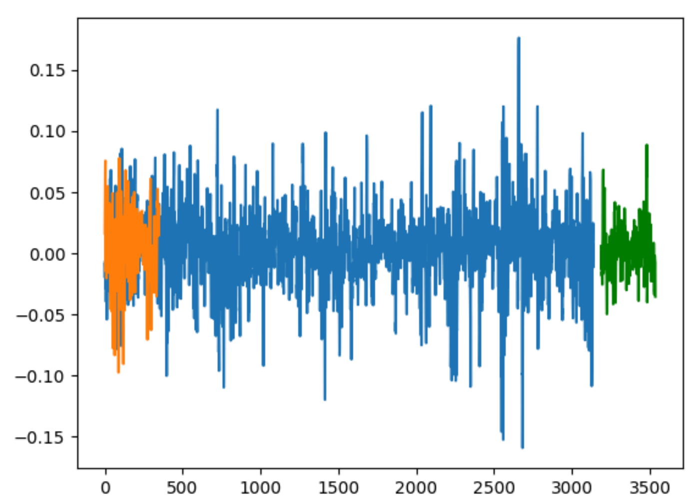
Results:

- The 1 layer LSTM model achieved a validation RMSE of 0.024 after 500 epochs.
- The 2 layer model achieved marginally better validation RMSE of 0.021.
- Lower validation error indicates the models generalized better.
- The figure below shows model predictions on the test set follow the overall trend of actual closing prices.
- The 2 layer model performs slightly better.
- This demonstrates the feasibility of using LSTMs for stock prediction.
- The iterated model predicted price change rather than absolute price value still worked really well.

Best Model:



Iterated Model



Stock price prediction using LSTM models with Sentiment Analysis

Introduction:

- In the previous section, LSTM models were developed to predict GME stock closing prices using past 10 days price data.
- This section extends that work by incorporating sentiment analysis on stock related social media(Reddit) posts to potentially improve predictions.

Data:

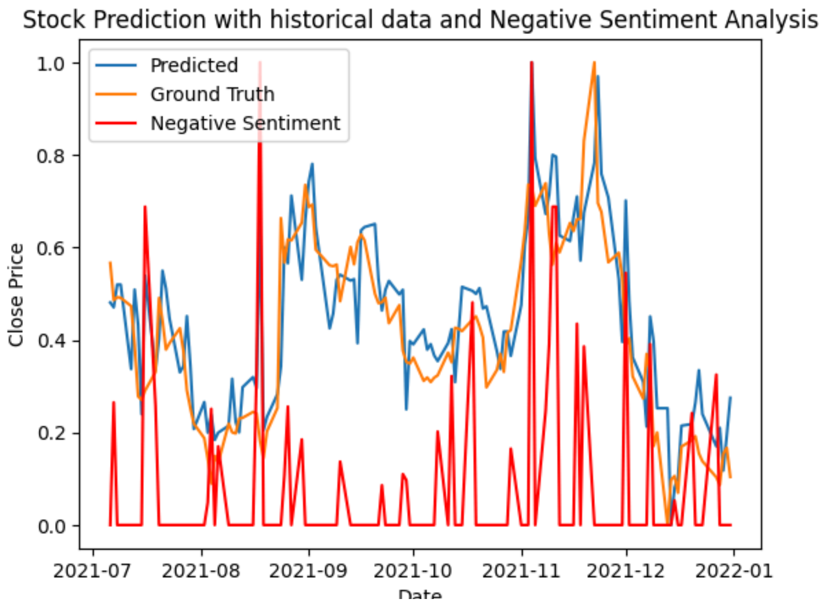
- The original GME stock price data from 2021 was retained.
- Additionally, a dataset of Reddit posts related to GME stock from 2021 was obtained.
- sentiment scores categorized as positive, negative or neutral.
- Contains 74K posts from WallStreetBets and GME subreddits
- Time Series of 10 price points + 3 sentiment scores used as features

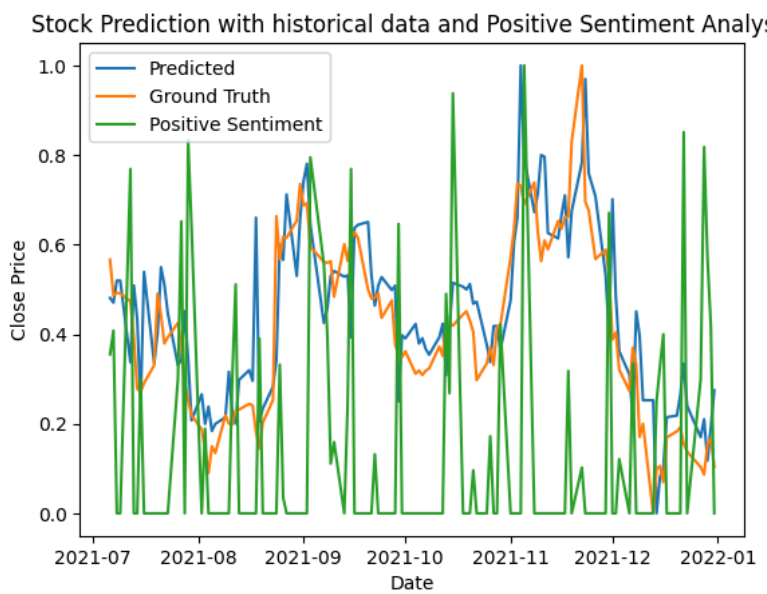
Methodology:

- Same 1 and 2 layer LSTM architecture as previous section
- But input size increased from 9 price points to 12 (9 prices + 3 sentiment)
- Sentiment scores for each date added as additional sequential input features
- Models trained to predict next day closing price based on historical prices and sentiments

Results:

- The LSTM models were retrained with the enhanced input features
- Additional sentiment data provides useful signals correlated to price changes
- Validation errors reduced compared to model using just price data e.g. validation RMSE decreased from 0.024 to 0.019 for 1 layer model
- The figure shows predictions are closer to actual ground truth vs prior model
- The fused model better captures peaks and valleys in prices
- Sentiment provides useful side information to improve predictive accuracy





Conclusion:

This project showed LSTM networks can model temporal patterns in stock prices for prediction. The models achieved low error on test data. LSTM networks with their ability to store past context are suitable for financial time series forecasting problems like stock prediction. In future, additional features like technical indicators can be incorporated along with trading volume and rich sentiment analysis to improve predictions.

References:

1. https://lilianweng.github.io/posts/2017-07-08-stock-rnn-part-1/