Stock Price Volatility Prediction with Long Short-Term Memory Neural Networks

Jason Sullivan
Department of Computer Science
Stanford University
jsull@stanford.edu

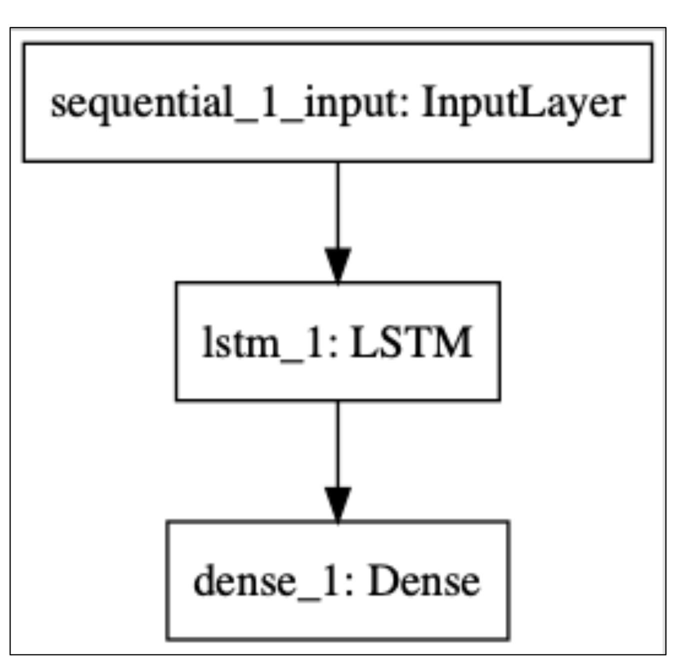
Motivation

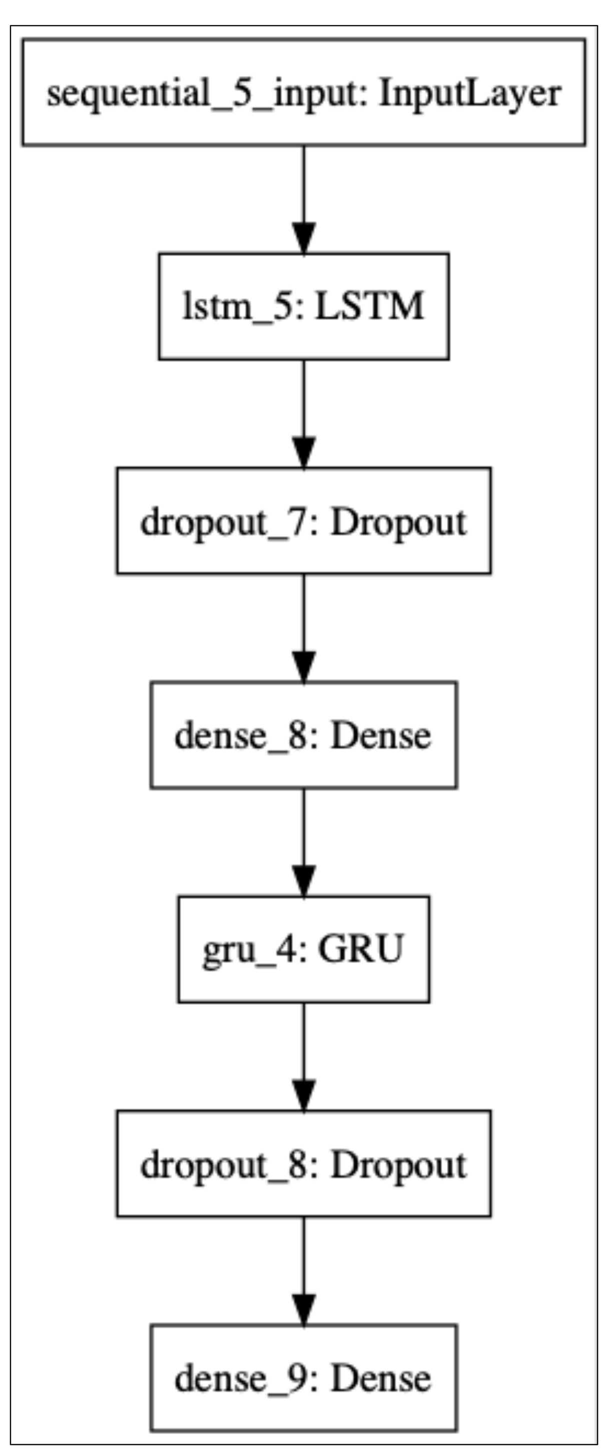
Stock price prediction in the financial markets is one of the most interesting open problems drawing new Computer Science graduates. A successful price prediction model can be highly profitable in trading equities in the public markets. A similarly interesting and profitable problem is that of volatility prediction. In predicting the volatility of a given stock, a trader can make bets or provide liquidity in the options markets. In this study, we employ a variation of a type of Recurrent Neural Network called Long-Short Term Memory (LSTM) [1] in order to predict stock price volatility in the US equity market.

Models

Our baseline model consists of a single LSTM layer with 5 hidden units, followed by a 1-unit output layer. We train this neural network with a batch size of 30 data points. Our chosen loss metric is Mean Squared Error, but we also track Mean Absolute Error and Mean Absolute Percentage Error. We use the Adam optimizer.

Our extended model is depicted below. It consists of a 20-unit LSTM layer, a 10-unit Dense layer with ReLu activation, and a 10-unit GRU layer. We also include 2 separate 10% Dropout layers.





Dataset

Our data source is from Kaggle (labeled "Huge Stock Market Dataset") [2] and provides over 18 years of daily Open, High, Low, Close, Volume, and Open Interest data for individual US stocks and ETFs. Lengths of time series vary by instrument based on when that instrument started trading in the market, and whether it still trades (or was acquired, delisted, etc.). The full dataset is 826MB, consisting of text files representing time series for over 8300 entities, including over 7000 single stocks and over 1300 ETFs—in this study, we focus on a universe of the 1000 largest and most liquid single stocks available in the dataset.

Features

Our features consist of several transformations of the given Open, High, Low, Close, Volume data, from which we extract Log Return, Log Volume Change, Log Trading Range (high vs. low for a given trading day), Previous 30-day Volatility, Previous 10-Day Volatility, and GARCH forward-looking 10-day volatility prediction as our features.

Financial time series is inherently non-stationary, and the common method of addressing this non-stationarity is to not use raw price series as inputs, but rather the series of price *changes* or percentage returns, and also to scale the data by taking the logarithm each element of the time series. For the GARCH model, we train a new model for each separate stock with parameters p=15 and q=15, with a horizon of 10 trading days [3].

For our target variable, we use the average volatility of the 10 days following a given data point, the motivation being that predicting a 1-day volatility doesn't make much of a difference and will be too noisy to train effectively, and predicting a 30-day volatility is not nearly as useful with a shorter-term trading horizon as part of one's trading strategy.

Results

The results are presented in the table below. The training and test sets contained 948,456 and 237,980 data points, respectively, which were input in batches of 30.

Training Set	MSE	MAE	MAPE
Baseline	4172.61	2.339	1975190.75
Extended	3883.5	2.314	8526497.0

Test Set	MSE	MAE	MAPE
Baseline	5167.51	2.459	8722592.0
Extended	4859.47	2.471	37653732.0

Discussion

The extended network outperforms during training, but the simpler network outperforms during testing. Notably, the extended network also has a greater degree of performance degradation in MSE and MAE in the test set relative to the training set. This suggests a greater degree of overfitting within the extended network. It also suggests that a greater deal of tuning is required on the deeper network in terms of regularization. In particular, the increased use of dropout layers could help reduce the variance problem. It would also be worth experimenting with early stopping to reduce variance.

Future Work

There are several axes on which this work could be extended including using a different financial data frequency, choosing a different asset class, incorporating a wider set of features including news sentiment, and further deepening and tuning the neural network architecture. It might also be more interesting and real-world-applicable to frame this as a classification problem rather than a regression problem, where the classes represent the presence of a price move past some statistical threshold (i.e. 3-stdv move) up or down, vs. neutral.

References

[1] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.
[2] https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs
[3] Bollerslev, Tim, 1986, Generalized autoregressive conditional heteroskedasticity, Journal of Econometrics 31, 307–327.