Predicting stock prices with LSTM Network

David Duemig ▪ dduemig@stanford.edu

https://www.youtube.com/watch?v=QQE_Yw5UUyc

CS 230 Poster Session, Deep Learning, Winter 2019

Introduction

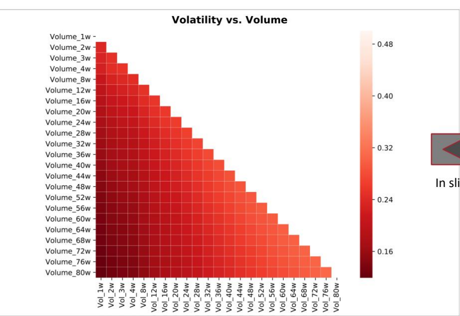
- The main goal of this project was to investigate whether modern DL techniques can be utilized to more efficiently predict the movements of the stock market
- Specifically, we applied an LSTM recurrent neural network to predict whether the S&P 500 will increase or decrease over the next trading month
- Stock price forecasting remains a difficult task even after simplifying the problem to a binary price trend and applying powerful deep learning models
- Nonetheless, we provide information which is useful in guiding future work, specifically in determining relevant feature sets and model architectures

Data

- The dataset provides daily quotes for the S&P 500 going back to 1980 and is downloaded using the Yahoo Finance API
- It comes with the "Open", "High", "Low", "Close", "Adj Close", and "Volume" for each day
- In total, the raw dataset consists of about 10,000 rows and 6 columns

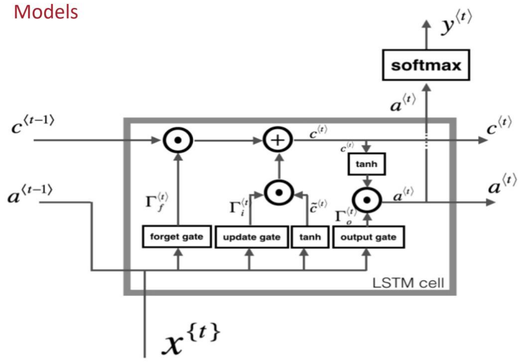
Features

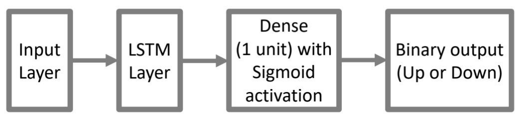
- All features are derived and include rolling (i) logarithmic returns, (ii) trading volumes, and (iii) volatilities for the past $i \in \{1, 2, 3, 4k \mid 1 \leq k \leq 20\}$ trading week(s)
- This results in a total of 69 features
- By combining these features, 7 feature sets were created
- The input features have been shown to have widespread predictive power



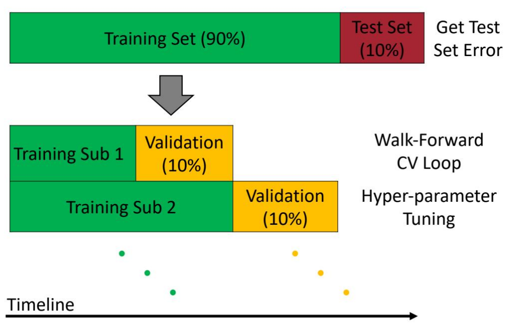
In slideshow...

Models





- Due to the nature of our features (returns, volatilities, and trading volumes of the past week(s), we use a **stateful** LSTM architecture with a batch size and a time step of 1
- By combining the feature sets, 7 models were created
- To make our models more robust, we use walk-forward-cross-validation with a dev size of 10% for our model development
- In order to provide an unbiased evaluation of our final model we withhold the last 10% of our data as a separate test set



Results

Model	Description	Dev/Test Acc. LSTM	Dev/Test Acc. Baseline	Test AUC LSTM	Test AUC Baseline
1	Feature Set 1	61.02%/ 64.38%	54.48% / 65.01%	0.52	0.52
2	Feature Set 2	55.62% / 65.01%	55.44% / 66.07%	0.52	0.54
3	Feature Set 3	56.49% / 65.12%	55.21% / 63.95%	0.54	0.48
4	Feature Sets 1 + 2	54.42% / 63.32%	52.06% / 60.04%	0.52	0.54
5	Feature Sets 1 + 3	59.34% / 65.96%	55.52% / 59.09%	0.46	0.53
6	Feature Sets 2 + 3	49.56% / 63.85%	54.19% / 65.54%	0.55	0.55
7	Feature Sets 1 + 2 + 3	53.13% / 65.54%	52.28% / 66.17%	0.53	0.50

Discussion

- From a trading perspective, we want to maximize accuracy
- However, there was a significant imbalance both in the training (62% "Up`s") and the test data (64% Up`s)
- Therefore the AUC score is a better choice for our model performance as it is insensitive to class imbalance
- All 7 models have a test AUC score close to 0.5, which is not a satisfactory result

Possible reason:

- ➢ Features do not contain enough information
- ➢ Input data is noisy, and it is difficult to generalize to other periods due to the non-stationarity of our data

Future

Potential future directions for this project may include:

- **Enriching our dataset**. We should include non-technical features such as fundamental companies' accounting data
- **Changing our prediction objective**. We could try a regression model instead of a classification model. Rather than prediction "Up" or "Down", we could sort our predictions in quantiles and try to predict bucket probability
- **Experimenting with different time horizons**. We can explore a variety of different window sizes (many high-frequency traders consume data in milliseconds)

References

[1] R.A. de Oliveira D.M.Q. Nelson, A.C.M. Pereira. Stock markets price movement prediction with lstm neural networks. In 2017 International Joint Conference on Neural Networks (IJCNN), pages 1419–1426, 2017.

[2] F. Cummins F.A. Gers, J. Schmidhuber. Learning to forget: Continual prediction with lstm. Technical Report, IDSIA-01-99, 1999.

[3] Colah, C. (2015), 'Understanding LSTM Networks', on Colah`s personal blog, available at http://colah.github.io/posts/2015-08-Understanding-LSTMs/