Deep Learning for Stock Price Forecasting

Kevin Li, Glenn Yu

kevinqli, glennyu@stanford.edu

Introduction

The goal of this project is to build an automated system to trade stocks. Stock traders examine various patterns in stock market data and use well-known motifs to determine their trade decisions. We believe with a well-trained neural network, we can automate these decision processes.

To narrow the problem, we focused on predicting daily price trend given historical data. As such, the **input** is an array of price features of the past 60 days and the **output** is whether the next price will increase or not.

Data Pipeline

- Daily price data for over 3000 stocks from Alpha Vantage
- 5 basic features for each price point (open, close, high, low, volume)
- 25 augmented features including technical indicators such as EMA, MACD, and Bollinger bands.
- For log-normal features like volume, we compute the log-transformation.
- Each feature is then normalized to have mean 0 and variance 1.
- Finally, each company's stock prices are split into 60-day windows for training and testing.

Models

Baseline Models: We experimented with different classification models, including ridge, SGD, and random forest, with hand-tuning of multiple parameters.

Shallow LSTM Network: We use a LSTM network with 30 hidden units, along with 4 fully connected dense layers and dropout.

Stacked LSTM Network: We use a two-layer LSTM network with 30 hidden units each, along with 4 fully connected dense layers and dropout.

1-D Convolutional NN: We use 9 1-D convolution layers, convolved in the time axis, along with 1 global max-pool layer (over time) and one fully connected dense layer..

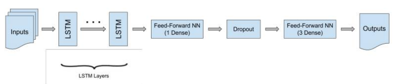


Figure 1. Graphical representation of the LSTM model.

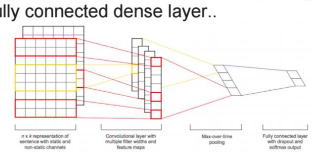


Figure 2. Graphical representation of the Conv-1D model.

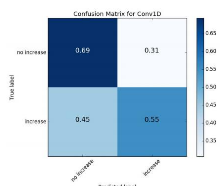
Results and Evaluation

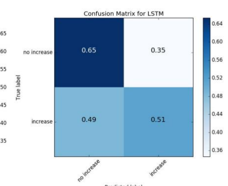
All DL models were trained using sigmoid cross entropy loss and run for 50 epochs. We used accuracy as the evaluation metric for the binary task of predicting no increase or increase. We had 113,908 training and 6,328 validation examples after a 90-5-5 split.

Label Distributions	Training	Validation
No increase	54.9%	55.2%
Increase	45.1%	44.8%

Table 1: Summary of model accuracies.

Model	Train Acc.	Validation Acc.
Ridge	58.03	57.79
Random Forest	98.4	55.09
SGD	54.77	55.06
Shallow LSTM	74.67	59.10
Deep LSTM	58.5	57.7
1-D CNN	56.2	57.7





Discussion and Analysis

- We searched through different hyperparameters, including learning rate, batch size, dropout rate, and L2 regularization, but did not achieve much higher results.
- We found that increasing the number of LSTM layers in the network generally decreased performance, perhaps because the deeper the model, the harder it is to train.
- None of the deep learning models were able to overfit the training set even with regularization, which suggests that the input may be too noisy for classification.
- Random Forest was able to overfit the training data likely because it is an ensemble model that comprises a bunch of strong classifiers, but it too is unable to generalize to the validation set.
- The deep learning models performed better than just predicting the most frequent label.
- LSTM models achieved higher accuracies than the CNN because it captures the sequential nature of time series data more effectively.

Conclusions and Future Work

- From this project, we see that it is very difficult to obtain signal from purely market data on even a simple task like price trend forecasting.
- For the future, we may want to include unstructured data from social media and news sources as helpful indicator features.
- We can also look for other kinds of financial data that might have better signal, such as market book transactions (for high frequency trading) or the economic sentiment of the general public (for predicting macro-economic trends)..

References

1. Borovykh, A., Bohte, S. and Oosterlee, C.W. (2017) Conditional Time Series Forecasting with Convolutional Neural Networks. *ArXiv e-prints*.
2. Flunkert, V., Salinas, D. and Gasthaus, J. (2017) DeepAR: Probabilistic Forecasting with Autoregressive Networks. *ArXiv e-prints*.
3. Jozefowicz, R., Zaremba, W. and Sutskever, B. (2015) An Empirical Exploration of Recurrent Network Architectures. *Proceedings of the 32nd International Conference on Machine Learning*. **37**.