Ranking financial assets using neural networks

Anton Ponomarev - aponom22@stanford.edu
Youtube link - https://youtu.be/89PIPxUSYQo

Motivation

Using deep learning to enhance the outcome of financial investments has been the “holy grail” for the industry over a number of years. While a lot of progress has been made, the major breakthroughs are yet to be made and the existing cutting-edge technology is still out of reach for the average investor. Vast majority of existing research[1,2,3] has been focused on predicting the direction and/or magnitude of future price movement. This usually requires access to large data sets of high-frequency price data or other information like financial reporting or news and social media content, again mostly unattainable for the average investor.

In this project we propose an approach to selecting an asset that is expected to yield highest return out of the pre-defined list of assets over a pre-defined investment horizon. This is done with the intention of softening the data and compute requirements while still resulting in a respectable predictive power, that can be of benefit to the average investor.

Data

We use daily closing prices for a selection of liquid publicly-traded financial assets (sourced from Yahoo Finance). The gaps are back-filled and the data is converted to log returns with appropriate normalization. Hyperparameters are:

- Number of assets, N
- Look back window, T
- Investment horizon, H

At time t, a training example can be represented like:

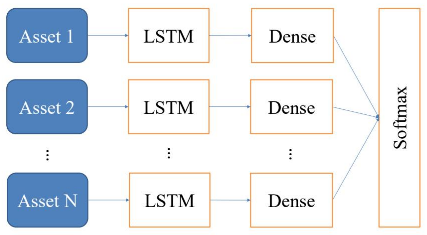
$$X_t^{[train]} = \begin{cases} x_t^{(1)}, x_{t-1}^{(1)}, x_{t-2}^{(1)}, \dots, x_{t-T}^{(1)} \\ x_t^{(2)}, x_{t-1}^{(2)}, x_{t-2}^{(2)}, \dots, x_{t-T}^{(2)} \\ \dots \\ x_t^{(N)}, x_{t-1}^{(N)}, x_{t-2}^{(N)}, \dots, x_{t-T}^{(N)} \end{cases}$$

$$Y_t^{[train]} = \max\left(\sum_{i=1}^{H} x_{t+i}^{(1)}, \sum_{i=1}^{H} x_{t+i}^{(2)}, \dots, \sum_{i=1}^{H} x_{t+i}^{(N)}\right)$$

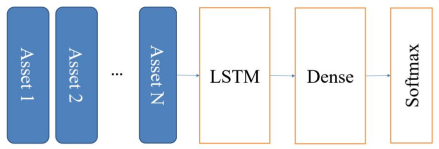
Experiments

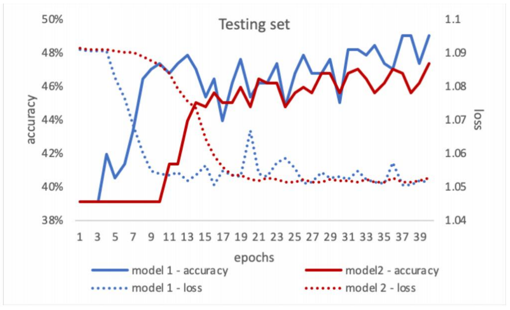
We have set up and tested a number of architectures, using just fully-connected layers, convolutional layers and LSTM. The latter proved to be the most successful, with two main architectures presented below. Since we use a one-hot encoding for the labels, a *categorical cross-entropy* loss was chosen as the most appropriate, although we have also experimented with others. For optimization, we chose *Adam* method which performed well but comparable to other methods like *Adadelta*.

Model 1



Model 2





Key observations

- ➢ Using neural networks, we can train the model to beat a naïve approach (e.g. using moving averages)
- ➢ Architectures utilizing LSTM showed best results
- ➢ We didn’t manage to overfit the training set and testing accuracy stayed close, suggesting that further deepening of the network can increase both
- ➢ Using just fully-connected or convolutional layers overfits the training set, and any type of regularization (via BatchNorm and Dropout) dramatically impairs generalization of the model

Model	Error		Accuracy	
	Training	Testing	Training	Testing
1	1.03	1.05	0.46	0.49
2	1.04	1.05	0.45	0.47

Data used: 3 assets, training set (95%): x(7104, 21, 3) y(7104,3), testing set (5%): x(353, 21, 3) y(353, 3), 100 epochs with batch size 32

Future research

- Adopt learning-to-rank algorithms from the field of information retrieval to generate a full ranking of assets. Google has just released a dedicated TensorFlow library[4].
- Re-frame the problem in the context of an agent seeking to figure out the optimal strategy for selecting best performing asset. Reinforcement learning and other nature-inspired algorithms (e.g. ant colony optimization) can be explored.

References

[1] X. Ding et al Deep Learning for Event-Driven Stock Prediction
[2] M. Dixon et al Classification-based Financial Markets Prediction using Deep Neural Networks, 2016
[3] Q. Song et al Stock portfolio selection using learning-to-rank algorithms with news sentiment, 2017
[4] R.K. Pasumarthi et al TF-Ranking: Scalable TensorFlow Library for Learinng-to-Rank, 2018