Introduction

The Efficient Markets Hypothesis states an efficient capital market fully reflects all relevant information in determining security prices [1]. Since any past prices and other information relevant to prediction are available to the market, it follows that any attempts at prediction will already be immediately taken advantage of; thus, an efficient market behaves in the manner of a Random Walk. However, given latency in market information availability and absorption, the varying liquidity of a market, and countless other real-life factors, the reality of the stock market deviates from the theoretical perfectly efficient model, thus allowing a small window for prediction.

Stock price prediction has typically been tackled by simple feed-forward neural networks and support vector regression, which have been shown to predict with high accuracy rates. I plan on implementing a deep Long-Short Term Memory network to accomplish the same task, with the hope that the memory capabilities may help prediction.

In addition to metrics of pure market data, I will be performing sentiment analysis on historical Twitter feeds of major global news agencies, using any tweets relevant to the companies and markets I am analyzing. This should give not only a global indicator for impending market trends, but also specific headlines concerning companies, such as product releases, scandals, and financial reports. I will attempt to implement this using LSTMs, but this is an area of experimentation for this project, as I will discuss in the Substance and Structure of the Project section.

This sentiment classifier will produce a sentiment prediction for each Tweet, which will then be used to generate features for the price prediction network.

Time permitting, I will also be implementing a simulated trading platform that will interface with the predicting network via an API. This platform will use expected utility functions that utilize incoming market predictions to make trading decisions.

Resources Required

My dataset will be acquired from the Bloomberg Terminal in the Marshall Library, which is free to use for all Cambridge students. This will provide me with advanced data about historical stock pricing and trading, in intervals as small as one minute, stretching back at least a full year.

I will also require use of my personal machine for the purposes of writing my project and maintaining my dataset, as well as for running and training my network.

Computer Specifications:

MSI GS63VR Laptop

Windows 10 Home

Intel i7-6700HQ CPU

16GB RAM

750GB SSD

Nvidia GTX 1060 6GB GDDR5

In case of failure: all source files will be on Github, and I will have a physical backup of all data files used. I can then continue working on MCS. Training the LSTMs on MCS machines may pose some difficulty, so I will ask for permission to use fellow students' workstations in this case.

Starting Point

I will be using Tensorflow for constructing the main LSTM network, and for constructing the LSTM network for twitter sentiment analysis. I will be working on mostly publicly available datasets; though I may have additional insight to the markets due to available information from the Bloomberg Terminal. All of the sentiment analysis datasets are fully publicly available.

Substance and Structure of the Project

The aim of the project is to build a deep neural network of LSTMs to try to predict stock prices based on historical data. This will leverage the supervised learning-based sentiment analysis of tweets from major news outlets as well as numerical analysis of stock pricing, volume, and general trading data.

The key part of my project will be the development of the predictive system. This includes research into layer structures for general neural networks and LSTMs, research into various optimizations such as early stopping, least absolute shrinkage and selection operator (LASSO), and other various regularization methods, and coding and testing of the system.

Parts of my LSTM framework for overall market prediction may be reused for my sentiment analysis feature generator. Because of the lack of labelled twitter datasets from news agencies, I am planning on performing supervised learning using a diverse database of ordinary tweets and newspaper articles in the hopes of capturing a useful model for extracting sentiment from news headlines on Twitter. The specific datasets that I will be employing to form this database are detailed below. This will be an area of experimentation to find a model for sentiment analysis that works best for the dataset. Other methods and algorithms worth considering are some form of unsupervised learning [2], which may be more suited to our unlabeled dataset, or other classifiers such as Naïve Bayes.

The result of the sentiment classifier will be parsed for any relevance to a company from the index I am studying. This will then be transformed into a one-hot-encoding style feature vector, with each feature corresponding to each company in our relevant index, and the corresponding feature value will be the predicted sentiment of the news article (1 for positive, 0 for negative). If the Tweet doesn't appear to have immediate relevance to a company, it will be used as a separate feature that is meant to showcase global news opinion.

If I manage to complete the above with sufficient time remaining, I will research economic expected utility functions and build a simulated trading platform that will interface with the prediction system. This platform will make investment decisions based on the utility functions and keep track of any stock shares currently held, liquid capital available, and net profit.

The project would therefore have the following main segments:

1. Familiarisation with the tools available to me, such as Tensorflow and the Bloomberg Terminal, and the Deep Learning algorithms I'll be implementing. Detailed design of the LSTM network structure, investigation into various tools for sentiment analysis that will fit our dataset.
2. Implementation and testing of the various sentiment analysis algorithms for breaking news headlines from the twitter feeds of news agencies, randomly choosing samples from the test dataset to manually classify and test against.
3. Development of a feature extractor for financial data, based on various statistical time-series calculations. Many of these features may already be computed by Bloomberg, so very few (if any) metric computations will need to be coded.
4. Development and testing of the main LSTM network that will utilize the tools built in (2) and (3) to perform stock price prediction.
5. (Time Permitting) Study financial expected utility that define the optimal trading strategies given a set of information including predicted return and risk. Build a simple simulated trading platform that will perform automated trades according to the optimal decision as defined by the expected utility, with the input being the predicted prices from the main LSTM network.
6. Evaluation of the project based on the accuracy of predictions using various goodness of fit tests, and if (5) was completed, the profit generated over a specified time period.
7. Writing the dissertation.

The datasets I will be using/exploring will be:

1. Historical stock trade price and volume data from Bloomberg, with an interval of 1-5 minutes, including any potentially helpful statistical figures pre-computed by Bloomberg, such as bid-ask spread.
2. Unlabelled Twitter feeds from global news sources, dating back to mid-2016, sourced from George Washington University [3].
3. Sentiment-labelled general Twitter feeds, from various sources such as Sanders Analytics, and the University of Michigan.
4. Sentiment-labelled congressional speeches corresponding to votes, sourced from Cornell University [4].
5. Sentiment-labelled movie reviews, from various sources of Cornell University and Stanford University [5][6][7].
6. Sentiment-labelled sentences, from the UC Irvine Machine Learning Repository [8].
7. Other datasets I may find useful, mostly for helping train the sentiment classifier.

Reference

[1] Malkiel, B. G. (1989). Efficient market hypothesis. The New Palgrave: Finance. Norton, New York, 127-134.

[2] Turney, P. D. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised. *CoRR*, *cs.LG/0212032*.

[3] Littman, J., Wrubel, L., Kerchner, D., Bromberg Gaber, Y. (2017). News Outlet Tweet Ids. Retrieved from https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/2FIFLH.

[4] Thomas, M., Pang, B., Lee, L. (2006). Get out the vote: Determining support or opposition from Congressional. CoRR, abs/cs/0607062.

[5] Pang, B., Lee, L., Vaithyanathan, S. (2002). Thumbs Up? Sentiment Classification Using Machine Learning Techniques. Retrieved from http://www.cs.cornell.edu/people/pabo/movie-review-data/.

[6] Pang, B., Lee, L. (2004) A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. Retrieved from http://www.cs.cornell.edu/people/pabo/movie-review-data/.

[7] Pang, B., Lee, L. (2005) Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. Retrieved from http://www.cs.cornell.edu/people/pabo/movie-review-data/.

[8] Kotzias et. Al (2015) From Group to Individual Labels using Deep Features. Retrieved from https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences.

Success Criteria

1. The whole prediction system must be fully functioning and able to output predictions based on input data.
2. The prediction system must perform better than a large set of random agents within a significance level of 5 percent.
3. The system must be able to make predictions in a time frame less than half that of the time interval between data points (30 seconds for a time interval of 1 minute).
4. The accuracy of the sentiment analysis classification based on verification against manually labelled test data must be significantly better than a random agent.

Supervision Arrangements

I will be meeting with my main supervisor, Dr. Holden, once a week until the end of the project. My second supervisor, Prof. Satchell, is a fellow of Trinity College specializing in econometrics, finance, risk measurement, and utility theory. I will be meeting Prof. Satchell regarding any financial market-based difficulties I incur, and should not have to leverage his help nearly as often.

Timetable and Milestones

Weeks 1 to 3 (October 23 – Nov 12)

Study of relevant machine learning algorithms. Familiarization with tools such as Tensorflow and Bloomberg. This will include generation of simple LSTM system. Build a walking skeleton of code modules to iron out early interfaces. This will include the sentiment classifier, the financial data feature extractor, and the main LSTM network. Research different layer structures and their applicability to this problem set. Devise an early design for the layer structure. Study of the predictive powers of various financial metrics.

Milestones: Finish walking skeleton of code modules (dummy functionality is acceptable).

Weeks 4 and 5 (Nov 13 – Nov 26)

Further literature study and discussion with Computer Science Supervisor to ensure early design of LSTM network structure is sound. Discussion with Economics Supervisor to ensure financial features are sound. Implementation of and experimentation with various algorithms for sentiment classification.

Milestones: Have working examples of one or more sentiment classifiers.

Week 6 (Nov 27 – approx. Dec 2)

Full implementation of various sentiment classifiers. Test with a small subset of manually classified news agency Twitter feeds.

Milestone: Fully implemented optimally-chosen sentiment classifier.

Weeks 7 to 10 (Dec 3 – Jan 1)

Christmas Break

Weeks 11 to 12 (Jan 2 – Jan 14)

Discuss with Computer Science Supervisor to help choose the most suitable sentiment classifier(s) for the project's datasets. Integrate sentiment classifier into the walking skeleton. Start implementation of LSTM network. Early testing on financial and news datasets. Implement financial feature extractor. Start integrating both into the project.

Milestone: Initial implementation of LSTM network, which should include various regularization techniques. The project should now be almost functional. Progress report submitted and entire project reviewed both personally and with Overseers.

Weeks 13 to 16 (Jan 15 – Feb 11)

Write initial chapters of Dissertation. Continue implementing and refining LSTM network. Manually classify a larger subset of the Twitter feeds to ensure evaluation of the sentiment analysis system is valid. Start initial evaluation of system. Time permitting, research the relevant economic background needed to build the simulated trading platform. Build the simulated trading platform.

Milestones: Preparation chapter of Dissertation complete, Implementation chapter near completion. LSTM network fully integrated. Code performs tolerably well together.

Weeks 17 to 22 (Feb 12 – March 25)

Attempt to write Introduction chapter and finish the Implementation chapter.

Milestones: Partially complete Introduction chapter, partially complete Implementation chapter.

Weeks 23 to 25 (March 26 – April 15)

Polish code. Finalize evaluations, and attempt to finish the dissertation.

Milestones: Finished evaluation (whether that be accuracy measurements or simulated net profit), Finished draft of Dissertation, Have Dissertation proof-read by Supervisors and DoS, as well as friends.

Weeks 26 and 27 (April 16 – April 29)

Finalize Dissertation. Aim to submit before the end of Week 27.

Milestones: Submission of Dissertation