Machine Learning Engineer Nanodegree

Capstone Proposal

Jacob Anthony

(jacobscottanthony@gmail.com)

February 18th 2020

Proposal

Domain Background

The domain of this proposal is finance and more specifically stock price prediction. Price prediction is a hard problem and one that is certainly not solved. Applying machine learning techniques to this problem is well studied (Krolner, Vanstone, & Finnie, 2010). The increase in interest in this area of study has coincided with a simultaneous reduction in computing cost due to Moore's law (Wikipedia, 2020), an increase in Artificial Intelligence research (Wikipedia, 2020) and an increase in value of global markets (Federal Reserve Bank of St. Louis, 2020). Financial markets are ripe for the application of Artificial Intelligence techniques given they are typically data-rich, are vast and varied, are difficult to model through quantitative means and offer a substantial financial reward to players able to successfully develop them.

My motivation for choosing this topic is my personal interest in the subject. I have tried implementing trading strategies before and have had this idea in the back of my mind since I started this program.

Problem Statement

Stock price prediction is a crucial skill for any investor who wishes to make informed decisions in the market. Predictions of some kind are a part of (virtually) all trading strategies. Supervised Machine Learning techniques that use artificial neural networks such as Long Short-Term Memory (LSTM) are well suited to these kinds of time-series prediction problems and can help to generate predictions of the market which in turn help to improve decision making for traders.

Datasets and Inputs

This proposal will make use of public financial data in the form of daily (open, high, low, close, volume) or intraday (open, high, low, close, volume) pricing data for the financial instruments considered. Published research in this area shows models being successfully trained on both daily and intraday data. Sources will include:

- Stock and Market Index's, daily resolution:
  - Alphavantage API
    - https://www.alphavantage.co/
- Stock and Market Index's, intraday resolution:
  - Purchased from:
    - https://pitrading.com/historical-stock-data.html
    - https://firstratedata.com/
    - https://www.backtestmarket.com/en/

Pricing and trading volume data are both time-series data. In addition to the time-series data, a number of leading and lagging market measures (also known as Technical Indicators) will be added to the state which will help to describe the past,

current and (possible) future state of the market at each time step. (Krolner, Vanstone, & Finnie, 2010) find the most commonly used indicators as:

- Simple moving average, SMA
  - https://www.investopedia.com/terms/s/sma.asp
- Exponential moving average, EMA
  - https://www.investopedia.com/terms/e/ema.asp
- Relative strength index, RSI
  - https://www.investopedia.com/terms/r/rsi.asp
- Rate of change, ROC
  - https://www.investopedia.com/terms/r/rateofchange.asp
- Moving average convergence / divergence, MACD
  - https://www.investopedia.com/terms/m/macd.asp
- Williams oscillator, WO
  - https://www.investopedia.com/terms/w/williamsr.asp
- Average true range, ATR
  - https://www.investopedia.com/terms/a/atr.asp

This set of technical indicators will be the starting point for the training set. It's noted that there are a large number of additional possibilities that could be considered for the agent (TradingView, 2020) & (Wikipedia, 2020).

The maximum time horizon for training I am going to consider will be limited by the data available. From the sources above, it'll be easy to collect daily data for most US Stocks going back to the early 2000's. At a daily resolution this will be ~ 20y x 250d = 5000 rows per instrument. If I increase this to minute resolution this becomes ~ 5000d x 8h x 60m = 2.4M rows per instrument. I will only use stocks currently found in the S&P500 index such as Ecolab Inc (NYSE: ECL).

Solution Statement

Long, Short Term Memory (LSTM) recurrent neural networks are an ideal solution to time series prediction (Wikipedia, 2020) as they have a type of memory. "[LSTMs] can learn to recognize an important input, store it in the long-term state, learn to preserve it for as long as it is needed, and learn to extract it whenever it is needed" (Géron, 2017).

A number of approaches to solving this problem have been considered such as purely statistical methods like autoregressive integrated moving average (ARIMA) and Support Vector Machines (SVMs). Facebook's Prophet forecasting API was also considered.

Due to its dominance in the literature and applicability to the problem statement, I've chosen to train an LSTM neural network to predict future stock prices.

Benchmark Model

I am proposing one benchmark model to compare the results achieved by this proposal.

- "Stock Prices Prediction using Deep Learning Models" (Liu, Chao, Lin, & Lin, 2015)
  - https://arxiv.org/pdf/1909.12227
  - This team trained an LSTM network to achieve a MAPE prediction accuracy of ~ 0.01 or 1% using daily OHLCV data for the S&P500 index.

Evaluation Metrics

There are a number of metrics to choose from when evaluating the performance of a neural network. From the literature considered, I'm going to use:

- Mean Average Percentage Error, MAPE
  - https://en.wikipedia.org/wiki/Symmetric_mean_absolute_percentage_error
- Mean Squared Error, MSE
  - https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html#sklearn.metrics.mean_squared_error
- Pearson Correlation Coefficient, PCC
  - (https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.pearsonr.html

Project Design

Below is the approximate workflow that I am planning to follow in completing this project.

- **Data Acquisition**
  - Acquire the daily and intraday pricing and volume data for all the stocks in the S&P500 index.
- **Data Pre-Processing**
  - Visualize and explore the data.
  - Re-add all the timestamps with zero volume (removed from dataset) so there is an even and continuous timestep from start to finish.
  - Pre-process the pricing and volume data into a continuous time series by removing the dates and replacing them with a whole number index.
  - Add all the technical indicators as mentioned above to the dataset.
  - Scale all variables in the dataset.
- **LSTM Network Architecture**
  - Using the Keras library, I'll specify a simple LSTM architecture to begin with.
- **LSTM Training**
  - Initialize all hyper-parameters using reasonable estimates from the referenced papers in this proposal.
- **LSTM Hyper-Parameter Tuning**
  - I am planning to start with manual grid-searching.
  - If this manual grid search is wasting too much time I will setup a more formal grid search of the variables of interest.

    - Stretch Goal: Experiment with Random Walk and Bayesian Optimization of the hyper-parameter tuning to see if this can further improve the grid search.
- **LSTM Performance Evaluation vs. Benchmark Model**
    - I will chart and discuss the performance of the network in training, validation and prediction.
- **Loop through Architecture, Training, Tuning and Performance Evaluation Steps**
    - Loop through steps as necessary or as additional tweaks are made.
- **Write up the Report**
    - Using the Udacity Project Rubric, write up the work completed and submit.

Stretch Goals:

- **Principal Component Analysis (PCA) for each Technical Indicator**
    - Conduct a PCA for each technical indicator in the data set with respect to *Close* price. Simplify the state array to the Technical Indicators that represent 80% of the PCA and see if this reduces noise in training without sacrificing model performance.

Bibliography

Federal Reserve Bank of St. Louis. (2020, February 2). *Wilshire 5000 Total Market Full Cap Index*. Retrieved from https://fred.stlouisfed.org/series/WILL5000INDFC

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn & TensorFlow.* O'Reilly Media Inc.

Krolner, B., Vanstone, B., & Finnie, G. (2010). Financial Time Series Forecasting with Machine Learning Techniques: A Survey. *ESANN 2010 Proceedings* (pp. 25-30). Bruges, Belgium: ESANN.

Lim, B., Zohren, S., & Roberts, S. (2019). Enhancing Time-Series Momentum Strategies Using Deep Neural Nets. *The Journal of Financial Data Sciences*, 19-38.

Liu, J., Chao, F., Lin, Y., & Lin, C. (2015). Stock Prices Prediction using Deep Learning Models.

Nan, A., Perumal, A., & Zaiane, O. (2020). Sentiment and Knowledge Based Algorithmic Trading with Deep Reinforcement Learning.

TradingView. (2020, February 2). *Category:Indicators and overlays*. Retrieved from https://www.tradingview.com/wiki/Category:Indicators_and_overlays

Wikipedia. (2020, February 2). *History of aritifical intelligence*. Retrieved from https://en.wikipedia.org/wiki/History_of_artificial_intelligence#Deep_learning,_big_data_and_artificial_general_intelligence:_2011%E2%80%93present

Wikipedia. (2020, 02 18). *Long short-term memory*. Retrieved from https://en.wikipedia.org/wiki/Long_short-term_memory

Wikipedia. (2020, February 2). *Moore's law*. Retrieved from https://en.wikipedia.org/wiki/Moore%27s_law

Wikipedia. (2020, February 2). *Technical Indicator*. Retrieved from https://en.wikipedia.org/wiki/Technical_indicator

Xiong, Z., Liu, X., Zhong, S., Yang, H., & Walid, A. (2018). Practical Deep Reinforcement Learning Approach for Stock Trading.