LSTM (Lucrative Stock Trading Machine)

Using Deep Learning to Predict the Stock Price Movement

Machine Learning Engineer Nano Degree
Capstone Project Proposal
Sloan Austermann

Domain Background

This purpose of this project is to create a deep learning algorithm that can learn patterns in from S&P500 price and volume data to determine which stocks in the index will have the highest chance of positive future daily returns and which stocks have the highest change of negative daily returns. This problem has historical roots in the trading of financial security, particular over short-term time horizons. There are two philosophies of analysis in the world of investing and trading, fundamental analysis and technical analysis. Technical analysis uses heuristics or mathematical calculations based on the price, volume or open interest of a security. Fundamental Analysis, on the other hand, is a method of measuring a security's intrinsic value through the examination of economic and financial factors. Macroeconomic factors like the interest or tax rates or microeconomic factors like the allocation of capital are both taken into consideration in attempt to calculate a stock's intrinsic value[1]. Momentum, for example, is one of the most widely known and used technical indicators because of its ability to achieve abnormal returns (returns above the market), over varying time intervals.

A 2015 study looked at momentum strategies on Global Multi-Asset data from 1800-2015 found that this strategy is not only effective on Equities but also Forex, Bonds, Commodities and other asset classes[2]. There many other technical indicators that all fall under the four main types of indicator: Trend, Momentum, Volume and Volatility. Relative Strength Index is an example of a momentum indicator, the Moving Average Convergence Divergence (MACD) is an example of a trend indicator, and Bollinger Bands are an example of a volatility indicator[3]. All of these indicators have proven to have success in trading strategies that executed over time frames of a few weeks to a few months. Fundamental analysis has is backed by economic theory and has been proven to be successful over long time periods. On the other hand, High Frequency Trading happens incredibly fast, executing trades in microsecond time intervals purely based off of price and volume data. The strategy being implemented in this project will attempt to use deep learning to find a successful strategy using a combination of factors over a medium range time frame of a 30-120 days.

---

[1] Investopedia, Technical Analysis, 2020.
[2] Christopher C. Geczy, Mikhail Samonoc, 215 Years of Global Multi-Asset Momentum, 2015.
[3] Harry Nicholls, 7 Popular Technical Indicators and How to Use Them, 2018.

As an undergraduate at the University of Notre Dame, I studied Mathematics and Economics, focusing on a concentration in financial econometrics. Studying economics through this financial lens taught me to appreciate the power of the stock market as a data processing machine. This academic background inspired and motivated the idea to apply machine learning to the financial market data in an attempt to better understand its seemingly enigmatic behavior.

Problem Statement

Since we are working with time-series data, the machine learning model developed in this project will be designed to solve a forecasting problem. Specifically, since we are working with stock price data, the model will attempt to predict the movement of the asset prices using a look-back window of historical data leading up to the time of prediction. The goal of applying machine learning trading is to determine which stock prices will increase in the future and which stock prices will decrease in the future. If the historical price data is one-hot-encoded to categorize the data into three categories based off of some daily return threshold, T, with all stocks that have a return greater than T over the prediction period categorized as a buy, stocks with a return below -T over the prediction period categorized as a sell, and everything else categorized as a hold. Assigning these values to each security for each day over the range of the historical data will turn our problem into one of classification. Rather than building a regression algorithm to predict the stock price of each security, we can classify them as buy, sell or hold based on whether we believe the price will go up or not.

Datasets and Inputs

The data used for this project is OHLCV data for the stocks trading on the S&P 500 index. This data was pulled from Yahoo Finance using the python library pandas-datareader. The data I collected and the code for I used to mine it are included in the GitHub Repo associated with this submission. The data is stored in the 'stocks_dfs' folder and the code used to get the data is in get_data.py and was used as part of a Sentdex's tutorial on python for financial analysis[4]. This

[4] Python Programming For Finance, Getting all company pricing data in the S&P 500 (2017)

data is structed in a panel and hierarchically indexed by date and stock ticker. The Adjusted Close and Volume data will be some of the primary feature inputs into the model. Other features will be technical indicators calculated from the Adjusted Close and Volume data for each stock. The technical indicators I will use will be using will be momentum of varying period lengths, delta Bollinger width, velocity, ATR and MACD. The input for the model will be an n-day window of historical data and the respectively calculated technical indicators that take place over the n days directly preceding the prediction period.

Solution Statement

Each security will have a target label (buy, hold, sell) for each day. The inputs for that label will be the n-day window of OHLCV and technical indicator data. The label will be determined based off of the price movement during the prediction window. The inputs for day n will be the sequence of day from n-60 to n-1. The label for day n will be:

'buy' if price (n+k) /price(n) > T.

'sell' if price(n+k)/price(n) < -T.

'hold' otherwise.

For some return threshold T and prediction window length k. The if the model will be tasked to accurately classify each stock's return over the prediction window as either greater than the threshold, less than the negative threshold or in between.

Benchmark Model

A 2011 study also attempted to predict the direction of stick price movement using machine learning technique applied to technical indicators of as inputs to their models. They train both vanilla MLP or ANN and a Support Vector Machine on the indicators calculated from Istanbul Stock Exchange data. They experiment resulted in the ANN performing with 75.74% accuracy and the SVM attaining 71.52% accuracy[5]. Since the vanilla ANN has proven to perform well on the problem, that would be a good benchmark model to start with in this project.

---

[5] Kara, Yakup & Boyacioglu, Melek & Baykan, Omer. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. Expert Systems with Applications. 38. 5311-5319. 10.1016/j.eswa.2010.10.027.

Evaluation Metrics

Accuracy will be the primary metric used to evaluate the performance of the model. The most important think in trading is picking the right trades. If the model picks profitable trades more often than not, then the model will be working sufficiently. A Confusion Matrix will also be helpful for evaluation here to determine if the model is misclassifying one label more often than another. Finally, model's performance can be tested by back testing the output of its trade picks in a simulated training test. Using the Zipline library, the output of the predicted training and testing values can be used to simulate a real trading algorithm powered by the model's predictions[6]. This performance of this algorithm can also be used as a performance metric. The ultimate would be to build a model with optimal risk/return profile and beat the market return without taking on too much additional risk.

Project Design

The goal of the project is to build a deep learning model to learn from the price movement and technical features of stock market data to accurately predict the movement of stock prices. Because of nature of the time-series financial data, a LSTM -Recurrent Neural Network will be the ideal model to forecast the price movement of the stocks. The recurrent loops in a LSTM network allow information to persist and for the model to learn from patterns that occur in the data over time. In order for the model to properly learn, the data has to be formatted correctly[7].

The preprocessing of the data will require for the datetime index of stock data to all be in the same time zone and sequentially ordered from earliest to most recent. The labels for each date and stock will be created by one hot encoding the daily returns of the point at some point in the future. This point will be defined by the prediction window. This parameter can be adjusted to determine which window length is results in the most profitable strategy. The inputs and the labels will be defined as described in the Dataset and Inputs and Solution Statement sections,

---

6 Zipline.io, Risk and Performance Metrics (2016).
7 Ravindra Kompella, Using LSTMs to forecast time-series (2018).

respectively. However, once the data is processed into the proper sequential, labeled form, it will be split into balanced groups so that the model does not over fit to predict one class more frequently than another because of and imbalance in the data.

The next step is to start training the models. The data will first be trained on a simple MLP to determine a benchmark accuracy. The data will then be trained on a LSTM – RNN to determine if an LSTM Network can improve upon the accuracy of the basic neural network. From there, we will adjust basic hyperparameters of the model such as the number and size of the layer and the learning rate, as well problem specific parameters such as the length of the look-back and prediction windows.

The goal of the model to should be to predict values that can be used as inputs to a profitable trading algorithm. The models predictions be used algorithmically to serve as buy and sell signal for stocks that we are most confident will move in a certain direction. The model's predictions can be back tested on historical stock data to evaluate the performance of an algorithm powered by the model.