

Capstone Project Proposal

DOMAIN BACKGROUND

“A price chart is a sequence of prices plotted over a specific time-frame. In statistical terms, charts are referred to as time series plots. “

The word “securities” refers to any tradable financial instrument or quantifiable index such as stocks, bonds, commodities, futures or market indices. Any security with price data over a period of time can be used to form a chart for analysis.

The chart is one of the important tools for traders to do analysis before they trade. From there, chart pattern is used to predict price movement. The successful traders can make profit on the market if they pick the high accuracy pattern and are able to locate the patterns on thousands of securities traded in the market.

PROBLEM STATEMENT

Traders have been using computers to help them in trading. It was Algorithm Trading that programmers implemented the rule-based trading strategy and hoped the computer speed could help to trade more. Since the improvement of Machine Learning in recent years, traders are applying Machine Learning to improve the strategy and identify trading opportunities. There is research done on predicting stock price with SVM model based on historical data and technical indicators. Some of the models predict the market based on financial news articles, reports with Bag of Words, Noun Phrases, and Named Entities. Then Deep Neural Networks have been very successful on image recognition recently, researchers tried to build a model on financial data.

There are many chart patterns for trading. This project just focuses on a single pattern when the price is going to change trend. Then the model should scan the historical chart; after the training on patterns; then identify the pattern. There is one problem which is how to define a trend change. Here, a successful trend change is that the price is able to reach a minimum price level from the point which indicated trend change. Then, the model can be used to classify a buy pattern and sell pattern of the thousands of securities; from stocks, to Forex pair simultaneously.

The CNN model is only to predict and locate the buy or sell pattern. The preprocess application described in the **PROJECT DESIGN** section will be used to calculate the **ENTRY**, **EXIT** and the **TARGET LEVEL**.

DATASET AND INPUT

The raw dataset will be downloaded from Yahoo Finance in daily/weekly/monthly time-frame with OPEN, HIGH, LOW, CLOSE price of the security per time-frame. From there, I have to scan the data and generate winning patterns and losing patterns with labels.

For each stock using 10 years of historical data, there should be 252 trading days X 10 years = 1,260 trading data samples. From there should be able to extract 50 winning patterns and 50 losing patterns data samples. The 10 years is from 2008 to 2017 which contains market crisis in 2008 and bull markets for the last few years. To increase the data set size, 500 stocks data can be used. In total, there should be 25,000 winning patterns and 25,000 losing patterns data samples for the model training.

The training data of the chart pattern is construct from a ten bar data. The label is 1 if this is a wining pattern or otherwise.

Here is an example of winning pattern:



The red line with the price level is the exit price if the trade is failed. The black line indicates the entry price of the trade. The blue dotted line indicates the target price for profit. The pattern is considered winning is that:

$$abs(target\ price - entry\ price) \geq n \times abs(entry\ price - exit\ price)$$

The '**target price**', '**entry price**' and '**exit price**' are calculated by an application that will generate the pattern data from the historical data files. The '**n**' is the parameter of how much profit the trader wants from this trade in terms of the possible lost. For now, '**n**' is defined as **5**. The size of the pattern provided for training will have 10 daily bars like the red rectangle with label whether this is a winning pattern (1) or losing pattern (0). The below diagram shows an example losing pattern.



The training data of each pattern has a window size, “ W ” of 10, of the daily OPEN, HIGH, LOW, CLOSE and Date. The Date is broken into 3 features which is day of week, day of month and day of years. Some traders have consider this information is useful for decision making. The window size “ W ” is parameterized that it can be evaluated with a bigger or smaller value during the training.

Open							
High							
Low							
Close							
Day of week							
Day of month							
Day of year							
	Bar 1	Bar 2					Bar 10

SOLUTION STATEMENT

The Convolutional Neural Networks (CNN) would be used as the modal to identify the chart pattern. A few pairs of Conv2D/MaxPooling layers will be used to apply filters and a full layer of 1 node to classify whether the input pattern is winning pattern.

Once the model is finished the training, the model will feed in a daily chart per year per securities. The process can be done overnight after the market was closed previous day.

BENCHMARK MODEL

The very common method to identify trend change for buy or sell is using 200 moving average and 50 moving average. The same training data would be used to find out the accuracy of this method which should be about a little bit over 50%.

If the accuracy of the modal is over 80%, it is a very high level of probability of success in trading community.

EVALUATION METRICS

The model is for trading in financial markets, therefore the evaluation of the model would do back test on historical data for profitability. The simulated trading can allocate a fix amount of money for trading in the account. If the account is profitable, the annual return rate will have to be calculated.

The next step is the forward testing on market, the model should running on real market data and execute the buy and sell signals based on a simulated market. Again, the profitability is calculated based on annual return rate.

PROJECT DESIGN

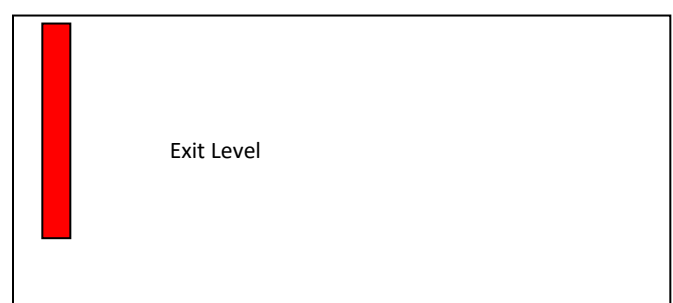
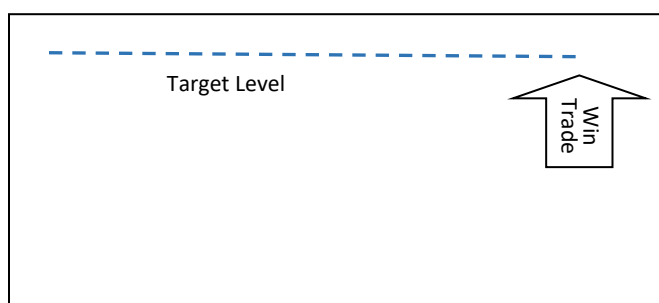
This project requires three applications:

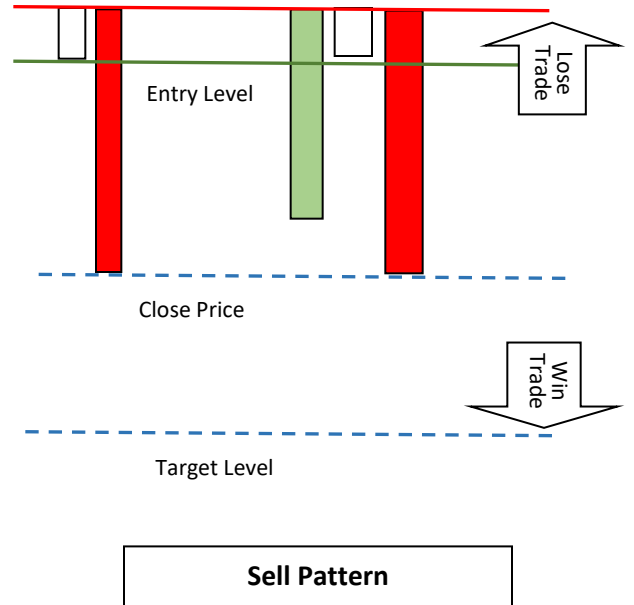
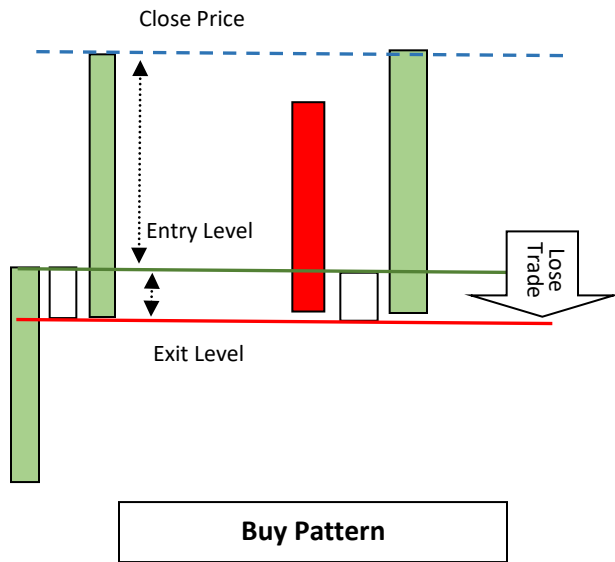
1. The preprocess application to locate winning pattern and losing pattern from historical price files.
2. An application to calculate the accuracy of the benchmark modal.
3. The CNN model implemented in Python with keras and TensorFlow backend. When the modal is in training, the GPU computing instance will be used. The same modal

THE PATTERN EXTRACT APPLICATION

The application is going to use the definition of winning pattern in the **DATASET AND INPUT** section. Each pattern generated is stored in one file according to sklearn.load_files format. The logic of this application is a known trading strategy. It tries to locate demand over supply to buy in an up-trend and supply over demand to sell in a down-trend market.

To form a **BUY PATTERN**, either a **rally/base/rally** or **drop/base/rally** bar pattern. The rally has to be three times larger than the base. Once the pattern is formed, and the price drop back to the **Entry Level** and bounce back up to the **Target Level**. It is considered a **Winning Trade** if $(\text{Target Level} - \text{Entry Level}) > 5 \times (\text{Entry Level} - \text{Exit Level})$. If the price drop back to the **Entry Level** and continue to drop below the **Exit Level**, it is considered a **Losing Trade**. The **SELL PATTERN** is just opposite direction of the **BUY PATTERN**.





THE BENCHMARK MODEL APPLICATION

This application is used to scan the historical data files to locate buy/sell signals according to **BENCHMARK MODEL** section. Then the accuracy is calculated for the comparison.

THE CNN MODEL

The model is constructed with a input layer, three Conv2D/MaxPooling2D pairs and a full lay of 1 node to indicate whether this is a winning (1) or losing (0) pattern. A weighting file will be generated after the training. Then the file should be used for model to predict patterns.