**Project Information**

|  |  |
| --- | --- |
| **Project:** | Stocks Trading Using RL |
| **Project Time-frame:** | October 2019 – December 2019 |
| **Summary:** | In the financial market, to decide whether to buy, sell, or wait for any financial instrument, one needs observations of the market. The profit is positive if the purchase takes place before the prices increase, otherwise it is negative. With this project, our goal is to get as much profit as possible |

### Background and Motivation

In the financial market, there are many financial instruments for which prices change over time. Trading involves frequent transactions, such as the buying and selling of stocks, commodities, currency pairs, or other instruments. The goal is to generate returns that outperform buy-and-hold investing. A trader would most likely look at some charts of a stock’s price action. From there, they would combine this visual information with their prior knowledge of similar price action to make an informed decision of which direction the stock is likely to move.

There are lots of financial consultants, investment funds, banks, and individual traders who have been trying to predict future price fluctuations, as this guarantees a lot of benefits. However, this is a very complex problem. To decide whether to buy, sell, or wait for any financial instrument, one needs careful observations of the market and then predict to find the best time to buy and sell and get maximum profit.

The above problem can be addressed using a Reinforcement Learning perspective where we have some observation of the market and we want to make a decision: buy, sell, or wait. The profit is positive when buying is done before the prices increase otherwise it is negative. In this project, we will aim to get as much profit as possible.

### Goals & Deliverables

The goal of this project is to start with replicating an existing model for trading functionality using Gym’s Env class API. Create an agent and environment and then investigate whether it will be possible for our agent to learn when the best time is to buy one single share and then close the position to maximize the profit. The purpose here is to show an increase in the existing feed-forward deep Q-network’s (FFDQN) performance.

The environment will have an agent that will be trained to do nothing, buy a share or close a position. If the agent has already got the share, nothing will be bought, otherwise a small percentage of the current price will be paid. The step-by-step reward will be equal to the last movement or the agent will receive a full reward at once only after the close action. We will train the agent using different learning algorithms and tune their hyperparameters to identify the best learning for our agent in the environment.

### Scope & Future Work

Currently, the scope of the system consists of open, high, low, and close prices given to an agent. The environment is implemented in the StocksEnv class. To improve the overall system, we will be tuning the RL model with:

* Different hyperparameter values
* Different architectures

For future work, we will try to tune the model including different datasets to see how the model performs in different conditions.

### Risks

The existing model is built with an assumption to convert every bar “open, high, low, and close” prices to three numbers as high, low, and close prices. However, this is a risk since we will have a loss of price information which may lead to a suboptimal solution.