Deep Reinforcement Learning for Optimal Stock Trading in Various Market Trends

Peter Li peter888@stanford.edu Tony Lee tonyhlee@stanford.edu

Introduction

- Stock Trading plays a crucial role in the growth of industry, commerce and the overall economy. However, it is deemed a high-risk investment financial activity.
- **Problem:** It is extremely challenging to obtain optimal trading strategy in such a dynamic market
- Deep reinforcement learning can navigate this large and complex space.
- **Goal:** Have our RL agent maximize return on investment given a test period and starting portfolio value.
- We explore various deep reinforcement learning based approaches and measure how well they perform in various market trends.

Building an Autonomous Stock Trading System

- Pull latest stock data from Alpha Vantage API and cache for future queries
- Given a period of time (start and end date), our system outputs an a sequence of orders that maximize return
- Our system is built on top of OpenAI Baselines with a custom MDP definition catered towards solving our stock trading problem
- Tensorflow backend
- Use CodaLab API to automatically publish experiments to a worksheet for reproducibility

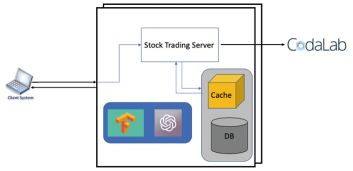


Figure 1. Architectural diagram of the stock trading system.

Experiments

- **Market:** Dow 30 Stocks (DJI) **Starting cash:** $10,000
- **Baseline:** Use entire starting cash and buy stocks using DJI
- **Oracle:** One-day lookahead of DJI and can buy and short daily
- **Hyperparameter Tuning:** Use validation set to find optimal step-size for DDPG (final step-size of 0.0001)
- **Number of Episodes:** 20
- **Fixed training period, but back test 4-month periods in various market trends:** Upward, Downward and Wave.
- **Experiment 1 (Economic Crisis)**

-Training Period: January 1, 2001 – April 30, 2008 -Upward Test Period: March 1, 2009 – June 30, 2009
-Wave Test Period: May 1, 2008 – August 31, 2008
-Downward Test Period: September 1, 2008 – December 31, 2008

- **Experiment 2**

-Training Period: January 1, 2001 – August 8, 2018 -Upward Test Period: January 1, 2019 – April 30, 2019
-Wave Test Period: May 1, 2019 – August 31, 2019
-Downward Test Period: September 1, 2018 – December 31, 2018

Future Work

- Explore training using various time periods
- Add more features to our existing reward function: frequency of trades, variance of portfolio value, etc.
- Explore more state-of-the-art RL algorithms (in particular Actor-Critic) and perform a more in-depth analysis and comparison between performances of the various algorithms
- Replicate results in a different stock market (e.g. NASDAQ)

MDP Definition

We modeled the stock trading process as a Markov Decision Process (MDP), considering the stochastic and interactive nature of the trading market:

- State:A set that includes the stock prices, the amount of stock holdings ***h*** and remaining balance.
- Action a: A set of actions on all ***D*** stocks. The available actions that can be performed for each stock: ***selling***, ***buying*** and ***holding***.
- Reward:The change of portfolio value when action ***a*** is taken at state ***s*** and arriving at new state ***s'***.
- Policy:The trading strategy of stock at state ***s***.
- ***Q(s, a)***:The expected reward achieved by action ***a*** at state ***s*** by following the policy.

Figure 2. Reinforcement learning for stock trading.

Some State-of-the-Art, Model-Free RL Algorithms

Deep Deterministic Policy Gradient (DDPG)

- Off-policy algorithm that learns Q-function and policy
- Same problem as gradient descent – Need to find a good step size!
- Allows agent to perform actions in continuous space with good performance

Trust Region Policy Optimization (TRPO)

- On-policy algorithm that trains a stochastic policy with KL-Divergence based constraint
- Samples action from the current policy with some randomness
- High-performant algorithm but can get stuck in local optima

Proximal Policy Optimization (PPO)

- On-policy algorithm that takes largest possible steps without overstepping
- Similar to TRPO, but uses only first-order optimization for easier implementation

Results and Analysis

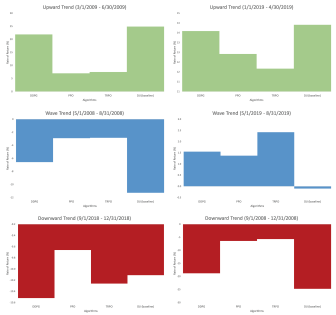
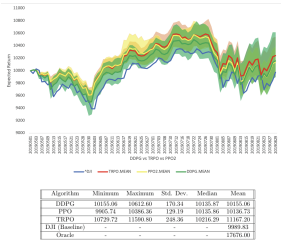


Figure 3. The performance of various reinforcement learning algorithms in different market trends compared to the baseline. The y-axis is the rate of return.

- RL algorithms did not perform as well or much better than the baseline and got completely outperformed by the oracle in both downward and upward economic trends
- A notable exception is PPO in a down market which outperformed the baseline by 18%.
- None of the RL agents yielded a profit in a down market
- Wave movements are more prevalent in the stock market than downward and upward spikes, therefore the trained agent better navigates in a wave-trending economy
- We speculate that training the agent in a long period of time (7+ years) caused overfitting on wave trends



Algorithm	Minimum	Maximum	Std. Dev.	Median	Mean
DDPG	10155.06	10612.60	170.34	10135.87	10155.06
PPO	9865.74	10386.36	129.19	10135.87	10136.73
TRPO	10729.72	11580.80	248.36	10216.29	11167.20
DJI (Baseline)	-	-	-	-	9989.83
Oracle	-	-	-	-	17676.00

Figure 1: Statistics for resulting portfolio values over 20 episodes of DDPG, PPO and TRPO during a wave trending market from May 1 to August 31, 2019.

Figure 4. The performance of various reinforcement learning algorithms in a particular trending market (May 1 to August 31, 2019). The table displays the statistics for resulting portfolio values over 20 episodes of DDPG, PPO and TRPO. The performance of expected return on the order sequence and variances is depicted in the line chart.