

Homework2: Stock Investment

CE6020 Artificial Intelligence (2022 Fall)

ncu-ai-2022fall-ta@googlegroups.com

Description

- **Task**

- 利用強化學習 (Reinforcement Learning) 進行股票投資決策
- **Environment:** AnyTrading

- **Data**

- 從臺灣證券交易所下載的台積電 (2330.TW) 每日交易資料。

- **Datetime:** 日期

- **Shares:** 成交總股數

- **Amount:** 成交總金額

- **Open:** 當日開盤價

- **High:** 當日最高價

- **Low:** 當日最低價

- **Close:** 當日收盤價

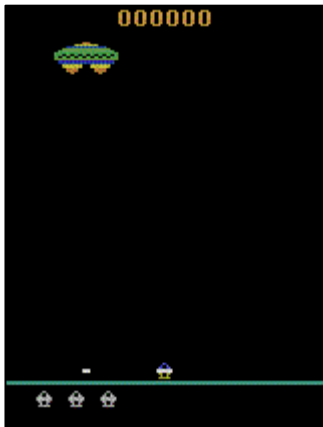
- **Change:** 當日漲跌價差

- **Turnover:** 成交總筆數

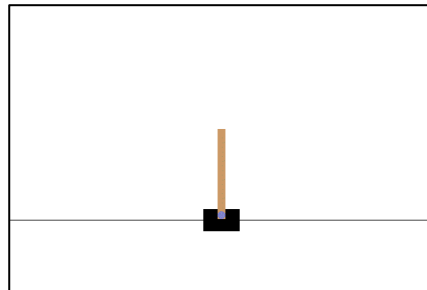
datetime	shares	amount	open	high	low	close	change	turnover
2010-01-04	39511138	2557720928	65.0	65.0	64.0	64.9	0.4	8255
2010-01-05	38394084	2464115096	65.0	65.1	63.9	64.5	-0.4	9205
2010-01-06	52734385	3390698544	64.5	64.9	63.7	64.9	0.4	12597
2010-01-07	53294614	3437221996	64.9	65.0	64.2	64.2	-0.7	11195
2010-01-08	48047497	3068341466	63.5	64.3	63.5	64.0	-0.2	9804
2010-01-11	37446686	2413758179	64.0	64.9	63.5	64.5	0.5	8066
2010-01-12	52444792	3337576396	64.4	64.4	63.3	63.6	-0.9	12865
2010-01-13	49806496	3129817596	63.0	63.1	62.6	62.8	-0.8	10609
2010-01-14	39254264	2469193916	63.6	63.6	63.0	63.2	0.4	8755
2010-01-15	50011109	3155482441	62.9	63.5	62.8	63.5	0.3	11048

OpenAI Gym

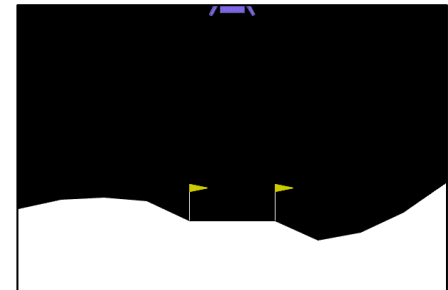
- 開源的強化學習 (Reinforcement Learning) 套件
 - 提供許多環境 (Environment) , 藉以處理環境模擬及獎勵等過程
- <https://www.gymnasium.dev/>



Assault



Cart Pole



Lunar Lander

AnyTrading

- 以 OpenAI Gym 為基底的股票投資環境

- **Environment 規則**

- 只有一家公司的股價可以參考
- 只能擁有一張股票
- 初始本金 = 你買入第一張股票的價格
- **Observation:** 第 $(i - 9)$ 日~ 第 i 日的股價資料 ($i \geq 9$)
- **Action:** 買一張股票 / 賣一張股票

- 若你已經擁有一張股票，而你選擇「買一張股票」，而環境不會讓你購買，反之亦然

- **Total Reward:** 每張股票的買賣價差之總和 (N = 你成功買過幾張股票)

- $\sum_{j=0}^N$ 第 j 張股票的賣出價格 - 第 j 張股票的買入價格

- **Total Profit:** 資金成長比 = $\frac{\text{最終本金}}{\text{初始本金}}$



Submission

- 請將檔案請包成一個 zip 壓縮檔，並繳交至 ee-class 平台
 - 壓縮檔名格式為 學號.zip (e.g. 111525000.zip)
 - 在你的 學號.zip 內至少要有下列 6 類檔案
 - report.pdf 你的報告
 - train.py 或 train.ipynb 訓練主程式
 - test.py 測試主程式 (助教會用這個程式評估你的模型結果)
 - 注意：執行測試主程式的時候只能讀取 model.ckpt，不能再進行額外訓練，否則將視為作弊！
 - model.ckpt 模型 / 參數檔案 (只能繳交一個模型讓助教評分)
 - README.md 說明要如何執行訓練主程式
 - requirements.txt 套件列表
 - Python 版本指定 3.7.x
- **Deadline: 2022/11/10 23:59**
 - 遲交：本次作業成績 = 原始成績 $\times (1 - (\text{遲交天數} \times 0.1))$
 - 檔名錯誤：本次作業成績 = 原始成績 - 5分

Grading Policy

- **Programming (50%)**

- $Score_{Testing} (35\%) = \begin{cases} 35\%, & \text{助教能正確執行你的程式} \\ \leq 25\%, & \text{助教不能正確執行你的程式} \end{cases}$

- $Score_{Ranking} (15\%) = \begin{cases} 15\%, & TotalProfit\ Top\ 1\sim 10 \\ 10\%, & TotalProfit\ Top\ 11\sim 20 \\ 5\%, & else \end{cases}$

- $ProgrammingScore(50\%) = Score_{Testing} (35\%) + Score_{Ranking} (15\%)$

- **Report (50%)**

- **Template Link:**

- https://docs.google.com/document/d/1j4AOR4jfU3ZoEg_bWhNWs4SKPeiwTAaqwDhFQnmB1SQ

- 請標明學號與姓名，並根據題號回答問題。

- 請輸出 A4 大小的 PDF 檔，檔名為 report.pdf (**Report.pdf 是不正確的**)

Report

1.(10%) Policy Gradient 方法

1. 請閱讀及跑過範例程式，並試著改進 reward 計算的方式。
2. 請說明你如何改進 reward 的算法，而不同的算法又如何影響訓練結果？

2.(15%) 試著修改與比較至少三項超參數（神經網路大小、一個 batch 中的回合數等），並說明你觀察到什麼。

3.(15%) 請同學們從 Q Learning、Actor-Critic、PPO、DDPG、TD3 等眾多 RL 方法中擇一實作，並說明你的實作細節。

4.(10%) 請具體比較（數據、作圖等）你實作的方法與 Policy Gradient 方法有何差異，並說明其各自的優缺點為何。

Grading Policy (cont.)

- Programming 評分細節

- 請同學務必參考 **Testing code example** 的輸入/輸出格式，並改寫成 `test.py`
 - 輸入格式：開始日期、結束日期
 - 輸出格式：您的結果檔案（須包含開始日期、結束日期、`Total_Reward`、`Total_Profit`、本次所有動作(action））
- 助教將從台積電 (2330.TW) 每日交易資料集抽樣進行評分
- 從資料集抽樣短、中、長期各 3 筆資料，以 **Total Profit** 作為該筆資料的評估標準
 - 短期=90天、中期=300天、長期=1000天（抽樣的區間將在 **Deadline** 後擇日公布）
 - 考慮到每次執行結果不同，同筆資料將執行 10 次並取平均
- 助教在批改程式部分時，會執行以下指令：

```
python3 test.py -s {StartDate} -e {EndDate} -f {OutputFile}
```

- StartDate: Stock investment start date index
- EndDate: Stock investment end date index
- OutputFile: Output action file path name
- 執行範例：`python3 test.py -s 500 -e 1000 -f output.txt`
- 執行環境：Ubuntu 20.04 (Python 3.7.15)

Overview

- 利用強化學習 (Reinforcement Learning) 進行股票投資決策
- 必須在 AnyTrading 這個環境進行股票投資
- 允許同學以範例程式碼為基礎進行修改
- Training code example:
<https://colab.research.google.com/drive/1zmx4INeyBV24krXjmW-h8O7cPDD-VFOS>
- Testing code example:
<https://colab.research.google.com/drive/1ZhWZkx5N77QWrfZ2MUa4s9Wt9dIFPFGC>
- **Deadline:** 2022/11/10 23:59

Reference / Resource

- [Gym Documentation](#)
- [Github: AminHP/gym-anytrading](#)
- [Github: Ryan906k9/stock_pg](#)
- [Github: poteman/ReinforcementLearning_for_stock](#)
 - 註：此程式的環境不是 AnyTrading，僅供參考