組員：S09350246 張子中、S09350211 陳昱樺

摘要

研究背景與動機：

在這個通貨膨脹的時代，金錢的使用變的日益重要，但若只是將錢存放在帳戶或口袋裡，在不知不覺中，錢的使用價值便會相對減少，所以越來越多人會利用投資理財、買股票來增加本身的資產，本次研究希望透過強化學習來打造能賺錢的機器，並預估其平均受益。

研究目的：

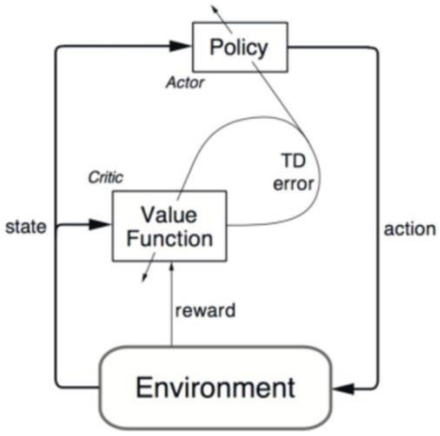
研究強化學習模型在投資股票中的效益。

研究方法

Policy

actor-critic policy …

訓練過程中，一個是 Value function，一個是 Policy，其中 Value function 輸入 state ，輸出 scalar。而 Policy 的話是輸入 state，輸出 action 的分配。因為 Value function 和 Policy 的 input 都是 state，因此前面幾個 layer 的話是可以共用的。



利用 value function 和 policy 兩類方法各自的優勢產生了集大成的 Actor-Critic 類方法。這是一個全能型的 agent，既能直接輸出策略，又能通過 value function 來實時評價當前策略的好壞。兩者都還在不斷更新，這種互補式的訓練方式會比單獨的策略網絡或者值函數網絡更有效。

MlpPolicy：實現 actor-critic 的 Policy 對象。 在這種情況下，將使用 2 層 64 層的多層感知器。另外還有視覺信息策略，例如 CnnPolicy 甚至 CnnLstmPolicy 可選。

Model

PPO(Proximal Policy Optimization, 近端策略優化)

PPO 算法是一種新型的 Policy Gradient 算法，Policy Gradient 算法對步長十分敏感，但是又難以選擇合適的步長，所以在訓練過程中新舊策略的的變化差異過大則不利於學習。PPO 提出了新的目標函數可以再多個訓練步驟實現小批量的更新，解決了 Policy Gradient 算法中步長難以確定的問題。

在 PPO 里面的 Important sampling 採樣過程能然是在同一個策略生成樣本，

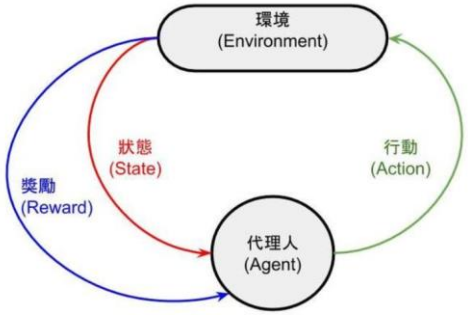
Proximal Policy Optimization (PPO)

$$\nabla f(x) = f(x)\nabla log f(x)$$

$$J_{PPO}^{\theta'}(\theta) = J^{\theta'}(\theta) - \beta KL(\theta, \theta')$$

$$J^{\theta'}(\theta) = E_{(s_t,a_t)\sim\pi_{\theta'}}\left[\frac{p_\theta(a_t|s_t)}{p_{\theta'}(a_t|s_t)} A^{\theta'}(s_t, a_t)\right]$$

並未使用其他策略生成的樣本，所以他是 on-policy 的，而 on-policy 指的是指與相同的環境下進行交互學習。

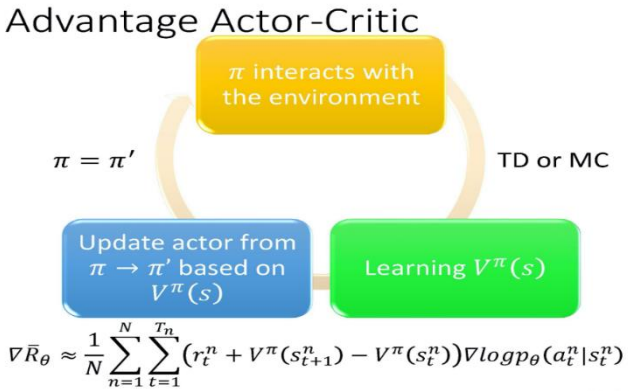


A2C(Advantage Actor Critic)

在強化學習中，一個 Agent 會透過採取行動，在環境里的不同狀態之間移動，並在移動中增加獎勵。

在這過程中會建立策略函數，可以讓代理人決定在目前所在的狀態中改採取何種行為。而策略是一個隨著時間而變動的函數，會因為每個時間點的狀態映對到的行為，所獲得的獎勵而不斷改變策略的選擇方式。每組狀態對行為的映對，所獲得的獎勵也會因為策略的變化而改變。建構一個能將獎勵與行為有效關聯的策略函數，在強化學習中被稱為**獎勵分配問題**（credit assignment problem）。

Actor 可以是類似於神經網絡的函數逼近器，其任務是針對給定狀態產生最佳動作。它可以是全連接的神經網絡，也可以是卷積或其他任何東西。Critic 是另一個函數逼近器，它接收 Actor 輸入的環境和動作作為輸入，將它們連接起來並輸出評分值（Q 值本質上是將來的最大獎勵。）



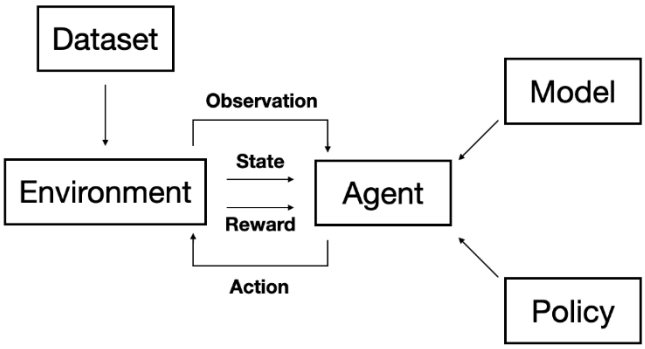
A2C 計算方式

資料集

我們採用 panda 的 API 從”Yahoo”爬取台積電股票(2330.Tw)近十年資料，日期由 2012-01-02 至 2022-5-27 ，其欄位包含「日期、最高價、最低價、開盤價、收盤價、成交量」，整個資料及為 2541rows*5columns。

實驗設計

Environment:



初始設定與限制:

MAX_ACCOUNT_BALANCE = 2147483647
MAX_NUM_SHARES = 2147483647
MAX_SHARE_PRICE = 5000
MAX_OPEN_POSITIONS = 5
MAX_STEPS = 20000
INITIAL_ACCOUNT_BALANCE = 10000

Reward:

Reward = balance*目前進度/ MAX_STEPS
短期投資方面：我們考慮當前價格的變動改變投資組合
長期投資方面：我們在每個步驟中，將帳戶餘額乘以到目前為止的進度
這麼做的目的是為了推遲 agent 在早期階段賺快錢，可使 agent 在過度深入探索相同策略前進行充分的探索，並使得 agent 走得越長遠獎勵越高

Observation:

包含近五天的 Date, Open, High, Low, Close, Volume
以及目前資金、總價值、持有股數、持有股均價

Reset:

為了學習如何交易我們的機器會常常重置，重置除了初始化資金表之外，我們讓下次機器開始訓練的位置為隨機數，只要達成以下條件環境就會重置：

1. Balance<=0
2. Balance>MAX_ACCOUNT_BALANC
3. currentStep>總 data 步數-6

Agent

Model: PPO,A2C（請參見研究方法）

Policy: MLP policy(2*64)（請參見研究方法）

State:

每一步 Agent 透過 model 參考 Observation 決定 action_space 採取行動，計算 reward 並返回下一個觀察結果，如果達成 reset 條件則記錄下當前資訊並啟動重置。

Action:

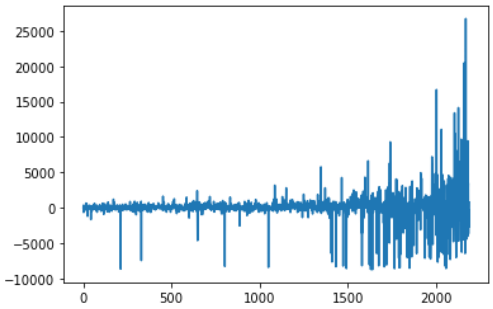
Agent 根據 action_space 做出行動，action_space 為 1*2 的矩陣
第 1 項為 action_type，第 2 項為 amount，若 action_type<1 則買入，若 1<action_type<2 則賣出，第二項 amount 決定買入賣出的量，隨後並更新目前資金表。

績效衡量

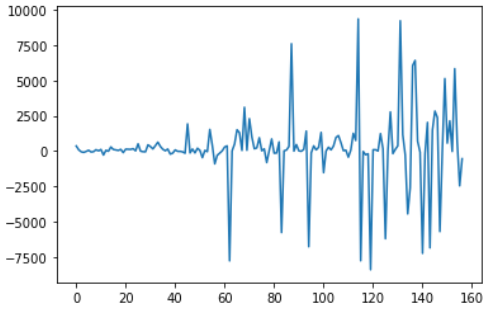
我們讓 Agent 學習 20 萬次，並記錄下每個重置前的利潤，我們將這些取得的利潤加總除以個數取得平均，以當做該模型獲利的能力。

研究成果

PPO



A2C



備註：X 軸為資料數，Y 軸為利潤，初始資產為 10000。

PPO:共 2190 筆資料，總利益 103477，平均利潤 47.25。

A2C:共 157 筆資料，總利益 20176，平均利潤 128.51。

雖然 A2C 資料數明顯較 PPO 少，但根據多次比較 A2C 的平均效益皆比 PPO 好，至於為什麼 A2C 資料會較少，可能是 A2C 模型在決策時有更長的投資策略，使得其在固定 2 萬次的訓練中得到的資料筆數偏少。

參考資料

劉上瑋 2017《深度增強學習在動態資產配置上之應用-以美國 ETF 為例》
https://ah.nccu.edu.tw/bitstream/140.119/114285/1/202901.pdf

定制股票交易 OpenAI Gym 强化学习环境
https://mp.weixin.qq.com/s?__biz=MzAxNTc0Mjg0Mg==&mid=2653291519&idx=1&sn=10a41e9889d065d5f330750750d43863

[Day-29] 增強式學習 (DQN) - 股票操作
https://ithelp.ithome.com.tw/articles/10228127

Sudharsan Ravichandiran 《用 Python 實作強化學習使用 Tensorflow 與 OpenAI Gym》

深度学习用于股票预测_用于自动股票交易的深度强化学习
https://blog.csdn.net/weixin_26704853/article/details/108515764

Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. 2020. Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy. In ICAIF '20: ACM International Conference on AI in Finance, Oct. 15–16, 2020, Manhattan, NY. ACM, New York, NY, USA.

人工智慧- Actor Critic：
https://www.wpgdadatong.com/tw/blog/detail?BID=B2535