利用強化學習實現金融配對交易策略

一、 摘要

儘管市場上針對配對交易有著各種不同的交易策略與統計工具，但是在傳統經理人的操作下，仍然存在著許多缺失。例如，傳統經理人對股價波動、兩股價差的預測都會大大影響配對交易的績效。此外，人為對金融市場的操作中，經常融入許多主觀的見解與情緒，最終做出不理性的決策。

透過強化學習演算法 (Q Learning)，搭配不同的神經網路 (Deep Q Network、Double Deep Q Network) 訓練出能夠對股價波動性與兩股價差進行預測的人工智慧代理人。透過兩個代理人各司其職，進而組成能夠實行配對交易策略的系統，並從中獲利。

參考的論文是用一般的 DQN，使用兩層 Dense 就得到 output，而且餵入的資料也僅僅只有一天，由於股票市場是連續的，我們設想每天的資料都會互相影響，因此設計了另一個用 Convolution 1D 的 DQN，一次看 5 天的資料，看是否能找到更多的特徵，比原 Paper 的表現還要更好。

二、 研究動機與研究問題

I. 配對交易的基本概念

配對交易(Pairs Trading)又稱為價差交易或是統計套利，它提供投資人在二種相關性資產間的交易契機，而所依循的價格數理邏輯很簡單，只要二種資產(或證券)在價格上具備高度相關，假設此股價的數學關連性存在，縱使短期這個數學關係中斷，經過一段時間後必將恢復，而中斷的時間就存在套利交易的空間了！

此策略不用對市場看多或看空，因此是一種中性策略。最常看到的就是外匯配對、貴金屬配對、股市指數配對、股票也可以配對。這樣的交易方式常被對沖基金(Hedge Fund)與金融機構自營部所運用。因為這樣的交易獲利仰賴的是統計學，所以公認是最簡單(不用分析)可機械化操作(不具恐懼與貪婪)的最佳策略。

- 優點：
  - 此為中性策略(Market Neutral)，無須判斷市場方向
  - 策略仰賴歷史統計與回歸均值的特性，不受個人主觀判斷影響
  - 只要設定好參數，買賣訊號出現即可機械化交易，也可以加入加減碼策略

- 風險較單一方向買入或放空來的低很多。

- 缺點：
  - 有些股票不容易找到好的配對股票
  - 股價比例可能突然產生長期趨勢，兩家公司競爭力明顯不對稱
  - 股價比例可能在布林通道邊緣來回穿越產生反覆訊號
  - 停損時機較難判斷

II. 傳統方式操作配對交易

- 確定股價具有相關性 High Correlation：
  將兩家公司股價重疊，從重疊的股價圖可以看出大略趨勢。除此之外，因為產業相同、產品互相競爭，所以兩家公司都受到景氣的影響大致相同。在選擇配對公司時，最好也選擇市值不要差異太大(Market Cap.)，兩家公司的營運已經上軌道，並且有多年的歷史股價的最好。也可以使用統計上的 Cointegration Test，計算出有 Pair 關係的股票。

- 計算兩家公司的股價比例歷史 Stock Price Ratio：
  將兩家公司，歷史資料同一天的股價相除，得到 Stock Price Ratio，看兩者股價的比率來決定進退場時機。

- 使用布林通道來確認訊號：
  * 布林通道：由上線(壓力線)、中線、下線(支撐線)組成
  *中線：20 期移動平均線
  上線：中線+2 個標準差
  下線：中線 -2 個標準差
  當股價超出上下線後，建立配對交易的訊號，並使用此時的股價比例來作等值交易(作空價格高的，買進價格低的，並依比例操作，使兩者等價)，股價比例回歸中線即為平倉的訊號。

III. 傳統方式操作配對交易的困境

傳統的配對交易已經行之有年，越來越多人使用這個策略投資，在高頻交易者也使用這個策略後，一般大眾要用這個方法獲利可說是難上加難，所以如果能用電腦加以輔助，這樣大家又能在同一個起跑點上，因此越來越多人選擇使用強化學習，希望能更早看出 Pair 的趨勢，繼續使用 Pairs Trading 來獲利。

IV. 應用強化學習於配對交易上

承如上述所說的，傳統響經理人在操作配對交易時，會納入考慮因素非常多。舉例來說，經理人可能會盡其所能的去考慮不同的時間長

度的兩檔股票歷史價格數據、各種不同的停損點或是交易時機，等等。

如果將各種不同因素做排列組合，將使得傳統經理人能夠選擇的依據高達上千種。在高達上千種可選取的依據與多變的金融環境下，傳統經理人勢必無法在所有情況下窮舉所有依據並查看績效。

此外，預測兩檔股票未來的價差，也是傳統經理人在操做配對交易必須考量的因素。然而，兩檔股票未來的價差受到種種因素影響，傳統經理人單純使用統計工具與公式進行計算，可能會導致最後的決定有所偏頗。

最後，傳統經理人在操作金融商品時，難免會有情緒的介入，導致無法理性的做出決策，讓結果不如預期。

綜合以上敘述，可以發現僅透過傳統經理人來操作配對交易時，仍然存在一些缺失。因此，希望透過強化學習訓練出能夠實踐配對交易的代理人。

要在多變的金融環境下，從上千種可選取的依據中，選出最好的依據的問題，可以視為強化學習中「一台多臂老虎機」的問題。讓代理人學習嘗試在不同的環境下，選擇適當的依據，並透過 Q-Learning 來提升預測兩檔股票未來價差的準確性，才能夠做出比較適當的行為，最後才能在配對交易中獲利。

透過強化學習訓練代理人操作配對交易的過程中並不會有情緒的介入，這樣可以使得整個決策過程更加理性與穩定。

三、 文獻回顧與探討

I. *Deep Reinforcement Learning Pairs Trading with a Double Deep Q-Network*

- 目的：
  本篇論文主要透過利用 Double Deep Q-Netwok 實作配對交易中預測兩股票未來價差的部分。也就是說，這個系統主要是在預測兩檔股票的「價差」的變化。當價差即將變大時，系統會輸出「購買訊號」；反之，當系統預期價差會變小時，就會輸出「賣出訊號」。

- 資料：
  從 S&P500 中透過兩種方式，篩選出 38 組股票。第一種方式：確保挑選出來的 Pair 具有 Mean-Reverting 特性，所以去

計算 500 檔股票兩兩之間的 Augmented Dickey-Fuller p-value，來知道兩兩股票之間的 Cointegration。第二種方式：確保挑選出來的 Pair 有一定的 Variance 才可以進行交易，所以將剩下的股票兩兩進行配對，一個 Pair 中的股票的標準差除以平均值必須大於等於 0.5。
因為系統為了讓累積報酬變得更高，就會一直去進行交易。為了讓系統更為保守，利用 Negative Rewards Multiplier (NRM) 讓 系統交易後得到的 Negative return 必須再乘以 NRM，使得系統的損失更大。如此一來，系統就會比較保守。

- 流程與組成：

  必須先建立環境，並確定 State 與 Action Space。同時，必須設計好環境給 Agent Reward 的公式。最後則是設計 Agent 的架構。

  - 環境：利用 OpenAI Gym 中的某一個金融環境。

  - State Space：
    - ✓ Current Spread of Pair
    - ✓ Daily Return of Spread
    - ✓ Spread Mean during Past 15 Days
    - ✓ Current Spread Mean / Spread Mean during Past 15 Days
    - ✓ Spread Mean during Past 10 Days
    - ✓ Current Spread Mean / Spread Mean during Past 10 Days
    - ✓ Spread Mean during Past 7 Days
    - ✓ Current Spread Mean / Spread Mean during Past 7 Days
    - ✓ Spread Mean during Past 5 Days
    - ✓ Current Spread Mean / Spread Mean during Past 5 Days

  - Action Space：
    - ✓ Long：預測 Spread 即將變大
    - ✓ No-Position：Risk Averse
    - ✓ Short：預測 Spread 即將變小

  - Reward Function：
    - ✓ Training Period
      - ◆ T = a * r * n
      - ◆ T = Training Reward

- a = Action Output by Agent
- r = Spread Return
- n = Negative Reward Multiplier

✓ Testing Period

- R = a * r
- R = Testing Reward
- a = Action Output by Agent
- r = Spread Return

可以發現 Negative Reward Multiplier 只出現在 Training 過程，主要是希望 Actor 在訓練時可以具有 Risk Averse 特性。

■ Agent 架構：Agent 由 Double Deep Q Network 實現。換言之，Agent 由 Evaluation Network (Critic) 與 Target Network (Critic) 所組成。Agent 的運作流程如下：

✓ Evaluation Network 接收環境給的 State 後，選出一個 Q Value 最大的 Action。
✓ 環境接收 Action 後，計算 Reward 與產生下一個 State。
✓ 將這一步的所有資訊（State0、Action0、Reward0、State1）存入 Replay Buffer 中。
✓ 從 Replay Buffer 中 Sample 一些 Experience，訓練 Evaluation Network。
✓ 將 Evaluation Network 的參數複製到 Target Network。
✓

■ Neural Network 架構：Evaluation Network 與 Target Network 的組成如下：

✓ Input Layer of 10 Features
✓ Fully Connected Layer of 50 Nodes with RELU Activation Function
✓ Another Fully Connected Layer of 50 Nodes with RELU Activation Function
✓ Output Layer of 3 Actions
✓ Model with MSE as Loss Function
✓ Model with Adam as Optimizer

- 結果：

  由下圖的統計圖表中可以發現當 Negative Reutrn Multiplier 為 1 時，Agent 在許多 Pair of Stock 的 Spread 預測上，都能有不錯的表現。但是當 Negative Return Multiplier 愈大時，象徵 Agent 犯錯時受到的懲罰會更大，導致 Agent 會傾向做出更保守的行為，也就是不去做預測，導致許多 Pair 的 Return 多為 0。

Negative Returns Multiplier	1	2.5	5	10	20	50	100	200	500	700	1000
Stock Pair											
BEN_COG	1.23	-0.27	0	1.13	0.27	0.18	0	0	-0.16	0	0.48
DISCA_RIG	-0.22	-0.54	-0.53	0	0	0	0.01	-0.14	0	0	0
DISCK_RIG	1.25	0	1.11	0.45	0.01	-0.14	0	0	-0.14	0	0
ADBE_CRM	1.07	0.09	-0.02	0	0	0	0	0	0	0	0
CF_HBI	0.25	-0.04	-0.14	0.31	0	0	0	0	0	0	0
ESV_GNW	-9.64	-11.54	-11.62	-2.3	-0.32	-11.5	0.89	-9.95	-11.5	11.67	-0.13
CNX_HBI	7.52	8.46	5.78	8.5	11.9	1.42	4.61	3.42	0.44	0	0
AMZN_CRM	0.56	-0.03	-0.01	-0.01	0	-0.07	0	0	0	0	0
MA_VFC	0.01	-0.18	-0.22	-0.26	-0.27	-0.06	0	0	0	0	0
FCX_GNW	-0.19	0	0	0	0	0	0	0	-0.05	0	0
CRM_NVDA	-0.54	-1.22	-3.29	2.94	-0.43	-0.96	-3.56	-3.15	-3.15	0	0
CF_FOSL	-1	-0.04	0.12	0	-0.01	0	0	0	0.02	0.12	0
FCX_HBI	25.67	26.55	17.34	16.12	20.87	22.52	22.27	21.28	20.92	-1.62	0.54
DISCK_ESV	0.08	0	0	-0.18	0	0	0	0	0	0	0
DISCK_NE	-0.08	0	0	0	0	0	0.09	0	0	0	0
DISCA_NE	-0.25	-0.17	0	0	0.12	0	0	0	0	0.03	0
DISCA_ESV	-0.56	-0.16	-0.21	0	0	0	0	0	0	0	0
ESV_RRC	-0.78	0.29	0.15	0.05	-0.28	-0.03	0.2	0	0.11	0.17	0
NBL_RIG	1.14	0.24	-0.04	-0.02	-0.02	0	0.04	0	0	0	0
CNX_GNW	-0.04	0.19	-0.06	0.14	-0.02	0	0.01	0.29	0.03	0	0
COG_DO	2.32	-0.71	0.51	0.54	0.14	-0.97	-0.22	-0.63	-0.41	0	1.06
HBI_NBL	0.24	0.01	0	0.07	0.03	0	0	0	0	0	0
HBI_MRO	27.41	8.6	16.15	29.83	10.68	3.3	13.16	0.6	-0.5	-6.23	0

- 探討：

  此篇文獻透過 Double Deep Q Network 訓練出能夠預測 Pair of Stock 的 Spread 趨勢，但是仍無法真正讓 Agent 在某個時間點對 Pair of Stock 進行更具體的操作。換言之，Agent 尚無法在某個時間點「買 A 賣 B」、「賣 A 買 B」或「不操作」。認為若此 Agent 可以預測 Pair of Stock 的 Spread，應該可以延伸此 Agent 實際對 Pair of Stock 進行更具體的操作，進而實行 Pair Trading 的理念。

四、 研究方法及步驟

I. 研究 Negative Return Multiplier 對 Spread Return 預測的影響（DQN 與 DDQN）

為了實作透過強化學習來訓練 Agent 能夠真正操配對交易，先模擬 *Deep Reinforcement Learning Pairs Trading with a Double Deep Q-Network* 中所述的環境以及 Agent 訓練方式，建立一個可以預測兩檔股票未來價差的 Agent。

- 資料收集：使用 AAPL 與 GOOG 從 2012/9 到 2017/9 的股價（收）資料。

- 建立環境：State Space、Action Space 與 Reward Function 皆採取 Paper 中的設定。

- 建立 Deep Q Network Agent：相較於 Paper 中使用的神經網路架構：

  - Input Layer of 10 Features
  - Fully Connected Layer of 50 Nodes with RELU Activation Function
  - Another Fully Connected Layer of 50 Nodes with RELU Activation Function
  - Output Layer of 3 Actions with LINEAR as Activation Function
  - Model with MSE as Loss Function
  - Model with Adam as Optimizer

  這裏以更簡單的神經網路架構實現相同的結果：

  - Input Layer of 10 Features
  - Fully Connected Layer of 20 Nodes with RELU Activation Function
  - Another Fully Connected Layer of 10 Nodes with RELU Activation Function
  - Output Layer of 3 Actions with LINEAR as Activation Function
  - Model with MSE as Loss Function
  - Model with Adam as Optimizer

- 建立 Double Deep Q Network Agent：相較於 Paper 中使用的神經網路架構：

  - Input Layer of 10 Features
  - Fully Connected Layer of 50 Nodes with RELU Activation Function
  - Another Fully Connected Layer of 50 Nodes with RELU Activation Function
  - Output Layer of 3 Actions with LINEAR as Activation Function
  - Model with MSE as Loss Function

- Model with Adam as Optimizer

這裏以更簡單的神經網路架構實現相同的結果：

- Input Layer of 10 Features
- Fully Connected Layer of 20 Nodes with RELU Activation Function
- Another Fully Connected Layer of 10 Nodes with RELU Activation Function
- Output Layer of 3 Actions with LINEAR as Activation Function
- Model with MSE as Loss Function
- Model with Adam as Optimizer

II. 研究 Negative Return Multiplier 對 Spread Return 預測的影響（將 DQN 前半部加上 Convolution Layer 1D）

- **神經網路架構：**

  - Input Layer of 10 Features * 5 feature length(5 天的資料)
  - Convolution 1D (10,3)
  - Flatten 拉平
  - Fully Connected Layer of 5 Nodes with RELU Activation Function
  - Output Layer of 3 Actions with LINEAR as Activation Function
  - Model with MSE as Loss Function
  - Model with Adam as Optimizer

  比較的神經網路架構(原 Paper 作法)：

  - Input Layer of 10 Features Convolution 1D (10,3)
  - Fully Connected Layer of 50 Nodes with RELU Activation Function
  - Fully Connected Layer of 50 Nodes with RELU Activation Function
  - Output Layer of 3 Actions with LINEAR Activation Function
  - Model with MSE as Loss Function
  - Model with Adam as Optimizer

III. 利用 Double Deep Q Network 建立一個系統實現配

對交易

- 資料收集：使用 AAPL 與 GOOG 從 2012/9 到 2017/9 的股價（收）資料。

- 建立環境：延伸 Paper 中使用的 State Space、Action Space。Reward Function 則是真實計算 System 進行 Pair Trading 操作時獲得的 Profit。

  - State Space:

    - ✓ current stock1 price
    - ✓ current stock1 state
    - ✓ number of units of stock1 which system holding
    - ✓ stock1 unit price
    - ✓ current stock2 price
    - ✓ current stock2 state
    - ✓ number of units of stock2 which system holding
    - ✓ stock2 unit price
    - ✓ current spread
    - ✓ spread return
    - ✓ spread mean during past 15 days
    - ✓ current spread / spread mean during past 15 days
    - ✓ spread mean during past 10 days
    - ✓ current spread / spread mean during past 10 days
    - ✓ spread mean during past 7 days
    - ✓ current spread / spread mean during past 7 days
    - ✓ spread mean during past 5 days
    - ✓ current spread / spread mean during past 5 days

  - Action Space:

    - ✓ [Pattern, [Stock1 Quantity, Stock2 Quantity]]

      - Pattern: 0 (Buy Stock1 Sell Stock2)
      - Pattern: 1 (Sell Stock1 But Stock2)
      - Pattern: 2 (No Operation)

- 建立 Double Deep Q Network for Pattern Agent :

  - Input Layer of 18 Features
  - Fully Connected Layer of 30 Nodes with RELU Activation Function
  - Another Fully Connected Layer of 24 Nodes with RELU Activation Function
  - Another Fully Connected Layer of 20 Nodes with RELU Activation Function
  - Another Fully Connected Layer of 12 Nodes with RELU Activation Function
  - Another Fully Connected Layer of 8 Nodes with RELU Activation Function
  - Output Layer of 3 Actions with LINEAR as Activation Function
  - Model with MSE as Loss Function
  - Model with Adam as Optimizer

- 建立 Double Deep Q Network for Quantity Agent :

  - Input Layer of 19 Features
  - Fully Connected Layer of 225 Nodes with RELU Activation Function
  - Another Fully Connected Layer of 196 Nodes with RELU Activation Function
  - Another Fully Connected Layer of 169 Nodes with RELU Activation Function
  - Another Fully Connected Layer of 144 Nodes with RELU Activation Function
  - Output Layer of 100 Actions with LINEAR as Activation Function
  - Model with MSE as Loss Function
  - Model with Adam as Optimizer

- 流程：

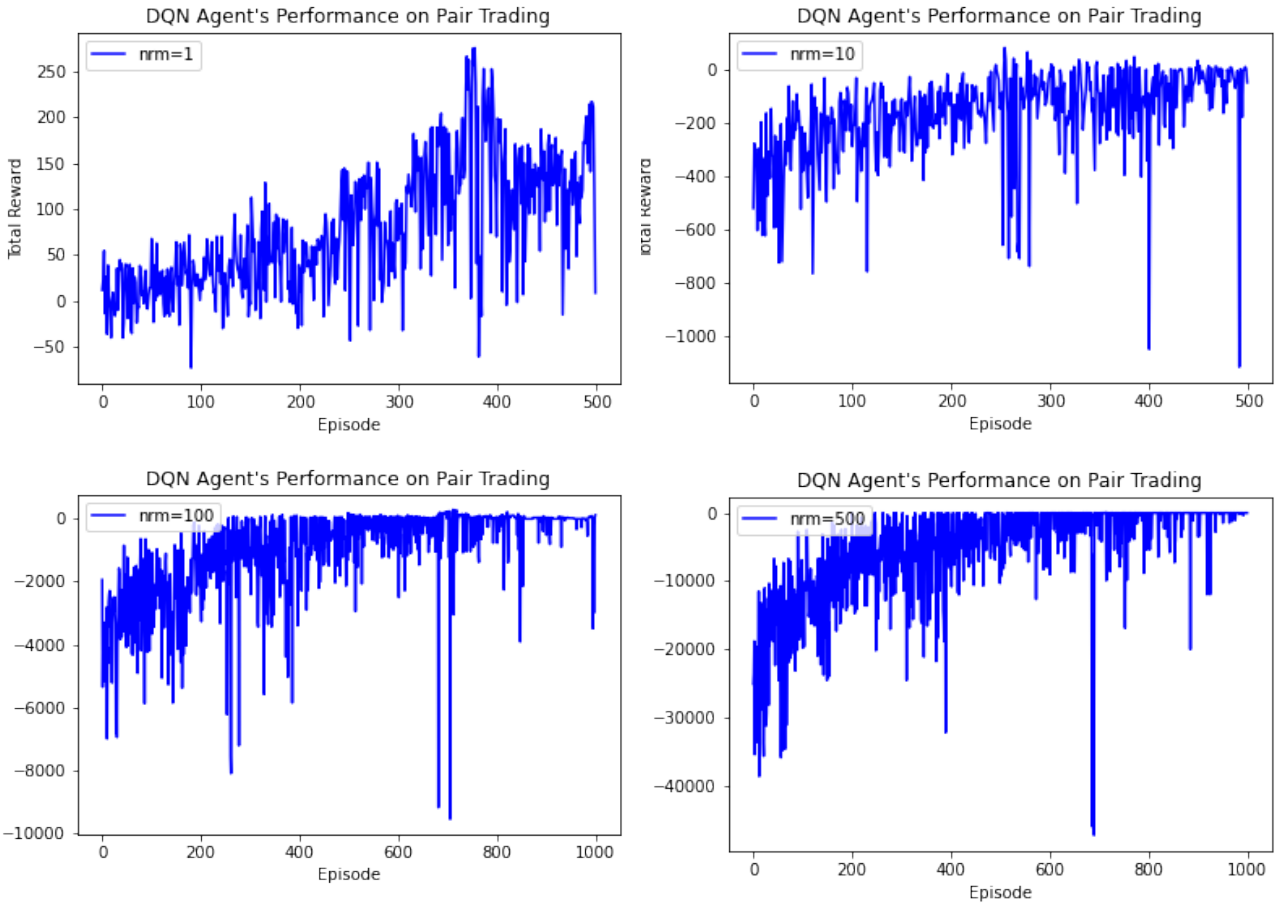
  - 建立一個 System 由 Pattern Agent 與 Quantity Agent 組成
  - 將環境的 State 提供給 System
  - System 得到 State 後，提供給 Pattern Agent 得到目前要採取的 Pattern。
  - 連同原本的 State 再加上 Pattern Agent 輸出的 Pattern 提供給 Quantity Agent 後，得到針對 Stock1 與 Stock2 要操做的數量。

- 將 Pattern、Quantity 1 與 Quantity 2 合成一個 Action 後，提供給環境。
- 環境得到 Action 後，會計算一個 Reward 並產生下一個 State。

五、 實驗結果

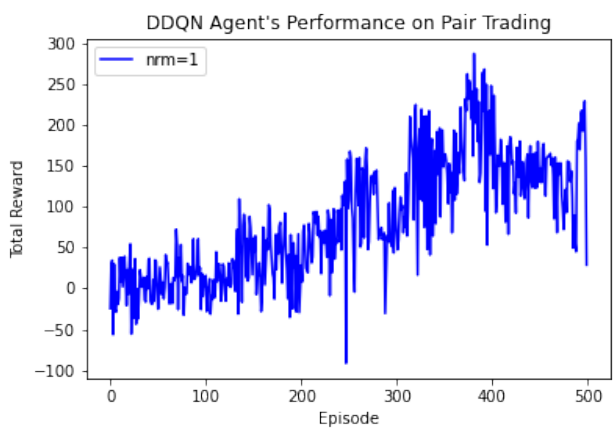
I. 研究 Negative Return Multiplier 對 Spread Return 預測的影響（DQN 與 DDQN）

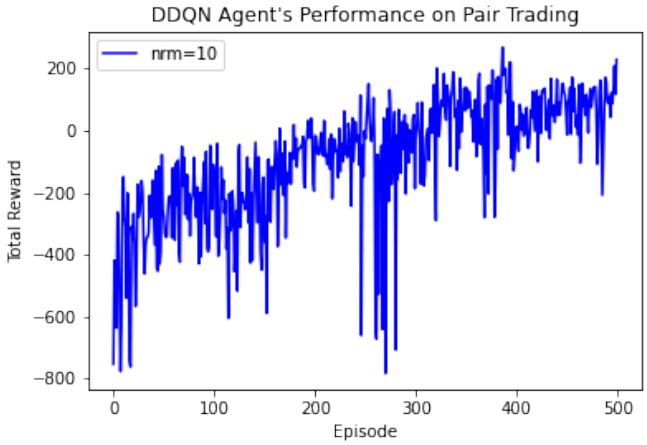
➢ Deep Q Network 結果：

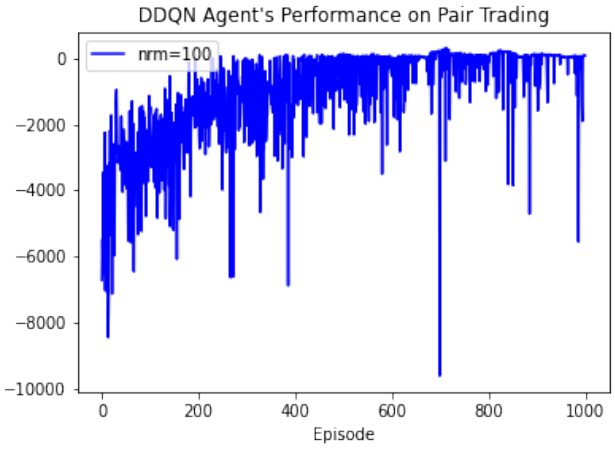


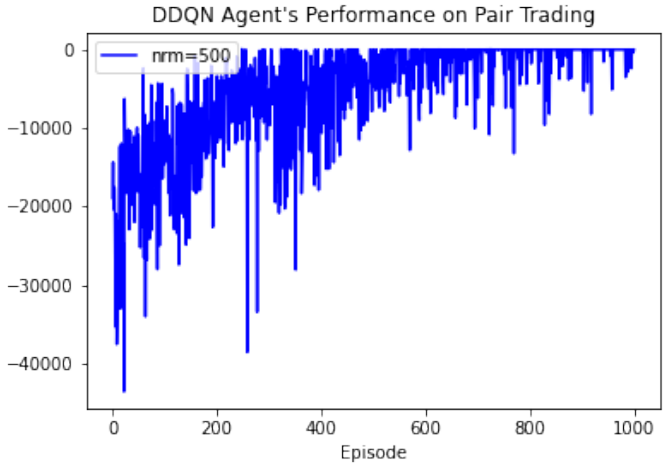
由上面四張圖可以發現，當 Negative Reward Multiplier 為 1 的時候，Agent 的 Performance 會愈來愈好。但是當 Negative Reward Multiplier 愈大時，Agent 會愈來愈保守，最後只學會不要做任何操作。

➢ Double Deep Q Network 結果



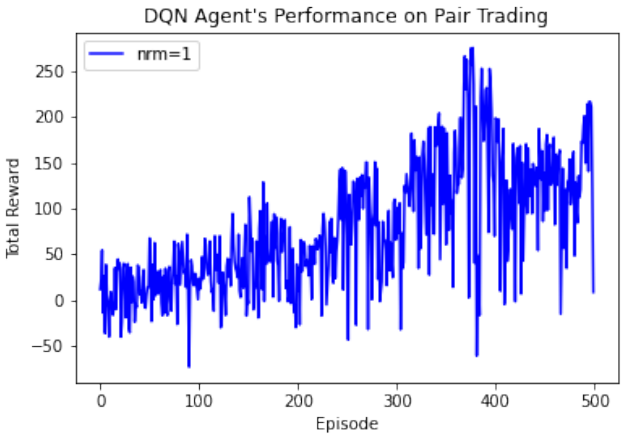


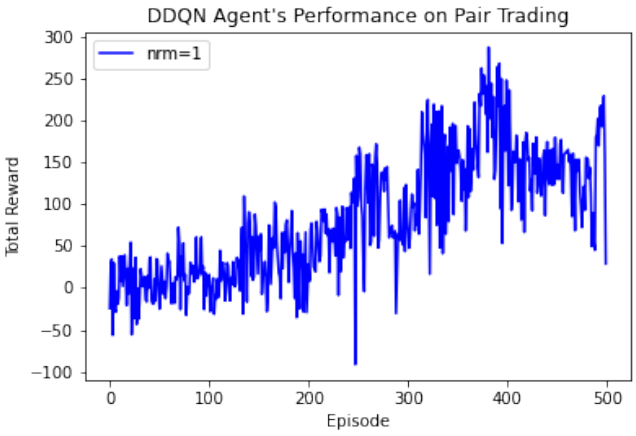




由上面四張圖可以發現與 Deep Q Network 一樣的結果。當 Negative Reward Multiplier 為 1 的時候，可以發現 Agent 的表現愈來愈好。但是當 Negative Reward Multiplier 變大時，表現出 Agent 具有 Risk-Averse 的特性。

➢ Deep Q Network vs Double Deep Q Network 結果比較





以上使用 Negative Return Multiplier 為例，可以發現 Double Deep Q Network 解決了 Deep Q Network 中高估 Action 的 Q Value 的問題。因此，Double Deep Q Network 的 Total Reward 相對比較收斂，且表現更好。

II. 研究 Negative Return Multiplier 對 Spread Return 預測的影響（將 DQN 前半部加上 Convolution Layer 1D)

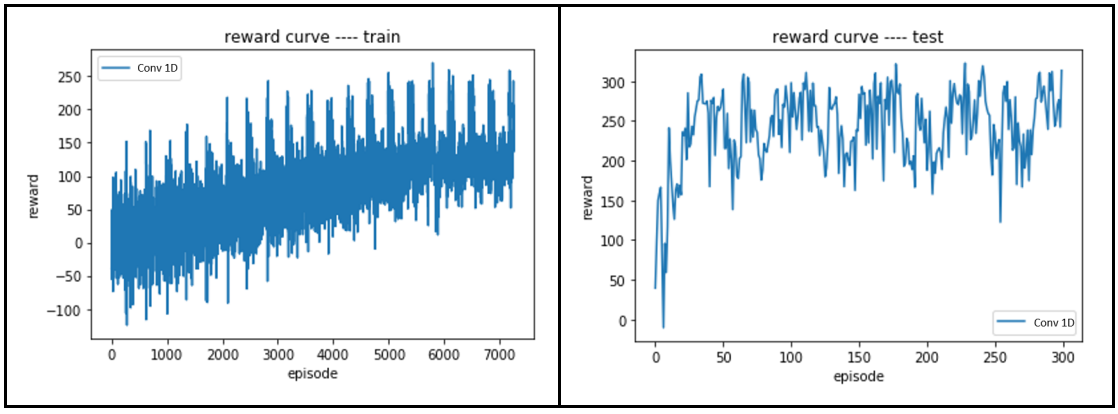
以下的實驗參數設定皆相同，有變的只有 nrm 的數值，Train 的資料為 現有資料的前 400 筆資料(2012/9/4~2014/4/8)；Test 的資料為 現有資料的後 100 筆資料(2017/4/12~2017/9/1)。

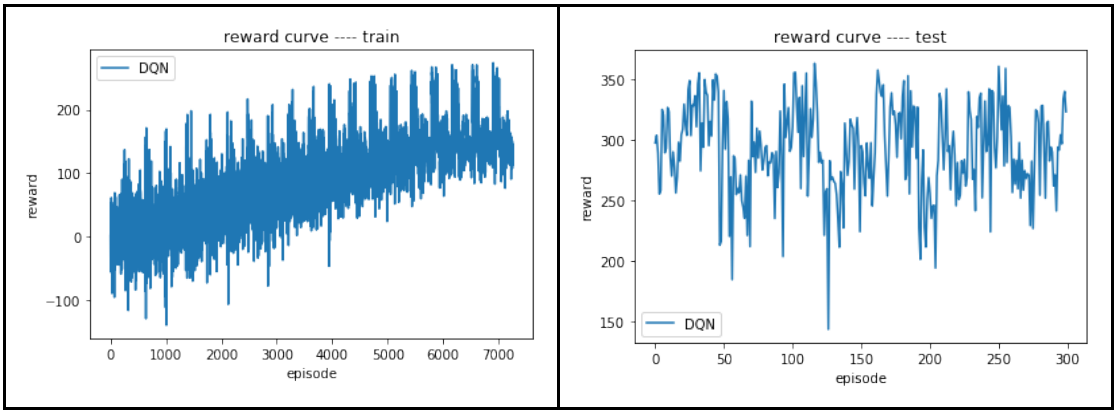
這樣設計的理由是，若能學習過去的資料後，就在較新的資料獲得不錯的成果，那就表示只要股票市場沒有太誇張的變動(Ex: 金融海嘯、武漢肺炎)，那麼就可以用相同的 Agent 去做決策。

Train 的部分為 400 筆資料重複訓練約 20 次，Test 的部分為 100 筆資料重複訓練約 5 次。

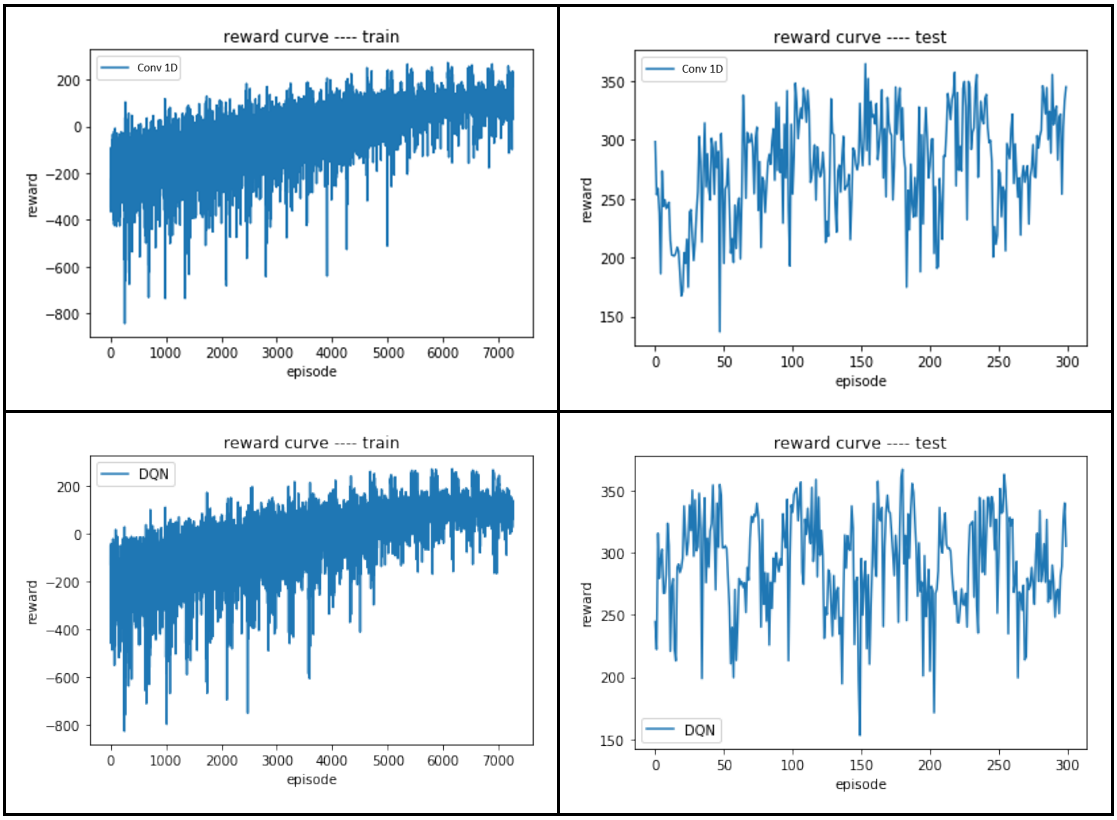
原先 Paper 的設計只有一般 Dense 的 DQN，但考慮到股價間可能會有時間關係，只看一天的資料或許不足，因此又設計了一個 Convolution 1D 的 DQN，觀察是否會有較佳表現。

NRM=1

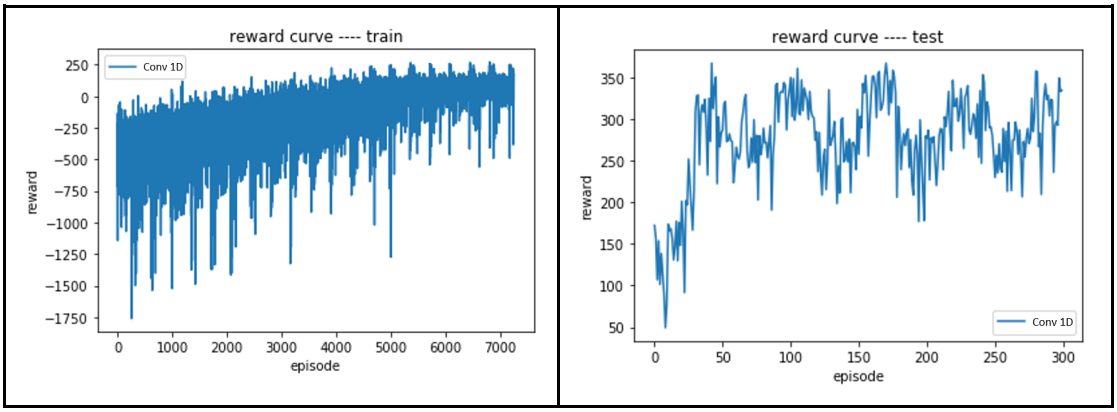


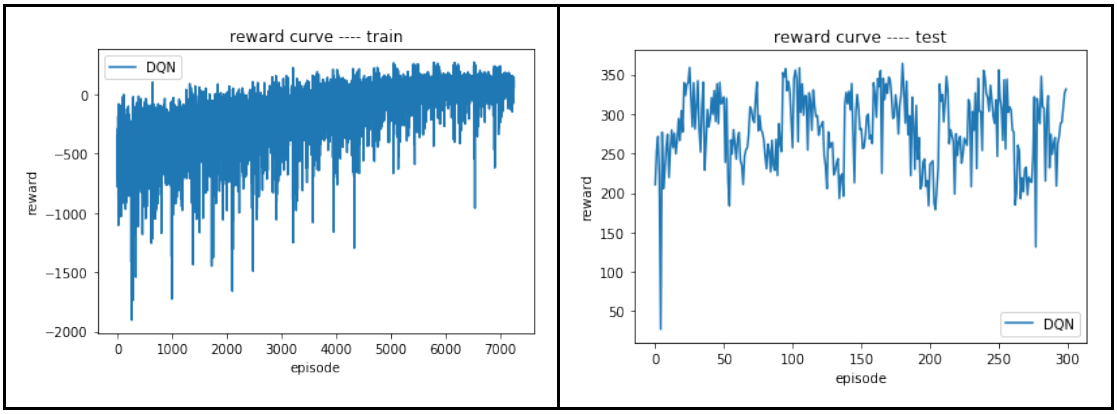

reward curve ---- train
DQN
reward
episode
reward curve ---- test
reward
episode
DQN

NRM=5

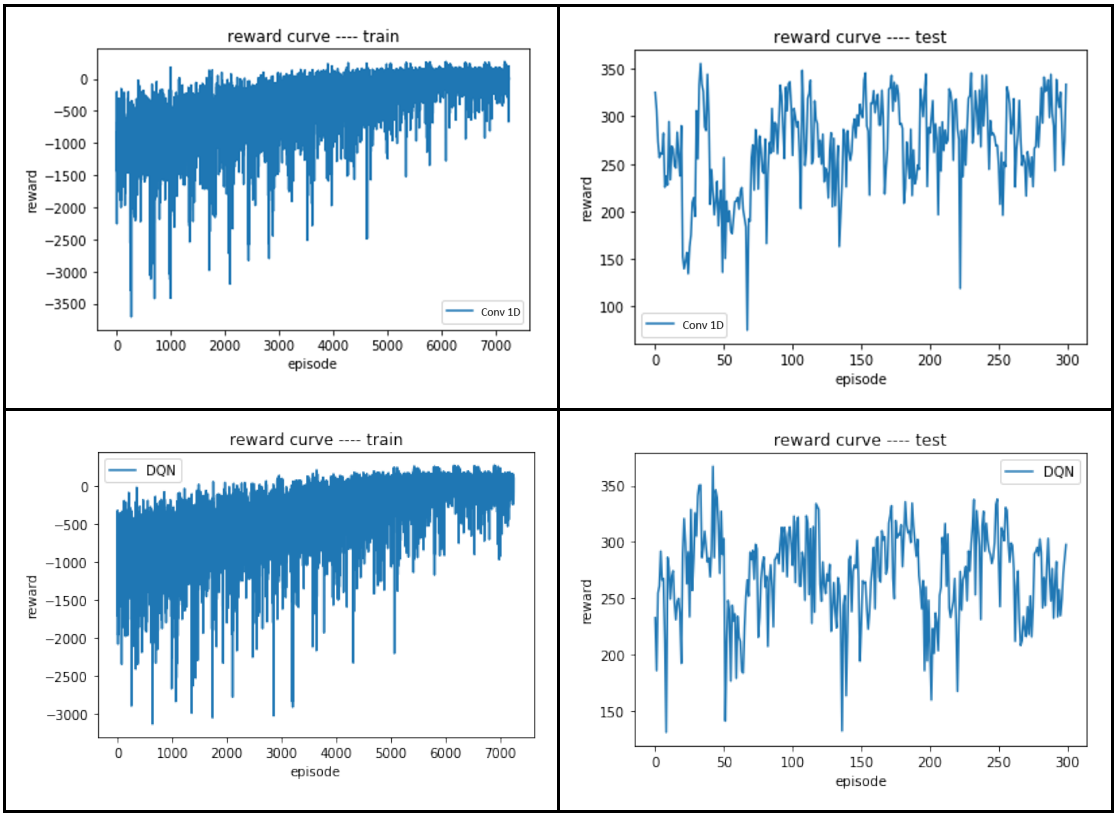

reward curve ---- train
Conv 1D
reward
episode
reward curve ---- test
Conv 1D
reward
episode
reward curve ---- train
DQN
reward
episode
reward curve ---- test
reward
episode
DQN

NRM=10

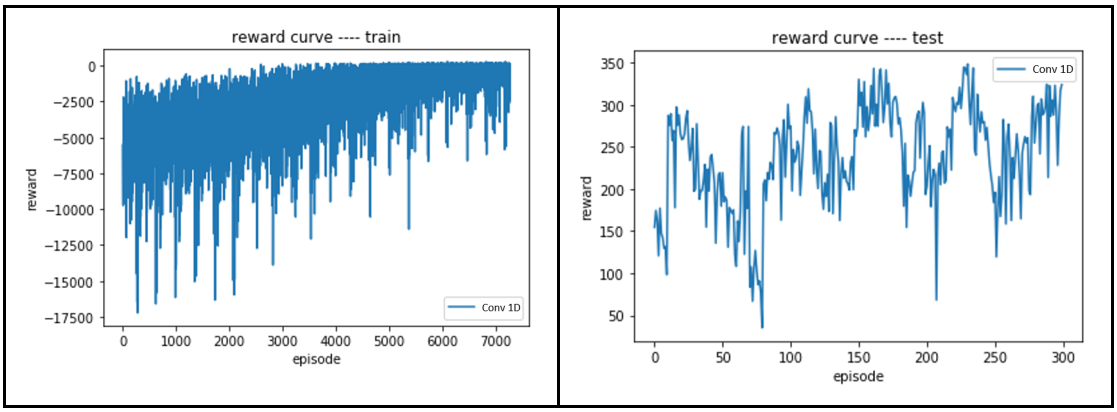

reward curve ---- train
Conv 1D
reward
episode
reward curve ---- test
reward
episode
Conv 1D

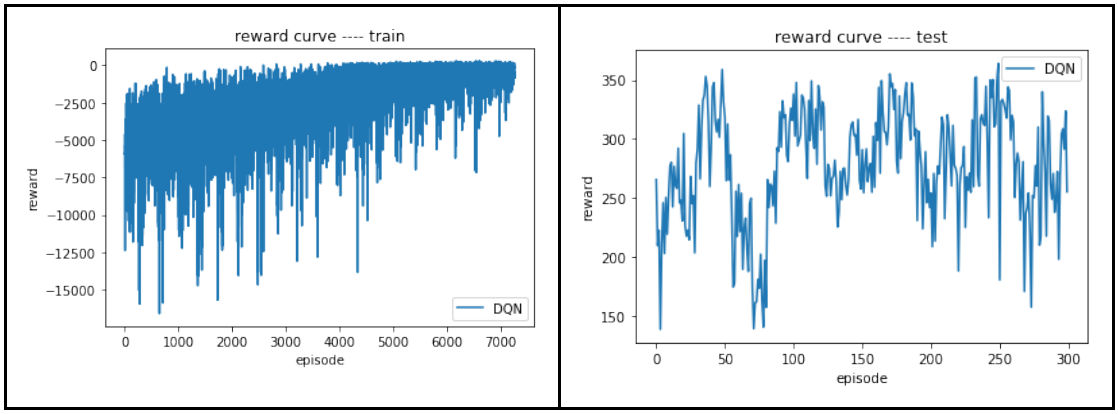


NRM=20

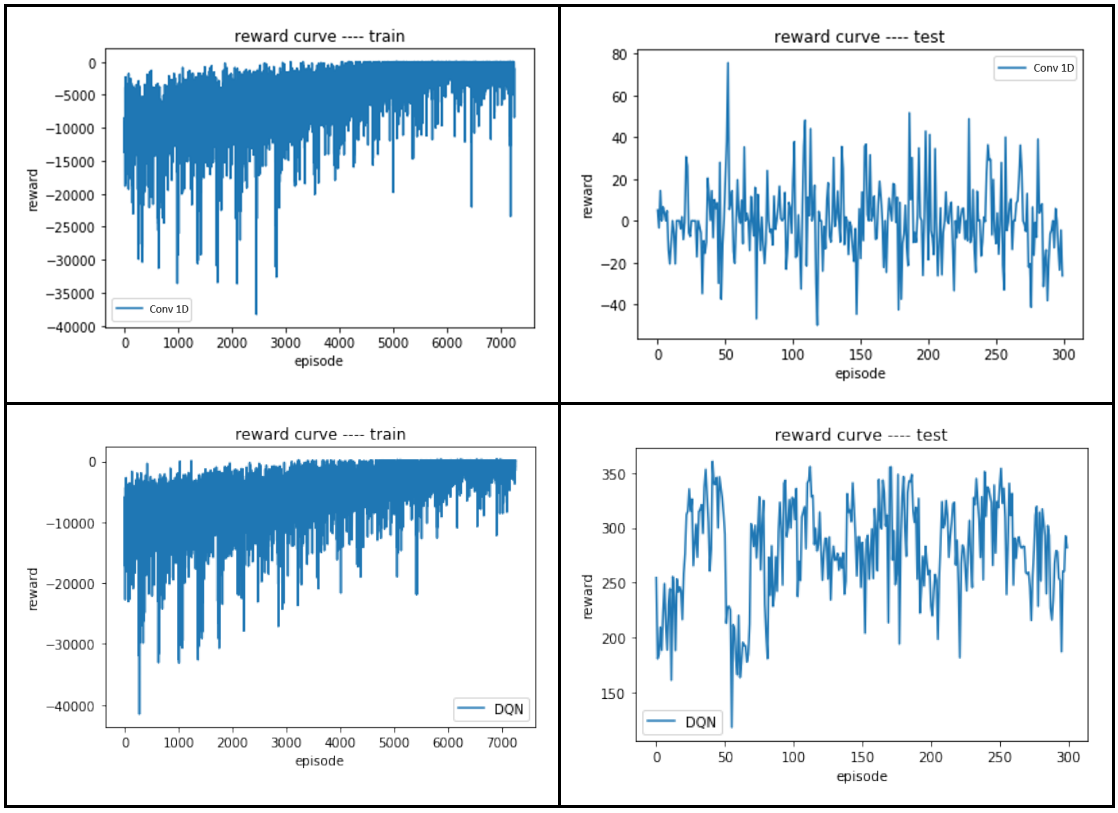


NRM=100





NRM=200



根據實驗結果，隨著 NRM 的提升， Train 部分的 Reward 有顯著的下降，但儘管 Reward 的值會負很多， Agent 仍然勇於嘗試做出正確的決測，直到最後 Agent 才會傾向一些 0 的 Action，但大方向還是仍然會選擇 1 或 -1 的選擇，也因此在 Test 的部分不會全為 0 的 Action。

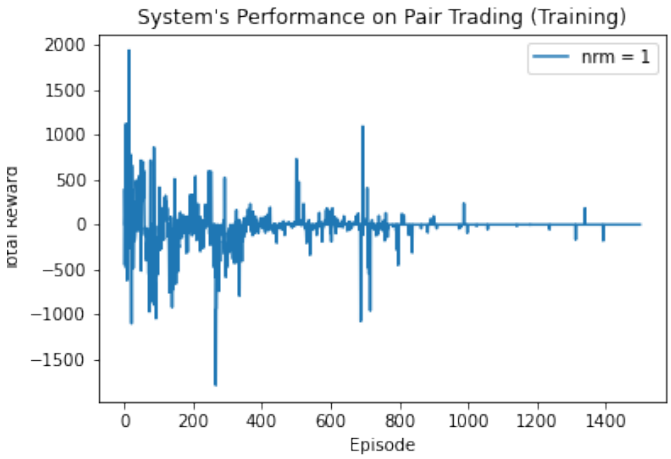
在不同 NRM 的條件下，一般 Dense 的 DQN 表現都較 Convolution 1D 好。

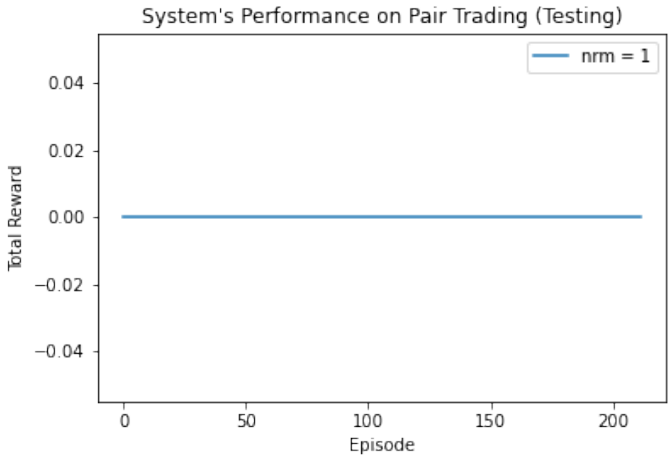
可以推論出，在複雜的股票市場中，或許用一般的 Dense 反而能得到較佳的結果，而不用一些其他較複雜的網路。

經由上述的實驗，我們可以透過 Agent 得知每個時間該做出何種決策才能獲利，但也發現了設計上的缺陷，在餵入兩檔股票資料時，只要不是同一支股票，股票間就會有 Spread 存在，因此並不一定要是 Pair 才會有結果，但是 Reward 的部分就不會有那麼明顯的震盪。

III. 利用 Double Deep Q Network 建立一個系統實現配對交易

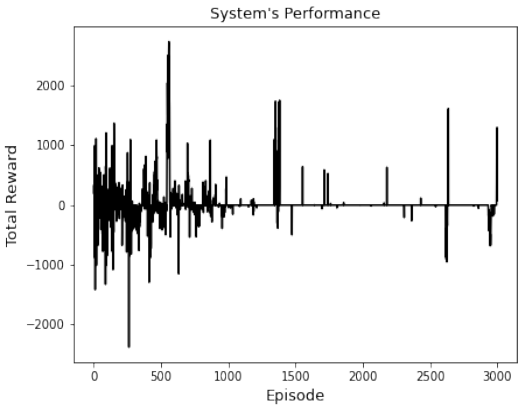
➢ 修改 Pattern Agent Reward Function 前:

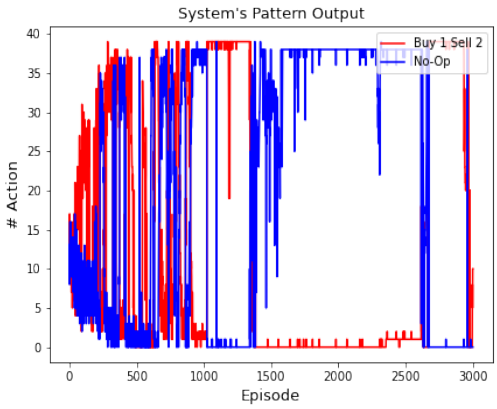




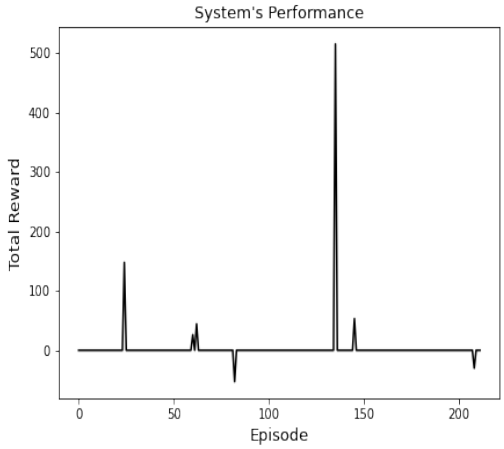
➢ 修改 Pattern Agent Reward Function 後:

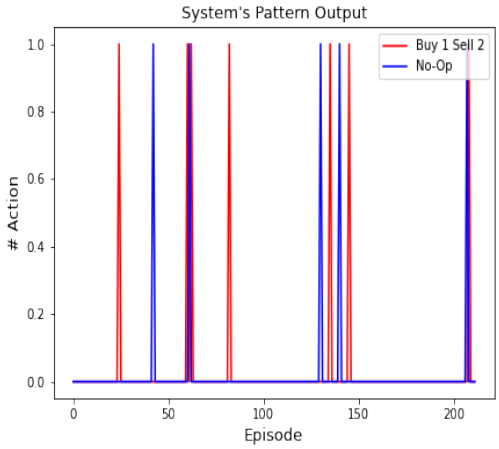
■ Training Period





■ Testing Period





六、 參考文獻

- Brim, A. (2020, January). Deep Reinforcement Learning Pairs Trading with a Double Deep Q-Network. In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 0222-0227). IEEE.

- Brim, A. (2020, January). Deep Reinforcement Learning Pairs Trading with a Double Deep Q-Network. In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 0222-0227). IEEE.

- Brim, A. (2019). Deep Reinforcement Learning Pairs Trading.