

國立陽明交通大學

資訊工程學系

畢業專題報告

運用強化學習演算法於股票市場中的配對交易

研 究 生：吳泓緯、陳鴻杰

指導教授：黃思皓、黃俊龍 博士

中 華 民 國 一 百 一 十 年 九 月

摘要

此研究的目的是在於應用強化學習 (Reinforcement Learning) 演算法於股票市場中的配對交易。由過往的研究中，可以歸納出四種進行配對交易的方法，分別為：距離法 (Distance Approach)、協整法 (Cointegration Approach)、時間序列法 (Time-Series Approach) 與隨機控制法 (Stochastics Control Approach)。然而，透過傳統的方法進行配對交易時，我們往往必須根據一些常用的統計指標，分析價差中存在的特徵，進而設定交易的門檻。然而，仰賴統計指標分析歷史價差時，往往會產生偏差或是遺漏某些重要的特徵。此外，傳統方法交易行為的觸發點完全取決於門檻，容易導致投資組合的淨值有很大的起伏。因此，我們以深度學習為基礎，利用神經網路從歷史價差中，萃取出關鍵的特徵；並導入強化學習演算法，訓練智能體依據特徵自行決定採取何種交易行為。最後，我們在環境中加入停損與停利的限制，使得智能體管理的投資組合淨值能夠更穩定的上升，在承受有限的風險下獲得更多報酬。在實驗二中，我們透過報酬率 (Return)、夏普指標 (Sharpe Ratio) 與最大回撤 (Maximum Drawdown)，分析傳統方法與強化學習方法進行配對交易時所獲得的報酬率，發現強化學習演算法的最大回撤比傳統方法少了 66%；夏普指標多了 27%。在實驗三中，我們也比較在環境中加入限制對於智能體的影響，發現加入停損或停利限制的智能體的報酬率比無限制的智能體多了 9%；最大回撤少了 12%；夏普指標多了 42%。

目 錄

摘要	
目錄	i
1 介紹	1
2 文獻探討	2
2.1 配對交易的傳統方法	2
2.1.1 距離法	3
2.1.2 協整法	3
2.1.3 時間序列法	3
2.1.4 隨機控制法	3
2.2 強化學習應用於股市交易上	3
2.3 強化學習應用於配對交易上	4
3 強化學習應用於配對交易	6
3.1 配對交易的概念	6
3.2 運用傳統方法進行配對交易	7
3.2.1 距離法	7
3.2.2 協整法	8
3.3 運用機器學習進行配對交易	9
3.4 運用強化學習進行配對交易	10
4 Learning System 實作	14
4.1 資料來源	14
4.2 模型定義	14
4.3 模型訓練	15
4.4 模型測試	17
5 實驗結果分析	18
5.1 篩選配對交易標的股票的策略比較	18
5.1.1 距離法	19

5.1.2 協整法	20
5.2 傳統方法和強化學習方法的獲利表現	22
5.3 強化學習演算法加入停損停利後的表現	26
6 Demo System 實作	31
6.1 前端實作	31
6.2 後端實作	32
6.3 使用者介面	34
7 結論與貢獻	37
參考文獻	38

Chapter 1

介紹

以傳統方法進行配對交易時，經常仰賴統計指標分析兩檔股票的歷史價差，往往會產生偏差或是遺漏某些重要的特徵。除此之外，在傳統方法的交易行為中，往往是將價差標準化後，設定固定的門檻，作為交易行為的觸發點。如此一來，容易導致投資組合的淨值有大幅的變動。

在配對交易的傳統方法中，Gatev et al. [5] 提出了距離法 (Distance Approach)，成為配對交易演算法的開山鼻祖。在距離法中，透過歐式距離計算兩檔股票走勢的相近程度。依據相近程度，挑選配對交易的標的組合。在交易訊號的產生階段，會先計算兩檔股票的價差，並且將價差正規化。再依據原始價差的偏差值，設定觸發交易行為的門檻。

在此研究中，我們成功將強化學習演算法應用在股市中的配對交易。透過神經網路的運算取代傳統人為的特徵工程，訓練智能體判斷價差走勢自動交易而非依據固定不變的門檻。從實驗結果中發現，我們的方法比傳統配對交易的方法，能夠獲得穩定的報酬。此外，我們也發現在強化學習演算法的環境中，加入停損與停利的限制，能夠有效提高智能體的報酬並降低波動。

最後，我們也以網頁平台設計 Demo System。透過 Bootstrap 與 Chart.js 設計互動與彈性的前端介面，後端部分採用 Django 框架與 PostgreSQL 資料庫。並將 Demo System 部署於 Heroku，讓任何人都可以透過網頁瀏覽的方式，觀察智能體在訓練與測試期間的交易數據。

Chapter 2

文獻探討

配對交易的概念相當直觀且容易實作。配對交易的實作可分為兩階段。第一階段為透過歷史價格選擇兩檔擁有同樣趨勢的股票。第二階段則是在預設時間區間內觀察兩檔股票的價格差異並且進行交易。在兩檔股票的價格偏離長期均值時，放空價格比較高的股票而買進價格較低的股票。配對交易中假設兩檔股票的價差具有均值回歸的特性，因此在價差收斂時對這兩檔股票進行反向操作並從中獲得利潤。表 2.1 中呈現的是配對交易傳統方法的相關研究以及資料來源。

表 2.1: 各配對交易方法相關的研究

配對交易方法	相關研究	採用資料	每一年的報酬率
Distance	Gatev et al (2006)	US CRSP 1962-2002	0.11
	Do and Faff (2010)	US CRSP 1962-2009	0.08
Cointegration	Vidyamurthy (2004)	-	-
	Rad et al. (2016)	US CRSP 1962-2014	0.10
Time-Series	Elliott et al. (2005)	-	-
	Cummins and Bucca (2012)	Energy futures 2003—2010	≥ 0.18
Stochastic Control	Jurek and Yang (2007)	Selected stocks 1962—2004	0.28—0.43
	Liu and Timmermann (2013)	Selected stocks 2006—2012	0.06—0.23

2.1 配對交易的傳統方法

配對交易的概念雖然直觀，但實作的方式多樣，大致可以分為以下四種方式，距離法 (Distance Approach)、協整法 (Cointegration Approach)、時間序列法 (Time-Series Approach) 與隨機控制法 (Stochastic Control Approach)。

2.1.1 距離法

距離法是配對交易最早提出的方法之一。此方法由 Gatev et al. [5] 所發表，並在配對交易領域達成了一個里程碑。距離法計算兩檔股票之間的距離指標來判斷這兩檔股票是否適合進行配對交易。距離指標象徵了兩檔股票在歷史價格中的相關程度。在交易階段，使用一個臨界點來觸發交易訊號。距離法簡單且擁有很高的透明度，適合大規模的實證應用。此外，透過 1962 年到 2002 年的美國全市場指數 (CRSP US Total Market Index) 進行回測分析，發現距離法也有良好的獲利能力。

2.1.2 協整法

利用了協整性測試依據兩檔股票之間的協整程度，挑選配對交易標的股票。此方法的核心在於挑選股票。大多數使用此方法的研究，在交易階段也使用了與距離法相同的「臨界點」概念來觸發交易訊號。相較於距離法，協整法被認為是比較有代表性的預測值，能夠穩定且精準的評估股票之間的關係。

2.1.3 時間序列法

在時間序列法中，例如 Rad et al. [10] 與 Cummins and Bucca [3] 的研究中，假設股票已經通過長時間觀察而成功配對，作者強調的是如何利用調整交易時期與分析時間序列去最佳化交易階段的總利潤。

2.1.4 隨機控制法

隨機控制法與時間序列法類似，都是假設股票已經完成配對。隨機控制法著重在如何判斷出最佳的投資組合策略。隨機控制理論中以過濾訊號中的雜訊且觀察股票的特性並判定哪種政策函數適用於該投資組合。

2.2 強化學習應用於股市交易上

強化學習是透過智能體與環境的互動過程中，讓智能體學習如何最大化獲得的獎勵期望值 (Expected Reward)。與監督式學習不一樣的地方在於強化學習演算法中，我們

不需要替樣本準備正確的標籤，而是智能體與環境互動的過程中，利用所得到的獎勵 (Reward)，學習採取正確的行為 (Action)。

近幾年，強化學習在金融領域的應用越來越多，主要是因為強化學習中的智能體能夠根據目前的狀態 (State) 自動且即時的採取正確的行為。Bertsimas and Lo [2] 透過強化學習來學習最佳的高頻交易策略，並進行交易。Nevmyvaka et al. [9] 基於 NASDAQ 過去 1.5 年的毫秒限價單，並使用強化學習來預測股價趨勢來最佳化交易的執行。此研究的核心在於利用毫秒為單位的買賣資料去估測未來商品的價錢並從中獲利。實際上，作者也表示大部分的經理人擁有龐大的股票組合，沒辦法在毫秒內針對每隻股票進行分析與操作。在實作上，智能體在每一個時間單位中所觀察到的狀態包含：持有的股票數量、交易數量限制與目前市場的買賣狀態。智能體的目標為在限制時間內把股票賣出且最佳化獲利。在時間區段結束時，最後未售出的股票以當下的價錢售出。

實際上，文獻 Song et al. [11] 透過夏普指標來評估交易策略的表現。其中，夏普指標代表額外承受一單位的風險所能獲得的額外報酬。

此外，Moody and Saffell [8] 使用了循環強化學習 (Recurrent Reinforcement Learning) 演算法學習如何進行資產分配，所使用的資料是 S&P 500 與美國短期國庫券的月份資料。此研究的目標是最佳化夏普指標的同時也最大化獲利。研究中指出，當我們使用夏普指標來評估時，無法分辨出一個投資組合的潛能。

Bertoluzzo and Corazza [1] 利用 Q-learning 與 Kernel-Based 強化學習演算法創造了自動交易的系統。作者以每週的股票價格與夏普指標作為智能體觀察到的環境狀態與獎勵。強化學習中智能體的行為則是「買進」、「賣出」與「持有」。在實作上，作者使用了三檔義大利股票的實際與虛擬價格來比較 Q-Learning 與 Kernel-Based 強化學習之間的表現。研究結果說明了 Q-Learning 基於 Kernel-based 強化學習表現更好，且發現利用夏普指標當作智能體的獎勵時，將限制智能體的表現。

2.3 強化學習應用於配對交易上

雖然近幾年越來越多研究說明了如何利用強化學習來進行投資組合分配與金融商品交易，但是強化學習在配對交易上的研究還是相對少。在近期的研究中，Fallahpour et al. [4]、Kim and Kim [6] 利用強化學習來選擇最佳的配對交易參數，參數中包含交易區間、歷史區間、停損門檻與停利門檻，交易階段則是透過這組參數對兩檔股票進行

交易。

此外，也有研究將配對交易轉換成是一個「多臂吃角子老虎機」(N-Arm Bandit) 的問題。在此問題中，智能體只會遇到一種環境狀態，透過反覆測試找出這組環境狀態下最好的參數組合。此研究是利用了 S&P 500 中的 25 檔股票的每分鐘資料來進行測試。最後的結果顯示，智能體能在沒考慮手續費的情況下獲得 0.12% 的利潤。在 Mnih et al. [7] 則是利用 Deep Q-learning 去最佳化配對交易的參數。經過訓練的強化學習智能體將可以透過挑選交易時間與設定停損來最佳化總獲利。透過此方法，強化學習智能體在交易 S&P 500 中的 25 檔股票時勝過傳統配對交易的方法。

Chapter 3

強化學習應用於配對交易

此研究在於了解強化學習演算法於股票市場配對交易中的表現狀況。我們聚焦於美國的電子產業，依照公司市值對該產業的所有公司進行排序，並挑選出市值前 30 大的企業。接著，我們採用距離法與協整法兩種策略，分別各挑出最佳的 6 組配對交易股票組合。並針對這 12 組股票進行 2015 年至 2019 年的回測，並各從距離法與協整法挑選年化報酬率最佳的 4 組股票進行機器學習與強化學習演算法的比較。

3.1 配對交易的概念

配對交易是一種基於統計套利的市場中性策略。統計套利指的是套利的基礎是建立在歷史數據的統計分析上，主要針對具有價格穩定性的標的。市場中性策略指的是在交易過程中，會同時建立多頭與空頭部位，以對消市場的風險。

具體而言，在配對交易的過程中會先從市場中挑出兩檔長期走勢相近的股票，當這組股票的價差偏離長期的均值時，則做空股價較高的股票，同時也買進股價較低的股票。等到這組股票的價差返回長期均值時，再進行反向的操作，藉此賺取價差變化而得到的報酬。

實際上，配對交易的核心概念也是基於投資的基本原則——買進被低估的賣出被高估的股票。為了判斷股票是否被低估或是高估，需要計算股票的內涵價值。然而，內涵價值的估計又基於對該檔股票的許多假設，如果這些假設不準確的話，對於該股票未來現金流的評估亦會不準確，進而影響內涵價值的估計。

配對交易中透過兩檔股票的價格相對性解決此問題。如果兩檔股票有許多相似的屬性與風險，則假設這兩檔股票的價格走勢會相近。透過兩檔股票的相對價格，可以得到相對高估與相對低估的時機，即不需要實際計算股票的內涵價值。

如果兩檔股票的價差擴大時，則高價者為相對高估的股票，低價者為相對低估的股票。在此錯誤定價的組合中，買進低價者，賣出高價者，並預期未來的價差將會縮小，錯誤的定價最終會自我修正。

3.2 運用傳統方法進行配對交易

此小節中，我們研究以傳統的方法進行配對交易。配對交易的傳統方法包含距離法 (Distance Approach)、協整法 (Cointegration Approach)、時間序列法 (Time-Series Approach) 與隨機控制法 (Stochastics Control Approach)。在此研究中，我們比較距離法與協整法兩種傳統方法在配對交易的獲利表現。

3.2.1 距離法

距離法因其簡易性而成為廣為人知的配對交易傳統策略。距離法可以分為兩個階段：股票組合的挑選與交易訊號的產生。在股票組合挑選的階段中，必須先對兩檔股票的歷史價格進行正規化。在公式 3.1 中透過正規化，將股票價格 P 的值域轉為 $[0, 1]$ 。

$$P_{normalized} = \frac{P - \min(P)}{\max(P) - \min(P)} \quad (3.1)$$

接著，利用公式 3.2 計算兩檔股票之間的歐式距離 SSD ，依據 SSD 的大小判斷兩檔股票價格走勢的相似程度。其中， P_t^1 與 P_t^2 是兩檔股票的價格。

$$SSD = \sum_{t=1}^N (P_t^1 - P_t^2)^2 \quad (3.2)$$

最後，利用公式 3.3 將兩檔股票的價格相減後得到價差 x_i ，並計算價差 x_i 的平均值 μ ，透過 x_i 與 μ 計算價差的偏差值 σ 。

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N - 1}} \quad (3.3)$$

在交易訊號的產生階段，以歷史價格的 $\min(X)$ 與 $\max(X)$ 對新的價格正規化，並計算價差。若價差超過了兩個歷史偏差值，則產生賣出訊號；若價差低於了兩個歷史偏差的負值，則產生買進訊號；若價差穿越偏差值為 0 時，則關閉已經開啓的部位。

3.2.2 協整法

協整法使用了更多統計分析的技術，成為配對交易的典型實作方法。協整亦可以分為兩個階段：股票組合的挑選與交易訊號的產生。

股票組合的挑選依據協整性的定義進行。在圖 3.1 中，兩個時間序列皆為 $I(1)$ 序列，透過線性組合後，形成新的 $I(0)$ 序列，則原來的兩個 $I(1)$ 序列滿足協整特性。若兩檔股票的價格具有協整的特性，則挑選為配對交易的股票組合。為了檢測兩檔股價的時間序列是否滿足協整特性，必須確保兩個時間序列的線性組合為 $I(0)$ 序列。檢測兩個時間序列是否滿足協整特性的常見做法為 Granger Engle Test 與 Johansen Test。

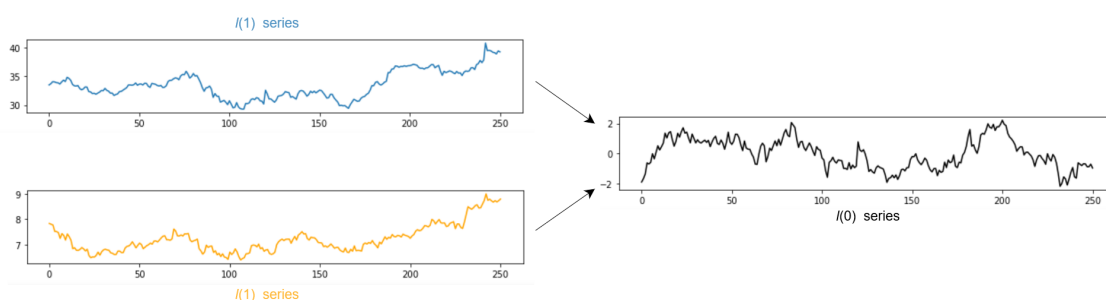


圖 3.1: 協整的特性

在此研究中，我們使用 Johansen Test 檢測協整特性。原因在於使用 Johansen Test，可以將每一個變數都視為獨立的，且使用了兩種統計指標（Eigenvalue Statistics 與 Trace Statistics）判斷變數之間的協整特性。若兩股票價格滿足協整的特性，經過 Johansen Test 後，能夠產生新的 $I(0)$ 序列 $z(t)$ 。在交易訊號的產生階段，根據 $z(t)$ 過去的均值與標準差對新的時間序列 $z(t+1)$ 進行標準化。當標準化後的 $z(t+1)$ 大於 1 時，則產生賣出訊號；若小於 -1 時，則產生買進訊號；若等於 0 時，則關閉已經開啓的部位。

3.3 運用機器學習進行配對交易

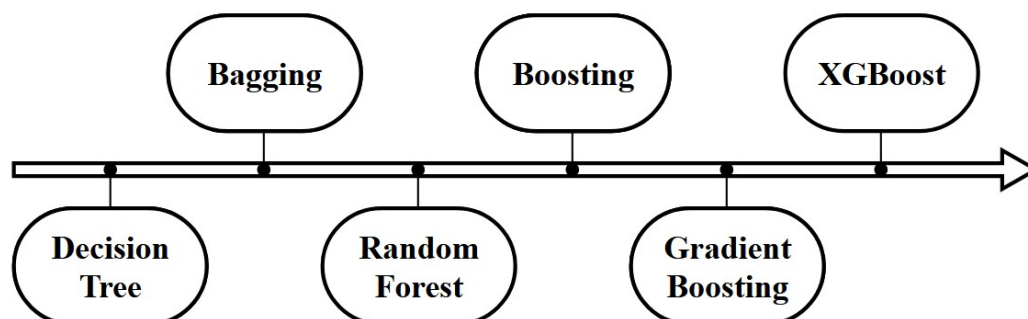


圖 3.2: XGboost 的演化歷程

以距離法 (Distance Approach) 與協整法 (Cointegration Approach) 選擇可能的配對交易股票組合後，並經過 2015 到 2019 的回測，我們從距離法與協整法各挑選出最佳的 2 組股票組合。在本研究中，我們以機器學習模型進行配對交易。我們將此問題定義為「時間序列的分類問題」(Time Series Classification)，透過機器學習模型觀察過去的歷史價格或價差，決定今日要採取的行為。我們採用的模型為集成學習演算法 (Ensemble Learning Algorithm) 中的極限梯度提升 (XGBoost)。圖 3.2 呈現 XGboost 是以決策數 (Decision Tree) 為基礎的集成學習演算法。集成學習演算法透過將許多弱分類模型結合再一起形成一個強分類模型，即使其中一個弱分類模型的預測結果有誤，其他的弱分類模型也能將結果校正回來，進而形成一個準確度更高的模型。集成學習演算法中的演算法可以分為三類: Bagging、Stacking 與 Boosting。Boosting 的特色在於將注意力集中在被錯誤分類的資料上，且模型與模型之間具有關聯性。在 Boosting 演算法中，會將原分類模型分類錯誤資料的權重提高，進而訓練新的分類模型。新的分類模型因而能夠學習到被錯誤分類資料的特性，而提升分類的結果。

在此研究中，我們針對以下四組股票組合：ACN & TSM、NFLX & TMUS、CMCSA & DHR 與 CMCSA & MA 進行 2015 年到 2019 年的配對交易測試。每一次測試前，使用過去 5 年的歷史價格訓練模型。並以網格搜索 (GridSearch) 最佳化模型的超參數 (Hyperparameter)。模型的輸入為過去 120 日的歷史價格，輸出為今日要採取的交易行為。根據模型輸出的交易行為，紀錄模型每日的資產價值，並透過報酬率、夏普指標

與最大回撤分析模型的表現。

透過夏普指標，能夠分析模型在承受單位風險下，能夠獲得多少超額報酬；透過最大回撤，則能夠了解投資組合的淨值由最高點的下跌程度，依此推測模型的風險控管能力。

3.4 運用強化學習進行配對交易

以距離法與協整法選擇可能的配對交易股票組合後，並經過 2015 年到 2019 年的回測，我們挑選出最佳的 4 組股票組合。在本研究中，我們以強化學習演算法進行配對交易。

如同 3.3 所述，我們將此問題定義為時間序列的分類問題，透過強化學習模型觀察過去的歷史價格或價差，決定今日要採取的行為。我們採用強化學習中的 Deep Q-Learning (DQN) 演算法訓練模型進行配對交易。並比較 DQN、Double DQN 與 Dueling Double DQN 三種模型的表現差異。

強化學習不同於監督式與非監督式學習，強化學習中包含兩個重要的元素：智能體與環境。在此研究中，環境為配對交易的模擬環境。在每一個時間單位，智能體觀察目前環境的狀態後，從行為空間 (Action Space) 選擇一個行為，環境根據智能體的行為計算獎勵回饋給智能體，智能體則能透過獎勵得知行為的好壞。智能體的目標為：最大化累積的期望獎勵，透過智能體不停地與環境互動，學習採取最好的行為獲得最多的獎勵。

• Deep Q-Learning (DQN)

在 Deep Q-Learning 中，定義 $Q^*(s, a)$ 為 Optimal Action-Value Function，表示在某一個狀態 s 下，採取行為 a 後，並在之後的狀態中，由模型 π 自行決定行為下，所能得到最大的期望獎勵，即：

$$Q^*(s, a) = \max \pi [R_t | s_t = s, a_t = a, \pi] \quad (3.4)$$

Optimal Action-Value Function 會滿足貝爾曼方程 (Bellman Equation) 的特性，因此

可以將 Optimal Action-Value Function 改寫為：

$$Q^*(s, a) = \mathbb{E}_{s' \sim \varepsilon}[r + \gamma \max_{a'} Q^*(s', a' | s, a)] \quad (3.5)$$

如此一來，就能以迭代的方式計算 Action-Value Function。在此研究中，我們以神經網路 (Q-Network) 作為 $Q(s, a)$ 的 Function Approximator，訓練 Q-Network 來降低損失函數 (Loss function) $L_i(\theta_i)$ ，使得 Q-Network 對於 State-Action Pair 的 Q-Value 預測更加準確。

$$L_i(\theta_i) = \mathbb{E}_{s, a \sim \rho(\cdot)} [(y_i - Q(s, a; \theta_i))^2] \quad (3.6)$$

其中 y_i 為目標值 (Target Q-Value)：

$$y_i = \mathbb{E}_{s' \sim \varepsilon}[r + \gamma \max_{a'} Q(s', a' | \theta_{i-1}) | s, a] \quad (3.7)$$

在訓練過程會將計算 y_i 的 Q-Network 參數固定下來，避免在訓練過程中，因為 Target Q-Value 浮動而難以收斂。此外，透過 Experience Replay Buffer 儲存每一個時間單位的資訊，包含： (s, a, r, s') ，並在訓練階段對 Experience Replay Buffer 隨機取樣，作為 Q-Network 的訓練樣本。如此一來，能避免使用連續樣本對 Q-Network 進行訓練。因為連續樣本之間具有關聯性，可能使 Q-Network 更新偏誤，導致某一個行為被採取的機會變得更高，透過隨機取樣打破連續樣本之間的連續性。演算法 3.1 為 Deep Q-Learning 的完整演算法。

Algorithm 3.1 Deep Q-Learning with Experience Replay

```
Initialize replay memory  $D$  to capacity  $N$ 
Initialize action-value function  $Q$  with random weights
for episode = 1,  $M$  do
  Initialize sequence  $s_1 = \{x_1\}$  and preprocessed sequenced  $\phi_1 = \phi(s_1)$ 
  for  $t = 1, T$  do
    With probability  $\epsilon$  select a random action  $a_t$ 
    otherwise select  $a_t = \max_a Q^*(\phi(s_t), a; \theta)$ 
    Execute action  $a_t$  in emulator and observed reward  $r_t$  and image  $x_{t+1}$ 
    Set  $s_{t+1} = s_t, a_t, x_{t+1}$  and preprocess  $\phi_{t+1} = \phi(s_{t+1})$ 
    Store transition  $(\phi_t, a_t, r_t, \phi_{t+1})$  in  $D$ 
    Sample random minibatch of transition  $(\phi_j, a_j, r_j, \phi_{j+1})$  from  $D$ 
    Set  $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$ 
  end for
end for
```

- **Double DQN**

然而，在 Deep Q-Learning 演算法中預測 Action 的 Q-Value 時，卻有高估的問題。

Van Hasselt et al. [12] 透過多種 Atari 遊戲比較 DQN 與 Double DQN，發現 DQN 在估計 Q-Value 時有高估的狀況，透過 Double DQN 能夠改善此問題。

在 Double DQN 中，存在兩個神經網路 Model Q 與 Target Model Q' 。計算 Target Q-Value 的過程中，DQN 僅由一個 Model 估計所有 Action 的 Q-Value 並選擇 Q-Value 最大的 Action；Double DQN 中，則是透過 Target Model Q' 選擇 Action 再交由 model Q 對 Action 的 Q-Value 進行估計。因此，Double DQN 的 Optimal Action-Value Function 修改為：

$$Q^*(s_t, a_t) \approx r_t + \gamma Q(s_{t+1}, \operatorname{argmax}_{a'} Q'(s_t, a_t)) \quad (3.8)$$

- **Dueling Double DQN**

在此研究中，為了使 Action 的 Q-Value 能夠更有效率的被更新，我們使用 Dueling Double DQN 的架構達到此目標。在圖 3.3 中，Dueling Double DQN 中，Model 的輸出由原來直接輸出 $Q(s, a)$ 變成輸出 $V(s)$ 與 $A(s, a)$ 。

$V(s)$ 象徵 Agent 由狀態 s 開始所能得到獎勵的期望值， $A(s, a)$ 則象徵在相同的狀態下，Action 相對好壞的關係。在 Dueling Double DQN 的最後將 $V(s)$ 與 $A(s, a)$

結合在一起形成 $Q(s, a)$ 。

爲了使模型更傾向於調整 $V(s)$ 而非 $A(s, a)$ ，在結合兩者的過程，針對 $A(s, a)$ 加入更多限制條件，形成以下的結合方式：

$$Q(s, a) = V(s) + (A(s, a) - \frac{1}{|A|} \sum_{a'} A(s, a)) \quad (3.9)$$

在此研究中，我們針對以下四組股票組合：ACN & TSM、NFLX & TMUS、CMCSA & DHR 與 CMCSA & MA 選取 2015 年到 2019 年與上述 3 種模型分別進行配對交易測試。每一次測試前，使用前 5 年的歷史格訓練模型。

模型的輸入爲過去 250 日的歷史價格，輸出爲今日要採取的交易行爲。根據模型輸出的交易行爲，紀錄模型每日的資產價值，並透過報酬率、夏普指標與最大回撤分析模型的表現。

Chapter 4

Learning System 實作

此研究的目的是在於透過強化學習演算法，訓練智能體進行配對交易，並研究智能體在加入停損與停利條件之下的獲利表現。本章節分為四個部分，依序介紹資料來源、模型定義、模型訓練與模型評估。

4.1 資料來源

此研究中使用到的歷史股價資訊皆來自 Yahoo Finance 平台。我們聚焦於美國電子資訊產業市值前 30 大且於 2010 年之前已上市之企業，共於 Yahoo Finance 平台挑選出 26 檔股，如表 4.1 所示。

表 4.1: 實驗使用的所有股票

股票		
AAPL	ACN	ADBE
AMZN	ASML	AVGO
CMCSA	CRM	CSCO
DHR	DIS	GE
GOOGL	INTC	MA
MSFT	NFLX	NVDA
ORCL	T	TMUS
TSLA	TSM	V
VZ	WMT	

每檔股票的區間皆為 2010 年到 2019 年，包含開盤價、最高價、最低價與收盤價，我們以收盤價象徵該股票的價格，進行後續的模型訓練與測試。

4.2 模型定義

本研究中分別使用 Double DQN 與 Dueling Double DQN 演算法，訓練智能體進行配對交易。首先，Double DQN 的模型架構如圖 4.1 所示。模型的前部分由許多卷積區塊

組成。卷積區塊中主要為 1D 卷積層與 1D 最大池化。相較於循環神經網路，我們發現利用 1D 卷積層來萃取時間序列中的重要特徵，不僅能夠縮短模型的複雜度與訓練時間，也能夠提升智能體進行時間序列分類時的準確度。

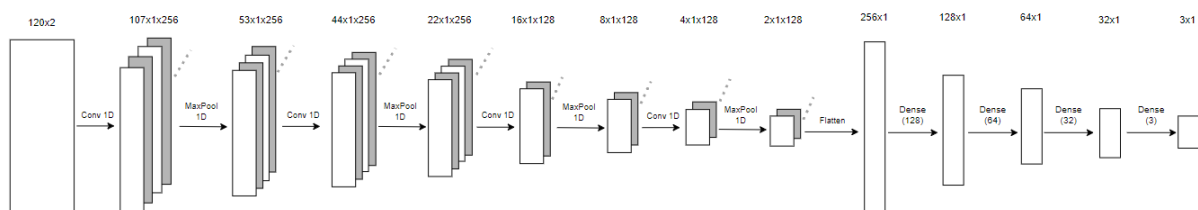


圖 4.1: Double DQN 架構圖

為了使行為的 Q Value 能夠更有效率的被更新，我們將原來模型架構的輸出層進行修改，形成 Dueling Double DQN 的架構，如圖 4.2 所示。

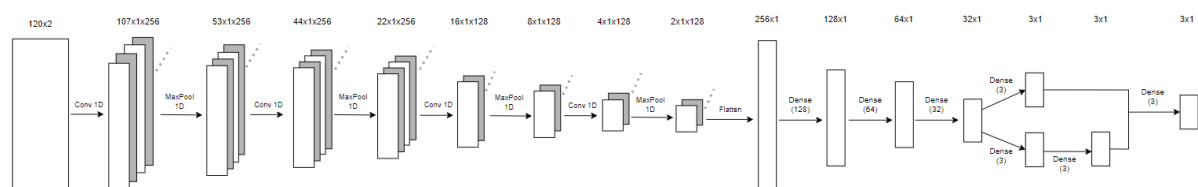


圖 4.2: Dueling Double DQN 架構圖

4.3 模型訓練

在模型訓練階段，我們以前五年的歷史股價作為訓練資料，第六年作為測試資料。並以移動式窗 (Rolling Window) 的方式依序進行訓練與測試。因此，針對單一組配對交易的股票組合，總共進行 6 次的訓練與測試，如表 4.1 所示。

表 4.2: 訓練期間與測試期間之年份

訓練期間	測試期間
2010—2014	2015
2011—2015	2016
2012—2016	2017
2013—2017	2018
2014—2018	2019

在模型訓練階段，我們會將模型訓練 2000 個 Episode。每一個 Episode 的長度皆為 30 天。智能體在決定每一天的行為 (買進、賣出或不動作) 時，皆會接收兩檔股票過去 120 天到 200 天 (視股票組合而定) 的歷史價格。環境將依據智能體採取的行為計算相對應的獎勵。獎勵函式定義如下：

$$Reward = PV(t+1) - PV(t)' \quad (4.1)$$

其中， $PV(t)$ 表示智能體在第 t 天執行行為「之前」的 PV ， $PV(t)'$ 表示智能體在第 t 天執行行為「之後」的 PV 。在獎勵函式中，我們不直接計算智能體執行行為前後的 PV 變化作為該行為的獎勵。因為在配對交易中，買進一檔股票時也會同時賣出另外一檔股票，使得 PV 幾乎不改變，甚至因為交易手續費，而使得 PV 變化多為虧損。因此，我們選擇計算智能體今日執行行為後到隔日執行行為前的 PV 變化作為該行為的獎勵。

在模型訓練時，我們發現直接透過 Double DQN 或是 Dueling Double DQN 演算法對整個模型進行訓練，模型的收斂速度較慢或是難以收斂，使得智能體難以學會做出正確的交易行為。因此，我們加入遷移學習 (Transfer Learning) 的技巧，先以監督式學習訓練模型，使得模型中的卷積區塊能夠學會從時間序列資料中萃取重要的資訊，再透過強化學習演算法，訓練智能體採取正確的交易行為。

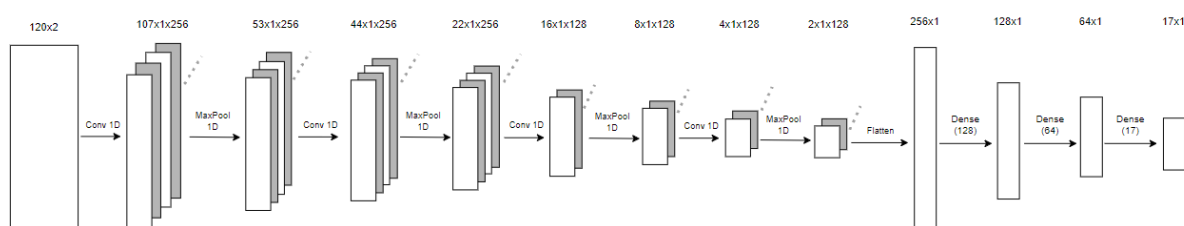


圖 4.3: Transfer Learning 架構圖

在遷移學習的過程中，我們先以監督式訓練模型進行時間序列的分類，模型架構如圖 4.3 所示，前半部由卷積區塊從輸入的時間序列資料中萃取重要的特徵，後半部則是透過全連接層 (Fully-Connected Layer) 進行分類。模型每一次皆會接收兩檔股票過去 120 天到 200 天 (視股票組合而定) 的歷史價格，並針對該時間序列進行分類。類別主要是依照兩檔股股票價差的漲跌而定。如表 4.3 中呈現的，若兩檔股票的今日價差相較於

昨日上漲了 5%，則模型在輸入該組股票過去 120 天到 200 天的歷史股價時，應將其分為 Class 1。

表 4.3: 範例類別

訓練期間	測試期間
Class 1	+5%
Class 2	+4.5%
Class 3	+4%
Class N	-5%

我們會事先分析訓練資料中兩檔股票的價差漲跌情況，使得每一個類別的數量相近，避免因為資料不平衡使得模型的訓練結果產生偏差。

以監督式訓練模型達到 95% 以上的準確率後，我們預期模型中的卷積區塊能夠從時間序列的資料中萃取出「判斷隔日價差漲跌」的重要特徵。進而將卷積區塊的參數固定下來，並修改輸出層的節點個數為 3 (象徵 3 個行為)，再以 Double DQN 或 Dueling Double DQN 演算法訓練智能體進行配對交易。

最後，我們也在環境中加入停損與停利條件，研究在此條件下對智能體獲利情形的表現。停損與停利的門檻為當前 PV 與初始 PV 的 $\pm 10\%$ 。

4.4 模型測試

在模型測試階段，以一年作為時間單位。在這一年中，智能體在決定每一天的行為 (買進、賣出或不動作) 時，皆會接收兩檔股票過去 120 天 200 天 (視股票組合而定) 的歷史價格。我們根據智能體每一天 PV 的變動情形，計算報酬、夏普指標與最大回撤進行後續的評估。

Chapter 5

實驗結果分析

在此研究中，我們進行以下三項實驗。首先，在第一個實驗中，比較兩種不同的配對交易傳統方法所選出的股票組合，並比較執行配對交易的獲利表現。接著，在第二個實驗中，加入機器學習模型與我們所設計的強化學習模型，執行配對交易並比較其獲利。最後，在第三個實驗中，針對強化學習模型加入停損與停利的限制條件，並觀察強化學習模型的獲利改變。

針對實驗一，我們透過報酬比較不同方法的獲利表現；在實驗二與實驗三中，除了使用報酬指標外，我們也納入最大回撤與夏普指標來比較不同模型的獲利表現。

5.1 篩選配對交易標的股票的策略比較

在此實驗中，我們比較距離法 (Distance Approach) 與協整法 (Cointegration Approach) 在配對交易的表現。聚焦於美國電子資訊產業市值前 30 大且於 2010 年之前已上市之企業，表 4.1 呈現最終所挑選出來的 26 檔股票。

5.1.1 距離法

在距離法中，透過歐式距離計算兩檔股票之間的相似程度。在圖 5.1 中，顏色愈深表示兩檔股票的距離愈相近。表 5.1 則是最後挑選出來最相近的 6 組股票。

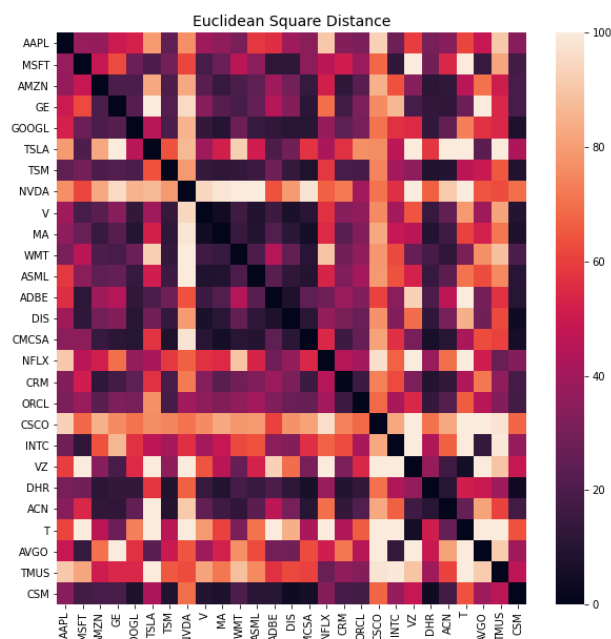


圖 5.1: 股票之歐式平方距離

表 5.1: 距離法所挑選出的股票組合

股票組合	股票一	股票二
1	MA	V
2	T	VZ
3	CMCSA	MA
4	CMCSA	DHR
5	DIS	V
6	ADBE	DIS

針對此 6 組股票，在 2015 年到 2019 年的時間區間，進行配對交易。表 5.2 呈現的是 6 組股票每一年所獲得的報酬。

表 5.2 中的每一個報酬為智能體經過過去 5 年的訓練後，並在該年度進行測試。智能體在年初擁有的投資組合現值 (Portfolio Value, PV) 為 1000，智能體每一日採取行為後將會更新淨值，並在年末計算報酬。其中，也可以發現 (CMCSA, MA) 與 (CMCSA,

表 5.2: 距離法實驗結果

	MA & V	T & VZ	CMCSA & MA	CMCSA & DHR	DIS & V	ADBE & DIS
2015	-0.0125	0.0053	-0.1637	-0.1144	-0.0969	-0.4886
2016	-0.0755	0.0522	0.5566	0.1743	0.4428	0.8294
2017	0.0753	-0.1069	-0.6482	0.4749	-0.4794	-0.7174
2018	0.0985	-0.5183	0.3924	0.335	0.2031	0.0695
2019	0.1161	-0.2383	1.2529	0.5728	0.6553	0.3666
AVG	0.04038	-0.1612	0.278	0.28852	0.14498	0.0119

DHR) 在五年的平均報酬中為最高與次高。此外，透過表 5.2 可以發現完全同產業的公司進行配對交易時不一定能獲得最佳的報酬。舉例來說，Mastercard (MA) 與 Visa (V) 同為付款解決方案公司，在全球發行許多信用卡；AT&T (T) 與 Verizon (VZ) 都屬於美國前四大電信公司。然而作為配對交易標的時，報酬卻不甚理想。Mastercard (MA) 與 Visa (V) 或是 AT&T (T) 與 Verizon (VZ) 因為屬於相同產業，擁有非常像似的價格走勢，因此透過距離法衡量時，非常適合作為配對交易的標的。然而，進行配對交易時，兩檔股票除了相似的價格走勢外，還必須在價差背離均值時，能夠即時回正。若衡量 Mastercard (MA) 與 Visa (V) 或 AT&T (T) 與 Verizon (VZ) 的協整時，發現此組股票並不適合進行配對交易。

5.1.2 協整法

以協整法挑選股票組合時，採用 Johansen Test 分析股票之間的協整特性。在 Trace Statistics 與 Eigen Statistics 時，以滿足 95% 的信心水準拒絕虛無假設為門檻，挑選出以下六組股票。其中，圖 5.2 呈現的是兩檔股票之間的協整特性。顏色愈淺表示兩檔股票的協整特性愈強烈。表 5.3 呈現出最終透過協整法所挑選出來的 6 組股票。

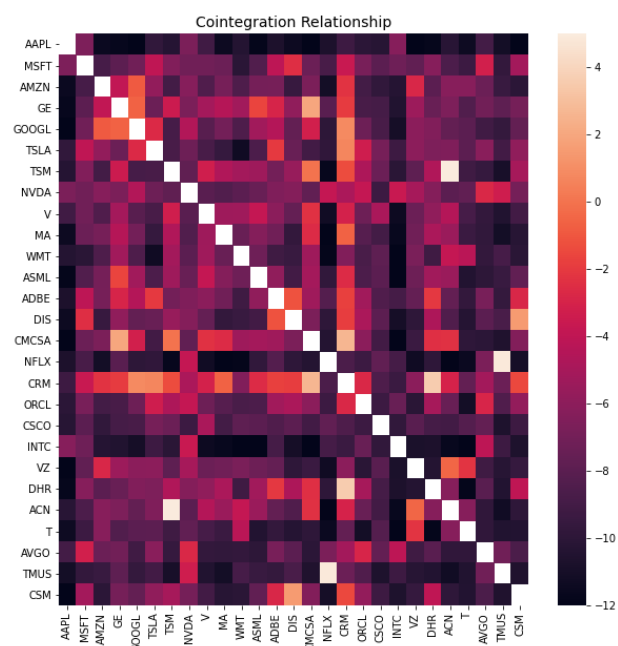


圖 5.2: 股票之協整特性

表 5.3: 協整法所挑選出的股票組合

股票組合	
TSM	ACN
NFLX	TMUS
CRM	DHR
CMCSA	DHR
CMCSA	CRM
DIS	CSM

針對此 6 組股票，在 2015 年到 2019 年的時間區間，進行配對交易。5.4 呈現的是 6 組股票每一年所獲得的報酬。

表 5.4: 距離法實驗結果

	TSM& ACN	NFLX & TMUS	CRM & DHR	CMCSA & CRM	GE & CMCSA	DIS & CSM
2015	0.2533	0.083	0.0828	-0.1925	-0.8316	0.5374
2016	0.4219	0.7748	-0.3727	-0.4917	-0.0804	0.2098
2017	0.248	-0.1851	0.112	0.3542	-4.504	-0.1553
2018	0.7705	0.444	0.7303	0.8024	-1.883	-0.7016
2019	-0.1797	3.396	0.7668	0.951	1.0759	0.6638
AVG	0.3028	0.90254	0.26384	0.28468	-1.24462	0.11082

由表 5.4 中，可以發現 (NFLX, TMUS) 與 (TSM, ACN) 為五年平均報酬的最高與次

高。由表中也可以發現，General Electric (GE) 與 Comcast (CMCSA) 每一年的報酬都有劇烈變化，5 年的報酬平均也是 6 組最差。若分析 General Electric (GE) 與 Comcast (CMCSA) 的股價關係，可以發現兩股價差在這 5 年間有劇烈的變化，價差的最高值與最低值相差了 6 倍。因此，即使兩檔股票的價格走勢滿足協整的門檻，進行配對交易時也可能出現暴漲暴跌的現象，導致最終以虧損收場。

我們各從距離法與協整法中挑選兩組報酬最佳的配對交易組合 (如表 5.5 所示)，作為後續實驗的資料來源。

表 5.5: 協整法所挑選出的股票組合

股票組合	
CMCSA	DHR
CMCSA	MA
NFLX	TMUS
TSM	ACN

5.2 傳統方法和強化學習方法的獲利表現

在此實驗中，我們比較傳統方法、機器學習方法與強化學習方法針對實驗一所挑選出來的四組股票 (如表 5.5 所示) 進行配對交易，並透過報酬、最大回撤與夏普指標，比較 4 種交易策略的績效。表 5.6 呈現的是 4 種策略針對 (TSM, ACN) 與 (NFLX, TMUS) 在每一個年度的績效比較。

表 5.6: 傳統方法、機器學習與強化學習於配對交易的表現比較

		TSM & ACN			NFLX & TMUS		
		return	mdd	sr	return	mdd	sr
2015	Rule-Based	0.2533	0.4754	0.6826	0.083	1.3814	0.4474
	XGBoost	0.1917	0.3828	0.591	1.0406	0.5174	1.1852
	DDQN	0.072	0.0038	0.8292	0.4341	0.1747	1.0428
	DDDQN	0.0436	0.0885	0.3739	0.1568	1.0354	0.1336
2016	Rule-Based	0.4219	0.6478	0.9154	0.7748	0.1701	1.4316
	XGBoost	0.3455	0.5018	0.7743	0.765	0.1701	1.4308
	DDQN	0.3038	0.0256	1.1703	0.1167	0.1524	0.7258
	DDDQN	0.3589	0.028	1.5418	0.069	0.0075	0.7009
2017	Rule-Based	0.248	0.0366	1.0675	-0.1851	2.7775	-1.164
	XGBoost	0.2884	0.0366	1.2034	-0.1851	2.7775	-1.164
	DDQN	0.1809	0.1553	0.7766	0.4587	0.0249	1.6525
	DDDQN	0.048	0.042	0.5894	1.5105	0.3177	1.4393
2018	Rule-Based	0.7705	0.05	2.4887	0.444	0.2483	0.8919
	XGBoost	0.713	0.05	2.0863	0.5037	0.2483	0.9092
	DDQN	0.2105	0.2758	0.7682	0.2362	0.24	0.7916
	DDDQN	0.6564	0.05	2.0714	-0.0027	0.5599	0.3464
2019	Rule-Based	-0.1797	0.863	0.7278	3.396	3.0711	0.9687
	XGBoost	-0.1214	0.8171	0.6388	3.4288	2.953	-0.4693
	DDQN	0.3179	0.2278	0.9832	0.3217	0.007	1.0807
	DDDQN	0.6884	0.0866	1.5196	1.372	0.9303	1.3933

由表 5.6 可以發現強化學習演算法 (DDQN 與 DDDQN) 在多數的年度中，在最大回撤指標上都有不錯的表現。換句話說，強化學習演算法相較於 Rule-Based 與 XGBoost 在進行配對交易時，承受相對較小的波動。可能的原因為透過強化學習演算法訓練智能體時，如果智能體進行愈多行為，得到負的獎勵的機率也愈高，促使智能體只有在相當有信心的情況下才會採取行為 (買進或賣出)，在多數情況下不採取行為 (繼續持有)，使得強化學習演算法 (DDQN、DDDQN) 擁有相對小的波動。

此外，從表 5.6 中還可以發現：當 Rule-Based 與 XGBoost 擁有負的報酬時，強化學習演算法卻能獲得相當好的報酬。舉例來說，在 2017 年的 (NFLX, TMUS) 或是 2019 年的 (TSM, ACN)。圖 5.3 為 2017 年四種模型在 (NFLX, TMUS) 的表現，可以發現該

年度的價差的波動情況較為劇烈，導致 Rule-Based 與 XGBoost 的資產價值劇烈波動，DDQN 與 DDDQN 則相對穩定上升。

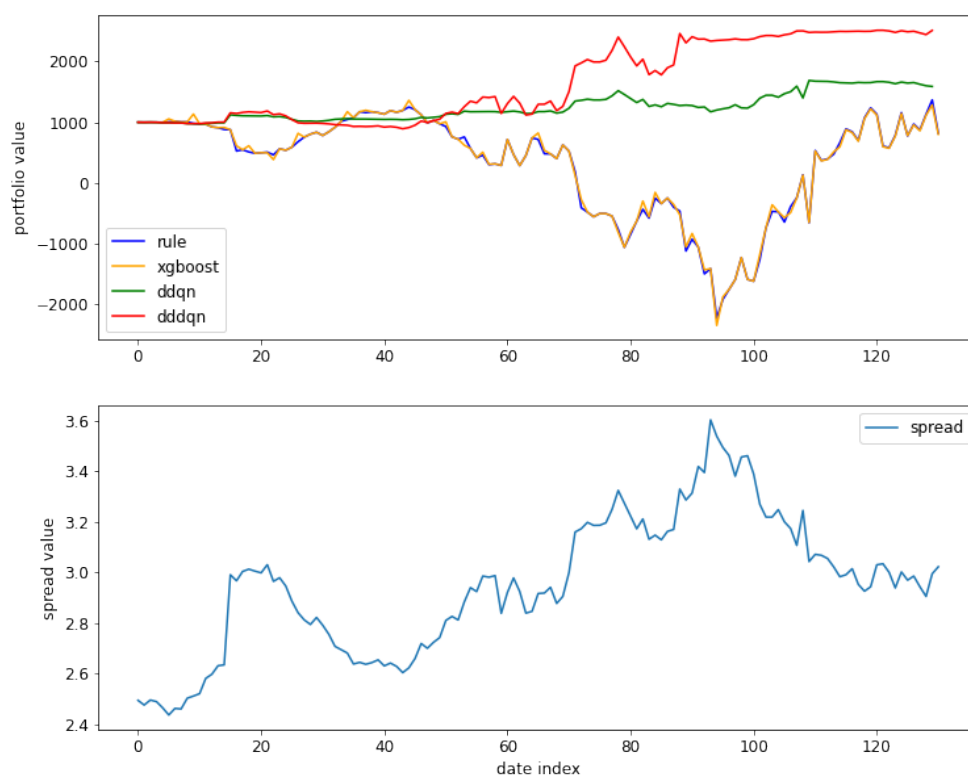


圖 5.3: 2017 年 NFLX 與 TMUS 的獲利表現

因此，可以推測透過強化學習演算法所訓練的智能體進行配對交易時，在報酬的波動上除了比較低之外，當該年度價差波動較為劇烈時，智能體也能維持較高的報酬。

表 5.7 呈現的是 4 種策略針對 (CMCSA, DHR) 與 (CMCSA, MA) 在每一個年度的績效比較，可以觀察到兩種現象：強化學習演算法在多數情況下，擁有較低的最大回撤；當 Rule-Based 與 XGBoost 演算法得到負的報酬時，強化學習演算法能夠獲得相對穩定的報酬。

表 5.7: 傳統方法、機器學習與強化學習於配對交易的表現比較

		CMCSA & DHR			CMCSA & MA		
		return	mdd	sr	return	mdd	sr
2015	Rule-Based	-0.1144	0.2911	-0.4039	-0.1637	0.2577	-0.7002
	XGBoost	-0.1006	0.2759	-0.3668	-0.1727	0.2494	-0.7981
	DDQN	0.1050	0.013	0.9879	0.003	0.0455	00.0918
	DDDQN	0.1151	0.0146	1.5993	-0.0612	0.128	-0.4994
2016	Rule-Based	0.1743	0.0247	1.5406	0.5566	0.0466	2.427
	XGBoost	0.1884	0.03	1.6486	0.5566	0.0466	2.427
	DDQN	0.0912	0.2296	0.4596	0.2341	0.1647	0.9201
	DDDQN	-0.0177	0.0384	-0.6584	0.1787	0.0058	1.3698
2017	Rule-Based	0.4749	0.3694	0.9433	-0.6482	2.4412	1.1543
	XGBoost	0.4815	0.3474	0.9712	-0.6338	2.4053	0.5497
	DDQN	0.3765	0.0105	1.6536	0.4587	0.1839	0.8409
	DDDQN	0.167	0.0593	0.9023	0.1327	0.0891	0.9829
2018	Rule-Based	0.335	0.3002	0.8771	0.3924	0.4244	0.7987
	XGBoost	0.335	0.3002	0.8771	0.4094	0.4189	0.8135
	DDQN	0.1918	0.025	1.2609	0.1409	0.1959	0.5509
	DDDQN	0.0496	0.0618	0.6794	0.5454	0.0807	1.2069
2019	Rule-Based	0.5728	0.0108	1.3091	1.2529	0.3808	1.373
	XGBoost	0.5476	0.0108	1.3289	1.2208	0.3949	1.3486
	DDQN	0.2576	0.0542	1.1206	0.3612	0.1845	0.9106
	DDDQN	0.1848	0.0394	1.2101	0.6618	0.192	1.3408

以 2017 年 (CMCSA, MA) 為例，由圖 5.4 可以觀察到兩檔股票的價差變化劇烈，連帶導致 Rule-Based 與 XGBoost 的資產價值波動較大，DDQN 與 DDDQN 則呈現穩定上升。由此可以推測強化學習方法在股票價差劇烈波動下，也能夠學習到適合的交易行為，穩定投資組合的價值波動。

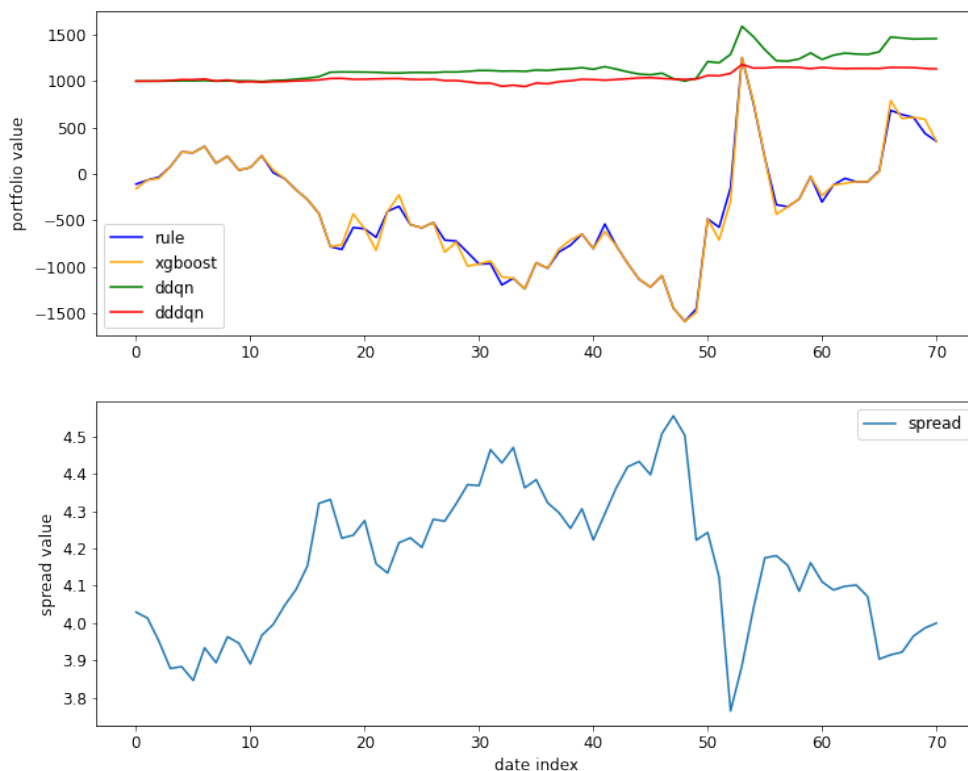


圖 5.4: 2017 年 CMCSA 與 MA 的獲利表現

5.3 強化學習演算法加入停損停利後的表現

在此實驗中，我們比較強化學習演算法 (Dueling Double DQN) 在加入停損與停利機制後的表現變化。我們比較 4 種強化學習模型：原始模型 (None)、加入停損機制的模型 (SL)、加入停利機制的模型 (TP) 與加入停損與停利機制的模型 (TP_SL)。其中，停損與停利的門檻皆固定為 10%。

表 5.8 呈現的是上述 4 種模型針對 (TSM, ACN) 與 (NFLX, TMUS) 在每一個年度的績效比較。

表 5.8: 加入停損於停利條件對強化學習智能體的影響

		TSM & ACN			NFLX & TMUS		
		return	mdd	sr	return	mdd	sr
2015	None	0.0436	0.0885	0.3739	0.1568	1.0354	0.1336
	TP	0.4027	0.3507	0.9685	0.3549	0.0101	1.3895
	SL	0.2151	0.0092	1.3879	0.4399	0.1135	1.9067
	TP&SL	0.1775	0.0036	1.3889	0.4528	0.0161	1.6523
2016	None	0.3589	0.028	1.5418	-0.1299	0.0075	0.7009
	TP	0.1904	0.0302	1.2179	0.5606	0.0532	0.9403
	SL	0.1719	0.0085	1.3486	0.0234	0.1931	0.2267
	TP&SL	0.2025	0.0303	1.0362	0.0461	0.1174	0.8867
2017	None	0.048	0.042	0.5894	0.7098	0.3177	1.4393
	TP	0.476	0.0929	1.5184	0.8935	0.0203	1.7525
	SL	0.1601	0.0383	1.3496	0.6488	0.0204	1.5112
	TP&SL	0.1614	0.0874	0.9288	0.8398	0.017	2.0326
2018	None	0.6564	0.05	2.0714	-0.0027	0.5599	0.3464
	TP	0.0714	0.0283	0.5252	-0.0133	0.87	0.8302
	SL	0.6422	0.0454	1.6753	0.6582	0.2677	1.008
	TP&SL	-0.0162	0.0501	0.0287	0.6696	0.3573	1.1188
2019	None	0.6884	0.0866	1.5196	1.372	0.9303	1.3933
	TP	1.3094	0.2648	1.8337	-0.6804	0.7328	-0.9601
	SL	0.7432	0.1643	1.77	-0.036	1.0671	0.3673
	TP&SL	0.2582	0.2362	0.7506	0.0869	0.4888	0.4253

由表 5.8 可以發現：加入停損或停利機制的模型 (TP、SL、TP_SL) 相較於原始模型 (None) 在最大回撤與夏普指標的表現更好。例如，在表 5.8 中，5 個年度與 2 組配對交易的標的組合所形成的 10 次實驗中，停損停利模型在 7 次實驗中的報酬、最大回撤與夏普指標表現都優於原始模型。特別的是，停損停利模型在 2016 年的 (NFLX, TMUS) 實驗中，最大回撤的表現劣於原始模型，但是報酬與夏普指標卻優於原始模型。若分析 (NFLX, TMUS) 歷年的價差波動程度，可以發現 2016 是 5 年中價差波動程度最小的一年。

圖 5.5 為該年度四種模型的資產價值變化，由圖中可以發現原始模型 (None) 在面對價差波動程度小時，報酬的波動程度相對也較小，然而卻也導致最終的報酬較低。相

對而言，該年度的停損停利模型 (TP) 在承受 5% 的最大回撤之下，能夠創造 56% 的報酬。由夏普指標指標也可以說明該年度的停損停利模型 (TP) 在承受 1% 的波動風險之下，能夠創造 94% 的報酬。因此，在強化學習演算法的環境中加入停損與停利機制，能夠促使智能體學會如何控管投資組合的波動風險，並在承受有限風險之下，創造更多報酬。

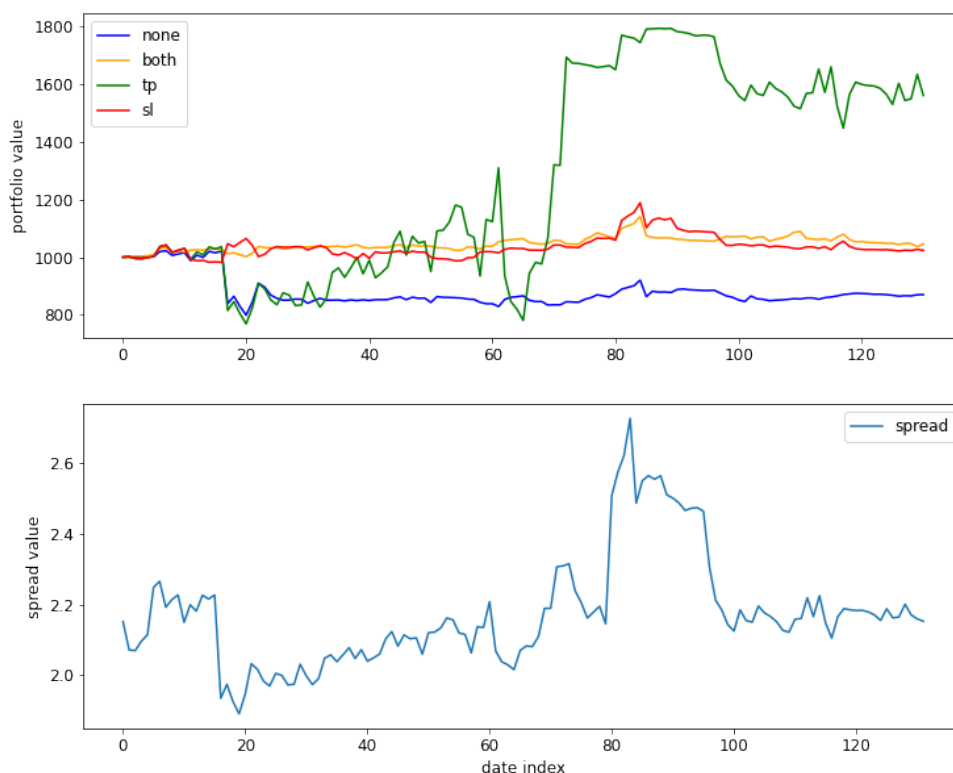


圖 5.5: 2016 年 NFLX 與 TMUS 的獲利表現

表 5.9 呈現的是上述 4 種模型針對 (CMCSA, DHR) 與 (CMCSA, MA) 在每一個年度的績效比較。

表 5.9: 加入停損於停利條件對強化學習智能體的影響

		CMCSA & DHR			CMCSA & MA		
		return	mdd	sr	return	mdd	sr
2015	None	0.1151	0.0146	1.5993	-0.0612	0.128	-0.4994
	TP	0.0098	0.0191	0.3183	-0.0132	0.034	-0.1723
	SL	0.1781	0.0146	1.6144	-0.0856	0.0901	-2.4026
	TP&SL	0.0199	0.0003	1.1494	0.1499	0.0614	0.9925
2016	None	-0.0177	0.0384	-0.6584	0.1787	0.0058	1.3698
	TP	-0.032	0.0333	-0.9788	0.237	0.093	1.0924
	SL	0.1235	0.0335	1.5028	0.0867	0.0227	1.267
	TP&SL	0.1254	0.027	1.0868	0.1569	0.0327	1.8287
2017	None	0.167	0.0593	0.9023	0.1427	0.0891	0.9829
	TP	0.1846	0.0849	1.3614	0.1391	0.0279	0.9228
	SL	0.0933	0.0716	0.7157	0.2317	0.1889	0.9062
	TP&SL	-0.005	0.0733	-0.0314	0.3716	0.1112	1.2001
2018	None	0.0496	0.0618	0.6794	0.2803	0.0807	1.2069
	TP	0.0884	0.016	1.2447	0.0288	0.0469	0.7685
	SL	0.0383	0.1225	0.3366	-0.0527	0.2356	-0.7264
	TP&SL	0.0767	0.016	1.0904	0.0998	0.0237	0.4898
2019	None	0.1848	0.0394	1.2101	0.6618	0.192	1.3408
	TP	0.0924	0.0089	0.819	1.067	0.2841	1.3278
	SL	0.263	0.0826	1.0679	0.1832	0.1168	0.6853
	TP&SL	-0.2382	0.2646	-1.1372	1.0432	0.3912	1.2042

在表 5.9 中同樣也可以看到加入停損停利模型 (TP、SL、TP_SL) 的優勢，甚至有突出的表現。在表 5.9 中，5 個年度與 2 組配對交易的標的組合所形成的 10 次實驗中，停損停利模型在 9 次的實驗中，報酬的表現都優於原始模型；在 8 次的實驗中，最大回撤的表現都優於原始模型；在 7 次的實驗中，夏普指標的表現都優於原始模型。特別的是，在 2018 年 (CMCSA, MA) 的表現上，加入停損停利後的模型 (TP、SL、TP_SL) 在 3 種指標上的績效明顯落後於原始模型 (None)。圖 5.6 呈現的是四種模型的績效表現，由圖中能發現加入停損停利後的模型 (TP、SL、TP_SL) 資產價值的波動較小，卻也導致最終的報酬低於原始模型 (None)。若分析兩檔股票價格的關係，可以發現在 2018 年兩檔股票價格的協整關係最為強烈。可能原因為擁有愈強烈協整關係的兩檔股票愈適

合透過沒有額外限制的演算法進行交易，因此加入停損停利的限制下，反而讓模型趨於保守，最終獲得較低的報酬與最大回撤。

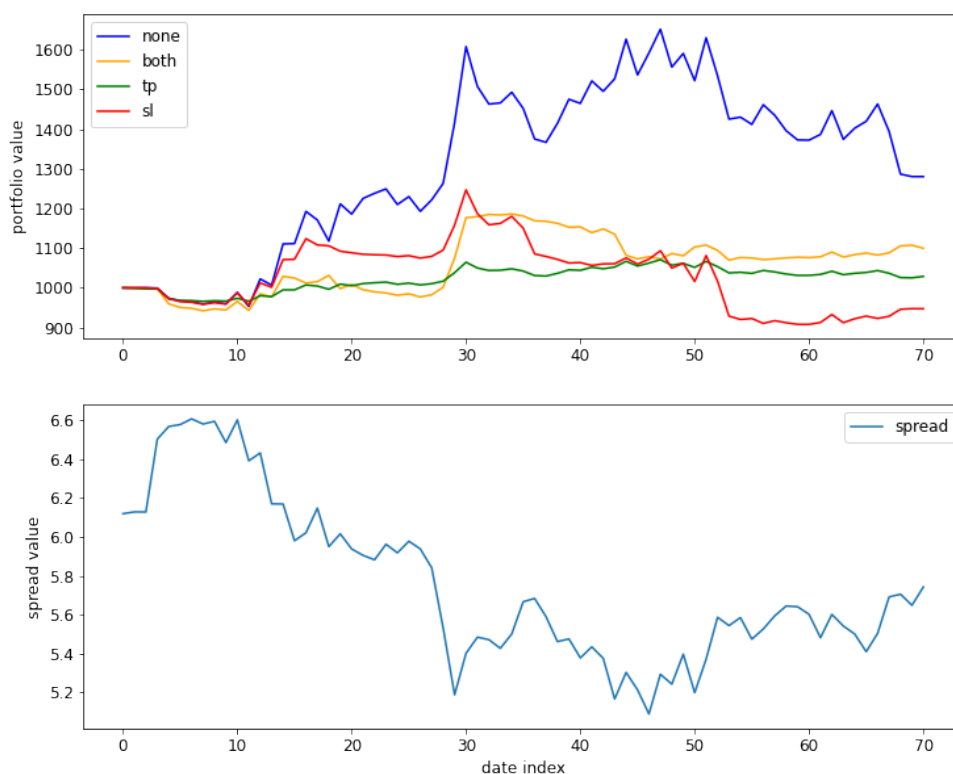


圖 5.6: 2018 年 CMCSA 與 MA 的獲利表現

由此實驗中可以發現，不管是透過距離法或是協整法所挑選出來的股票，在加入停損停利模型後的表現，明顯優於原始模型。在透過強化學習演算法訓練智能體期間，加入停損與停利限制，能夠使得智能體在測試期間能夠學會控制風險並最大化報酬。

Chapter 6

Demo System 實作

本研究開發了一套 Demo System 讓使用者能透過網頁平台，選擇配對交易標的組合、訓練區間、測試區間與交易策略後，透過 Dueling Double DQN 演算法訓練該智能體，並將數據以折線圖與表格形式顯示於頁面上。由於智能體所需的訓練時間太長，沒辦法即時的顯示結果到網頁上。因此，智能體的訓練與測試結果都是預先收集好並上傳到資料庫中。如圖 6.1 所示，Demo System 主要由三個元素所組成：前端 (Frontend)、資料庫 (Database) 與後端 (Backend)。本章節將依序介紹各個元素的實作細節。

6.1 前端實作

前端開發所使用的語言為 HTML、CSS 與 JavaScript。HTML (Hypertext Markup Language) 是一種描述 Hypertext 的 Markup Language。基本上，目前盛行的瀏覽器都可以讀取 HTML，還染出頁面的架構。透過 HTML 中的元素，可以在網頁中加入各種物件。例如，表格、圖片、表單、連結、文字、段落，等等。利用 CSS (Cascading Style Sheets) 則可以在由 HTML 定義的元素上添加樣式。例如：決定文字的色彩、大小與字型或是多個元素的排版形式。目前發行的 CSS 版本有 CSS1 到 CSS4.15；在本研究中，我們透過 CSS3 開發 Demo System。原因在於 CSS3 已經相容與大部分瀏覽器，能夠讓更多不同平台的使用者瀏覽 Demo System 頁面。最後，我們也透過 JavaScript 程式語言實作網頁上的動態與互動效果。JavaScript 支援物件導向 (Object-Oriented Programming)、指令式編程 (Imperative Programming) 與函數式編程 (Functional Programming)。JavaScript 普遍使用於網頁設計，許多網頁都是透過 JavaScript 呈現出動態與互動的效果。與 CSS3 都已被大多數的瀏覽器支援。

為了美化前端介面，我們使用 Bootstrap 框架。Bootstrap 是一個由 HTML、CSS 與 JavaScript 寫成的前端框架。透過 Bootstrap 可以不需要撰寫大量的 CSS 卻能達到專業前端設計的效果。Bootstrap 的核心目標是讓網頁開發人員能輕易的創造一個響應式網站

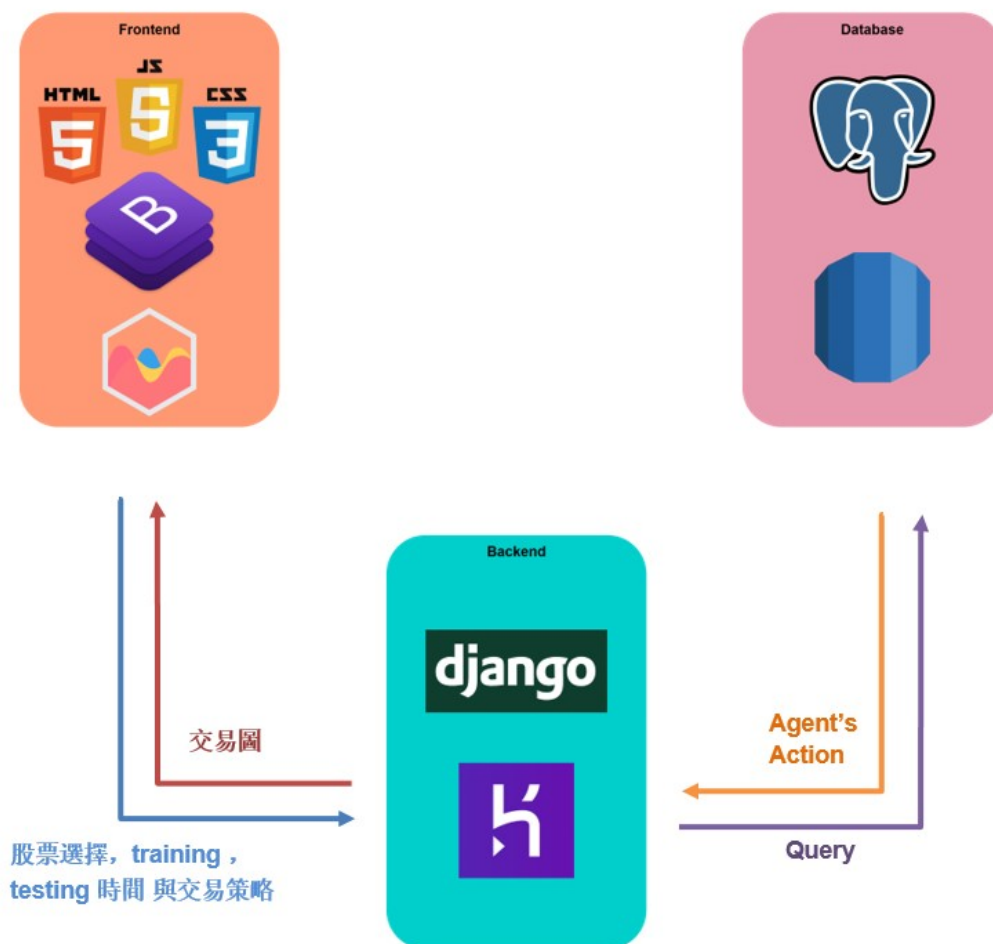


圖 6.1: 網頁架構圖

(Responsive Website)，也就是網頁的排版可以自動適應螢幕大小而變化。實作上，只需要在 HTML 元素上配置恰當的 Bootstrap Class。此外，我們在 JavaScript 中使用 Chart.js 套件製作互動式的圖表。Chart.js 是一個具有高度彈性的畫圖工具，能讓多種不同的圖表混搭且支援動畫效果，輕易的製作出精美的圖表。Chart.js 也支援常見的多數統計圖表。例如：長條圖 (Bar Chart)、折線圖 (Line Chart)，圓餅圖 (Pie Chart)。

6.2 後端實作

在本研究中，我們透過 Django 作為 Demo System 的後端。Django 是免費且開源的 Python 網路框架 (Web Framework)。Django 利用 MTV (Model-Template-View) 架構，簡化網路應用程式 (Web Application) 的開發複雜度。我們使用 Django 的原因在於我們

可以透過 Python 程式語言建立 Django Web Application，同時使用 Python 程式語言開發 RL 演算法與配對交易。此外，Django 也提供了強大且豐富的函數庫，使得 URL Mapping、連結資料庫與網路安全性都更容易實作，讓我們能夠更專注在介面與功能的開發。

在圖 6.2 中，呈現出 Django Web Application 的完整架構。最上層為瀏覽器 (Web Browser)，屬於前端的部分。依序向下為模板 (Template)、視圖 (View)、模型 (Model) 與資料庫 (Database)。Web Browser 向伺服器發送請求時，Django 將依照 URLConf 將不同的 URL 對應到不同的 View，並執行該 View Function。View Function 可以透過 Template 回傳新的頁面或是透過 Model 從 Database 中取得資料。

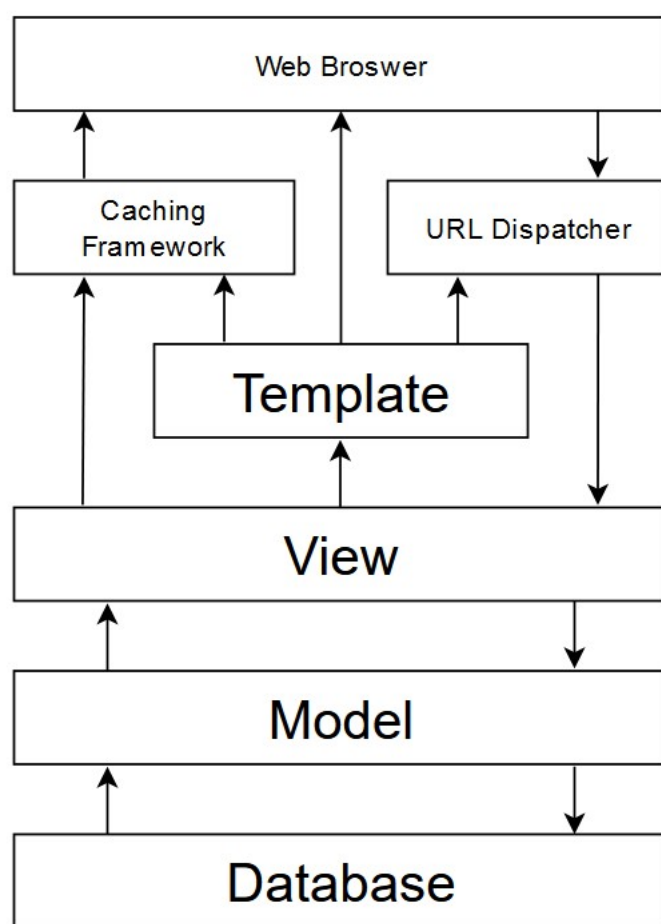


圖 6.2: Django 架構圖

為了使任何人可以存取到我們的 Demo System，我們將 Django Web Application 部署於 Heroku。Heroku 是一個平台即服務 (Platform as a Service) 的提供商，透過友善的介面與簡易的指令包裝 AWS 所提供的雲端運算資源。由於我們的 Demo System 無需很大

的網路流量，Heroku 所提供的免費資源正好符合了我們的需求。

6.3 使用者介面

在此小節中，將會依序介紹 Demo System 的頁面以及區塊。首先是 Demo System 的首頁。如圖 6.3 所示，在首頁中使用者可以選擇配對交易的標的組合、訓練區間、測試區間與交易頻率。按下「Trade」按鈕後，將會顯示智能體在訓練期間與測試期間的交易數據。

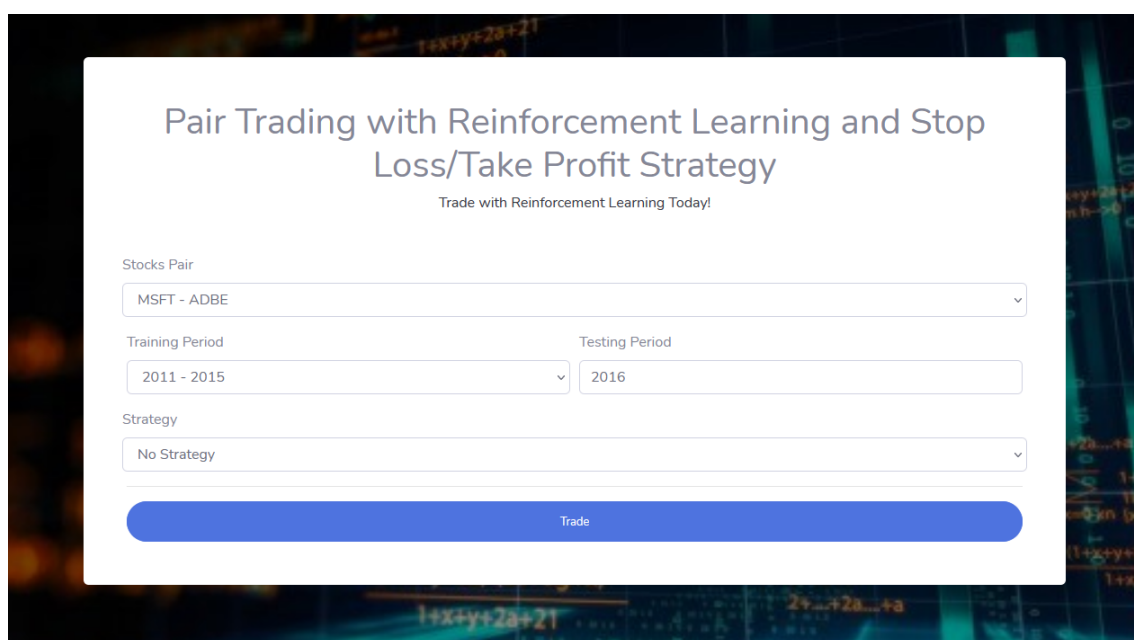
The image shows a web interface titled "Pair Trading with Reinforcement Learning and Stop Loss/Take Profit Strategy". Below the title is a subtitle "Trade with Reinforcement Learning Today!". The interface contains several input fields: "Stocks Pair" with a dropdown menu showing "MSFT - ADBE", "Training Period" with a dropdown menu showing "2011 - 2015", "Testing Period" with a text input field showing "2016", and "Strategy" with a dropdown menu showing "No Strategy". At the bottom of the form is a large blue button labeled "Trade". The background of the interface is a dark blue grid with some mathematical formulas like $1+x+y+2a+21$ and $2+...+2a+...+a$.

圖 6.3: Demo System 首頁

在交易數據的頁面中，包含訓練期間與測試期間的交易數據。首先說明訓練期間的交易數據。如圖 6.4 所示，兩大折線圖為配對交易組合中的兩檔股票在訓練期間的歷史價格。其中，紅色的點表示「買進」，綠色的點則是代表「賣出」。透過滑鼠在圖表縮放，能夠聚焦於某一段訓練期間。



圖 6.4: 訓練期間股價走勢

接著，在圖 6.5 中呈現的是訓練期間交易數據的總結。「FINAL PORTFOLIO VALUE」呈現的是智能體最終獲得的現值；「PROFIT」呈現的是智能體最終獲得的利潤；「NUMBER OF ACTIONS」則是呈現智能體的總交易次數。在 Demo System 中也可以看到智能體執行 Action 的詳細記錄。在表中可以觀察智能體每一天的交易行為與交易量。

FINAL PORTFOLIO VALUE

\$2426.45

\$

PROFIT %

242

NUMBER OF ACTIONS

1138

Actions

# Action	Date	Buy	Sell	Buy Volume	Sell Volume
1	June 23, 2011	ADBE	MSFT	0.0	0.0
2	June 24, 2011	MSFT	ADBE	1.0	-1.233333
3	June 27, 2011	MSFT	ADBE	1.0	-1.225794
4	June 28, 2011	ADBE	MSFT	0.0	0.0
5	June 29, 2011	ADBE	MSFT	0.0	0.0

圖 6.5: 交易數據的總結

在測試區間的區塊中，呈現與訓練區間一樣的資訊，但是針對智能體在測試區間的表現，如圖 6.6 所示。

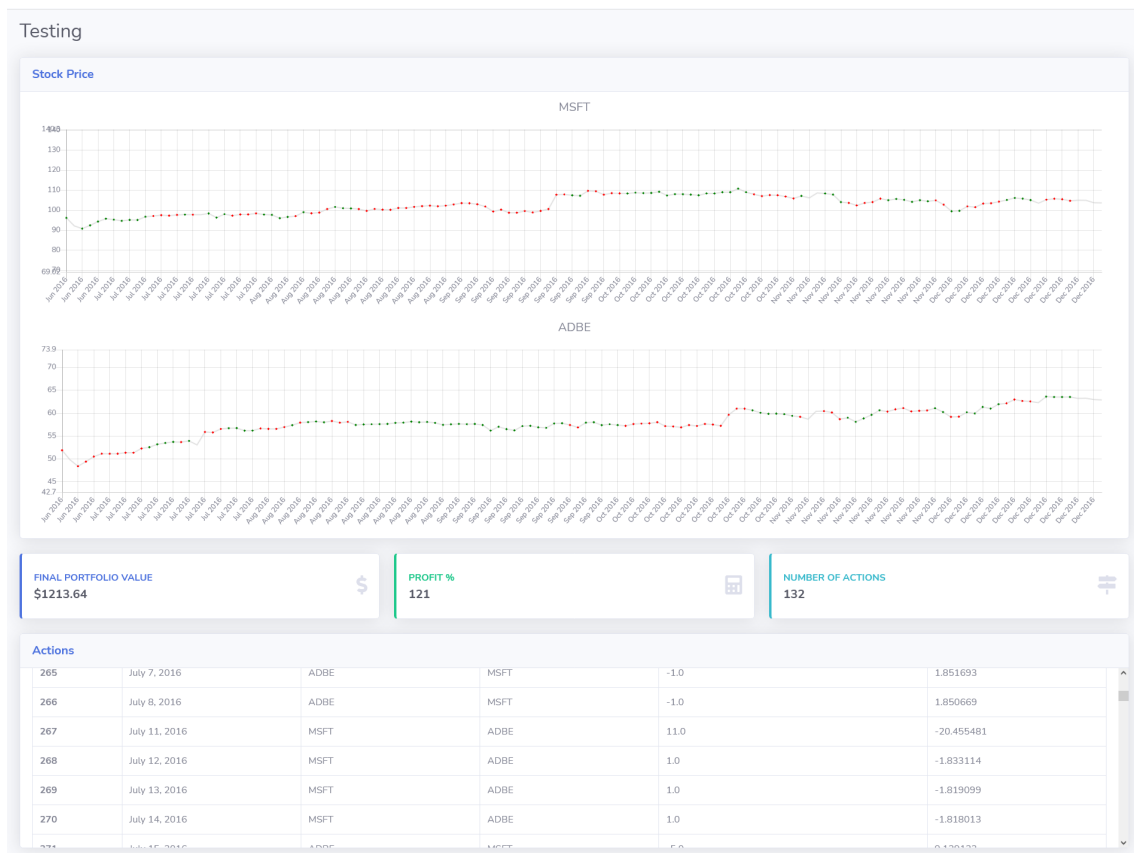


圖 6.6: 測試區間的交易數據

Chapter 7

結論與貢獻

此研究在於了解強化學習演算法應用於股市中配對交易的表現情形。我們聚焦於美國的電子產業，依據市值總共挑選出 26 間企業進行後續實驗。在實驗一中，我們透過距離法與協整法各挑選出 6 組股票組合，並進行 2015 年到 2019 年的回測。我們發現透過單一策略 (距離法或協整法) 所挑選出來的配對交易組合存在偏差。例如：距離法注重價格走勢的相似程度，容易挑選出屬於完全相同產業的公司 (Mastercard 與 Visa)。然而，這並不表示兩檔股票的價差一定可以促使配對交易獲利。

在實驗二中，我們比較傳統方法、機器學習方法與強化學習演算法針對實驗一最終所挑選出來的 4 組股票進行回測。透過報酬、最大回撤與夏普指標，我們發現強化學習演算法針對距離法所挑選出來的股票有較為優異的表現。此外，強化學習演算法也比傳統方法或是機器學習方法造成相對小的報酬波動。

在實驗三中，我們針對強化學習演算法的訓練環境加入停損與停利的機制並研究其績效表現。我們發現在智能體的訓練過程若加入停損與停利的限制，將會有效提升智能體在報酬、最大回撤與夏普指標的表現。換句話說，智能體更能夠降低投資組合淨值的波動程度，並在承受單位風險時獲取更多報酬。

在未來的發展中，我們期望進行智能體的訓練與配對交易的回測時，能夠與真實環境夠加貼近。除了考慮交易的手續費外，也將滑價成本 (Slippage Effect) 納入。此外，除了提供兩檔股票的歷史股價之外，我們也希望提供智能體更多市場資訊。例如，該產業具指標性的指數 (Index)、經濟與市場指標 (Indicator) 或是市場的情緒 (Sentiment)。最後，除了透過 Value-Based 的強化學習演算法訓練智能體，我們也希望透過 Actor-Based 亦或是兩者的結合的演算法。期許透過這些改進，能讓智能體在配對交易回測中，並透過多項指標的衡量下，穩定的勝過傳統方法。

參 考 文 獻

- [1] Francesco Bertoluzzo and Marco Corazza. Testing different reinforcement learning configurations for financial trading: Introduction and applications. *Procedia Economics and Finance*, 3:68–77, 2012.
- [2] Dimitris Bertsimas and Andrew W Lo. Optimal control of execution costs. *Journal of Financial Markets*, 1(1):1–50, 1998.
- [3] Mark Cummins and Andrea Bucca. Quantitative spread trading on crude oil and refined products markets. *Quantitative Finance*, 12(12):1857–1875, 2012.
- [4] Saeid Fallahpour, Hasan Hakimian, Khalil Taheri, and Ehsan Ramezanifar. Pairs trading strategy optimization using the reinforcement learning method: a cointegration approach. *Soft Computing*, 20(12):5051–5066, 2016.
- [5] Evan Gatev, William N Goetzmann, and K Geert Rouwenhorst. Pairs trading: Performance of a relative-value arbitrage rule. *The Review of Financial Studies*, 19(3):797–827, 2006.
- [6] Taewook Kim and Ha Young Kim. Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries. *Complexity*, 2019, 2019.
- [7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- [8] John Moody and Matthew Saffell. Learning to trade via direct reinforcement. *IEEE transactions on neural Networks*, 12(4):875–889, 2001.
- [9] Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd international conference on Machine learning*, pages 673–680, 2006.

- [10] Hossein Rad, Rand Kwong Yew Low, and Robert Faff. The profitability of pairs trading strategies: distance, cointegration and copula methods. *Quantitative Finance*, 16(10): 1541–1558, 2016.
- [11] Qiang Song, Saud Almahdi, and Steve Y Yang. Entropy based measure sentiment analysis in the financial market. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–5. IEEE, 2017.
- [12] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2016.