

國立清華大學

工業工程與工程管理學系

碩士論文

結合神經網路與簡化群體演算法

建構彈性網格交易模型

**Combined Neural Network with Simplified Swarm
Optimization for Building Flexible Grid Trading
Model**

系別 工業工程與工程管理學系 組別 工業工程組

學號姓名 109034552 謝予欣 Yu-Hsin Hsieh

指導教授 葉維彰 博士 Dr. Wei-Chang Yeh

中 華 民 國 一 百 一 十 一 年 七 月

摘要

現今隨著網際網路技術及資訊科技發展日益蓬勃，演算法交易——將交易策略邏輯預先撰寫成電腦程式以進行線上自動交易也隨之興盛，尤其在漲跌劇烈的交易市場中，使用能因應市場局勢進行自動買賣的量化交易程式更是在金融市場中引起各大金融機構及政府的關注。如何利用程式在瞬息萬變的金融市場中自動且具自我調適能力的進行交易已成為近年全體金融市場競相發展及追求的熱門研究議題。

本研究提出一套全新的線上自適應交易演算法可應用於貨幣市場、加密貨幣市場、期貨市場等金融交易市場中。於演算法前段將以簡化群體演算法進行本研究新提出的彈性網格參數優化，後段將資料匯入人工神經網路模型進行訓練，以助於在未來的各種情況下能自動選擇出相對應適合的參數組合，進行彈性網格的建構與交易。

本套全新的交易演算法得以於劇烈變動的金融市場中，適應市場多變的情境，自動適時的做出交易策略的調整並且掛單買賣，在獲利的同時，兼具風險控制考量，屬於一套兼具穩健性及獲利性之平衡型交易演算法。

關鍵字：演算法交易、簡化群體演算法、人工智慧、深度學習、神經網路、網格交易

Abstract

In our modern society, with the development of Internet and information system, pre-programming algorithmic trading strategies for online automatic trading has also flourished, especially in the rapidly fluctuating trading market. Using quantitative trading programs that can automatically trade in response to market conditions has attracted the attention of major financial institutions and governments in the financial market. How to use self-adaptive programs to automatically trade in the ever-changing financial market has become a popular research topic for the development and pursuit of all financial markets in recent years.

This research proposes an online self-adaptive trading algorithm that can be applied to financial markets such as stock market, currency markets, cryptocurrency markets, futures markets, etc. In the first part of the algorithm, the simplified swarm optimization will be used to optimize the parameters of the newly proposed flexible grid in this research. Then the data will be imported into the artificial neural network model for training in the latter part, helping the trading model automatically select the appropriate parameters for construction flexible grid corresponding to the market conditions.

The greatest contribution of the research is to provide a whole new trading algorithm that can adapt to the dynamic trading market, automatically make suitable adjustments to current trading strategy and place real-time orders. The algorithm controlling both profit and risk, which can seem as a balanced trading algorithm that are robust and profitable.

Keywords: Algorithmic trading, simplified swarm optimization, deep learning, simplified swarm optimization, artificial neural network, grid trading

目錄

摘要.....	I
Abstract.....	II
目錄.....	III
第一章、緒論.....	1
1.1 研究背景.....	1
1.2 研究動機.....	2
1.3 研究目的.....	3
1.4 研究架構.....	3
第二章、文獻回顧.....	5
2.1 量化交易.....	5
2.2 網格交易.....	7
2.3 簡化群體演算法.....	9
2.4 深度學習.....	12
2.5 近期研究概況.....	19
第三章、研究方法.....	20
3.1 網格交易之運行.....	20
3.2 彈性網格概念及架構.....	27
3.3 以簡化群體演算法尋找最適參數.....	30
3.4 訓練人工神經網路自動調整彈性網格參數.....	33
第四章、研究結果.....	35
4.1 以固定參數驗證彈性網格之表現.....	35
4.2 SSO 參數設定.....	37

4.3 以 SSO 選取參數驗證彈性網格之表現	39
4.4 訓練人工神經網路自動調整彈性網格參數	41
第五章、結論	49
參考文獻	50



圖目錄

圖 1- 1 整體研究架構.....	4
圖 2- 1 等差網格交易示意圖.....	8
圖 2- 2 等比網格交易示意圖.....	9
圖 2- 3 SSO 更新流程圖[26].....	11
圖 2- 4 人工智慧、機器學習與深度學習之間之關係圖.....	12
圖 2- 5 簡單人工神經網路架構圖.....	14
圖 2- 6 常見激勵函數.....	15
圖 2- 7 反向傳播法進行方式.....	16
圖 2- 8 簡單 RNN 模型架構.....	17
圖 2- 9 LSTM 模型架構[46].....	18
圖 3- 1 網格交易運行圖(1).....	23
圖 3- 2 網格交易運行圖(2).....	23
圖 3- 3 網格交易運行圖(3).....	24
圖 3- 4 網格交易運行圖(4).....	24
圖 3- 5 網格交易運行圖(5).....	25
圖 3- 6 網格交易運行流程圖.....	27
圖 3- 7 彈性網格交易模型示意圖.....	29
圖 3- 8 研究流程示意圖.....	30
圖 3- 9 編解碼結構.....	32
圖 3- 10 人工神經網路架構.....	34

表目錄

表 1 網格交易模型符號及定義.....	22
表 2 網格交易參數上下界.....	32
表 3 SSO 符號及定義	33
表 4 彈性網格與等差網格、等比網格之報酬率比較(固定參數版).....	36
表 5 彈性網格與等差網格、等比網格之累積財富比較(固定參數版).....	36
表 6 彈性網格與等差網格、等比網格之夏普率比較(固定參數版).....	37
表 7 不同參數組合下最佳及最差解(實驗一).....	37
表 8 不同參數組合下最佳及最差解(實驗二).....	38
表 9 不同參數組合下最佳及最差解(實驗三).....	38
表 10 彈性網格與等差網格、等比網格之報酬率比較(SSO 參數版).....	39
表 11 彈性網格與等差網格、等比網格之累積財富比較(SSO 參數版).....	40
表 12 彈性網格與等差網格、等比網格之夏普率比較(SSO 參數版).....	40
表 13 全連接神經網路超參數及架構相關設定.....	42
表 14 長短期記憶神經網路超參數及架構相關設定.....	42
表 15 全連接神經網路與長短期記憶模型之均方誤差比較.....	43
表 16 全連接神經網路與長短期記憶模型之判定係數比較.....	44
表 17 投資報酬率之比較.....	45
表 18 最大回落之比較.....	46
表 19 波動性之比較.....	47
表 20 夏普率之比較.....	47

第一章、緒論

1.1 研究背景

從 20 世紀後期至今，在網路及電腦運算技術不斷更迭發展下，進而也改變了金融市場交易方式及交易策略的制定，越來越多金融機構及各大交易從傳統的現場交易逐漸轉為電子化的遠端交易，甚至能以事先編程好的電腦程式進行自動化的交易(algorithmic trading)。據顯示，目前金融發展最為發達的美國股票市場，已有高達六至七成的交易是由已編程好的程式進行自動化交易，而在這波應用電腦運算技術與金融市場的浪潮中，量化交易(quantitative trading)與電腦演算的結合發展又尤為蓬勃[1]。

金融市場最主要之功能為提供交易商品的當前市場價格，予以市場參與者從中進行交易並且從中獲利。而量化交易興起於 20 世紀後期的股票市場，近年來被大量應用於股票、貨幣及期貨等市場之自動化交易系統，透過觀察過去大量歷史數據，以統計學及數學模型求取市場未來高機率會發生的情況，並制定相對應的一套模型化交易邏輯進行市場交易，以期帶來的超額收益。

量化交易最大的特色為使用一套固定的邏輯進行交易，以求穩定且持續的獲得高於平均回報的超額收益，最常被應用於股票及期貨交易市場。量化交易發展至今，已具有完整的系統性，目前已有多種模型可被廣泛應用於各式的金融活動中，常見的有金融商品挑選、投資組合配置、交易時機決定等等方面。

另外量化交易所考量的面向也越來越多元，包括市場組成結構、商品估值、投資標的發展性評估、市場情緒…等，皆可以拿來進一步分析及參考。配合資訊時代的發展，越來越多相關的資訊被完整的收集及保留了下來，能提供量化交易

模型去參考與分析，補足人腦無法快速消化巨量資料的不足，並且能更加客觀且不受市場情緒影響的進行決策。

1.2 研究動機

如今隨著時代的變遷，國際貿易與網路促使了現今活絡的自由交易市場，其降低了交易成本，擴大了交易規模及範疇，金融產業與科技的關係也日益加深，愈發成熟普及[2, 3]。如何利用日漸發達的電腦運算，作為金融市場交易的輔助工具，甚至成為交易者獲利的致勝關鍵已儼然成為近年金融界爭相關注及發展的研究重點。

初期的電子商務應用，多注重如何將實體交易形式轉型為電子交易形式，包含將交易資訊以電子文件形式進行輸入及儲存，方便日後查找及分析；將交易程序，包含開戶、匯款、提款等等簡單交易從人工服務改為電子服務，大量減少了人力成本。

而近年來，電子金融在有了初步的成果後，金融業者們希望除了能以電腦完成簡單的交易程序，能進一步的利用成熟的電腦程式運算，協助人們完成更複雜的交易決策。尤其近年人工智慧的日益蓬勃，有了許多突破性發展[4-7]，如何利用電腦程式，模仿人類的思考邏輯，輔助交易進行，甚至是思考的比人類更快更廣更深，已成為了近年金融領域中，諸多人致力於開發及研究的方向。

交易相關的程式演算法於 1980 年代末期興起至今，初期較多採人為或以統計工具進行分析及決策後，再以電腦程式進行交易。但時至今日，近十到二十年間，越來越多量化交易將決策過程模型化，與交易邏輯一同設計進入交易演算法中，使演算法更加智能及自動化，成為一門高實用度之熱門研究，受到許多政府及金融機構關注及實踐[8]。

時至今日交易演算法研究領域中更是有了許多豐富多元的研究成果，如利用均數回歸進行股票投資權重配置的調整[9]，以長短期記憶模型(Long Short-term Memory, LSTM)預測市場走勢結合網格交易法(grid trading method, GTM)進行貨幣交易[10]，以箱型理論(box theory)結合支持向量機(support vector machine, SVM)輔助股票交易決策[11]，基於一目平衡圖(Ichimoku Kinkohyo)分析進行外匯交易[12]，以市場趨勢代替固定時間序列作為交易決策實行單位之交易策略[13]等。不難看出，以電腦程式輔助交易，甚至直接做出交易決策及執行交易，已成為金融發展中勢不可擋的趨勢。

1.3 研究目的

繼前文所提到之電子商務及量化交易的崛起，人們的經濟活動與科技之間的關係越來越密不可分，許多交易皆仰賴程式來完成。而在未來趨勢中，程式不只是單純幫助人們紀錄交易、完成轉帳，甚至能幫助人們自動做出交易決策，包含決定交易的時機與價格。在前述種種背景之下，本篇論文的目的為：

1. 提供一套全新的網格交易演算法，改善市場上現有網格交易模型過早進場及退場之弊病。
2. 此交易演算法可隨著時間及市場條件的改變，自行適應外在環境的變化，進行模型的自我調整，減少投資人在交易市場中所花費之心力。
3. 透過一套有邏輯規則的交易模型，降低投資人主觀的交易決策所帶來的非理性決策。
4. 平衡風險與獲利之間的關係，在一定的合理風險下獲得良好的報酬。

1.4 研究架構

在第一章緒論中，闡述了交易演算法(trading algorithm)的發展近況及其重要性及影響力，同時講述了本研究之研究背景、動機與目的。

第二章中闡述了量化交易演算法(quantitative trading algorithm)的定義及特色，簡介近年廣泛被使用的網格交易策略，簡化群體演算法(Simplified Swarm Optimization, SSO)的運作邏輯與應用，及人工智慧(artificial intelligence, AI)近年於深度學習(deep learning, DL)方面的發展，包括人工神經網路(artificial neural networks, ANN)的介紹及應用。

第三章會描述模型建構及訓練過程，包括以下幾個部分：網格交易的運行機制、彈性網格的概念及建構、利用簡化群體演算法優化不同狀況下的彈性網格參數、利用市場情況與簡化群體演算法所優化之網格參數組合進行神經網路的學習，最後以訓練後的神經網路模型來對不同市場情況進行參數挑選，建構出彈性網格模型用以進行交易。

在第四章中我們會以今日量化交易中最具代表性的股票市場為標竿資料，進行實驗，並與原始市面普遍所使用的等比及等差網格進行結果比較，比較項目包含年化報酬率(annualized percentage yield)、夏普率(Sharpe ratio)及最大交易回落(maximum drawdown)等交易績效評估指標。

最後結論部分，會將此研究所提出模型之表現及所帶來之貢獻作總結，並提供一些未來發展及優化的可能方向，整體研究架構如圖 1-1 所示。



圖 1-1 整體研究架構

第二章、文獻回顧

2.1 量化交易

量化交易最主要就是將人們的投資決策規則化後，編寫成一固定可量化的運作邏輯，屏除人的一切情緒干擾及主觀想法的影響，將此邏輯套用於金融市場活動中，以期獲得高於平均收益的超額利潤。模型及規則建構的過程中，可利用大量的過往歷史資料進行模型驗證，並且在實行階段使用電腦程式進行實際的自動化金融活動。

量化交易的建構過程時常是基於統計學，透過巨量的過往資料計算出最大概率的未來情況，進而做出相對應的反應。過程中往往會透過到機率、期望值、標準差，計算出對應的投資報酬率、 α 值、夏普值等相關評估預期獲利及風險的指標，來進行量化邏輯及模型的優化及表現評估。

於量化交易中又分為主要幾個交易策略，以下將一一分別簡述其概念及內涵：

α 策略(alpha strategies)：

α 策略的核心概念是在眾多的投資標的中(比方說多支股票或是各種貨幣中)，挑選出較為優良的項目作多，並且做空相對較差的項目。此種策略屬於交易頻率相對較低的方法，風險低但收益也相對偏低。

套利策略(arbitrage strategies)[14]：

套利策略的核心為利用市場的價格差異獲利，如利用相同商品在不同市場的價格差異或同市場不同商品的價格差異來獲取價差利潤。另有一類套利是利用商品價值會隨著時間而產生價格差異，如現有一商品市價 90 元，但估計一星期後其市值會成長為 92 元，則此時量化交易模型便會發出交易訊號買進此商品，此類演算法通常會有一商品價格估價模型，為結果好壞之主要關鍵。

通常此模型之基本假設為商品價值具有均數回歸(mean reversion)現象[15]，若價格顯著偏移長期平均水平時，就會發出交易訊號，藉此獲利。此策略常用於價格具浮動性之商品，若商品長期處於價格穩定狀態則不適用此策略。

均值回歸策略(mean reversion strategies)[15]：

均值回歸是由觀察金融市場長期以來的變化所提出的理論，它假定商品價格以長期來看，即使短期內會有所浮動漲跌，最終都應回復到其一穩定均值。通常於量化交易上應用此策略會以程式尋找具有長期均值特徵的市場，並且在其偏離時發出訊號，計算獲益率。如果價格往下偏離均值，則發出買入訊號；若價格高於均值一定量之後，發出賣出訊號，以此邏輯尋找獲利機會。

趨勢追隨策略(trend following)[16]：

趨勢追隨為量化交易中另一大主流策略，為最直接的方法之一，其只追求在一開始時識別市場中的趨勢並追隨此趨勢直到最後。

趨向策略[17]：

透過資訊收集，以可能對市場造成影響之消息、事件或是數據等作為依據預測市場未來走向，從中尋找獲利機會。主要分為兩種：事件策略(event strategies)[18]及流動性偵測策略[19, 20] (liquidity detection strategies)。

事件策略採事件觸發機制，其會收集大型金融機構及國家政府對於未來總體經濟或產業所發表之報告書、公司財報及聲明、國際情勢等資訊，依過去市場對於相關消息所造成之影響作為預測依據，以經驗法則及統計相關分析得出未來市場趨勢並先於市場大眾做出即時反應為此種策略主要獲利之方法。

而流動性偵測策略則是去找出市場中潛在的關鍵交易者意圖，關鍵交易者可以是買方或是賣方，其行為於整體市場具有一定的影響力，足以為市場價格帶來波動。事先於價格波動前偵測關鍵交易者可能做出之買賣行為，藉此早一步做出相對應行動為此策略之主旨。

總的來說，量化交易能以程式快速處理現代社會各個系統所收集來的資料，取代了傳統交易上只能依照個人經驗與片面少量的資訊作為參考依據，更全面的對於市場情勢進行分析及判斷。同時遵照一套固定的邏輯及數學模型進行交易活動，可以避免受情緒受外在影響，進而減少交易操作上的誤判。

2.2 網格交易

網格交易(grid trading method)最基本的定義為在特定價格區間內平均的掛單，進行買低賣高的交易策略[21]。網格交易原理簡單卻能有效的在價格波動時賺取差價利潤，而此策略於近年來常被應用於諸多金融市場中進行套利。

網格交易有兩大部分關鍵決定此策略之成效，第一為價格波動性，其中包括漲跌兩方向：若價格為一路上漲，手中持有現貨者獲利會比網格交易者多；若價格為一路下跌，做現貨交易者基本上無法有太多獲利機會，惟期貨買空者較能在價格一路往下時有獲利能力；若市場有漲有跌者，採用網格交易策略者會擁有最大的獲利能力，其中波動越頻繁越大，獲利率會隨之增加。而在絕大多數之市場中，價格多為高度浮動的，且往往具有價格平均回歸性(price mean reversion)[22-24]，故網格交易策略如今被越來越多市場交易者所採用。

而另一關鍵決定網格交易之獲利大小為其參數設定，包括網格平均、網格上界、網格下界、網格格數、啟動價格、停損點、停利點等等，以上參數之設定需依照產品價格波動大小、交易成本、風險承受能力、本金多寡做為設置考量，直接的影響著網格交易最終的表現。

2.2.1 等差網格與等比網格

網格交易主要分為兩種：等差網格及等比網格，而目前實務上較多人採用之網格為等差網格。

等差網格之設置會以初始價 P_0 為基準，並設定網格上界 G_{ul} 、網格下界 G_{ll} 及網格數 n ，而網格間距會以公式 2.1 計算得出。

$$G_s = \frac{G_{ul} - G_{ll}}{n} \quad \text{公式(2.1)}$$

設定完網格模型後，就會以此網格為交易依據，隨著價格的漲跌進行金融商品的買賣，過程可參考下方圖 2-1。

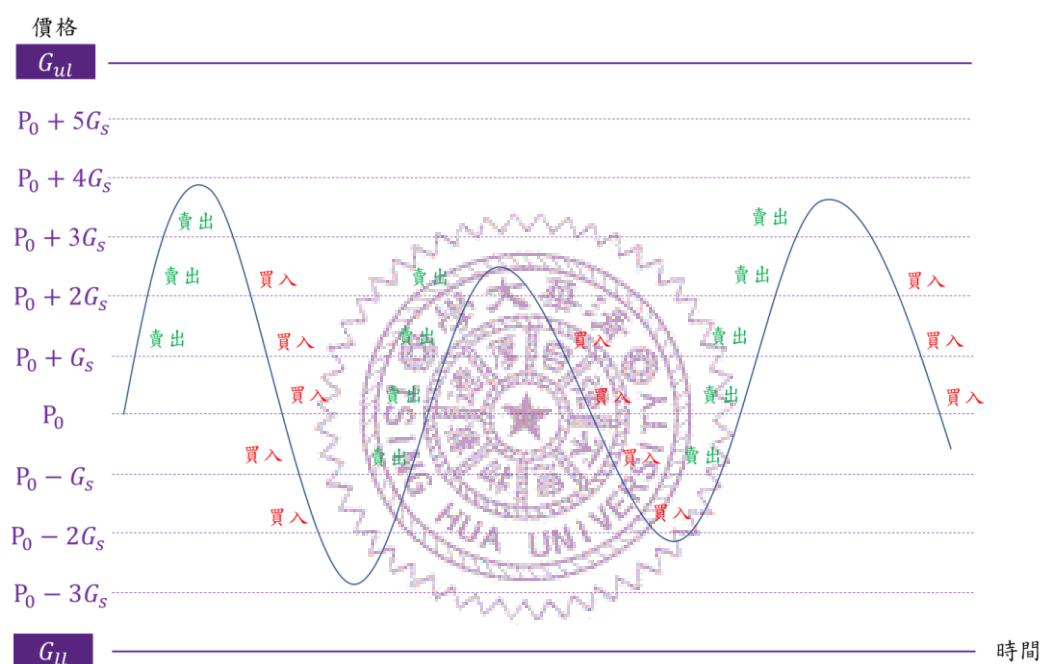


圖 2-1 等差網格交易示意圖

而等比網格之 G_s 則為一固定比率，而網格間距會以此比率計算得出，其概念可參考下方圖 2-2。

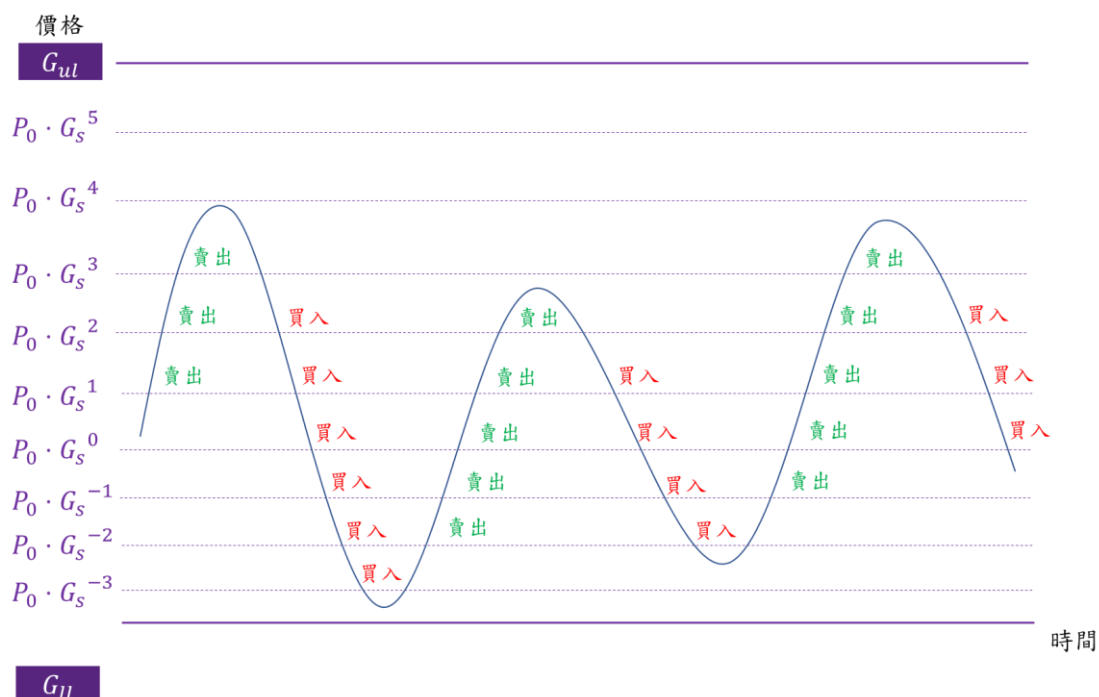


圖 2-2 等比網格交易示意圖

註：詳細網格交易過程與機制會於第三章中說明。

2.3 簡化群體演算法

粒子群演算法(Particle Swarm Optimization, PSO)是由 Kennedy, Eberhart et al. [25] 於 1990 年時，為了描繪出鳥群優美又難以預測的飛行軌跡所發想而出，屬於啟發式演算法中的一支，可在合理的時間內求出對於問題相對適當的解，常被應用於複雜且難以直觀求取最佳解的問題中。

然而 PSO 在求解離散型問題時會有過早收斂及效果不佳的問題，因此 Yeh [26] 於 2008 年以大道至簡的核心理念，提出簡化群體演算法(Simplified Swarm Optimization, SSO)以改善 PSO 之不足，在簡化演算法的同時，提升了求解效率，目前已廣為應用於多個不同領域之問題求解[27-29]。而求取最佳參數組合為 SSO 應用中的其中一塊，可同時考量並調整多個參數，以求取表現最佳的參數組合，曾被用來調整卷積神經網路超參數組合[30]。

PSO 的更新機制需要兩個隨機及三個預設值(w 、 C_1 和 C_2)進行更新，如公式(2.2)與(2.3)所示，可看出其求解過程受區域最佳解(Personal Best, $pbest$)及全域最佳解(global best, $gbest$)所影響。SSO 延續 $pbest$ 及 $gbest$ 的概念並加入隨機數[30]，使解能有機會跳脫出局部最佳解，提升了解的多元性。

$$v_{ij}^t = w \cdot v_{ij}^{t-1} + c_1 \cdot \rho_1 \cdot (P_{ij}^{t-1} - x_{ij}^{t-1}) + c_2 \cdot \rho_2 \cdot (g_j^{t-1} - x_{ij}^{t-1}) \quad \text{公式(2.2)}$$

$$x_{ij}^t = x_{ij}^{t-1} + v_{ij}^t \quad \text{公式(2.3)}$$

SSO 與其他啟發式演算法最大的差別在於其特別的更新機制，有三個尤為重要的參數設定，分別為： C_g 、 C_p 、 C_w ，其中 $C_g > C_p > C_w$ 。更新機制如公式(2.4)及圖 2-3 所示，依隨機數 ρ 與 C_g 、 C_p 和 C_w 之間的對應關係，決定下一代的解 x_{ij}^t ，分別可能為 $pbest$ 、 $gbest$ 、當前解或一隨機數。

$$x_{ij}^{t+1} = \begin{cases} g_j, & \text{if } \rho_{ij}^t \in [0, C_g) \\ p_{ij}, & \text{if } \rho_{ij}^t \in [C_g, C_p) \\ x_{ij}^t, & \text{if } \rho_{ij}^t \in [C_p, C_w) \\ x, & \text{if } \rho_{ij}^t \in [C_w, 1) \end{cases} \quad \text{公式(2.4)}$$

其中，令 $x_i^t = x_{i1}^t, x_{i2}^t \dots x_{ij}^t$ ，以 x_{ij}^t 表示為第 t 次迭代中第 i 組解，具有 j 個變數； ρ 為介於 $[0,1]$ 間均勻分配之隨機變數。其更新方式為，當 ρ 介在 $[0, C_w)$ 間時，變數 x_{ij}^t 會維持前一代的解 x_{ij}^{t-1} ；當 ρ 介在 $[C_w, C_p)$ 間時，則會由 p_{ij}^{t-1} 區域最佳解($pbest$)取代，也就是該變數過去迭代中最優解；當 ρ 介在 $[C_p, C_g)$ 間時，則由 g_i 全域最佳解($gbest$)，也就是目前所有解中表現最好的解取代；當 ρ 介在 $[C_g, 1]$ 間時，則由 x ，一個在變數上下界中產生的隨機數取代，目的是要降低受困於局部最佳解的機會，同時還可以提升解的多樣性。

而在更新機制中，會針對不同問題與情境設置不同的 C_g 、 C_p 、 C_w ，其對於最後解的品質有著極大的影響力，在過去諸多的SSO相關研究中會使用直交表(Orthogonal Array, OA)來挑選較適合的參數配置[29-31]。

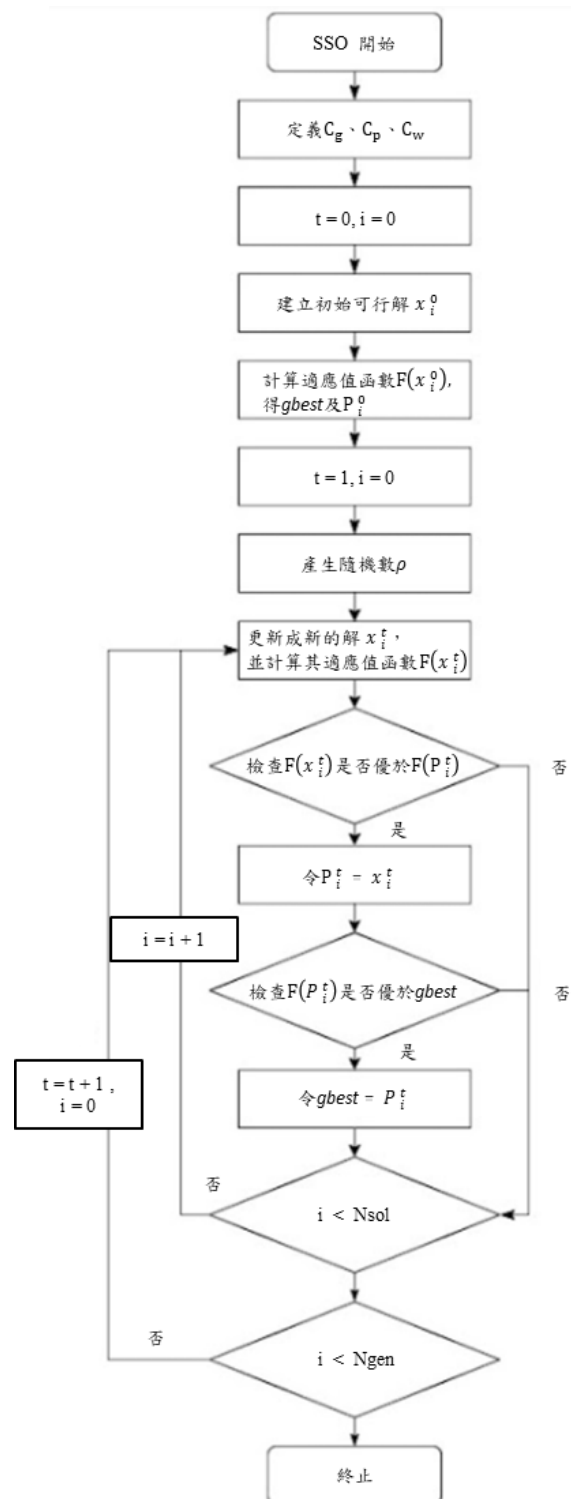


圖 2- 3 SSO 更新流程圖[26]

2.4 深度學習

近期深度學習與人工神經網路於金融活動上的應用越來越多元廣泛，其中應用最多的莫過於在股市、匯率、金融指數等方面的預測[32]，目前仍有許多研究嘗試以各種神經網路模型協助人們在金融預測及決策方面[33-35]，做出更加理性準確的判斷。

2.4.1 人工神經網路

人工智慧(artificial intelligence, AI)為近期全世界先進國家都積極發展的科技領域，主旨為希望將人類思考及判斷的能力通過運算賦予給機器，使機器或電腦能表現出如人類一般的智慧[36]。

人工智慧領域中又包含了機器學習(machine learning, ML)，而深層學習(deep learning, DL)為機器學習中的一個分支(三者之間的關係可參照下方圖2-4)。深度學習近年來被廣泛應用於醫學、工業、交通等領域[37]，可幫助語音辨識及機器視覺的完成，是一種以人工神經網路(artificial neural network, ANN)為架構的機器學習演算法，透過資料的特徵進行學習。

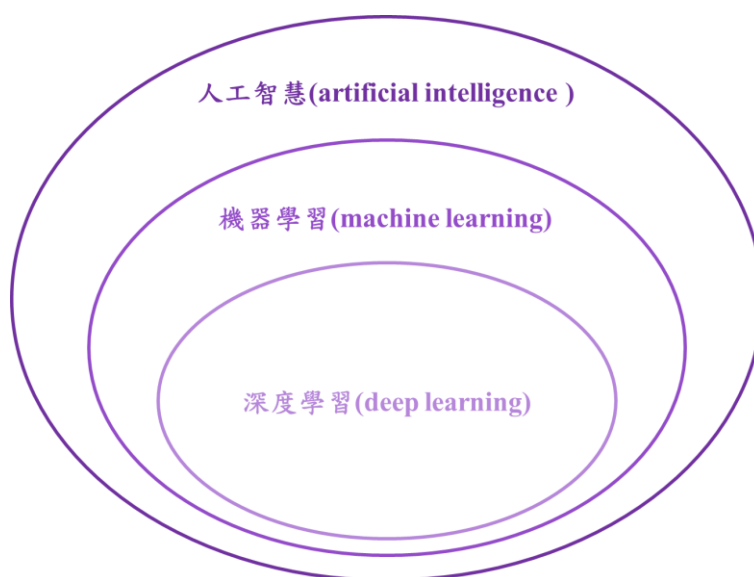


圖 2-4 人工智慧、機器學習與深度學習之間之關係圖

人工神經網路之計算架構最早由 WS. McCulloch 及 W. Pitts[38]提出，後續經多位傑出學者不斷的更迭改進[39, 40]，成為了人工智慧中的著名機器學習模型。人工神經網路為仿生物神經網路結構而產出的模型，試圖模仿生物神經傳導及大腦思考的模式，以大量的人工神經元為單位，相互連結並進行運算，可依訓練資料進行自我調適，並以半監督式(semi-supervised)或非監督式(un-supervised)方式進行學習[41]。

人工神經網路的基本架構分為三個部分，分別為輸入層(input layer)、隱藏層(hidden layer)及輸出層(output layer)，可參考圖 2-5。輸入層的每個節點對應到的是所輸入的預測變數 x_j ， j 為輸入層節點編號，如 x_1 為輸入層之第一個節點。輸出層各個節點對應到的是目標變數 y_j ，為輸出層結點編號，例 y_1 為輸出層之第一個節點。夾在兩者之間的是隱藏層，其中節點以 h_{ij} 表示， i 代表第幾層， j 代表該層的幾個節點，如 h_{23} 為第一層隱藏層之第三個節點。此處須注意當 $i=1$ 時，代表的是位於整體人工神經網路最前端的輸入層，隱藏層的開始是從 $i=2$ 時。

而 w_{i,j_1,j_2} 所代表的是第 i 與 $i+1$ 層中，上層第 j_1 個節點與下層第 j_2 個節點間的權重，如下圖 2-5 中，節點 h_{21} 與節點 h_{31} 間的權重以 $w_{2,1,1}$ 表示。 b_i 所代表的是第 i 與 $i+1$ 層間的誤差，第一層隱藏層與第二層隱藏層間的誤差以 b_2 表示。

隱藏層層數可以是單層或單層以上，甚至到上百層不等，層與層之間的節點會相互連接，形成複雜的神經網路，而人工神經網路之所以會被稱為深度學習的原因就在於此。

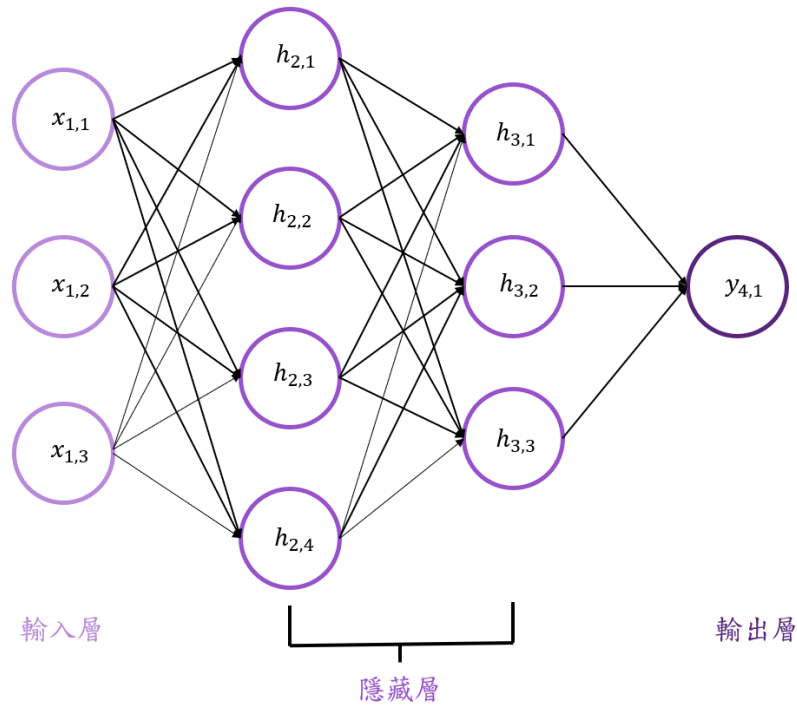


圖 2-5 簡單人工神經網路架構圖

除了輸入層的節點外，人工神經網路中的每個節點都會與其前面的數個節點相互連接（此種模型又稱之為全連接神經網路，fully connected neural network, FNN），而每個節點都具有一對應權重 w ，而此節點的值為前節點所得出之值與其權重之乘積，再經一激勵函數 $f(x)$ (activation function)計算得出。

其中隱藏層節點值與輸出值以公式 2.5 及公式 2.6 計算得出：

$$h_{i+1,j_2} = \sum_{j_1} h_{i,j_1} \times w_{i,j_1,j_2} + b_i \quad \text{公式(2.5)}$$

$$y_{i+1,j_2} = \sum_{j_1} h_{i,j_1} \times w_{i,j_1,j_2} + b_i \quad \text{公式(2.6)}$$

激勵函數為一非線性函數，若無激勵函數的引進，ANN 架構會以單純的線性方式進行交互運算，節點之間關係為線性，但實務中，大部分輸入與輸出變數之間的關係為非線性，若無激勵函數的輔助，ANN 所訓練出之模型便很難具有實質的意義及實際解問題的能力[42]。

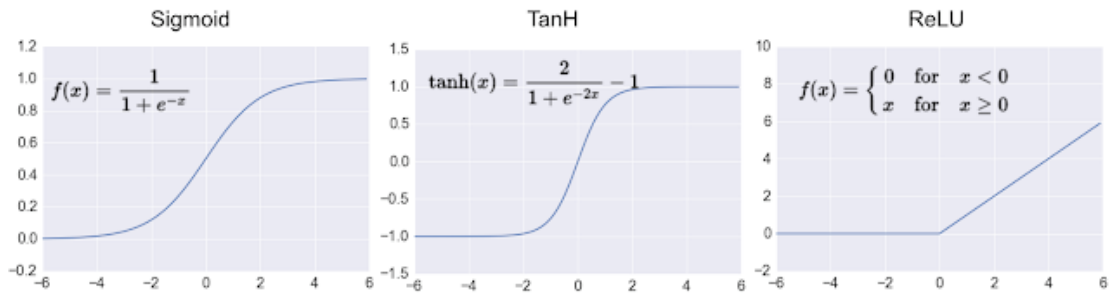


圖 2-6 常見激勵函數

2.4.2 反向傳播

在深度學習中，反向傳播法[43]為一個使模型完整極為重要的關鍵，常被用來訓練及優化人工神經網路。利用反向傳播，可以有效率的找到損失函數對於權重的梯度，後續再以梯度下降法（gradient descent）[44]來對各個權重求解。而其中損失函數中的「損失」意指「實際值和預測值的殘差」。反向傳播最主要的概念，就是將誤差值回傳，使權重可以利用誤差大小進行梯度下降法來求取及更新更適合的權重，進一步的降低誤差，優化權重。

而在反向傳播法中，又可分為兩大階段：前向傳播與反向傳播，進行流程可參考下方圖 2-3。

1. 第一階段：前向傳播

以隨機產出，或是目前所有的權重進行由輸入層、隱藏層到輸出層的傳播計算，計算出最終輸出值。

2. 第二階段：反向傳播

由損失函數計算出網路輸出值與目標值間的差，將差值回傳，以此作為優化依據，並以梯度下降法進行權重優化。

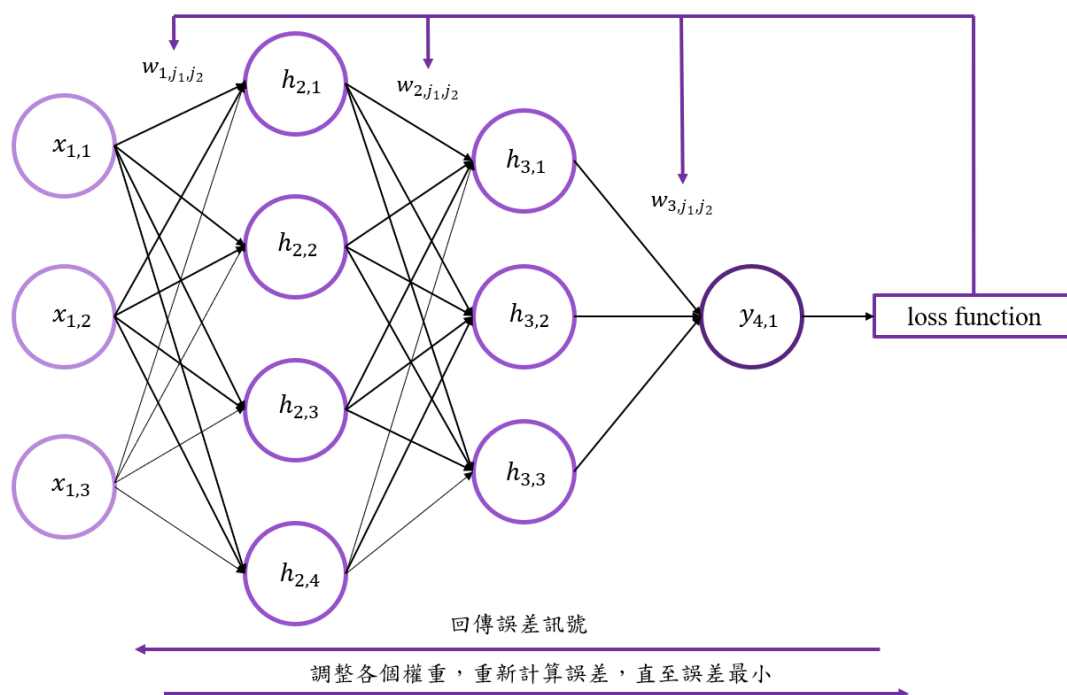


圖 2-7 反向傳播法進行方式

2.4.3 遞迴神經網路

遞迴神經網路(recurrent neural networks, RNN)為 ANN 中的一種，可處理與時間序列相關的問題，也就是說當下或未來的情況與過去發生的事情具有關聯性。RNN 將問題的特徵以權重方式儲存在網路架構中的各個神經元中，用過去發生過之歷史資料，預測未來會出現的情況，其架構如圖 2-8 所示。其中 A 為神經網路，透過讀取輸入層之 t 時間的狀態 x ，然後由輸出層輸出一個預測值 h_t ，循環此過程可以使得信息從當前時間單位傳遞到下一時間單位。詳細結點值之計算公式可參考公式 2.7 與公式 2.8[45]。

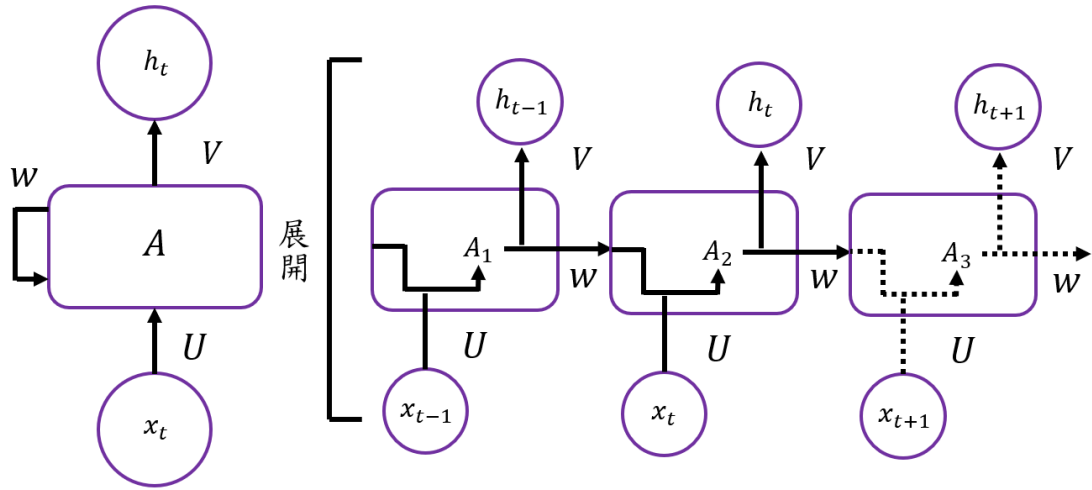


圖 2- 8 簡單 RNN 模型架構

隱藏層節點值計算：

$$h(t) = f(Ux(t) + Wh(t-1)) \quad \text{公式(2.7)}$$

輸出層節點值計算：

$$y(t) = f(Vh(t)) \quad \text{公式(2.8)}$$

其中 $h(t)$ 為 t 時之隱藏層之值， $y(t)$ 為 t 時之輸出層之值， $f(t)$ 為激勵函數， U 為輸入層與隱藏層間之權重， W 為隱藏層與隱藏層間之權重， V 為隱藏層與輸出層間之權重。

RNN 因其以上特徵，目前已廣泛的運用於語音辨識、文本分析、天氣與股價預測等領域中，為現今深度學習中一廣為人知的人工神經網路模型。

2.4.4 長短期記憶模型

雖說 RNN 可以用先前的歷史資料推算未來狀態表現，也就是具有資訊記憶性，但僅限於短期記憶，較久以前的所發生的資訊往往會隨著模型的訓練遭到遺忘。長短期記憶模型(Long Short-term Memory, LSTM)為改善 RNN 短期記憶性所產生出來的模型，其主要由四個單位所組成，分別為記憶單元(memory

cell)及輸入閘(input gate)、輸出閘(output gate)、遺忘閘(forget gate)三個閘門所組成，其架構可參考圖 2-9 [46]。

輸入閘調控此次是否輸入記憶單元中，記憶單元負責儲存計算出之數值，遺忘閘控制將記憶清除，輸出閘控制是否將此次運算結果輸出(公式 2.9-公式 2.14)。

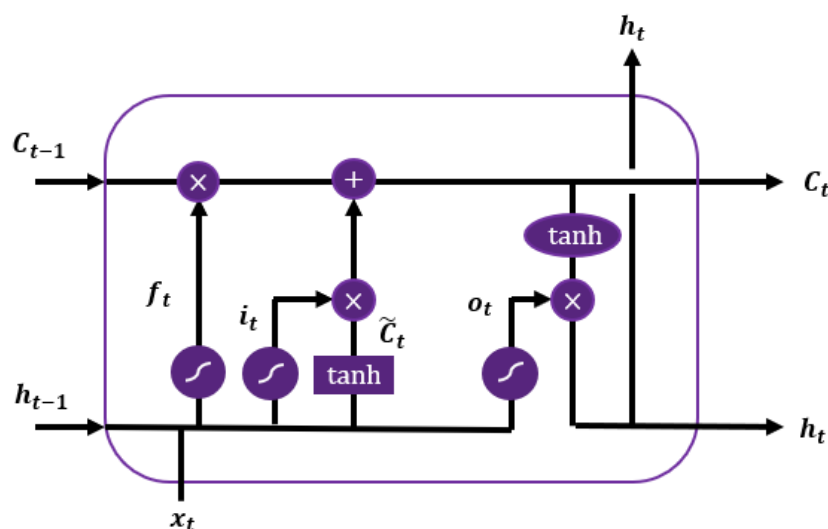


圖 2-9 LSTM 模型架構[46]

決定上層與本層資訊有多少需要被遺忘：

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{公式(2.9)}$$

決定須被留下的資訊並更新記憶單元：

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{公式(2.10)}$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad \text{公式(2.11)}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad \text{公式(2.12)}$$

最後決定被輸出之資訊

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{公式(2.13)}$$

$$h_t = o_t \times \tanh(C_t) \quad \text{公式(2.14)}$$

2.5 近期研究概況

近期將機器學習應用於量化交易上的方法主要有強化學習(reinforcement learning)、支持向量機(support vector machine)、基因演算法(genetic algorithms)、隨機森林(random forest)、人工神經網路(artificial neural networks)等。

而在神經網路這塊，目前在金融領域方面已有大量的運用及研究，例如應用多層神經網路與多層感知機在金價銀價預測上，在結果上優於傳統線性 ARMA 預測模型[47]；使用 RNN 進行市場預測，訓練多個不同神經網路，並在一開始進行分析，選擇目前市場情況所適用的神經網路進行預測[48]；以 LSTM 進行市場預測，並依結果去建構適當的投資組合，結果優於以線性回歸與 SVM 預測結果所建構的投資組合[49]；以增強學習與 LSTM 建構決策模型，輸入市場金融資訊，輸出對應目前市場情況所適合之交易決策[50]；以增強學習與 RNN 建構交易模型，結果穩健，能在相對穩定的風險下獲取不錯的報酬。[51]

綜上所述，本研究會使用全新提出的網格交易模型，以簡化群體演算法求出各種市場情勢所適合的參數作為輸入值及標籤，訓練全連接神經網路及長短期記憶模型，最後得出一量化交易模型能在輸入現有市場情勢後，自動計算及調整出最佳交易參數進行交易。

而比較方法除了會與現有市面上之等差網格及等比網格比較，還會和細田悟一提出之一目均衡圖之技術分析[12]及以長短期記憶模型預測市場走勢結合網格交易法[10]來做比較，其中值得注意的是此處的網格交易所設定的網格高與網格寬是以等差網格為基礎所設定而出。

第三章、研究方法

在第三章中依序會介紹：網格交易的初始設定及運作機制(3.1 節)，以 SSO 求取等差與等比網格參數作為結果比較的方法(3.2 節)，介紹彈性網格概念與設置方法(3.3 節)，並在 3.4 節中以 SSO 求取不同情況下的最佳彈性網格參數，作為 3.5 節深度學習中的訓練依據，以期最後能訓練出一神經網路能透過輸入近期市場資訊，自動輸出最適的網格參數進行交易，以求從市場波動中獲取超額報酬。

3.1 網格交易之運行

接續 2.2 提到的基本網格交易概念，本節會更加詳細的說明本研究中會如何實踐網格交易，包括初始的參數設定與計算，及後續的運作機制與流程。

3.1.1 網格交易初始設定

在運作一網格交易模型前，共有五個基礎參數需要設定，分別是總投入資金 F_0 、初始價格 P_0 、網格上界 G_{ul} 、網格下界 G_{ll} 及網格數 n 。其中總投入資金與初始價格在本研究中被設定為控制變數，也就是說在結果比較時，任意方法中使用的總投入資金與初始價格會設定為一相同定值，在相同的基準上進行比較。

而在網格交易模型運行前，若網格為等差網格需先以第二章中公式 2.1 計算出單格價差 G_s ，也就是網格間距。 G_s 在等差網格時為一定值，但在等比網格中會為一固定比例，例如當 $P_0 = 100$, $G_s = 1.1$ 時，網格下一格之值為

$P_0 \times G_s^1 = 110$ ，在下一格為 $P_0 \times G_s^2 = 121$ ，而此比例可由以下公式 3.1 計算而出。

$$G_s = \sqrt[n]{\frac{G_{ul} - G_{ll}}{G_{ll}} + 1} \quad \text{公式(3.1)}$$

另外我們還需推算出在網格交易前應該使用多少資金先行購買現貨，也就是起始購買現貨 S_0 ，以供價格上漲時賣出獲利；以及手頭上須保留多少資金以供價格下跌時買入現貨，在此以 C_0 表示起始持有現金。也就是說總投入資金會被分為兩部分，以下以式子 3.2 表示。而後續每期持有資金及現貨以 S_j 及 C_j 表示，其中 j 代表第 j 期。

$$F_0 = S_0 + C_0 \quad \text{公式(3.2)}$$

而要計算出起始購買現貨 S_0 與起始持有現金 C_0 ，我們還需先計算出起始上方格數 n_u 與起始下方格數 n_l ，兩者總和相當於總網格數 n ，如公式 3.3 所示，而兩者之值可分別以公式 3.4 及公式 3.5 得出(此處以等差為例)。

$$n = n_u + n_l \quad \text{公式(3.3)}$$

$$n_u = (G_{ul} - P_0) \div G_s \quad \text{公式(3.4)}$$

$$n_l = (P_0 - G_{ll}) \div G_s \quad \text{公式(3.5)}$$

計算出起始上方格數 n_u 與起始下方格數 n_l 後，再進一步以公式 3.6 與公式 3.7 計算起始購買現貨 S_0 與起始持有現金 C_0 ，此時單格交易量 G_v 尚且為一未知數。

$$S_0 = G_v \times n_u \times P_0 \quad \text{公式(3.6)}$$

$$C_0 = G_v \times [(P_0 - G_s) + G_{ll}] \div 2 \times n_l \quad \text{公式(3.7)}$$

最後再將公式 3.6 與 3.7 代回公式 3.2 就能得出單格交易量 G_v ，計算可參考公式 3.8。

$$G_v = F_0 \div \{[(P_0 - G_s) + G_{ll}] \div 2 \times n_l + n_u \times P_0\} \quad \text{公式(3.8)}$$

最後再以公式 3.9 (等差網格)或公式 3.10 (等比網格)計算出網格各格的價格 g_i , i from 1 to n 。

$$g_i = G_{ll} + G_s \times (i - 1) \quad \text{公式(3.9)}$$

$$g_i = G_{ll} \times G_s^{(i-1)} \quad \text{公式(3.10)}$$

至此，我們可用以上公式，透過初始網格參數設置，計算出一網格交易模型運作時所需的種種參數，包括：單格價差 G_s 、單格交易量 G_v 、起始購買現貨量 S_0 、起始持有現金 C_0 、起始上方格數 n_u 、起始下方格數 n_l 及各格價格 g_i ，符號整理可參考下表。

符號	定義
F_0	初始總投入資金
P_0	商品初始使市場價格
n	網格總數
G_{ul}	網格上限
G_{ll}	網格下限
G_s	單格價差，在等差網格時為兩網格線之間的數值差，在等比網格時為兩網格之間的倍數差，而彈性網格之定義會於 3.2 小節詳細說明。
G_v	單格交易量，單格所需賣出或買入之現貨量，為一定值。
S_0	起始購買現貨量
C_0	起始持有現金
n_u	起始上方格數
n_l	起始下方格數
g_i	各格價格， i 從 1 到 n

表 1 網格交易模型符號及定義

3.1.2 網格交易運作機制

以下會搭配圖示詳細說明網格交易模型在初始參數設定完之後，如何隨著市場價格進行更新及調整，以及結束時如何進行結算。

- 1、網格初始運行時，會以當前價格為基準，將以上的網格價格掛上賣單，以下網格掛上買單。

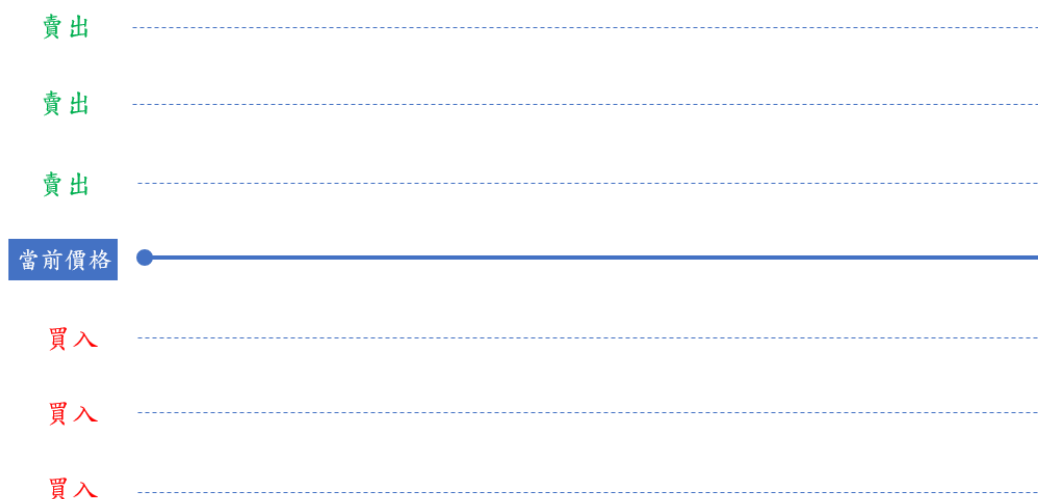


圖 3-1 網格交易運行圖(1)

- 2、若價格往上漲至碰到第一格網格線，則做出賣出動作，更新持有現貨量及資金，並且在原網格位置，掛上買單。

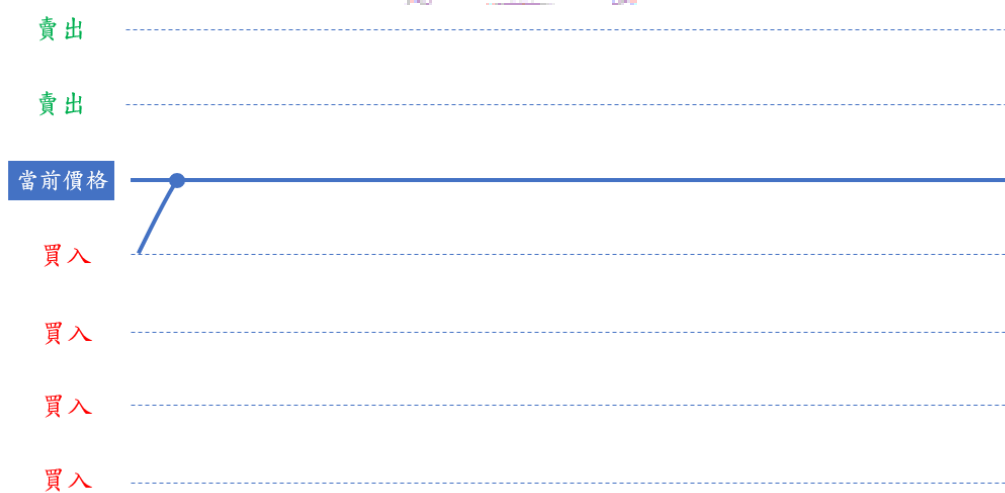


圖 3-2 網格交易運行圖(2)

- 3、若價格往下降至回到初始網格線，則做出買入動作，更新持有現貨量及資金，並且在原網格位置，掛上賣單。

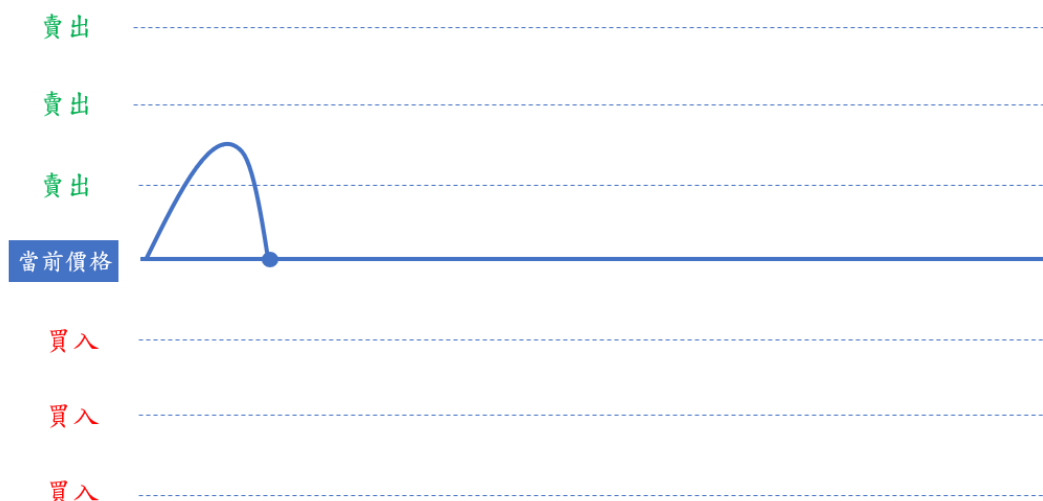


圖 3-3 網格交易運行圖(3)

- 4、若價格持續往下降至下一網格線，持續做出買入動作，更新持有現貨量及資金，並於原網格位置，掛上賣單。

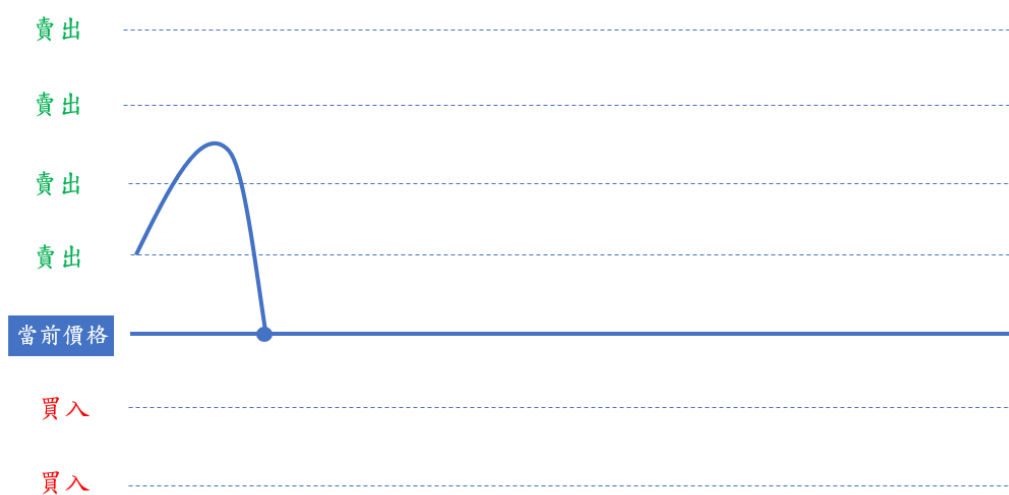


圖 3-4 網格交易運行圖(4)

- 5、持續以上機制進行交易，詳情可見圖 3-5，價格雖回到網格交易起點，卻已成功套利了 7 次，相當於獲得了 7 格網格價差利潤。

整體流程可參考下方圖 3-6 網格交易運行流程圖，而 S_0 與 C_0 經過 j 期價格輸入後，其更新過程虛擬碼如上。

在每次輸入新價格，更新完持有現貨及持有資金後，還需做重新掛單的動作，其虛擬碼如下：

Pseudocode for Grid Transaction Operation II	
if $P_{x+1} > P_x$	1
$A1 = []$ (設一空 array 來儲存需要更新掛為買單的價格)	2
y from 0 to i	3
find all $g_y \geq P_x$ and $g_y < P_{x+1}$ ，存入 A1	4
將 A1 中紀錄之 g_y 撤下賣單，掛上買單	5
if $P_{x+1} < P_x$	6
$A2 = []$ (設一空 array 來儲存需要更新掛為賣單的價格)	7
y from 0 to i	8
find all $g_y \leq P_x$ and $g_y > P_{x+1}$ ，存入 A2	9
將 A2 中紀錄之 g_y 撤下買單，掛上賣單	10

若此次價格更新無碰到任何掛單價格，則不觸發更新，此處須注意，在輸入新價格完成本次交易後，最後買賣之價格上不會掛上買單或賣單，亦即此價格於下次更新中，不會觸發任何交易活動。

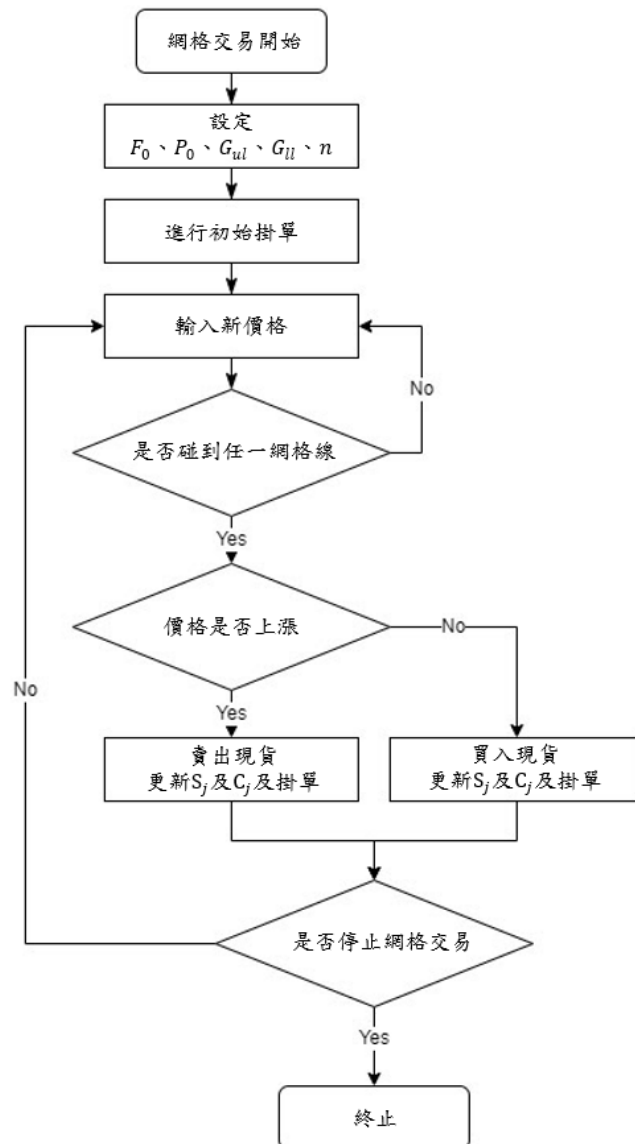


圖 3-6 網格交易運行流程圖

3.2 彈性網格概念及架構

基於現今金融市場所使用的等差及等比網格交易模型，本研究提出了一種全新且具有自適性的網格交易模型—彈性網格，希望透過更加具有彈性的網格架構，配合簡化群體演算法與深度學習，建構出一能隨著市場變化自行調整參數，並能適應各種外在條件的網格交易模型。

在等差網格交易模型的架構下，於市場震盪時，能發揮其最大的模型效益，即使價格回到原點，仍能從震盪的市場局勢中套利，主要基礎採用第二章量化交易中提到的均價回歸策略(可參考 2.1 小節)；而等比網格交易則在震盪上漲中能獲得更出色的報酬，概念結合了均價回歸及趨勢跟隨兩種策略，詳細網格架構可參考 2.3 小節。

而本研究提出的彈性網格，同時擷取了等差網格及等比網格的優勢，希望不管在市場橫盤整理、上漲甚至是下跌中，表現都能優於以往的傳統網格交易架構。

彈性網格於初始設定時，也需要設定其總投入資金 F_0 、初始價格 P_0 、網格上界 G_{ul} 、網格下界 G_{ll} 及起始上方格數 n_u 與起始下方格數 n_l ，但其初始設定時較其他模型不同的地方主要有兩點：

1. 公式 3.3 中的起始上方格數 n_u 與起始下方格數 n_l 不再是以公式 3.4 及公式 3.5 進行計算，而是可以於初始進行設定。
2. 將網格切分以初始價格 P_0 為界，切分為上下兩部份。上部分與下部分可各自設定網格數，並具有各自網格間距比例，上網格建具比例為 G_{su} 與下網格間距 G_{sl} ，這裡需要注意的是 G_{su} 要為一個大於 0 且小於 1 的數，而 G_{sl} 需為一個大於 1 的數。其特色在於上部份的網格間距，價格越高時會越來越小，也就是說，交易頻率會越來越頻繁；同理，下半部份的網格間距當價格越低時，也會越來越小，越來越密。

具體間格值計算方法，可參考公式 3.11 與公式 3.12，整體架構可參考下方圖 3-7。

$$G_{su} = \frac{1}{n_u \sqrt{\frac{G_{ul} - P_0}{P_0} + 1}} \quad \text{公式(3.11)}$$

$$G_{sl} = \frac{n_l \sqrt{\frac{P_0 - G_{ll}}{G_{ll}} + 1}}{1} \quad \text{公式(3.12)}$$

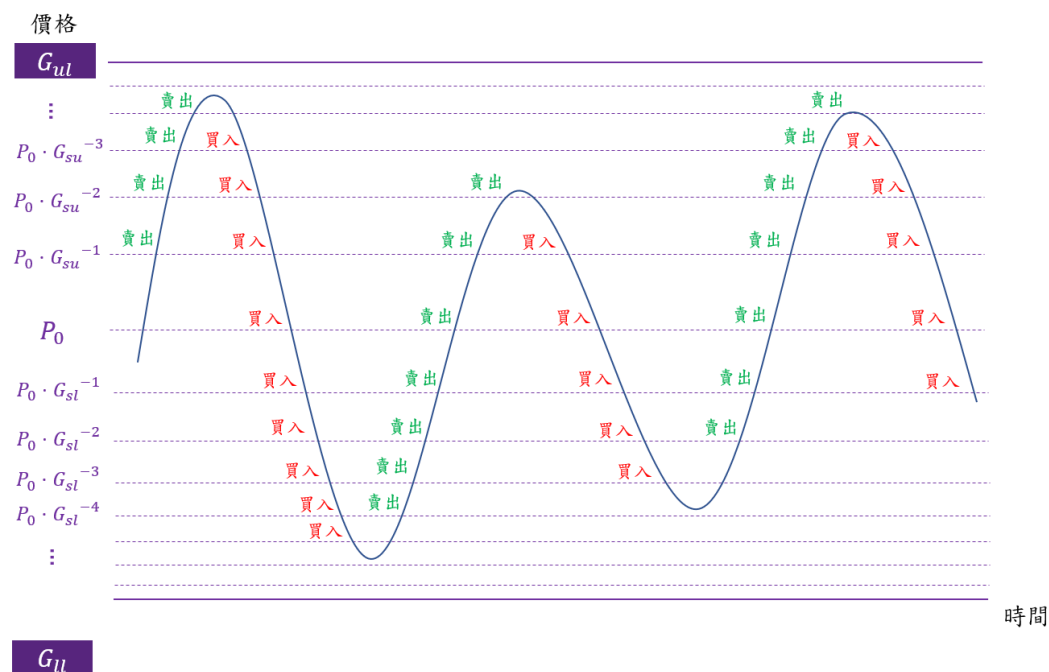


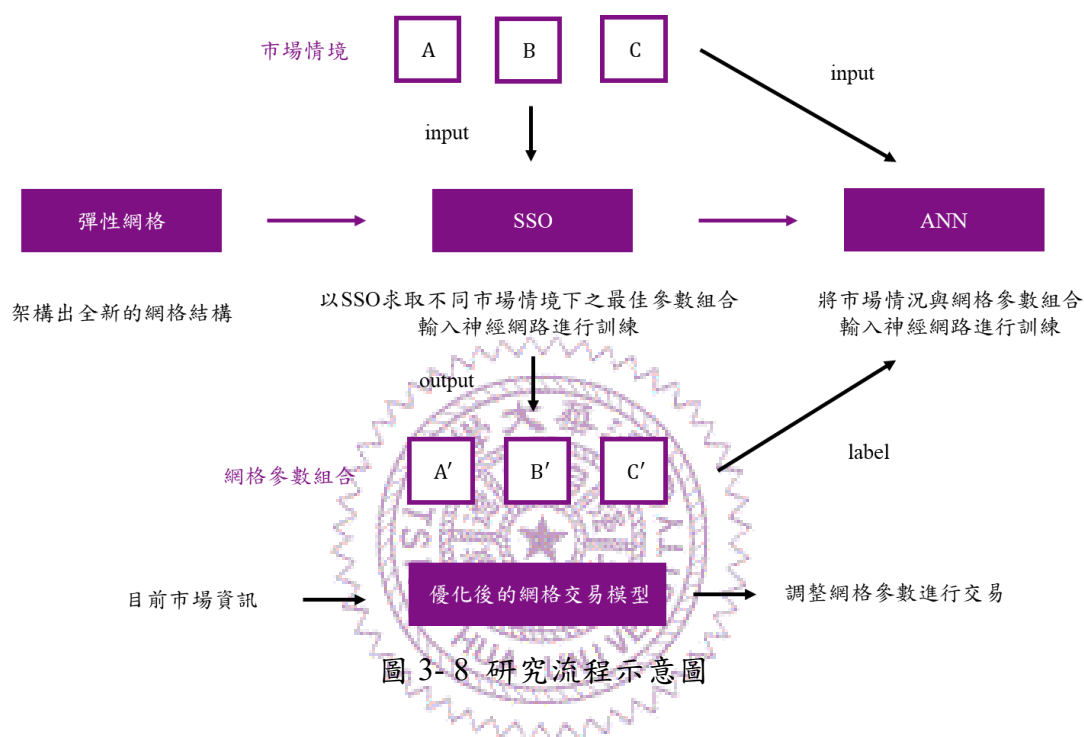
圖 3-7 彈性網格交易模型示意圖

彈性網格的設計過程中的核心概念為希望在高價時執行較頻繁的賣出動作，而在價格越低時越頻繁的買入較便宜的現貨。彈性網格於模型中間區段與等差網格有較為相似的架構，而模型下半部較近似於等比網格，另外模型上半部有別於目前常見的等比等差網格交易模型，在越高價格時網格密集度越高。此模型的建構主旨在於擷取等差與等比的優點，並且不管市場走勢為行盤震盪、持續上漲或下跌時，都能擁有優於目前現有模型的表現。

然要能真正發揮彈性網格模型架構的優勢，良好的參數設定與配合市場局勢的自我調整是決定模型成敗至關重要的決定性因素，故本研究後續會以簡化群體演算法決定彈性網格在不同情況下所適合的參數(過程會於 3.3 小節詳述)，並將計算出之參數組合與對應的市場情境輸入人工神經網路中進行模型訓練(過程會於 3.4 小節詳述)。最後研究會產出一訓練完成的深度學習模型，只需輸入當前市場情境，就能自動調整至最適合目前市場情況之網格參數。

3.3 以簡化群體演算法尋找最適參數

本小節會以前一小節建構出之彈性網格作為基礎，利用簡化群體演算法調整彈性網格在不同市場情況下的最適解，並在 3.4 節中將市場情況與最適參數作為訓練集輸入進人工神經網路進行模型訓練。流程可參考下方圖 3-8。



3.3.1 目標式及限制式

本研究最終所追求的主要目標為最大化投資報酬，故於簡化群體演算法中設市場投資商品價格經過 j 期更新後，最終目標式設置如下：

$$\max S_j \times P_j + C_j \quad \text{公式(3.13)}$$

其中 S_j 代表最後一期所持有現貨量， P_j 為最後一期時商品市場價格， C_j 為最後一期時之持有資金。而 S_0 與 C_0 經過 j 期價格輸入後，其更新過程可參考 3.1.2 小節。

另外彈性網格交易模型中，須保證每次交易獲利大於該次交易成本 $h\%$ (通常為交易手續費率)，故須設置以下限制式對， i 從 0 到 n ：

$$s.t. \ g_{i+1} - g_i > h\% \times g_{i+1} \quad \text{公式(3.14)}$$

其中 g_i 之計算可參考公式 3.9 及公式 3.10，代入後可看出此限制式實際是對網格參數 G_{ul} 、 G_{ll} 、 n_u 與 n_l 做出限制。

另一方面，我們還會對網格上下限 G_{ul} 、 G_{ll} 、 n_u 及 n_l 作出額外的限制， x 從 0 到 j ：

$$s.t. \text{ (實驗一)} \ P_0 \times 105\% < G_{ul} < P_0 \times 130\% \quad \text{公式(3.15)}$$

$$P_0 \times 70\% < G_{ll} < P_0 \times 95\% \quad \text{公式(3.16)}$$

$$s.t. \text{ (實驗二)} \ P_0 \times 105\% < G_{ul} < P_0 \times 150\% \quad \text{公式(3.15)}$$

$$P_0 \times 50\% < G_{ll} < P_0 \times 95\% \quad \text{公式(3.16)}$$

$$10 < n_l < [P_0 \times 100 / (\text{maximum } P_x \times 1.3)] - 10 \quad \text{公式(3.18)}$$

以上四個限制是為保有誤差空間，避免後續人工神經網路訓練過度擬合，造成價格容易超出網格上下限而失去套利機會，並且將參數控制在合理範圍。

3.3.2 解編碼方式

根據 3.3.1 小節所提到，一網格交易模型之建構須由總投入資金 F_0 、初始價格 P_0 、網格上界 G_{ul} 、網格下界 G_{ll} 及網格數 n 來設定，而在本研究中， F_0 及 P_0 為控制變數，故須利用 SSO 求解的參數分別為 G_{ul} 、 G_{ll} 及 n ，而在彈性網格中，網格數 n 有可分為上半網格數 n_u 與下半網格數 n_l 。

而其中解範圍可參考下表，其中 P_0 商品之起始價格。

使用範圍	參數	變數上界	變數下界
實驗一	網格上界 G_{ul}	$P_0 \times 130\%$	$P_0 \times 105\%$
	網格下界 G_{ll}	$P_0 \times 95\%$	$P_0 \times 70\%$
實驗二	網格上界 G_{ul}	$P_0 \times 150\%$	$P_0 \times 105\%$
	網格下界 G_{ll}	$P_0 \times 95\%$	$P_0 \times 50\%$
通用	上網格數 n_l	$[P_0 \times 100 / (\text{maximum } P_x \times 1.3)] - 10$	10
	下網格數 n_u	$[P_0 \times 100 / (\text{maximum } P_x \times 1.3)] - 10$	10

表 2 網格交易參數上下界

而在本研究編解碼時，將 x_1 設定為網格上界 G_{ul} ， x_2 設定為網格下界 G_{ll} ， x_3 設定為上網格數 n_u ， x_4 設定為下網格數 n_l 。

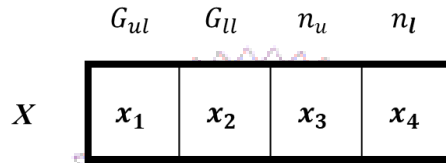


圖 3-9 編解碼結構

舉例來說，當 $X = (150, 100, 10, 30)$ 的時候，代表網格上界為 150，網格下界為 100，上網格數為 10，下網格數為 30，總網格數為 40。

3.3.3 更新機制參數設定與範圍

本研究使用 SSO 進行求解，其中 ρ_{ij}^t 為界於 0 到 1 之間的隨機亂數，而問題的解會根據此數進行更新：當 ρ_{ij}^t 界於 0 到 C_g 時， x_{ij}^{t+1} 會以 $gBest$ 作為解；當 ρ_{ij}^t 落在 C_g 與 C_p 之間時， x_{ij}^{t+1} 以 $pBest$ 作為解；而當 ρ_{ij}^t 界於 C_p 與 C_w 之間時， x_{ij}^{t+1} 將更新為 x_{ij}^t ，也就上一代解；而在 ρ_{ij}^t 界於 C_w 與 1 之間時，會隨機在上下界間產生一隨機新解。其更新機制可參考公式 2.4。

下表為 SSO 運行中所需用到的符號及定義整理：

符號	定義
N_{var}	變數的數量，在此研究中為網格上界 G_{ul} 、網格下界 G_{ll} 及網格數 n 。
N_{sol}	解的總數量。
N_{gen}	最大迭代數，亦即演算法終止條件，適當的設定值應在有限時間內使解能收斂於一適當值。
X_i^t	$X_i^t = (G_{uli}^t, G_{lli}^t, n_i^t)$ 表示第 t 次迭代中的第 i 個解，其中 $t = 1, 2, \dots, N_{gen}$ ， $i = 1, 2, \dots, N_{sol}$ 。
$\widehat{G}_{ul}^p, \widehat{G}_{ul}^g, \widehat{G}_{ll}^p, \widehat{G}_{ll}^g, \widehat{n}^p, \widehat{n}^g$	各變數在更新過程中的 $pBest$ 及 $gBest$ 。
C_g, C_p, C_w	於 SSO 中三個用以決定更新值的關鍵參數，可依不同情況做相對應的調整。
LB	$LB = (lb_1, lb_2, \dots, lb_{N_{var}})$ 為各變數的下界，即 $x_{ij}^t \geq lb_j$ 。
UB	$UB = (ub_1, ub_2, \dots, ub_{N_{var}})$ 為各變數的上界，即 $x_{ij}^t \leq ub_j$ 。

表 3 SSO 符號及定義

3.4 訓練人工神經網路自動調整彈性網格參數

利用 SSO 求取各種市場情況下之最佳網格配置後，再將市場情況與計算出之網格參數作為訓練人工神經網路訓練之題目與解答。其中一市場情況會以以下幾項數值進行詮釋，輸入神經網路之中，分別為：期間最高價、期間最低價、平均市場價格、平均交易量、價格變化(起始價格減去最終價格)、交易量變化(起始交易量減去最終交易量)、價格標準差、交易量標準差。

而在神經網路輸出端會輸出建構一網格交易模型所需之相關參數，其中包括：上網格數、下網格數、網格上界、網格下界。具體人工神經網路架構可參考下圖。

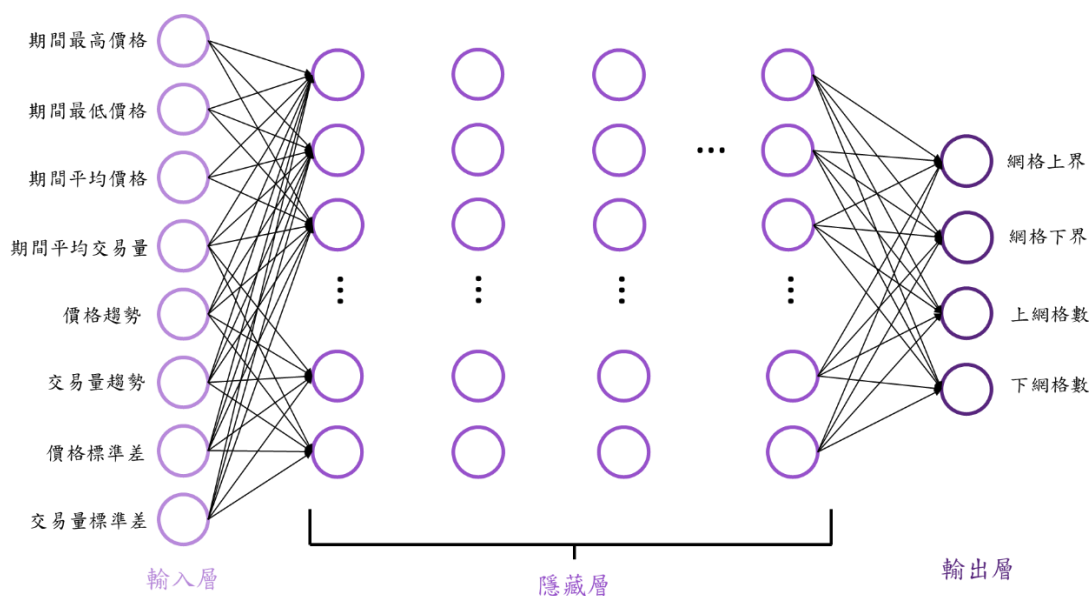


圖 3- 10 人工神經網路架構

在神經網路訓練過程裡，人工神經網路節點中之權重會透過損失函數計算出之誤差值不斷進行自我調適，直至收斂到一誤差最小之狀態即訓練完成。之後我們就能透過此經過訓練之神經網路，輸入近期市場狀態及走勢，生成最佳網格交易模型參數，自動建構一彈性網格模型進行市場交易活動。

而在本研究中，我們將使用兩種神經網路進行訓練，分別為全連接神經網路(Fully-connect Neural Network, FNN)及近年時常被用來進行時間序列預測之長短期記憶神經網路(Long Short-Term Memory, LSTM)，以觀察及比較何種神經網路在學習網格參數上有較佳的表現。

其中因為模型的架構不同，在全連接神經網路模型的資料處理上，會將過隨機的重新排列，使市場情況與對應參數的配對被以單一情況的方式進行訓練，而非視為時間序列問題。而在長短期記憶模型的資料上，基於模型為處理時間序列問題而生，在輸入資料時，資料就會以時間順序的方式進行輸入與訓練。在此處也可觀察兩種資料處理的方式在問題解決的成果上，何種比較出色。

第四章、研究結果

本研究用於驗證與比較的資料集為標準普爾 500 指數 (Standard & Poor's 500, S&P 500)、那斯達克綜合指數(NASDAQ Composite)、道瓊工業平均指數 (Dow Jones Industrial Average, DJIA)、歐洲 STOXX 50 指數(Euro Stoxx 50)、上海綜合指數(Shanghai Composite)，共計五個大盤指數從 2011 到 2022 年之資料。

4.1 以固定參數驗證彈性網格之表現

首先我們以相同的網格數及網格上下限作為基礎，將彈性網格的表現與等差網格、等比網格相互比較，結果可參考表 4-表 6。其中網格數是以式子 4.1 所計算而出，以貼近真實投資情況，並確保資金使用上的充足性。而依上下界條件又將每個資料集分為兩組進行實驗，第一組以初始價格的 1.3 倍作為網格上界，並以初始價格的 0.7 倍作為網格下限；而第二組則以初始價格的 1.5 倍作為網格上界，並以初始價格的 0.5 倍作為網格下限。

從表 4.中的結果可驗證在五個不同綜合指數十年間的波動下，以相同的網格參數進行交易，彈性網格報酬率、累積財富及夏普率上表現明顯最佳。整體而言，彈性網格因其交易結構適當地延緩了進場及出場時機，而獲得了相對較高的投資報酬率與夏普率。

網格種類	彈性	等差	等比
S&P 500			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	31.692 %	27.934 %	22.234 %
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	43.727 %	39.556 %	30.174 %
Nasdaq 100			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	53.795 %	49.261 %	40.369 %
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	71.888 %	66.024 %	51.180 %
Dow Jones Industrial Average			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	21.935 %	18.332 %	13.629 %
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	33.541 %	29.741 %	21.648 %
Euro Stoxx 50			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	0.601 %	-4.122 %	-7.284 %
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	17.499 %	12.520 %	7.087 %
Shanghai Composite			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	-33.904 %	-38.944 %	-38.534 %
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	-12.955 %	-19.936 %	-18.290 %

表 4 彈性網格與等差網格、等比網格之報酬率比較(固定參數版)

網格種類	彈性	等差	等比
S&P 500			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	13169	12793	12223
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	14373	13956	13017
Nasdaq 100			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	15380	14926	14037
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	17189	16602	15118
Dow Jones Industrial Average			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	12194	11833	11363
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	13354	12974	12165
Euro Stoxx 50			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	10060	9588	9272
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	11750	11252	10709
Shanghai Composite			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	6610	6106	6147
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	8705	8006	8171

表 5 彈性網格與等差網格、等比網格之累積財富比較(固定參數版)

網格種類	彈性	等差	等比
S&P 500			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	0.118	0.103	0.092
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	0.160	0.145	0.137
Nasdaq 100			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	0.187	0.172	0.161
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	0.234	0.219	0.215
Dow Jones Industrial Average			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	0.079	0.065	0.054
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	0.119	0.105	0.095
Euro Stoxx 50			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	0.002	-0.011	-0.021
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	0.046	0.033	0.023
Shanghai Composite			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	-0.082	-0.095	-0.106
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	-0.030	-0.047	-0.053

表 6 彈性網格與等差網格、等比網格之夏普率比較(固定參數版)

4.2 SSO 參數設定

接著我們使用了部分資料集去挑選適當的 SSO 參數以進行後續的所有實驗。

第一組實驗將 C_g, C_p, C_w 如表 2. 所配置，其意義是將隨機數 ρ 的範圍依 7:1:1:1 切分為四個部分，其中比例 7 的部分在實驗中輪流分配給 $gbest$ 、 $pbest$ 、 x_{ij}^{t+1} 及全新解，以檢測何種解對於產生品質較好的解有更加關鍵的影響力。

(C_g, C_p, C_w)	最大比例解	ROI (max)	ROI (min)
(0.7,0.8,0.9)	gbest	69.74%	66.83%
(0.1,0.8,0.9)	pbest	69.62%	67.18%
(0.1,0.2,0.9)	x_{ij}^{t+1}	69.69%	66.01%
(0.1,0.2,0.3)	new random	69.41%	66.69%

表 7 不同參數組合下最佳及最差解(實驗一)

由表 7 可知，gbest 最大時能產生出品質較好的解，同時又具有穩健性，故於最終參數配置上會將 gbest 的機率設定為最大。

在確定 gbest 會被設定為最大機率後，接著再將隨機數 ρ 的範圍依 5:3:1:1 切分為四個部分，其中因上一步驟已確定 gbest 為產生品質好的解的關鍵因素，所以將以 gbest 作為解的機率設為最大的 0.5，而 0.3 的機會會輪流分配給 pbest、 x_{ij}^{t+1} 及全新解，依實驗結果決定哪一種解為產生品質好之解的第二關鍵因素，結果可參考表 8。

由表 8 可知，新的隨機解較大時能產生出品質較好的解，於最差報酬上也可看出其表現具有穩健性，故於最終參數配置上會將新的隨機解的機率設定為次大。

(C_g, C_p, C_w)	最大比例解	ROI (max)	ROI (min)
(0.5,0.8,0.9)	pbest	69.77%	66.22%
(0.5,0.6,0.9)	x_{ij}^{t+1}	69.77%	66.91%
(0.5,0.6,0.7)	new random	69.86%	67.83%

表 8 不同參數組合下最佳及最差解(實驗二)

在確定新的隨機解會被設定為次大機率後，接著再將隨機數 ρ 的範圍依 3:3:3:1 切分為四個部分，其後步驟依前兩步驟類推，得出表 5，可看出 x_{ij}^{t+1} 及 pbest 對於解的品質差異性不大，故會將兩機率同設定為最小。

(C_g, C_p, C_w)	最大比例解	ROI (max)	ROI (min)
(0.3,0.6,0.7)	pbest	69.85%	67.99%
(0.3,0.4,0.7)	x_{ij}^{t+1}	69.90%	67.81%

表 9 不同參數組合下最佳及最差解(實驗三)

4.3 以 SSO 選取參數驗證彈性網格之表現

將 SSO 參數設置結束後，我們將彈性網格、等差網格及等比網格分別接上 SSO，以 10 run，每 run 20 個 generation，每個 generation 100 組 solution 的設定進行尋解。與前面彈性網格架構驗證一樣，我們同樣用不同風險程度的上下界進行了兩組實驗，其結果比較可參考表 10 – 表 12。

網格種類	彈性	等差	等比
S&P 500			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	82.859 %	66.384 %	58.394 %
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	90.269 %	77.198 %	60.774 %
Nasdaq 100			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	111.649 %	66.384 %	82.662 %
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	127.268 %	110.179 %	86.208 %
Dow Jones Industrial Average			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	74.509 %	60.591 %	51.893 %
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	80.559 %	70.063 %	54.760 %
Euro Stoxx 50			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	77.220 %	56.650 %	48.365 %
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	88.844 %	72.293 %	55.856 %
Shanghai Composite			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	58.389 %	39.178 %	33.363 %
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	80.954 %	58.934 %	43.639 %

表 10 彈性網格與等差網格、等比網格之報酬率比較(SSO 參數版)

由表 10 到表 12 結果可知在應用 SSO 求解網格交易參數上，彈性網格相較於等差及等比網格，在投資報酬率、累積財富與夏普率上仍為表現最佳的模型。另外將接上 SSO 後的網格交易結果與固定參數版本比較，可以看出報酬率很明顯的上升了許多。整體投資報酬率結果上，彈性網格最佳，等差次之，等比最差。

網格種類	彈性	等差	等比
S&P 500			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	18286	16638	15839
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	19027	17720	16077
Nasdaq 100			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	21165	16638	18266
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	22727	21018	18621
Dow Jones Industrial Average			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	17451	16059	15189
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	18056	17006	15476
Euro Stoxx 50			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	17722	15665	14837
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	18884	17229	15586
Shanghai Composite			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	15839	13918	13336
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	18095	15893	14364

表 11 彈性網格與等差網格、等比網格之累積財富比較(SSO 參數版)

網格種類	彈性	等差	等比
S&P 500			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	0.439	0.350	0.351
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	0.442	0.391	0.388
Nasdaq 100			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	0.498	0.350	0.437
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	0.524	0.464	0.462
Dow Jones Industrial Average			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	0.377	0.310	0.301
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	0.412	0.344	0.341
Euro Stoxx 50			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	0.300	0.219	0.211
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	0.331	0.265	0.257
Shanghai Composite			
$(G_{ul}, G_{ll}) = (P_0 \times 1.3, P_0 \times 0.7)$	0.189	0.134	0.131
$(G_{ul}, G_{ll}) = (P_0 \times 1.5, P_0 \times 0.5)$	0.220	0.179	0.172

表 12 彈性網格與等差網格、等比網格之夏普率比較(SSO 參數版)

4.4 訓練人工神經網路自動調整彈性網格參數

證實了彈性網格配合 SSO 搜尋交易參數的表現出色後，我們以彈性網格搭配 SSO 記錄下各個指數十年間的最佳交易參數，每 30 天求出一組交易參數。為擴增後續訓練及資料，我們將模型的移動步伐設定為 5，在各個指數中，分別得出約 500 筆參數資料進行人工神經網路的訓練及驗證，其中將前九年的資料作為訓練集，並將最後一年的資料作為驗證集。

而本研究訓練集之資料因具有時間上的連續性，為避免影響模型訓練結果，在全連接神經網路中，資料會經過重新隨機排序，再進行訓練。而在長短期記憶神經網路上，因其模型設計，其輸入訓練之資料須具備前後順序性，故就不將訓練資料做重新隨機排序的動作。

而全連接神經網路與長短期記憶神經網路架構經過多次實驗後，最終架構及超參數之設定如表 13 及表 14 所示。

其中輸入參數分別為：期間最高價、期間最低價、期間平均價、期交平均交易量、交易量前後差、價格前後差、價格標準差、交易量標準差，共八個變數來描述一個市場情境，並將輸出變數設為網格交易所用之網格上界、網格下界，上網格數及下網格數。

而其中隱藏層數設定為三層，優化器設為近年較常見、收斂速度快、尋解表現優的 adam，激勵函數使用 sigmoid 與 relu，損失函數則使用回歸問題所常用的 mean squared error。

項目	設定值
輸入變數數量	8
隱藏層數	3
輸出變數數量	4
隱藏層節點數	500
優化器	adam
激勵函數	sigmoid
損失函數	mean squared error
迭代數	300
批量大小	40

表 13 全連接神經網路超參數及架構相關設定

項目	設定值
輸入變數數量	8
隱藏層數	3
輸出變數數量	4
隱藏層節點數	256, 128, 64
優化器	adam
激勵函數	relu
損失函數	mean squared error
迭代數	300
批量大小	32

表 14 長短期記憶神經網路超參數及架構相關設定

經過以上設定之神經網路訓練後，其訓練及比較結果以表 15 及表 16 呈現。從表 15 均方根差(公式 4.1)結果來看，除了在 Nasdaq 100 指數中的網格上下限，LSTM 有較小的均方根誤差，其他的指數中的四個輸出變數之均方根差皆為 FNN 表現較佳。

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad \text{公式(4.1)}$$

神經網路種類	FNN	LSTM
S&P 500		
網格上界 G_{ul}	359.87	431.70
網格下界 G_{ll}	244.51	329.58
上網格數 n_l	14.39	15.87
下網格數 n_u	17.55	21.02
Nasdaq 100		
網格上界 G_{ul}	2173.88	719.85
網格下界 G_{ll}	1296.80	1078.14
上網格數 n_l	11.53	17.20
下網格數 n_u	15.68	23.83
Dow Jones Industrial Average		
網格上界 G_{ul}	2452.86	4944.37
網格下界 G_{ll}	1418.22	4426.61
上網格數 n_l	12.81	15.39
下網格數 n_u	17.86	20.05
Euro Stoxx 50		
網格上界 G_{ul}	324.07	578.32
網格下界 G_{ll}	184.36	326.22
上網格數 n_l	11.49	14.21
下網格數 n_u	14.11	17.03
Shanghai Composite		
網格上界 G_{ul}	255.34	304.74
網格下界 G_{ll}	183.31	161.83
上網格數 n_l	9.22	16.73
下網格數 n_u	14.63	21.42

表 15 全連接神經網路與長短期記憶模型之均方誤差比較

而從表 16 兩模型的判定係數(R squared)，一種衡量回歸模型適配性的指標，也可解讀成模型的解釋程度，計算可參考公式 4.2。

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = \frac{\sum_{i=1} (y_i - f_i)^2}{\sum_{i=1} (y_i - \bar{y})^2} \quad \text{公式(4.2)}$$

在表 16 中可發現除了 Nasdaq 100 指數外，其餘大部分時候皆為 FNN 對於四個輸出變數有最佳的適配性。

神經網路種類	FNN	LSTM
S&P 500		
網格上界 G_{ul}	97.410%	90.490%
網格下界 G_{ll}	99.586%	97.240%
上網格數 n_l	94.972%	99.469%
下網格數 n_u	99.392%	94.892%
Nasdaq 100		
網格上界 G_{ul}	96.415%	98.075%
網格下界 G_{ll}	97.088%	99.915%
上網格數 n_l	94.942%	99.635%
下網格數 n_u	94.737%	98.019%
Dow Jones Industrial Average		
網格上界 G_{ul}	97.704%	95.970%
網格下界 G_{ll}	99.771%	93.429%
上網格數 n_l	99.855%	93.685%
下網格數 n_u	99.559%	91.285%
Euro Stoxx 50		
網格上界 G_{ul}	96.183%	94.392%
網格下界 G_{ll}	99.324%	98.063%
上網格數 n_l	99.297%	96.858%
下網格數 n_u	98.405%	99.760%
Shanghai Composite		
網格上界 G_{ul}	97.157%	96.577%
網格下界 G_{ll}	96.749%	95.545%
上網格數 n_l	93.165%	93.881%
下網格數 n_u	99.661%	99.003%

表 16 全連接神經網路與長短期記憶模型之判定係數比較

以上結果原因可能為 LSTM 較擅長處理連續型之時間序列問題，但在股票市場的變化上，價格浮動性大，造成浮動的原因複雜，其中包括經濟政策的改變、國際情勢、戰爭、疫情等等，是以使用遞迴神經網路處理金融相關問題時，常會發生過擬合之問題，loss 值在訓練集低，而在測試集時高，造成模型泛化度低的現象發生。

而結構相對簡單的 FNN，在訓練集的部分我們將資料進行了隨機排序，隱藏資料的時間順序，以期能提升模型的泛化性，解決過擬合問題，就結果論，在最佳化網格參數時，這樣的做法確實獲得了較佳的表現。

以下我們會用報酬率(return of investment, ROI)、最大回落(maximum drawdown, MDD)、波動性(volatility)及夏普率(Sharpe ratio)四個數值來檢視本研究之模型表現。而在方法上會有最初買入且最後賣出(Buy and Sell, B&S)、最初賣出且最後買入(Sell and Buy, S&B)、網格交易系統機器人(GTSbot) [10]、一目均衡圖(Ichimoku Kinkohyo, IK) [12]、以 FNN 訓練之彈性網格(Flexible Grid with Fully-Connected Neural Network, FG-FNN)、以 LSTM 訓練之彈性網格(Flexible Grid with Long short-term memory, FG-LSTM)、等差網格、等比網格、彈性網格，共 9 種方法進行比較。

首先投資報酬率之公式可參考公式 4.3，投資報酬率為投資學中最常見之量化投資績效之指標，計算投資所得收益與成本間的比，通常以年化報酬獲總報酬之方式呈現。其中已將交易成本，包含交易手續費，納入計算。結果可參考表 17。

$$ROI = (\text{淨利潤} - \text{投資成本}) \times 100\% \quad \text{公式(4.2)}$$

	S&P	Nasdaq	DJI	Euro Stoxx	Shanghai
B&S	14.392%	1.519%	9.880%	28.340%	-15.581%
S&B	-14.392%	-1.519%	-9.880%	-28.340%	15.581%
GTSbot	6.408%	4.680%	6.594%	1.191%	-0.466%
IWOC	24.111%	23.171%	7.143%	32.087%	-18.758%
FG-FNN	11.520%	11.733%	8.849%	12.977%	-3.125%
FG-LSTM	4.639%	1.823%	2.940%	7.734%	-9.283%
等差	5.194%	-2.972%	3.480%	10.748%	-4.495%
等比	4.536%	-3.625%	3.276%	10.764%	-4.892%
彈性	5.589%	-1.895%	4.270%	11.477%	-4.309%

表 17 投資報酬率之比較

在報酬率上，以 FNN 彈性網格在每個指數中，皆能達到前三高的收益，尤其可看出在價格下跌的趨勢下，FNN 彈性網格最具有控制整體損失的能力。

第二個要比較的是最大回落，也就是在所有期間，收益下跌最劇烈的期間所下跌的收益量，屬於評估投資風險的指標之一，結果可參考表 18。

	S&P	Nasdaq	DJI	Euro Stoxx	Shanghai
B&S	7.960%	12.570%	7.290%	4.029%	8.529%
S&B	7.491%	10.370%	6.580%	7.313%	3.103%
GTSbot	0.524%	5.335%	1.000%	0.510%	1.126%
IWOC	10.345%	12.421%	8.606%	5.467%	7.640%
FG-FNN	6.455%	5.596%	7.491%	2.034%	4.105%
FG-LSTM	1.580%	49.031%	4.809%	2.966%	3.551%
等差	5.252%	8.594%	4.078%	1.513%	4.074%
等比	5.243%	10.096%	4.057%	1.529%	4.059%
彈性	4.737%	8.780%	4.163%	4.144%	4.208%

表 18 最大回落之比較

在最大回落方面，可看出投資報酬率越大的方法，最大回落也相對較高，驗證了在投資學中，高風險高報酬的理論。FNN 彈性網格在最大回落上之表現其原因有二：在投資報酬率相對較高的情況下，風險相對也會較高，再加上每期參數皆為神經網路所推算出，若在其中一期所預測出之參數特別差，很有可能造成單一期上的較大回落。是否能將此最大回落狀況視為離群值的單一狀況，還需用波動性來驗證。

在量化交易上，會透過其波動性去評估風險及模型穩定性，其計算方式可參考公式 4.3，結果可參考表 19。

$$Vol = \sigma [return] \quad \text{公式(4.3)}$$

表中可看到 FNN 彈性網格波動性的表現在各個指數上都是前三小的，也就是說相對於其他方法，FNN 彈性網格的表現十分平穩，也可推測出在最大回落上的

表現不佳，可能只是單一事件，在大多數的情況下，可是其為一種表現穩健的投資模型。其中 LSTM 彈性網格在標普與那斯達克兩個資料集中，表現為所有模型中最佳的，為一風險較小，較為穩健的投資模型。

	S&P	Nasdaq	DJI	Euro Stoxx	Shanghai
B&S	0.06126	0.07671	0.03568	0.06221	0.01840
S&B	0.06126	0.07671	0.03568	0.06221	0.01840
GTSbot	0.01710	0.03589	0.01795	0.00267	0.00190
IWOC	0.07689	0.08398	0.04564	0.07337	0.03568
FG-FNN	0.01687	0.02076	0.02149	0.01730	0.01414
FG-LSTM	0.01672	0.01637	0.02203	0.01679	0.02030
等差	0.02450	0.03579	0.02129	0.01786	0.01590
等比	0.02473	0.03629	0.02155	0.01828	0.01629
彈性	0.02503	0.03665	0.02162	0.01862	0.01624

表 19 波動性之比較

最後看到夏普率的部分，其為一種計算投資報酬率與風險之間比率的模型績效指標，也可解釋為每承受一單位風險下，可換得之報酬，其計算方法可參考公式 4.4，結果可參考表 20。

$$Sharpe = \frac{return}{\sigma [return]} \quad \text{公式(4.3)}$$

	S&P	Nasdaq	DJI	Euro Stoxx	Shanghai
B&S	2.349	0.198	2.769	4.556	-8.466
S&B	-2.349	-0.198	-2.769	-4.556	8.466
GTSbot	3.748	1.304	3.673	4.466	-2.446
IWOC	3.136	2.759	1.565	4.373	-5.258
FG-FNN	6.828	5.651	4.119	7.499	-2.210
FG-LSTM	2.774	1.114	1.335	4.606	-4.573
等差	2.120	-0.830	1.634	6.017	-2.827
等比	1.834	-0.999	1.520	5.887	-3.004
彈性	2.233	-0.517	1.975	6.165	-2.653

表 20 夏普率之比較

透過表 20 可看出，FNN 彈性網格在四個指數上，夏普率的表現是所有方法中最佳的；而在下跌的市場情況下，除了放空之外，其夏普率也是最大的。由此可知，每承受一單位的風險下，FNN 彈性網格所換得之報酬為最高的。



第五章、結論

本研究除了提出一種全新的網格交易架構，有效的提升原有模型的投資報酬率，改善了過早進場及出場的弊病，並利用了簡化群體演算法及人工神經網路，協助網格交易的參數選擇，並賦予模型適應市場的能力。

在模型表現上，我們以五大市場指數作為驗證資料，其涵蓋範圍包含美國、歐洲及中國。FNN 彈性網格在夏普率上表現十分優異，在擁有出色的投資報酬率下，同時具有模型穩健性，能適當的控制風險與報酬之間的平衡。

以下整理了幾點未來相關研究可延伸及改善的方向：

1. 可分析及研究在各種市場情況下(例如價格持續上升、價格持續下降、價格震盪)所適用之網格模型為何，並透過判斷市場情勢，選擇適當的模型進行交易，以期能獲得更好的模型表現。
2. 在最大回落上，目前模型表現上有進步的空間，也就是說目前模型在少數特殊情況下會有特別差的表現發生，未來可研究如何避免此種情況發生。
3. 目前模型為了控制風險，將單邊網格數控制在最少 10 格，未來若期望擁有更高的報酬率，可嘗試調整最少網格數。
4. 近期在金融預測上，生成對抗網路(Generative Adversarial Network, GAN)儼然有逐漸超越長短期記憶模型的趨勢，未來可考慮使用 GAN 來改良長短期記憶模型目前的不足。

我們證明了在量化交易上，結合人工智慧的技術，能為原有的交易模型帶來突破，並能適應各種瞬息萬變的市場情勢。期待未來能有更多的相關延伸研究，能讓交易模型在做參數調整時，更準確的貼近人類的決策，甚至能更優於人類的判斷，屏除人性上的衝動與市場情緒，理性的進行市場交易決策，以獲取更好的投資效益。

參考文獻

1. IOSCO, J., *Regulatory Issues Raised by the Impact of Technological Changes on Market Integrity and Efficiency*. Consultation Report, 2011.
2. Allen, F., J. McAndrews, and P. Strahan, *E-finance: an introduction*. Journal of financial services research, 2002. **22**(1): p. 5-27.
3. Shahrokhi, M., *E-finance: status, innovations, resources and future challenges*. Managerial Finance, 2008.
4. Turing, A.M., *Computing machinery and intelligence*, in *Parsing the turing test*. 2009, Springer. p. 23-65.
5. Hutson, M., *How researchers are teaching AI to learn like a child*. Science, 2018. **10**.
6. Minsky, M. and S.A. Papert, *Perceptrons: An introduction to computational geometry*. 2017: MIT press.
7. Silver, D., et al., *Mastering the game of Go with deep neural networks and tree search*. nature, 2016. **529**(7587): p. 484-489.
8. Aldridge, I., *High-frequency trading: a practical guide to algorithmic strategies and trading systems*. Vol. 604. 2013: John Wiley & Sons.
9. Lai, Z.-R., et al., *Reweighted price relative tracking system for automatic portfolio optimization*. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018. **50**(11): p. 4349-4361.
10. Rundo, F., et al., *Grid trading system robot (gtsbot): A novel mathematical algorithm for trading fx market*. Applied Sciences, 2019. **9**(9): p. 1796.
11. Wen, Q., et al., *Automatic stock decision support system based on box theory and SVM algorithm*. Expert systems with Applications, 2010. **37**(2): p. 1015-1022.
12. Deng, S. and A. Sakurai. *Short-term foreign exchange rate trading based on the support/resistance level of Ichimoku Kinkohyo*. in *2014 International Conference on Information Science, Electronics and Electrical Engineering*. 2014. IEEE.
13. Ye, A., et al. *Developing sustainable trading strategies using directional changes with high frequency data*. in *2017 IEEE International Conference on Big Data (Big Data)*. 2017. IEEE.
14. Krauss, C., *Statistical arbitrage pairs trading strategies: Review and outlook*. Journal of Economic Surveys, 2017. **31**(2): p. 513-545.

15. Poterba, J.M. and L.H. Summers, *Mean reversion in stock prices: Evidence and implications*. Journal of financial economics, 1988. **22**(1): p. 27-59.
16. Hurst, B., Y.H. Ooi, and L.H. Pedersen, *A century of evidence on trend-following investing*. The Journal of Portfolio Management, 2017. **44**(1): p. 15-29.
17. Kampouridis, M. and F.E. Otero, *Evolving trading strategies using directional changes*. Expert Systems with Applications, 2017. **73**: p. 145-160.
18. MacKinlay, A.C., *Event studies in economics and finance*. Journal of economic literature, 1997. **35**(1): p. 13-39.
19. Hautsch, N. and R. Huang, *On the dark side of the market: identifying and analyzing hidden order placements*. Available at SSRN 2004231, 2012.
20. Pardo, A. and R. Pascual, *On the hidden side of liquidity*. The European Journal of Finance, 2012. **18**(10): p. 949-967.
21. DuPoly, A. *The Expert4x, No stop, Hedged, Grid Trading System and The Hedged, Multi-Currency, Forex Trading System*. 2008
22. Jegadeesh, N., *Evidence of predictable behavior of security returns*. The Journal of finance, 1990. **45**(3): p. 881-898.
23. Jegadeesh, N., *Seasonality in stock price mean reversion: Evidence from the US and the UK*. The Journal of Finance, 1991. **46**(4): p. 1427-1444.
24. Li, B., et al., *PAMR: Passive aggressive mean reversion strategy for portfolio selection*. Machine learning, 2012. **87**(2): p. 221-258.
25. Kennedy, J. and R. Eberhart. *Particle swarm optimization*. in *Proceedings of ICNN'95-international conference on neural networks*. 1995. IEEE.
26. Yeh, W.-C., *A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems*. Expert Systems with Applications, 2009. **36**(5): p. 9192-9200.
27. Yeh, W.-C., *Novel swarm optimization for mining classification rules on thyroid gland data*. Information Sciences, 2012. **197**: p. 65-76.
28. Yeh, W.-C., W.-W. Chang, and Y.Y. Chung, *A new hybrid approach for mining breast cancer pattern using discrete particle swarm optimization and statistical method*. Expert Systems with Applications, 2009. **36**(4): p. 8204-8211.
29. Huang, C.-L., *A particle-based simplified swarm optimization algorithm for reliability redundancy allocation problems*. Reliability Engineering & System Safety, 2015. **142**: p. 221-230.
30. Yeh, W.-C., *Orthogonal simplified swarm optimization for the series-parallel redundancy allocation problem with a mix of components*. Knowledge-Based Systems, 2014. **64**: p. 1-12.

31. Yeh, W.-C., *A novel boundary swarm optimization method for reliability redundancy allocation problems*. Reliability Engineering & System Safety, 2019. **192**: p. 106060.
32. Huang, W., et al., *Neural networks in finance and economics forecasting*. International Journal of Information Technology & Decision Making, 2007. **6**(01): p. 113-140.
33. Qi, S., et al., *The exploration of internet finance by using neural network*. Journal of Computational and Applied Mathematics, 2020. **369**: p. 112630.
34. Siامي-ناميني, S. and A.S. Namin, *Forecasting economics and financial time series: ARIMA vs. LSTM*. arXiv preprint arXiv:1803.06386, 2018.
35. Cao, J., Z. Li, and J. Li, *Financial time series forecasting model based on CEEMDAN and LSTM*. Physica A: Statistical Mechanics and its Applications, 2019. **519**: p. 127-139.
36. Russell, S. and P. Norvig, *Artificial intelligence: a modern approach*. 2002.
37. Deng, L. and D. Yu, *Deep learning: methods and applications*. Foundations and trends in signal processing, 2014. **7**(3-4): p. 197-387.
38. McCulloch, W.S. and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*. The bulletin of mathematical biophysics, 1943. **5**(4): p. 115-133.
39. Rosenblatt, F., *The perceptron: a probabilistic model for information storage and organization in the brain*. Psychological review, 1958. **65**(6): p. 386.
40. Werbos, P., *Beyond regression: "new tools for prediction and analysis in the behavioral sciences"*. Ph. D. dissertation, Harvard University, 1974.
41. Wang, S.-C., *Artificial neural network*, in *Interdisciplinary computing in java programming*. 2003, Springer. p. 81-100.
42. Leshno, M., et al., *Multilayer feedforward networks with a nonpolynomial activation function can approximate any function*. Neural networks, 1993. **6**(6): p. 861-867.
43. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors*. nature, 1986. **323**(6088): p. 533-536.
44. Ruder, S., *An overview of gradient descent optimization algorithms*. arXiv preprint arXiv:1609.04747, 2016.
45. Medsker, L.R. and L. Jain, *Recurrent neural networks*. Design and Applications, 2001. **5**: p. 64-67.
46. Graves, A., *Long short-term memory*. Supervised sequence labelling with recurrent neural networks, 2012: p. 37-45.
47. Dunis, C.L. and A. Nathani, *Quantitative trading of gold and silver using nonlinear models*. Neural Network World, 2007. **17**(2): p. 93.

48. Bloch, D.A., *Recipe for quantitative trading with machine learning*. Available at SSRN 3232143, 2018.
49. Ta, V.-D., C.-M. Liu, and D.A. Tadesse, *Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading*. Applied Sciences, 2020. **10**(2): p. 437.
50. Jia, W., et al. *Quantitative trading on stock market based on deep reinforcement learning*. in *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019. IEEE.
51. Liu, Y., et al. *Adaptive quantitative trading: An imitative deep reinforcement learning approach*. in *Proceedings of the AAAI conference on artificial intelligence*. 2020.

