

國立屏東大學

資訊科學系

碩士論文

結合頻譜轉換的啟發式卷積神經網路



Heuristic Convolutional Neural Networks with
Spectral Transform

研究生：李重瑩

指導教授：施釗德 博士

中華民國一百一十一年一月

國立屏東大學資訊科學系

碩 士 論 文

結合頻譜轉換的啟發式卷積神經網路

本論文經審查及口試合格特此證明

論文口試委員會主席

林家樑

委員

王朱福

委員

施劍河

指導教授

施劍德

系主任

蔡進聰

研究生

李重雲

撰

中 華 民 國 111 年 01 月 14 日

誌謝

研究所，在很小的時候夢想的地方，因為有著充沛的實驗資源跟空間，還兼併工作的性質，在當時對於缺乏的我常在追求部分實驗知識上感到被阻礙，卻隨著學歷的長河，早早就認清了現實，若非家裡允許我讀個輔系延一年，那數度低空飛過的驚悚小說依舊恍如隔日、歷歷在目。而後在大學末期，逢林家樑教授的鼓勵下秉持發彩光的機會卡牌（SSR 的誼），一舉將夢想變理想，開始敢於去嘗試拚搏從未妄想的另一條升學管道。

感謝指導教授施釗德教授，在碩士生涯的期間看著我在數個主題方向之間衝浪起伏；在多個題目之間甩尾過彎，即便如此不安定的情況下依舊給予我充足的時間探索，並解答我在研究方面諸多細節上的疑問。也感謝王朱福教授在碩士期間多次的提醒和叮嚀，讓我在長期的時程中維持方向不迷失。最後感謝黃奕欽助理教授曾在我研究內容疑義上提供關鍵解答與研究建議，讓我在研究主題中能定下船錨。最後，感謝吳俊廷、連士謙、鄭宇軒、朱家聖和王聖堯在學期間莫名激盪出的奇葩技能樹和奇幻遭遇，如毒梟般散播迷因散播愛，這種曾經是學弟、升學後變同學的感覺真神奇。

最後，感謝縣溪彼岸等候的父母，在就學貸款林立的升學途徑上，與您達成的協議我完成了，謝謝你們！讓我能有稍微不一樣的序曲踏上業界，這樣就足矣，非常謝謝你們！

2021.11

摘要

本論文嘗試採用爬山演算法自動構築卷積神經網路，並應用於手寫數字辨識。我們應用離散傅立葉轉換和離散餘弦轉換的神經網路層提升準確度。具體而言，本文在爬山演算法自動構築卷積神經網路的基礎下結合頻譜轉換層。結果發現在三個 2D 資料集準確度平均值相較於不使用頻譜轉換層提升 3.28 %、3.6 % 和 8.57 %。證實以爬山演算法自動構築的卷積神經網路，在結合頻譜轉換層後，能夠有效的提升手寫數字辨識的準確度。

關鍵詞：頻譜、卷積神經網路、爬山演算法

Abstract

In this thesis, we use the Hill-Climbing Algorithm to build Convolutional Neural Networks automatically and apply them to the handwritten digit recognition problem. Network layers based on Discrete Fourier Transform and Discrete Cosine Transform are used to improve accuracy. By adding spectral transform layers to the convolutional neural network, simulation results show that the accuracy increases by a mean of 3.28 %, 3.6 % and 8.57 % across 3 different 2D datasets. Use the Hill-Climbing Algorithm to build Convolutional Neural Network with spectral transform layers can indeed enhance the accuracy on the handwritten digit recognition problem.

Keyword : Spectrum, Convolutional Neural Network, Hill-Climbing algorithm

目錄

誌謝.....	i
摘要.....	ii
Abstract	iii
目錄.....	iv
圖目錄.....	vi
表目錄.....	viii
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究動機與目的.....	2
1.3 論文架構.....	2
第二章 文獻探討.....	3
2.1 類神經網路 (Artificial Neural Network).....	3
2.2 卷積神經網路.....	9
2.3 神經網路架構搜尋 (Neural Architecture Search).....	13
2.4 啟發式演算法 (Heuristic Algorithm).....	16
2.5 傅立葉轉換 (Fourier Transform).....	18
第三章 研究方法.....	23
3.1 研究架構.....	23
3.2 網路層類別.....	24
3.3 爬山演算法.....	25
3.4 頻譜轉換層.....	25
第四章 研究結果和分析.....	26

4.1 資料集簡介.....	26
4.2 MNIST	27
4.3 QMNIST	30
4.4 KMNIST	32
4.5 Fashion-MNIST	36
4.6 本章總結.....	39
第五章 研究結論.....	40
5.1 結論.....	40
5.2 未來研究建議.....	41
參考文獻.....	42

圖目錄

圖 1	神經元結構示意圖	3
圖 2	神經元架構圖	4
圖 3	激勵函數 ReLU (左)、Sigmoid (中) 和 Tanh (右)	6
圖 4	多層感知器架構圖	7
圖 5	研究架構圖[3]	8
圖 6	卷積神經網路示意圖	9
圖 7	卷積流程圖	10
圖 8	池化流程圖	11
圖 9	COVID-19 深度學習檢測流程圖[6]	12
圖 10	神經網路搜尋架構圖[9]	13
圖 11	神經網路搜尋架構圖[10]	13
圖 12	神經網路搜尋基礎架構圖	14
圖 13	ENAS 基礎架構圖[11]	15
圖 14	神經網路搜尋示意圖[12]	15
圖 15	實驗流程圖	16
圖 16	波型圖	18
圖 17	頻譜強度示意圖	19
圖 18	複數平面圖	20
圖 19	多頻率分量疊加的複數平面圖	21
圖 20	圖像轉頻譜模擬圖	22
圖 21	轉頻譜並濾波後的圖像[15]	22
圖 22	分組示意圖	23

圖 23	實驗流程圖	23
圖 24	爬山演算法流程圖	26
圖 25	MNIST 各組準確度盒狀圖	27
圖 26	MNIST 最佳準確度模型成長折線圖	29
圖 27	QMNIST 各組準確度盒狀圖	30
圖 28	QMNIST 最佳模型成長折線圖	32
圖 29	KMNIST 各組準確度盒狀圖	33
圖 30	KMNIST 最高準確度模型成長折線圖	35
圖 31	KMNIST 最佳模型成長折線圖	36
圖 32	Fashion-MNIST 各組準確度盒狀圖	36
圖 33	Fashion-MNIST 最佳模型成長折線圖	38

表目錄

表 1	爬山演算法.....	17
表 2	基因演算法.....	18
表 3	資料集範例表.....	26
表 4	MNIST 各組綜合統計結果	28
表 5	MNIST 各組參數數量統計	28
表 6	MNIST 最佳模型構建過程	29
表 7	QMNIST 各組綜合統計結果	31
表 8	QMNIST 各組參數數量統計	31
表 9	QMNIST 最佳模型構建過程	32
表 10	KMNIST 各組綜合統計結果	34
表 11	KMNIST 各組參數數量統計.....	34
表 12	KMNIST 最高準確度構建過程	34
表 13	KMNIST 最佳模型構建過程	35
表 14	Fashion-MNIST 各組綜合統計結果	37
表 15	Fashion-MNIST 各組參數數量統計	37
表 16	Fashion-MNIST 最高準確度構建過程	37

第一章 緒論

1.1 研究背景

自從神經網路範疇再度掀起滔天巨浪的發展以來，神經網路相關的論文數量呈現驚人的成長，也隨著硬體技術飛躍式的變革，各式各樣的新型神經網路如雨後春筍般冒出，同期間流行的加密貨幣技術亦相當依賴硬體設備，為各國的經濟系統開出一條雙向的流通管道，硬體的計算速度更甚於過去，這些依賴龐大計算量的理論得以實踐，各文獻研究分別從各種角度下手，有的追求鶴立雞群，帶來引頸期盼的突破，有的在標新立異間競逐，予以問題不同的選擇方向。

神經網路執行判斷時亦符合直覺理解，將所見比喻為輸入，而神經網路會將最高概率的判斷結果作為輸出。而訓練後的神經網路模型，保留整個神經網路模型或參數，考慮硬體設備之間計算能力不要過於懸殊，具有方便轉移、不必重新訓練的優點，打破部分問題依舊受限於暴力破解法的侷限。

至今，三大神經網路仍然在各處為應用問題上提供解決方案的基礎：

- 卷積神經網路 (CNN, Convolutional Neural Network)。
- 遞迴神經網路 (RNN, Recurrent Neural Network)。
- 強化學習 (Reinforcement Learning)。

其中，卷積神經網路 (CNN, Convolutional Neural Network) 以擅長處理影像辨識、視訊分析、自然語言處理等等，在辨別圖像上有強大的能力，計算成本上也小於另外兩種神經網路，以及容易概觀其運作過程等特色，如同廣為人知的 AlphaGo，即在圍棋領域頗負盛名。AlexNet、LeNet-5、ResNets、VGG (Visual Geometry Group) 也都是以此為主要所建立的深度學習模型。

1.2 研究動機與目的

在諸多手動設計的神經網路架構中，成績上都有不錯的突破，也進入了自動化產生神經網路的趨勢，企圖藉由電腦協助找到最佳的神經網路架構，但是因為探索眾多元素排列組合所形成的龐大計算成本太過可觀，就算是份量較小的神經網路，對於計算能力較弱的設備，依舊是只能打退堂鼓，時間過於漫長的缺點也不利於普及，故需要一些方法緩解壓力。在數種調整方案中，有一類透過長期觀察自然界變化過程而得出的啟發式演算法，能在前提下先進行相關附加條件限定，之後依此演算法多次探索神經網路架構的演化方向，一次次趨近最佳化模型，又有多項屬性能夠客製化，減少數次嘗試方案間所浪費的時間，也兼具原先運作設備的效能考量。如爬山演算法即屬於此例，且理解與實作上相對容易，具相當的彈性，適合作為初步嘗試階段的作法。

考慮以神經網路作圖像辨識，其中部分資料集的圖像均來自照片，而照片是由畫面中多個光線的投影所組成的，因此當聲波可以由一維的波形圖表達(時域)，又能夠由傅立葉分析轉換得到零散的多個頻率分布圖(頻域)和相位圖，相較於數種為了設計出能凸顯圖像特徵的網路層，嘗試引進一些因多樣結合或者延伸改良的待驗證理論，傅立葉分析的相關理論已經行之有年，並且廣泛應用於各項領域，驗證上具相當的共識。話說回來，光線亦具有聲波的特性，並以此推廣至二維的照片，也能找到相應的頻率分布圖和相位圖，足以成為顯示該圖像特徵的另一種管道。因此若將頻域圖作為網路層加入神經網路中，相信有助於神經網路學習。

本研究以啟發式演算法裡的爬山演算法作篩選策略，並以卷積神經網路常見的各種網路層組合外加頻譜轉換層組成搜尋空間，綜合隨機搜尋做搜尋策略，試圖尋找二維圖像辨識的近似最佳解模型。

1.3 論文架構

本篇論文架構如下：第一章陳述研究背景、動機和目的。第二章文獻探討，檢視相關作品、研究和文獻的介紹與發展。第三章研究方法講述實驗的架構和流程。第四章展示實驗結果並且分析實驗後的數據。第五章結論與未來展望，針對前述的研究目標做出回應。

第二章 文獻探討

本章以關於類神經網路、卷積神經網路、神經網路架構搜尋、啟發式演算法、傳立葉轉換的相關文獻作探討。

2.1 類神經網路 (Artificial Neural Network)

起初在鑽研生物神經網路模型 (Biological Neural Network) 時，專家們解析出人類大腦是由諸多神經元 (Neuron) 所構築，因而得以展現大腦強大的平行計算能力以及分類訊息的能耐。如圖 1，神經元結構區分為：

- 細胞體 (Cell Body)：維持細胞本身的運作，如同一個小單位的系統。
- 樹突 (Dendrites)：接收來自其他神經元軸突的訊號。
- 軸突 (Axon)：送出神經元訊號。

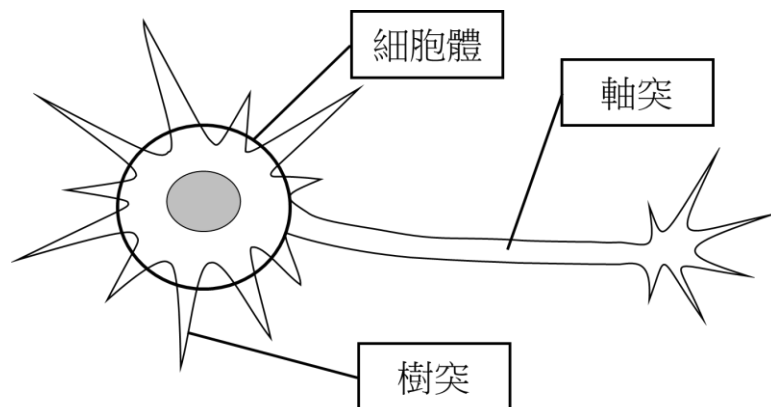


圖 1 神經元結構示意圖

神經元之間會與彼此或動器 (Effector) 之間以直接或間接的形式連結，組成所謂的神經網路。當接收到來自外在環境的刺激，該感官或受器就會傳送神經訊號至大腦，並由大腦決策後再送出訊號至指定位置部位顯現出對於該刺激的反應，像是有所學習般能對目標分辨或分類。至此，人們便試圖在電腦上實作這樣的概念，參照神經元的架構和原理，將單一神經元的運作模式以函式的形式呈現，做出類神經網路。把輸入喻為來自外在環境的刺激，如思考般對經驗中印象深刻的訊息有所觸發，經過一套固定的加權流程，綜合多個來自相異方面的輸入做出判斷，最終輸出分析的結果。

相較於線性的迴歸分析就可以發現，類神經網路能藉由學習多個樣本，以非線性的形式趨近和實際資料的關係。這樣的特色是即便在訓練前完全都不知道資料與答案的關聯，以此做為分析資料的方法亦能有所應用，有時還能將神經網路歷次訓練結果視覺化，方便觀察訓練過程的模型各層轉變，利於探討訓練過程中的各種情形所對應的圖形變化，既省去預先猜測數種模型適用性評估，還能隨著屢次更新的樣本特徵做出學習校正，無論是彈性上或是精準度上，類神經網路皆能成為更佳的选择。

最早的類神經網路謂為感知器 (Perceptron)，似下圖 2，由 Frank Rosenblatt 於 1957 年發明，是一種二元線性分類器，被視為最簡單最小形式的前饋神經網路 (Feedforward Neural Network)，僅一個加總計算和激勵函數組成一層並輸出結果，屬於單一神經元，而後 Marvin Minsky 和 Seymour Papert 分析感知器功能及局限時察覺，雖然當時的感知器能做到 AND、OR 和 NOT 的邏輯閘功能，但卻獨漏 XOR 功能。直至 Rumelhart 等人提出反向傳播 (Back-Propagation)[1]，才注意到多層感知機能解決單一神經元缺乏的 XOR 功能，研究熱潮再度如沸水般滿溢，並且得到廣泛的應用，比如機器學習 (Machine Learning) 的支援向量機 (Support Vector Machine)、深度學習 (Deep Learning) 等蓬勃發展。

神經網路的神經元架構如下：

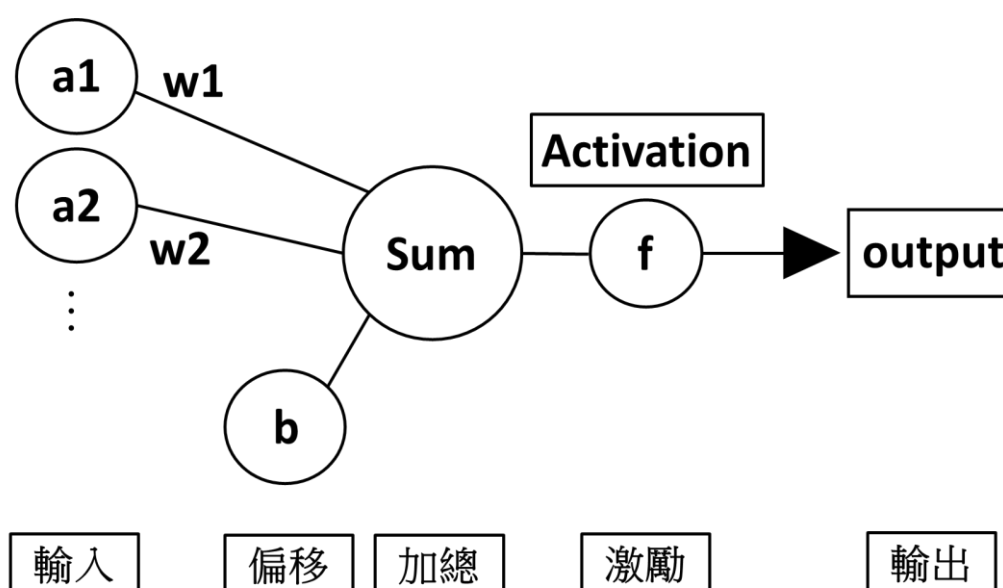


圖 2 神經元架構圖

將圖 2 轉換成公式敘述如下：

$$a_1 \times w_1 + a_2 \times w_2 + \dots + b = \left(\sum_{n=1}^n a_n \times w_n \right) + b \quad (1)$$

接著將公式 1 的結果引入激勵函數 f ：

$$f\left(\left(\sum_{n=1}^n a_n \times w_n\right) + b\right) \quad (2)$$

而激勵函數 f 的判斷條件：

$$f(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 \text{ or } -1 & \text{otherwise} \end{cases} \quad (3)$$

圖 2 和公式各參數描述如下：

- a_n 為輸入值，亦所謂特徵 (Feature)。
- w_n 為權重值 (Weight Value)。
- b 為偏移量 (Bias)。
- f 為激勵函數 (Activation Function)、轉換函數 (Transfer Function)，大部分是非線性函數，以總和作輸入，並將輸出作為判定的結果。

綜上公式 1 到公式 3，激勵函數的工作如同水杯的杯口，決定水杯內的水是否滿溢出來，判斷來自公式 1 的輸出值是否有大於臨界值或閾值，需針對實際情況選擇適當的激勵函數，如下列圖 3(左) 就是常見的 ReLU (Rectified Linear Units)，輸出介於 0 到輸入值之間，凡是輸入小於 0 直接以 0 作輸出，否則保持原輸入數值；Sigmoid 在處理負數的輸入上，不會完全把負數歸零輸出，而是把負數輸入下的輸出限在 0 到 0.5 間，而正數時的輸出亦規範在 0.5~1。即所有的輸入都被強制限在 0 到 1 之間，如公式 4 和圖 3(中)；形狀與前者相似的 Tanh，則是對於正數的輸入予以 0 到 1 的輸出，反之，對於負數輸入予以 0 到 -1 的回應。雙邊的輸入欲趨近 0，輸出的值欲趨向於 0，示意於圖 3(右)。

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

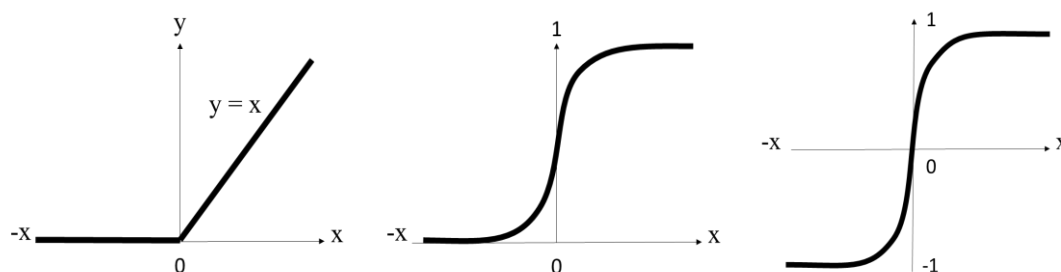


圖 3 激勵函數 ReLU (左)、Sigmoid (中) 和 Tanh (右)

先前曾提及單一神經元受限於 XOR 功能的線性不可分問題。多層感知器 (Multilayer Perceptron) 便因此發展出來，在輸入和預測輸出之間，模型的描述亦更加細緻，學習能力提升，具錯誤率收斂的能力，延伸出機器學習、深度學習至人工智慧那趨之若鶩的研究浪潮。依圖 4 的完全連接神經網路架構，類神經網路架構基本可分成：

- 輸入層 (Input Layer)：如圖 4 最左邊的 c 節點，該層為 c1 到 c3 節點組成，有諸多神經元(Neuron)接受大量資料和訊息輸入，數學式表示上以輸入向量呈現，通常僅一層。
- 隱藏層 (Hidden Layer)：見圖 4 中間的數個空白節點，該層涵蓋所有白色節點，由輸入層和輸出層之間的眾多神經元和連結組成。可以是一層(就是單層感知器)抑或是多層。節點(神經元)數目不定，但節點數目越多，如同更多人思考，成效也會有所提升，關係上好比邊際效用遞減法則，在抵達某個臨界數量後，即便隨著節點數目增加，提升的成效會逐步降低，說明過多的節點數目在成效提升上不會顯著。
- 輸出層 (Output Layer)：如下圖 4 最右邊的 d 節點，該層為 d1 到 d3 節點組成，訊息在數個神經元連結中彼此接力轉傳，從隱藏層節點分析判斷後，如選舉的開票般輸出結果，並可以將各節點結果在數學式上以向量形式輸出，並且以數值最大的作為分類或預測結果。

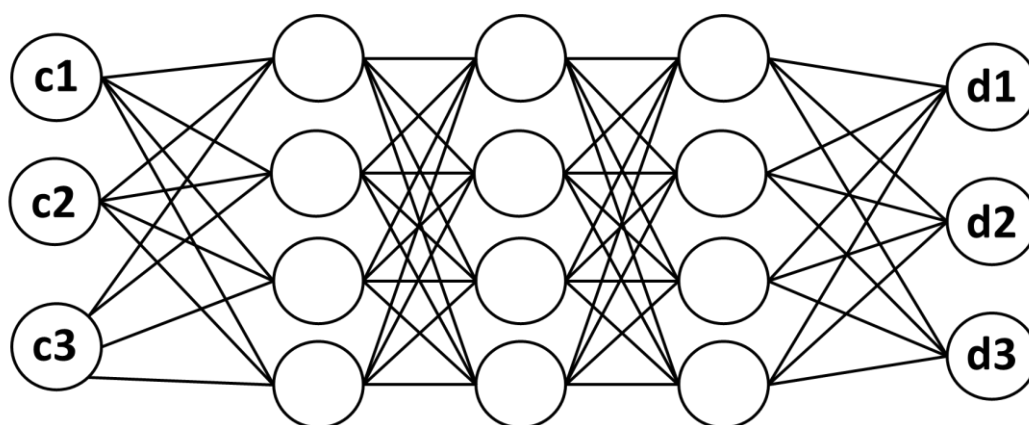


圖 4 多層感知器架構圖

類神經網路藉由訓練數據進行自我調整，對各層的權重進行修正 (Learning) 並構築模型的學習過程，謂之訓練 (Training Algorithm)。具體的學習過程則因感知器結構和模型不同而有所相異，常用如反向傳播演算法 (Backpropagation)。而類神經網路在學習過程還可以依數據是否有供答案參照而有所分別，像是監督式學習網路 (Supervised Learning Network)，著名的如卷積神經網路 CNN (Convolutional Neural Network)、遞迴神經網路 RNN (Recurrent Neural Network)，以及無監督式學習網路 (Unsupervised Learning Network)，例如說生成對抗網路 GAN (Generative Adversarial Network)、強化學習 (Reinforcement Learning)。

類神經網路在自動學習後，倘若要做出調整或是效能提升，基本上無從改進該學習成果，因為那些權重對直觀的意義上皆太過分散，因此藉由人為手動改進類神經網路權重其實不容易，一般會搭配組合最佳化的演算法進行“修剪 (Pruning)”，例如基因演算法 (Genetic Algorithms)。

在應用上，Jiabao Sun 等人[2]就針對癌症、第一型糖尿病、第二型糖尿病和衰老基因，採用三種類神經網路：多層感知器 (Multi-layer Perceptron)、Functional Link Artificial Neural Network 和支持向量機 (Support Vector Machines) 嘗試進行基因方面的疾病預測，最終三種類神經網路在準確度、精度、召回率上皆相近。其中支持向量機在四種疾病上皆獲得最佳準度，多層感知器次之，而 Functional Link Artificial Neural Network 則是花費時間最短，具最高的效率，在四種疾病的預測上皆低於零點一秒鐘，與其他組相比有相當的差距。

Christopher Pedersen 等人[3]，嘗試從不可控制的遊戲特徵（如收集金幣數）和可控制的遊戲關卡特徵（例如懸崖數量）六種情感偏好，以神經進化偏好學習作函數趨近。遊戲採納 Markus Persson 的《Infinite Super Mario Bros》修改版。首先找尋線性相關性，然後嘗試簡單的非線性模型，最後組成更複雜但功能更強大的非線性模型。因此，應用簡易的單層感知器（Single-layer Perceptron）和多層感知器（Multi-layer Perceptron）作為學習遊戲特徵（Artificial Neural Network 的輸入）與遊戲的情感偏好（Artificial Neural Network 的輸出）之間的關係。由於該類學習問題沒有可微分的輸出誤差函數，得透過神經網路的人工進化（神經進化）偏好學習實作學習[4]。所以搭配基因演算法，應用適應函數測量實驗玩家的情感表現和相應模型輸出間的差異，嘗試找到最理想的模型演化參數上使用 1000 個母代，持續進化 100 代，高斯突變發生概率為 5 %。流程如圖 5。

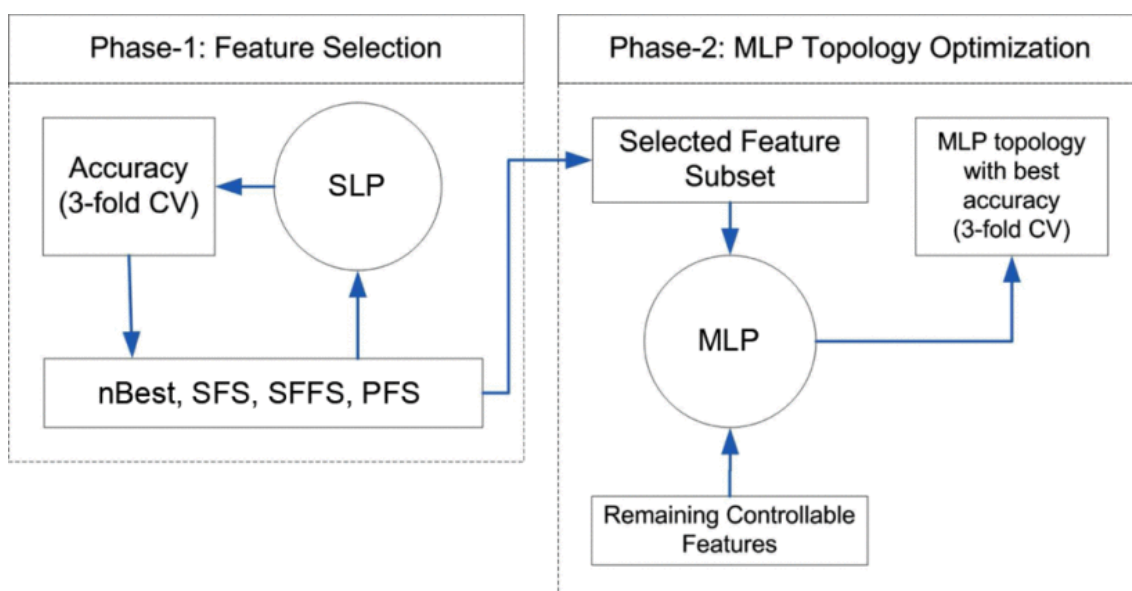


圖 5 研究架構圖[3]

該研究使用最佳單一特徵選擇（nBest）、序列前向選擇（SFS，sequential forward selection）、序列浮動前向選擇（SFFS，sequential floating forward selection）和感知器特徵選擇（PFS，perceptron feature selection）四種方案作比較。希望透過關卡或遊戲考慮每個特徵間彼此的獨立性，而能設計出許多有意義的附加可控特徵。像是與敵人和物品的數量、類型和分佈，關卡中各個位置之間的高度差等有關的特徵；在關卡設計之外，有可能設計與遊戲的物理特性（例如重力和慣性）以及遊戲的其他設計方面（例如“死亡”或贏得關卡的含義）有關的可控制特徵。

2.2 卷積神經網路

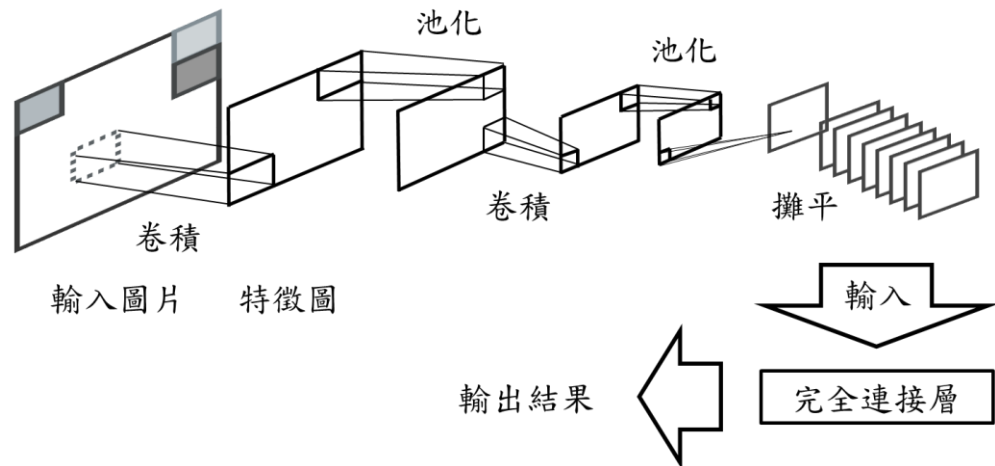


圖 6 卷積神經網路示意圖

卷積神經網路 CNN (Convolutional Neural Network)[5]，如圖 6，專長辨識大型圖像，憑藉對圖像局部區域進行卷積 (Convolution) 和池化 (Pooling) 來擷取圖像特徵，之後攤平轉換成一維輸入完全連接層，配合神經網路的訓練達到能夠進行辨識的作用。ImageNet 每年舉辦的競賽 (ILSVRC) 就有不少大型的卷積神經網路取勝，如著名的深度學習 VGG (Visual Geometry Group) 以及 ResNet。VGG[7] 有六種結構，又以 VGG16、VGG19 最廣為人知，後面的數字是因為 VGG16 以總共 13 個卷積層、3 個全連接層加上 5 個池化層組成，16 指的就是權重層數總和 (13+3)，所以 19 實際上就是多了三個卷積層。對輸入圖像均會做減去訓練集平均值的預處理。以 VGG16 為例，與圖 6 稍有差異的是卷積跟池化並未完全交叉，而是連續兩到三個卷積才銜接一個池化，目的是要減少參數數量，防止學習曲線因為過於貼近學習資料所造成的過擬合 (Overfit) 情形。ResNet[8] 則以前者為基礎，加入了殘差學習 (Residual Learning) 的概念，打破以往逐層傳送輸入，除了將特定層的輸出予以下一層之外，還額外繞過二或三層與之後的輸出加總，作為下一層輸入傳遞，將此架構組稱作一個 Building Block 或 Bottleneck Block，形似短路的電路。此架構解決了深度網路因為直接增加層數超過特定臨界導致錯誤率提升的情形，時常在訓練資料階段就發生，有別於過擬合及梯度消失 (Vanishing Gradient)，稱為退化問題 (Degradation Problem)。整體架構前期大多憑藉較大的卷積核移動間距或池化層降低圖像解析度，並提升通道數 (Channel)，接著才進行殘差學習。因此其架構後面就會有相對龐大的數字，如 ResNet34、ResNet50 等。

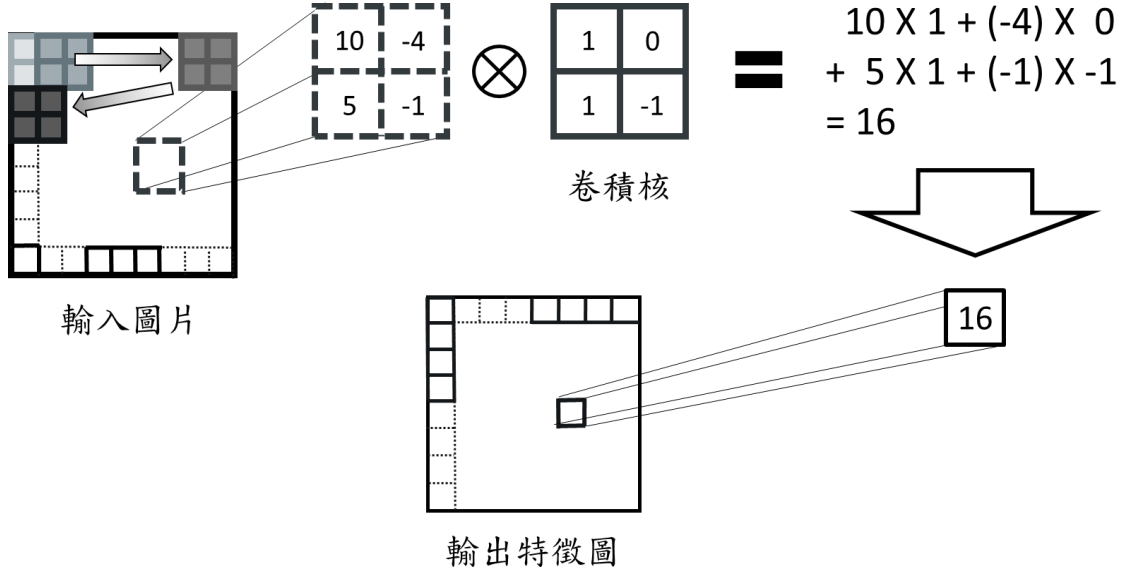


圖 7 卷積流程圖

卷積 (Convolution) 為一套固定的運算流程，輸入進來的圖像皆須經過此運算數次。見上圖 7，在運算過程中有一個要素稱為“卷積核 (Kernel)”或者“過濾器 (Filter)”，一般會以一個小於輸入圖像大小的方型表格理解，表格邊長自定義，並且在相同的卷積層裡，卷積核的邊長不會因為屢次計算而變更，可以為 2 或 3 甚至更高，即“卷積核大小 (Kernel size)”因而卷積核面積就是 2×2 或 3×3 ，卷積核內部的每一格都會有各自的權重值。每當要計算時，倘若未額外添加框距，就會從輸入圖像的左上角貼齊兩個邊覆蓋，開始由左邊逐次往右移，完成一排再往下新的一排，移動的距離亦是預定義的，相鄰兩次移動之間的覆蓋範圍可以重疊，以此類推跑完整張輸入圖像。如同兩向量的內積運算，將所比對區域的像素對應到卷積核的權重值相乘並加總得出像素，也就是單一次卷積的“特徵 (Feature)”。如此一來就會得到一張面積略小於輸入圖像的“特徵圖 (Feature map)”，如公式 5：

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1 \quad (5)$$

其中 n_{in} 和 n_{out} 分別為輸入 (包含框距) 以及輸出圖像的邊長或寬度， k 為卷積核大小、 p 為框距、 s 是移動距離。該特徵圖後續就會作為辨識圖像時比對圖像特徵的參考依據。

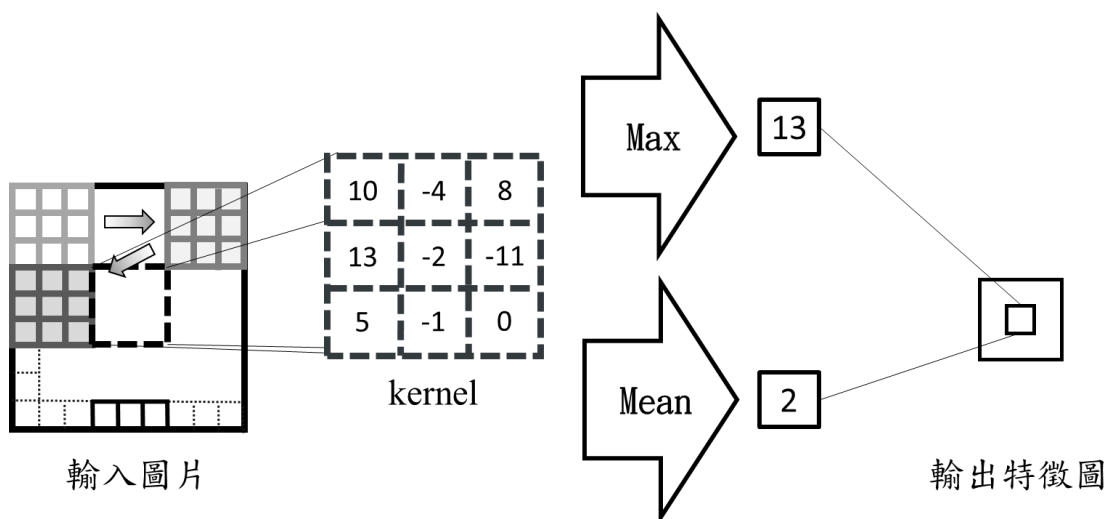


圖 8 池化流程圖

池化 (Pooling)，意同採樣 (Subsampling)，如圖 8。與先前所敘述的卷積運算一樣都需要一個類似卷積核的過濾器參與，但是該卷積核內沒有各自對應的權重值，意同沒有加權的過濾器，每次移動長度一般會預設等同於過濾器大小。池化理解上如同用顯微鏡檢視圖像從高倍率轉至低倍率一般，擴大了觀察範圍，對於輸入圖像逐一進行局部運算，和前述卷積運算流程差異在於運算最後的步驟不再是採用類似內積的形式求值，該步驟需要在運算方式上再額外預先選擇，主流的如 Max Pooling、Mean Pooling。在神經網路中，越複雜或者參數數量越多的神經網路模型容易造成過擬合的情形，池化層能有效減少神經網路模型的參數，也減低計算量，而且依然可以保留圖像重要的特徵，亦能使神經網路模型對圖像細微的變換或是一些失真的細節變得更穩定，意義上可以看成在維持該圖像局部最大特徵的前提下，壓縮資料達到降維的目標，讓相對微小的特徵能夠有誤差空間。以 Max Pooling 來說，即同一區域裡，數個特徵彼此競爭最大特徵，獲勝的像素將會作為代表並忽略其他像素，成為輸出特徵圖對應位置的像素。而 Mean Pooling 就是在局部的眾多特徵或者是像素之間取平均值，並以此作為局部特徵的代表，考量上有所不同於 Max Pooling，兩種保留特徵的選擇各有不同的優勢。綜上所述，池化運算由於全程沒有權重涉入，因此不參與訓練過程，僅僅對輸入做轉換，具減少計算量並且降低維度的優點，也能防止過度擬合的問題，對於太過細微的變化並不影響最終的輸出結果。

2019 年，中國武漢爆發病毒性肺炎，世界衛生組織宣佈為全球大型流行病，將病毒命名為嚴重急性呼吸系統綜合症冠狀病毒 2 (SARS-CoV-2)，以其產生的肺炎則稱為 COVID-19。最廣泛認可的篩選是採 RT-PCR 測試但成本高昂，作為第一線檢測尚嫌不足，另一種方式是以電腦斷層掃描 (CT) 協助判斷肺部病變與否。Raghul M 等人[6]使用深度學習模型，以電腦斷層掃描 (CT) 進行 COVID-19 相關的肺部病變判別，採用了 ResNet18、ResNet50 和 VGG16 進行遷移學習。最後 VGG16 模型具有最高的 95 % 的分組準確度，實驗流程如圖 9。

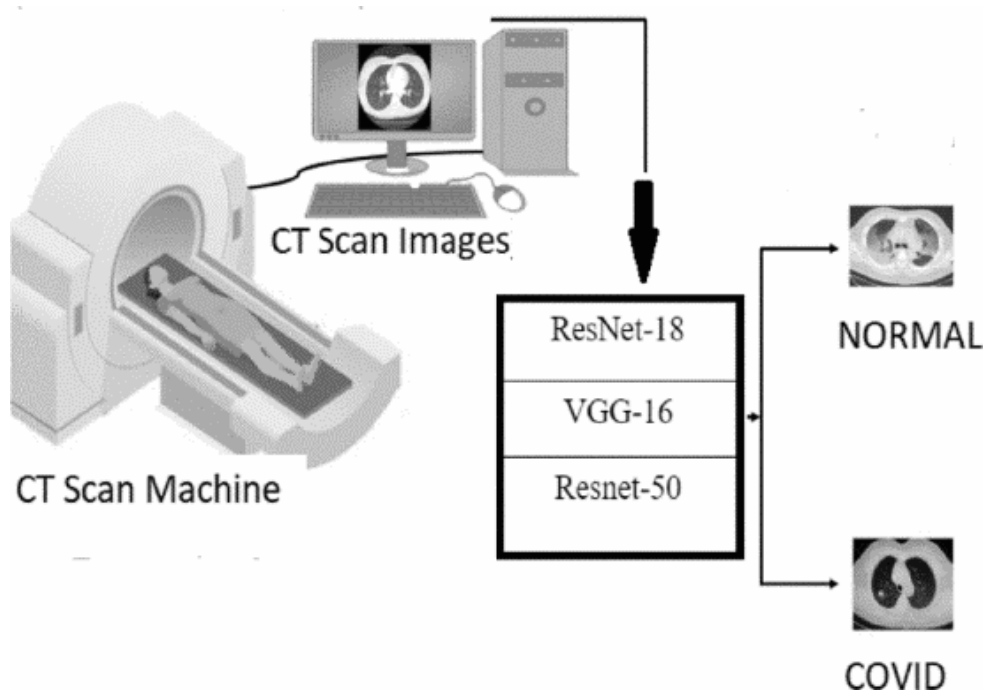


圖 9 COVID-19 深度學習檢測流程圖[6]

2.3 神經網路架構搜尋 (Neural Architecture Search)

如同著名的圍棋神經網路 AlphaGo，其架構就有採用到強化學習 (Reinforced-Learning) 作為短期內巨幅進步的推手。因此，文獻[9]便也應用強化學習作為自動搜尋神經網路架構的方式，提出 MetaQNN 以產生最佳表現的卷積神經網路模型，在 SVHN、CIFAR-10 和 MNIST 三種數據集上進行實驗，表現甚至超越當時幾個手工設計出來、相對龐大的神經網路，架構如圖 10。文獻[10]，先採用遞迴神經網路 (RNN) 產生子網路並且進行訓練和評估，以此循環更新控制端的引數，而控制端引進強化學習接收回饋的精準度作為獎勵機制以達成自我學習，整體架構如圖 11，在策略梯度 (Policy Gradient) 上可能推出更佳遞迴神經網路模型，而後在 CIFAR-10 數據集上勝過類似的神經網路模型。而兩者的共通點儘管皆成效卓越，但是計算成本上，即便是在採用相對小的資料集或者是降低訓練代數亦還是太過龐大。

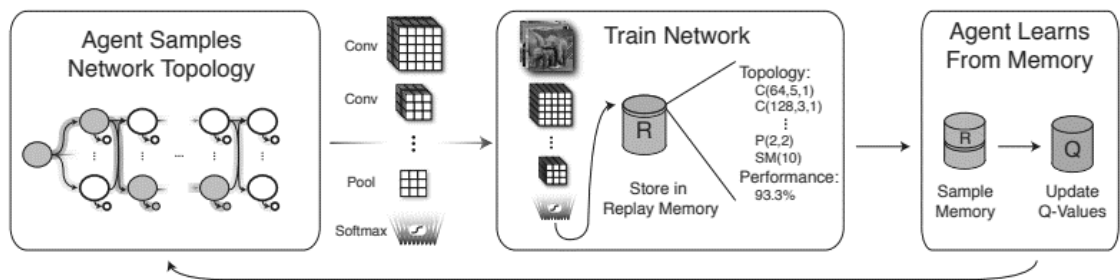


圖 10 神經網路搜尋架構圖[9]

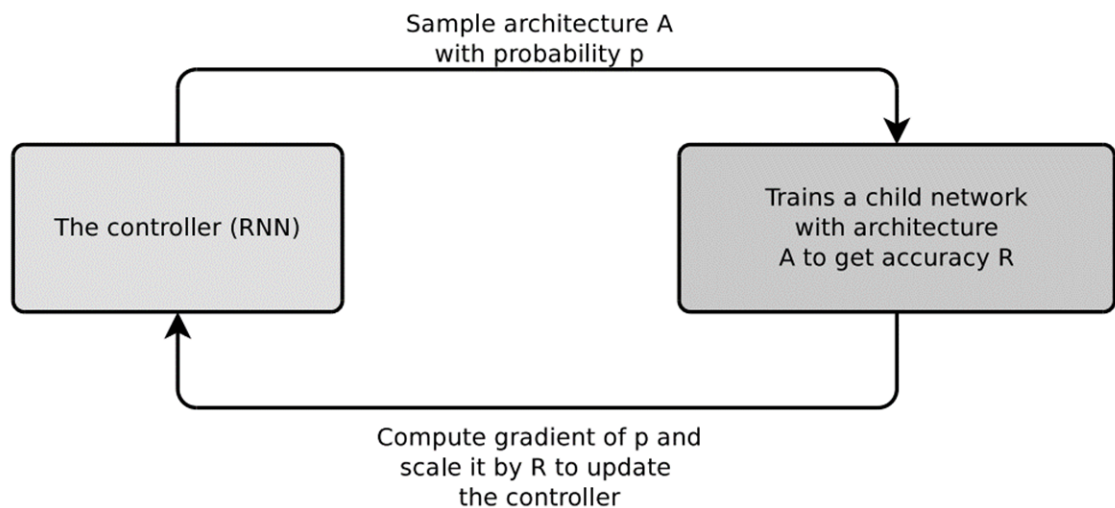


圖 11 神經網路搜尋架構圖[10]

因此，要舒緩 NAS (Neural Architecture Search) 如此龐大的計算成本，各樣式的解決方案接踵而至，但仍然可以描繪成如圖 12 的架構，由搜尋空間 (Search Space)、搜尋策略 (Search Strategy) 和性能評估 (Performance Estimation Strategy) 三個部份組成。詳述如下：

- 搜尋空間 (Search Space)：該架構下的網路層總數、網路層的超參數 (卷積核大小、通道數、卷積層維度等等)和網路層種類，這些可預先訂好的元素所組成的空間。
- 搜尋策略 (Search Strategy)：在架構的尋找上所採用的規劃，如何降低因暴力破解所造成的巨大計算成本，讓神經網路架構有個穩定的成長傾向，選定策略執行是相當重要的，省略太過無謂的、影響力低的非必要流程。好比在超參數優化 (Hyperparameter Optimization) 的問題上，可以選用網格搜索 (Grid Search)、貝葉斯優化 (Bayesian Optimization) 等選項。
- 性能評估 (Performance Estimation Strategy)：評斷當前要測試的神經網路架構的表現。條件可以依照實際需求做調整，像是模型訓練後的準確度、參數數量等，可以是複數條件，但相對也會因此增加模型被排除的機會而導致搜索時間延長的情形。

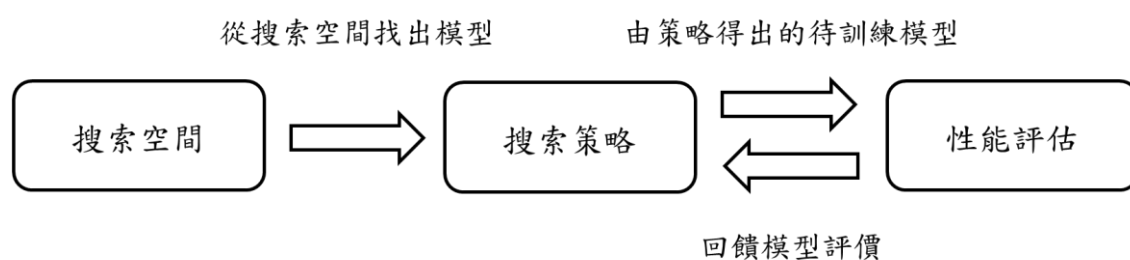


圖 12 神經網路搜尋基礎架構圖

Hieu Pham 等人[11]，提出了 ENAS (Efficient NAS)，ENAS 將搜索空間表示成一個有向無環圖 (DAG)，將原來 NAS 的搜尋過程喻為在大圖中尋找子圖，其中的任一子圖皆為一個神經網路架構，每一個節點表示局部計算，讓搜索出來的所有子架構共享權重，如此一來就極大地縮短了搜尋時間，相較於原來的 NAS 架構，以 10 個 GPU 搜尋 CIFAR-10 就花費 8 到 10 天，該文獻僅使用一張 GPU 費時 10 小時就完成了，示意如下圖 13。

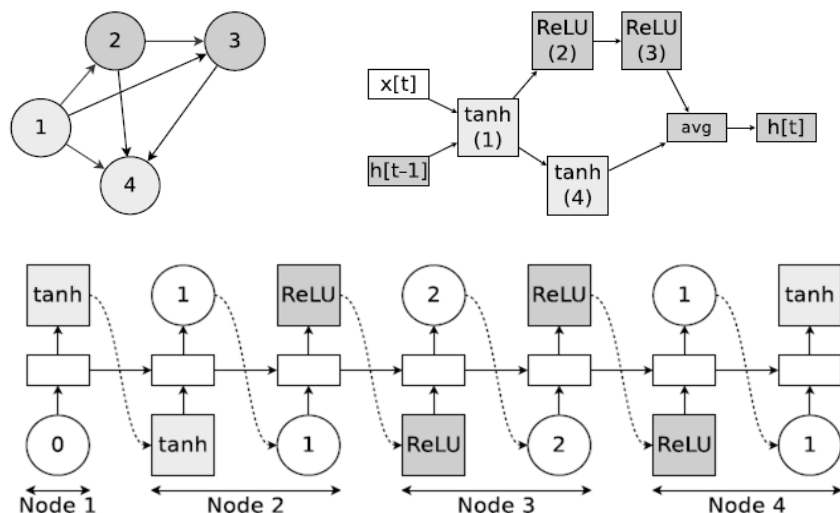


圖 13 ENAS 基礎架構圖[11]

Chenxi Liu 等人[12]，則採用逐層漸進式的搜索，特別的是，文獻額外應用了 Controller (Predictor) 預測當前神經網路架構眾組合的準確率高低，並且以輔佐的角度協助淘汰，其預測限制了子網路架構的成長，僅保留一小部分的組合群體去進行訓練後更新 Controller (Predictor)，以此類推下一層的加入，如下圖 14。

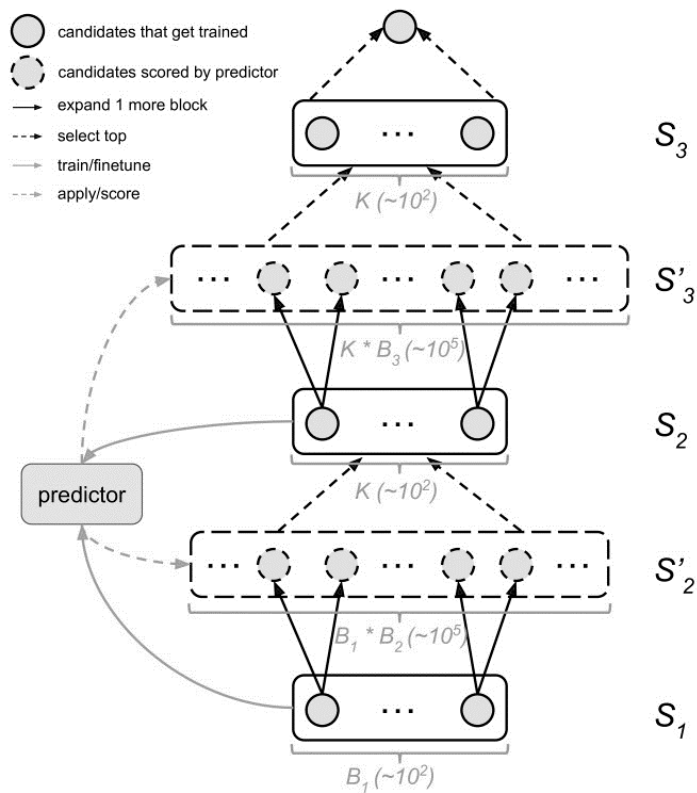


圖 14 神經網路搜尋示意圖[12]

2.4 啟發式演算法 (Heuristic Algorithm)

對於最佳解問題，時常必須面臨如何在由數個元素所組成的搜索空間中有效的搜尋，迴避窮舉法複雜的計算成本，針對該問題提供的一切訊息，和可接受的時間下的搜索方案，盡管未必能夠解決問題，而且不保證是最佳解決方案，在歷次執行的結果下也會有所差異的方法，稱其為啟發式方法 (Heuristic Function)，簡稱啟發式 (Heuristic)。而當該啟發式方法成為框架的形式，不再受限於單一問題，就成為元啟發式 (Metaheuristic)。特別的是，元啟發式還得在全域解和區域解中間尋求平衡，一來防止全域搜尋上的收斂障礙，二來避免陷入區域最佳解的泥沼。文獻[13]提及啟發式演算法是憑藉疊加的經驗，在多次的實驗中找出前進的方向，兼具隨機性與確定性，並以此作為屢次謀求進步的跳板，成為近似解的演算法。以下將概述常見的最大值求解方法 -- 爬山演算法和基因演算法。

爬山演算法 (Hill-Climbing Algorithm)，如圖 15 所演示，是一種相當容易理解的演算法，屬於區域擇優的方法，相較於深度優先搜尋 (Depth-First Search)，爬山演算靠著在搜索範圍內數個鄰近駐點 (問題解) 間的回饋，從圖 15 的起點 S 至更高點 S'，若下一個駐點更佳，則轉移至下一個駐點，如此循環。但是在山頂 (圖 15 的黑色三角形) 時容易因搜索範圍內尚未找到範圍外更高的點而駐足與此，即卡在區域最佳解，抑或是在山谷 (圖 15 的灰色三角形) 因兩側皆有相近的坡度，形成來回擺盪的情形，虛擬碼如表 1。文獻[14]就曾在 MNIST 上以爬山演算法嘗試作為卷積神經網路自動構築的搜索策略，模型平均準確度落在 91 % 到 92 %。

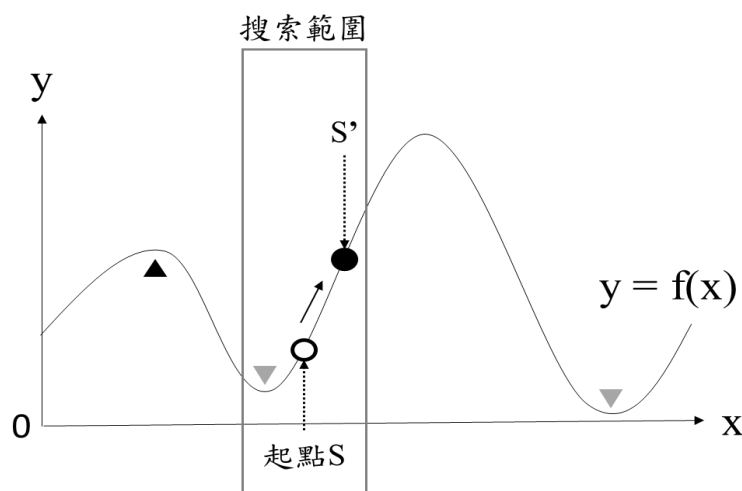


圖 15 實驗流程圖

表 1 爬山演算法

Hill-Climbing Algorithm (S)
<pre> S ← 任意起始解 while(存在鄰近解 S' 且高於 S) S ← S' end return S end </pre>

基因演算法 (Genetic Algorithm)，或又稱遺傳演算法，也屬於進化演算法 (Evolutionary Algorithm) 的一員，受生物界的演化流程啟發，遵守物競天擇、適者生存原則。將問題的解對應成基因編碼，並將基因組成染色體，先是求出適應值檢視群體表現，利用適應值及基因的演化，透過每一代基因的變化，淘汰過低適應值的染色體，保留高適應值的菁英染色體，進行選擇、配對、突變、更新群體適應值的過程，完成數次迭代來趨近最佳解，基因演算法採用間接處理和設計變數相互對應在字串上，而非直接處理變數的本身，函數本身求解空間是否連續，能否微分皆不受影響，勝過一般最佳化方法需要問題目標函數與限制函數皆為連續或可微分的條件才能繼續工作，如此一來跳脫搜尋空間的限制；只利用適應度函數作為判斷的依據，不需要困難的數學運算，虛擬碼如下頁表 2。

和前述爬山演算法從單一點起手不同，基因演算法允許從多個起點開始，並且可以隨著歷代演化更新且縮減群體，亦可以視需要修改成在維持群體特定數量的情形下持續補充新染色體提升變化的可能，更利於突破適應值的區域最佳解情形。又或者因為歷代的答案之間概率上會維持前代基因保留下來的優勢和特色，在突變階段針對優秀基因予以保留不參與，免於因突變所帶來的逆轉影響，導致前功盡棄。核心概念上即維持基因持續變化的探索下，增加更高適應值基因出現的概率，保留當下適應值更優良的基因，雖然要探索的範圍也更廣，計算成本提升，但是卻保有方向性，在足夠長的總代數下終將會走向極度趨近最佳解的答案。

表 2 基因演算法

Genetic Algorithm
產生初代群體
計算初代染色體的適應值
while(未達終止條件)
選擇
配對
突變
計算該代染色體的適應值
end
end

2.5 傅立葉轉換 (Fourier Transform)

聆聽透過聲音，可以直覺理解高音會有較刺耳、尖細的印象；低音則有沉穩、鬱悶的感受。但繪製成單一頻率的波型圖話就會如圖 16 所示，若左方為低音的波型圖，右方則是相對高音的波形圖，兩者之間的差異在於同一時間內（下方的兩條粗線），曲線的密集程度。密集度越高聲音越高，反之密集度越低則聲音越低。

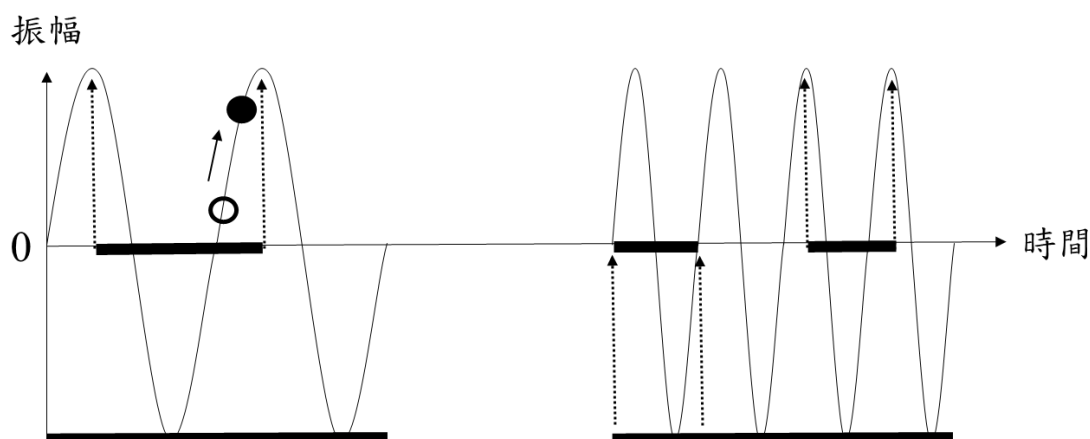


圖 16 波型圖

並且圖 16 中可以發現，波型就是在一定高度範圍內以固定的曲線上下震盪的曲線，震幅 (Amplitude) 最高點稱為波峰，反之為波谷。若以兩個波峰所對應的橫軸線段為基礎，排除兩側曲線線形重複的線段所殘留下來的路徑就是所謂的單位週期，定義為一個點從該路徑的起點到終點所花費的時間（兩個最高點之間的粗線距離），時間橫軸上的三條粗線段。換個角度看，倘若是檢視單位時間內一個點在曲線上完成的週期次數，指的就是頻率 (Frequency)，兩者彼此互為倒數。如此一來，就能想像男聲與女聲之間，其聲音在頻率上差異。而波形的起始數值，就是所謂的相位 (Phase)。

傅立葉於 1807 年的論文 “On the Propagation of Heat in Solid Bodies”，提及關於傅立葉級數 (Fourier Series)，描述任意複雜連續的週期訊號，皆可由三角函數的正弦和餘弦的組成來表示。而這些分離出來的單一頻率，稱為頻率分量。與單一頻率的波形圖 16 表示，即所謂的時域，然而同樣的聲音還可以裡另一種角度繪製，形成一張僅僅以頻率分量描述的強度頻譜，如下圖 17，左方對應圖 16 的低頻時域，單位時間完成了兩個週期；右方則是相對的高頻時域，單位時間完成了四個週期。由於雙方波形皆屬於單一頻率，且時域圖裡差了兩倍，固僅有唯一卻相異的頻率分量也會是相差兩倍。

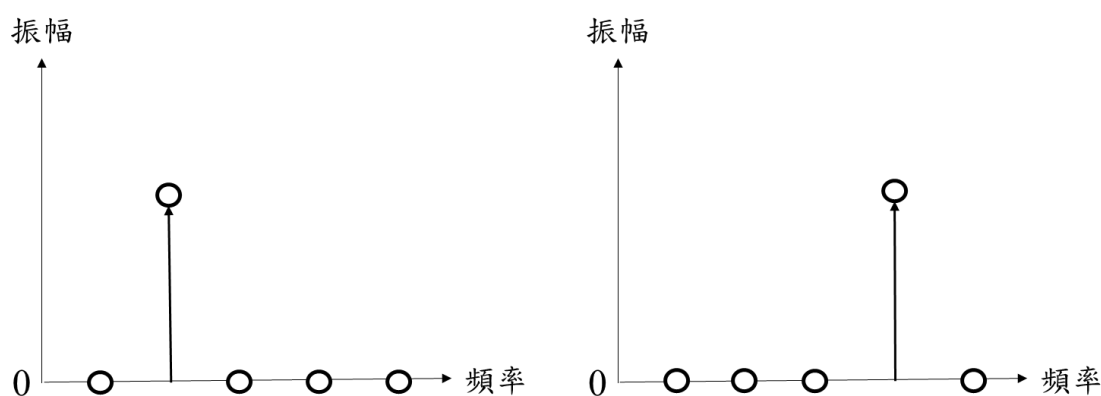


圖 17 頻譜強度示意圖

此外，對於一段聲音的描述除了時域圖的角度能唯一表示外，只靠著頻譜圖表示是不夠的，因為空有頻率分量的組成，並未能知曉是從該頻率波形的哪個位置開始，例如：波峰或者是波谷，所以必須配合相位頻譜一併表達。然而聲音時常並非單一頻率，反而經常是由多個頻率分量疊加形成。

複數一般會以左方純數值的形式（實部）加上右方數值乘上虛數 i 的形式（虛部）表示該複數。而複數亦可繪製在平面座標上，見圖 18 的複數平面，縱軸為虛數軸，其中 $i = \sqrt{-1}$ ，橫軸為實數軸。此外，複數在這裡還可以採用另一種形式表達，即是從原點到複數所對應的點連成的直線長度 A ，和該直線與橫軸的正數一側（角度的起始參考基準，為 0 度）的夾角 θ 。夾角以圓點為圓心且直線為半徑逆時針旋轉，則角度為正值，反之順時針旋轉即為負值，單位有角度和弧度可以彼此轉換。當要轉換成負數的形式時，僅需要將直線長度分別乘上 $\cos(\theta)$ 與 $\sin(\theta)$ ，就會分別得到該複數實部以及虛部的數值。

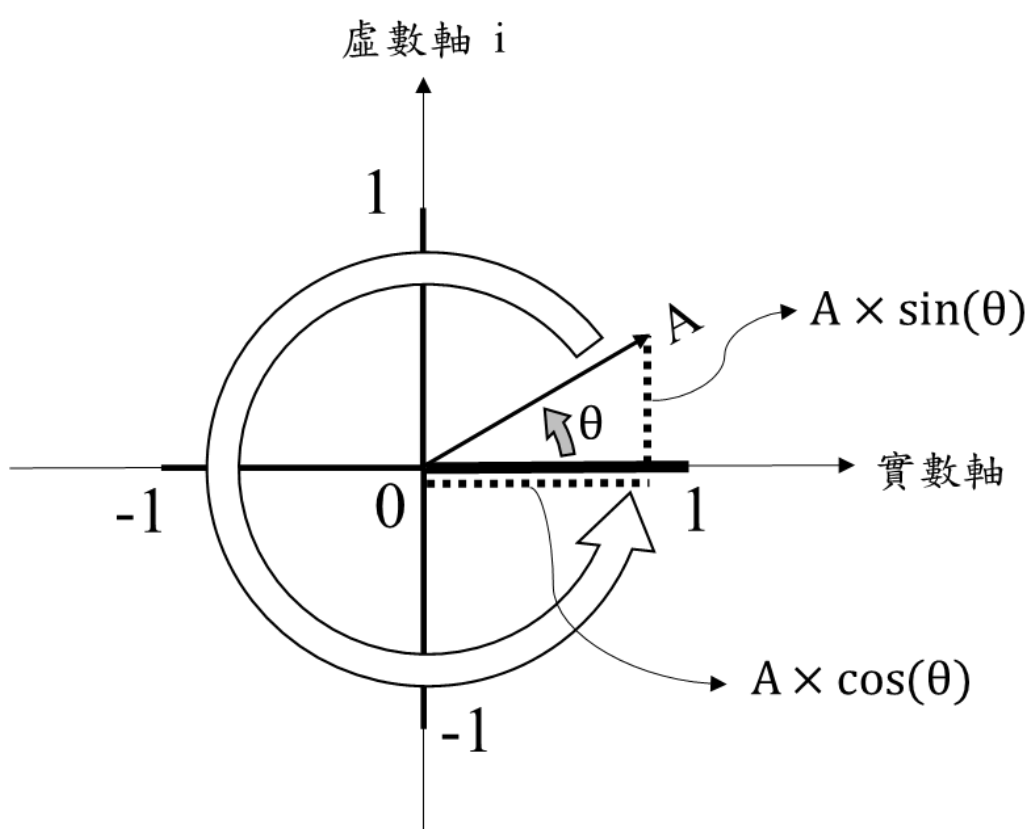


圖 18 複數平面圖

考慮進相位 φ ($\varphi = \omega \times$ 在時域的起始時間，等同在複數平面上的起始角度) 和角頻率 ω ($\omega = 2\pi/T = 2\pi \times f$, T 為週期, f 為頻率), 單一頻率在時域的曲線就能以公式 6 表示。

$$\text{時域振幅} = A \times \cos(\omega t + \varphi) = A \times \cos\left(\frac{2\pi t}{T} + \varphi\right) \quad (6)$$

雖然僅採用 \cos ，但仍可以與 \sin 進行相互轉換。也就是說，時域就是複數平面上，直線持續繞著原點旋轉，對應時間的投影。直線長度就是時域上波形振幅的最大值。特別的是，強度頻譜圖中，頻率為 0 的分量，由於沒有直線長度，因此完全不會影響波型，卻影響著整體波型相對於橫軸的上下移動，稱為直流分量。

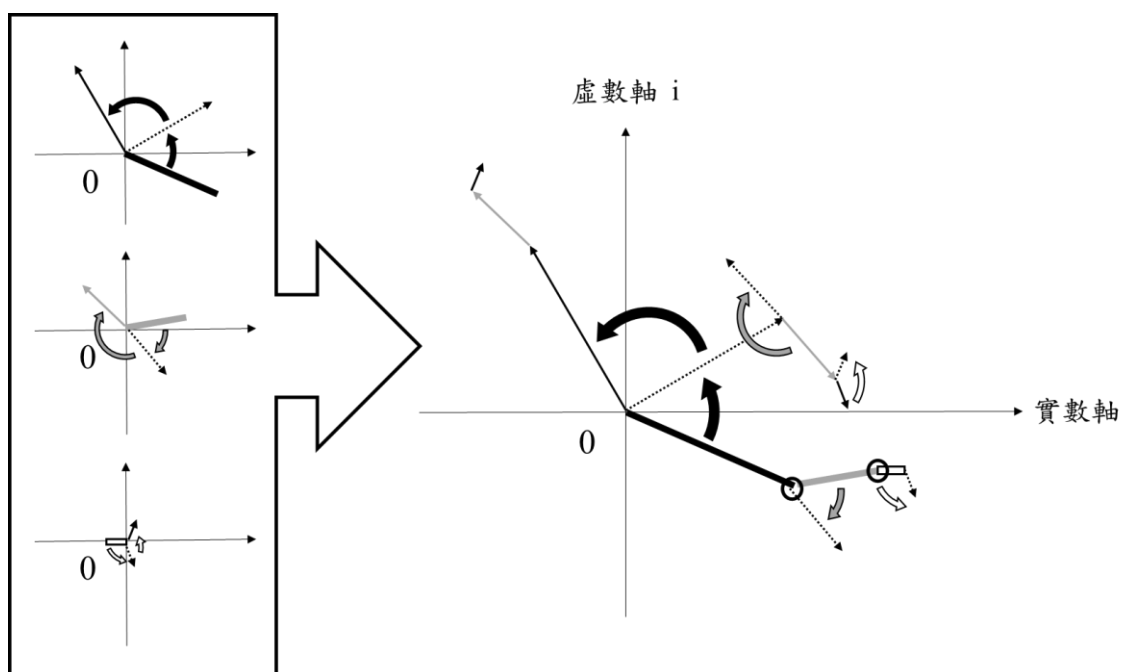


圖 19 多頻率分量疊加的複數平面圖

要描述更複雜的波形，多個頻率分量疊加在複數平面的形式如圖 19，且各項在公式 6 的參數彼此獨立運作、不會中途異動，但彼此之間半徑必定維持頭尾相接，並且將各時間對應的點投射至時域，連起來形成看似短時間內不規則的曲線，其實僅僅只是週期較長而已。在週期函數特色的前提下，週期外的曲線均可以透過週期函數再次描繪，只要找到描述該週期的式子，就能帶入數值去進行推算。但若是針對特定頻率，透過公式 7 的傅立葉級數分解就能作分析或者是疊加的推算。因此時常會先將該不規則的波形直接當作單一次週期去進行分析，抑或者是週期無限長的波段理解，即傅立葉轉換 (Fourier Transform)。

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \times \cos\left(\frac{2\pi \cdot n}{T} t\right) + b_n \times \cos\left(\frac{2\pi \cdot n}{T} t\right) \right) \quad (7)$$

$$f(u, v) = C_u C_v \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (8)$$

當傅立葉轉換經過的二維推廣就會如公式 8，代入大小 $M \times N$ 的圖像座標 (x, y) 、頻率變數 (u, v) 、正規化係數 $C_u C_v$ (可以是 $\frac{1}{\sqrt{MN}}$ 、 $1/MN$ 或者 1) 計算後，就會形成圖 20 左上的頻譜圖，中央為高頻，往四個角落的方向為低頻。而為了與聲音的頻譜同方向呈現，預設的處理流程 (如程式框架)，會把斜對角的兩象限區域互換，如圖 20 左下，以低頻為中央，成為可見的白色十字線或白點。以前述的頻率比喻，圖 16 左方的低頻如同圖 20 中上，圖 16 右方的相對高頻好比圖 20 的右上，兩者分別對應各自正下方的頻譜圖。能觀察出頻率越高，轉換成頻譜圖對應的光點間隔越大。且頻譜圖不會因為原圖像經切割成數塊換位而受影響，具平移不變性，但是會受旋轉影響。在影像處理上有諸多廣泛應用，像是濾波處理，如圖 21 所示，該人偶圖像就經過轉換成頻譜後進行濾波處理，得到包裹相位圖進行相位展開。

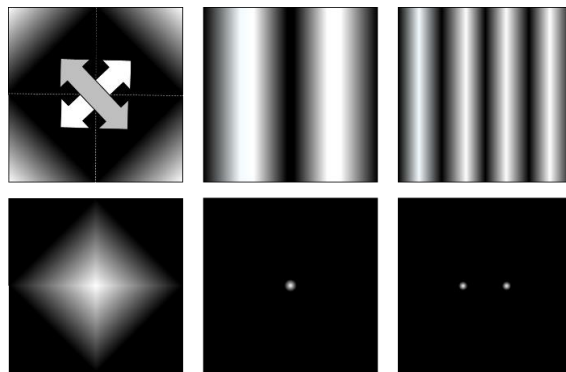


圖 20 圖像轉頻譜模擬圖

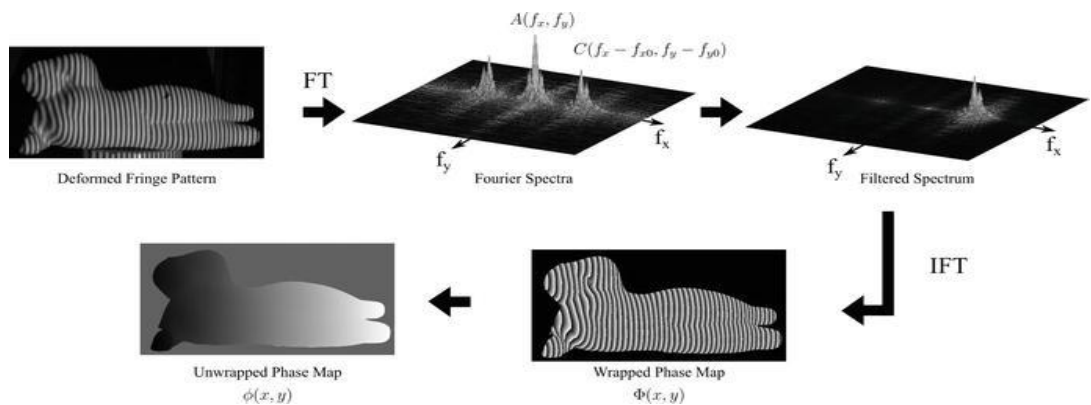


圖 21 轉頻譜並濾波後的圖像[15]

第三章 研究方法

3.1 研究架構

本研究架構設定在總代數極限內，從初始模型輸入爬山演算法開始，任一代模型連續維持且未被替換，當超過指定代數後就提前輸出最終模型，過程中網路層類別和通道數皆為有限選項的隨機配置，如此反覆執行爬山演算法直至找到最終模型，並且以三種開放的手寫字符資料集 (MNIST、QMNIIST、KMNIST) 與一個立體圖像資料集 (Fashion-MNIST) 訓練模型。見圖 22，每個資料集下依構建方式的差異分三種實驗組和控制組，隨機組和控制組構件方式相同，差異在於是否採用頻譜轉換層，中間的 CNN 基礎層將於下一節詳述，灰色層表示當前最新操作的網路層，虛線的區間表示其他已加入的網路層，鏤空的部分為新網路層插入的區間，每組分別執行如圖 23 的流程 100 次後統計並分析，實驗全程以 Pytorch 實作。

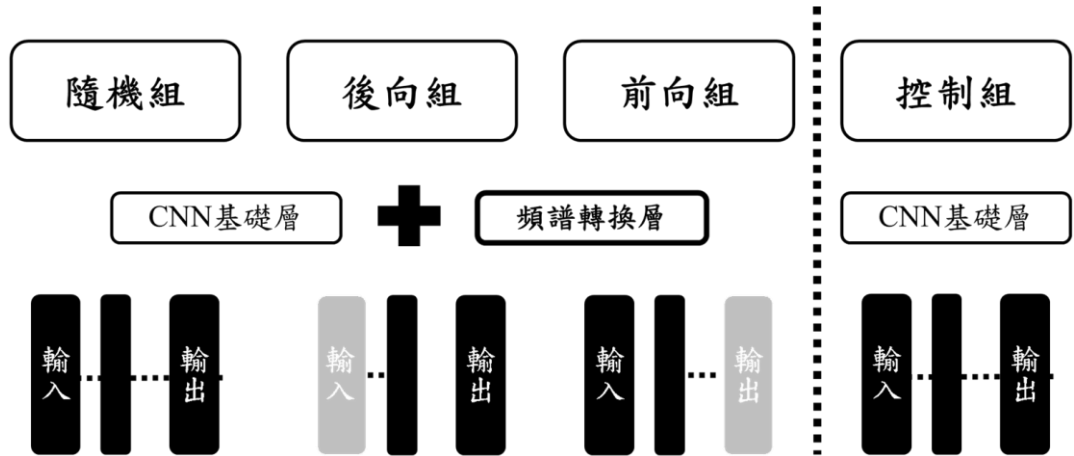


圖 22 分組示意圖

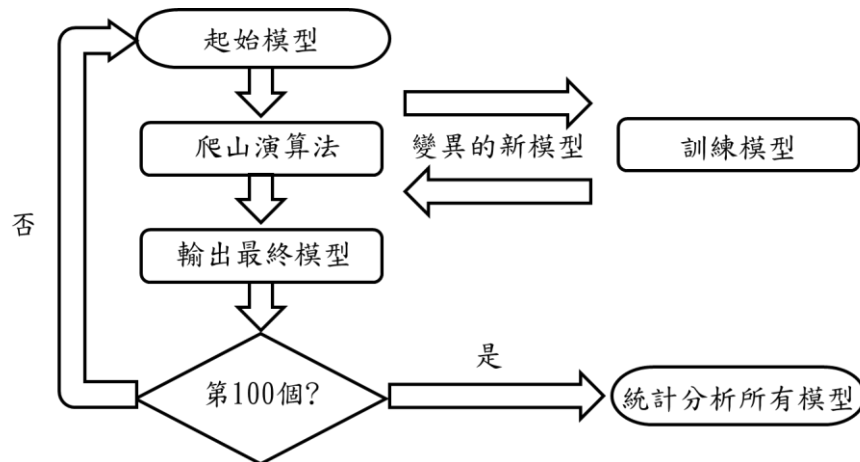


圖 23 實驗流程圖

3.2 網路層類別

本研究的實驗全程採用 Python 的框架 Pytorch 完成，初始模型涵蓋輸入以及一個 Flatten 層（輸出層），作用上會將任意維度的輸入強制展開成一維輸出，輸入層通道數皆為 28×28 ，輸出通道數皆為 10，全程限制自動構築在初始模型內進行，每次隨機選取任一種進行網路層置換或者添加的行為，初始模型的所有層皆為保留層不被行為影響，可以選用的網路層如下：

1. ReLU：對輸入進行激勵函數的運算，輸入通道數上由框架自行調整。
2. Linear：好比一字排開的單一神經元，輸入數目 N 的資訊進行運算後以數目 M 的輸出，意義上等同線性方程 $Nx = M$ ， x 為權重，即線性層。
3. 二維卷積層 (Con)：對輸入進行卷積運算。
4. 平均池化層 (Pool)：考慮到倘若採用最大池化雖然會強化較優勢的特徵，卻也會有過度偏頗的可能，於是採用平均池化層，讓影響力次之和較小的特徵也能參與。
5. Linear + ReLU：為各式卷積神經網路常見的組合，稍微減少搜索空間。
6. 卷積層 + 平均池化層 (Con + Pool)：為各式卷積神經網路常見的組合，稍微減少搜索空間。

對於超參數的部分，卷積核大小有 8、16，通道數目則有 1、2、4、8、16、32，每次皆隨機選取，倘若逢維度不符則放棄並重新選取網路層。六個項目網路層的英文簡寫皆延用於第四章的模型構建過程。

3.3 爬山演算法

本研究設定總代數極限為 10000 代、單一代模型維持連續 50 代且成績未被超越就提前輸出模型，從初始模型開始，分為加入或者取代網路層兩種動作，在初始模型的輸出入層之間，層級的選擇和序位皆隨機，以有限數量的網路層種類和通道數中隨機擇一，然後訓練所產生的新模型，接著從測試集得到該模型的準確度決定是否取代原模型，如此反覆爬山演算法直至找到最終模型，如流程圖 24。

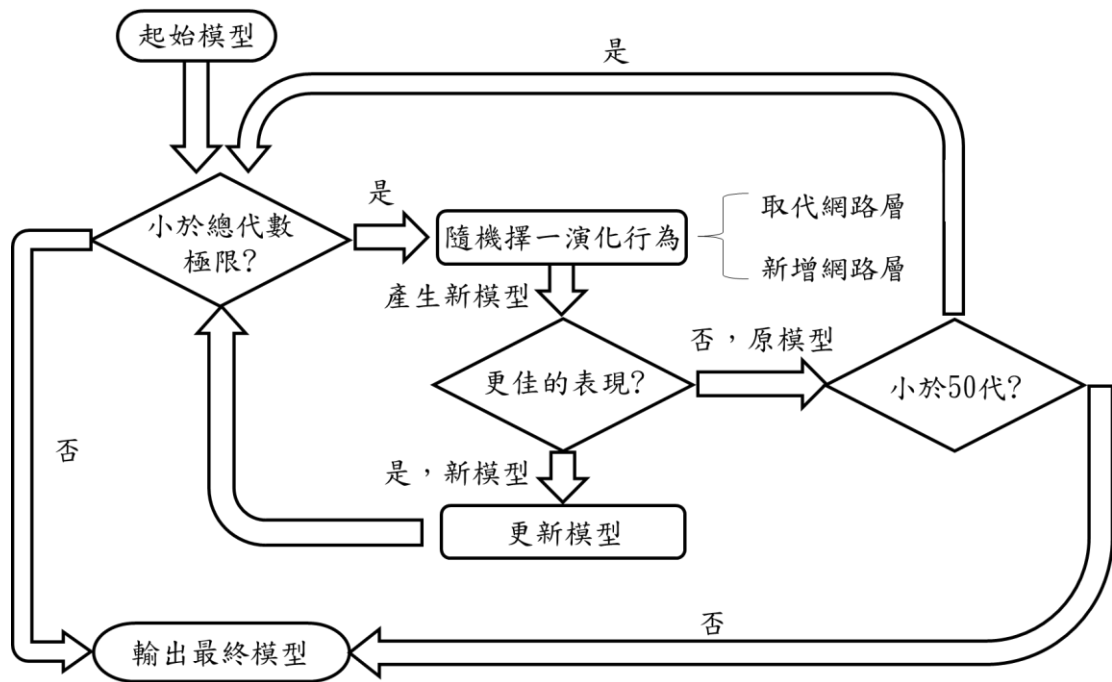


圖 24 爬山演算法流程圖

3.4 頻譜轉換層

由文獻[17]所釋出的開源程式碼，以離散餘弦轉換和離散傅立葉轉換的形式，本質上類似文獻[18]的空間轉換網路，引介圖像或特徵圖進行矩陣轉換成頻譜圖。此外，文獻[19]發現該形式能引導特徵有更加密集的表现。通常用於不規則圖案之特徵提取，像是組織結構和其他醫學相關圖像檢測。過程中所有轉換皆為可微分，對於有正常梯度下降基礎的演算法中皆可被訓練，因此能直接合併進正常的卷積神經網路的主框架裡，優勢在於它需要的專家知識較少，能自動調整到給定的輸入域，讓神經網路學習更有效。本研究採用其中的離散餘弦轉換和離散傅立葉轉換兩種網路層作為額外加入的選項，檢視產生出的卷積神經網路的辨識表現。在之後的模型構件中分別以 Dct、Dft 表示。

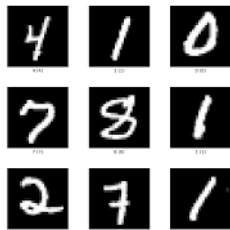
第四章 研究結果和分析

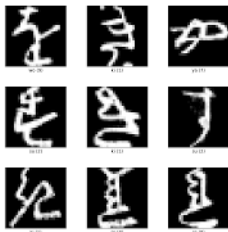
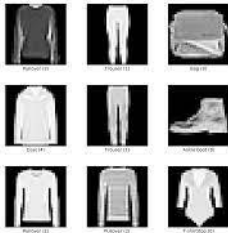
本研究針對手寫文字辨識訓練卷積神經網路，設定初始模型做為起點，採用爬山演算法在多處隨機且有限的選擇中讓模型自動構築。並添加兩種頻譜轉換層成效。資料集上採用三種手寫文字辨識做測試。延伸實驗探討立體內容的圖像是否也具有相同的成效。依據構築方式分為三組與未有頻譜轉換層的控制組，每組分別構築出模型 100 個，以盒狀圖檢視準確度分布，三條粗黑橫線由上往下表示該組最大值、中位數、最小值，盒子的上下界分別表示序位 75 % (三分位數) 與 25 % (一分位數) 的準確度，再從中選出準確度最高的模型作為該組代表，找到該資料集表現最佳的模型，觀察模型成長過程，亦會檢視各組參數數量、準確度的分布情形等等統計結果。

4.1 資料集簡介

MNIST 是深度學習最常用的資料集，為手寫數字 0 到 9 的標準資料集，包含大小為 28×28 像素的灰階圖片，有 60000 張用於訓練的圖片和 10000 張用於測試的圖片。以手寫資料集 MNIST 為基礎下，[15]在 QMNIST 預處理上盡可能接近 MNIST 的製作步驟，提供額外 50000 個模仿 MNIST 所產生的測試圖片集。KMNIST (Kuzushiji-MNIST) 則是古日文的數據集，由[16]於 2018 年提出，選擇一個字符來代表 10 行平假名中的每一行，有 60000 個訓練圖片和 10000 個測試圖片。Fashion-MNIST 亦用於 10 種不同類型的時尚配飾圖片，從 T-shirt 到腳踝靴一應俱全，有 60000 個訓練圖片和 10000 個測試圖片，與前述資料集差別在於，該資料集圖片內容屬於三維的平面影像。範例圖彙整如下表 3。

表 3 資料集範例表

資料集名稱	資料集範例圖
MNIST、QMNIST	

KMNIST	
Fashion-MNIST	

4.2 MNIST

如圖 25 所示，額外添加網路層的前三組在整體準確度上明顯勝過控制組，顯示頻譜轉換層不僅僅作用在表現最佳的單一次模型上。準確度分布上相較控制組來的零散，僅隨機組和前向組的盒子部分較為靠前。最後由隨機組產生出最佳表現的卷積神經網路，以最低參數數量 4714 且準確度高達 96.63 %。

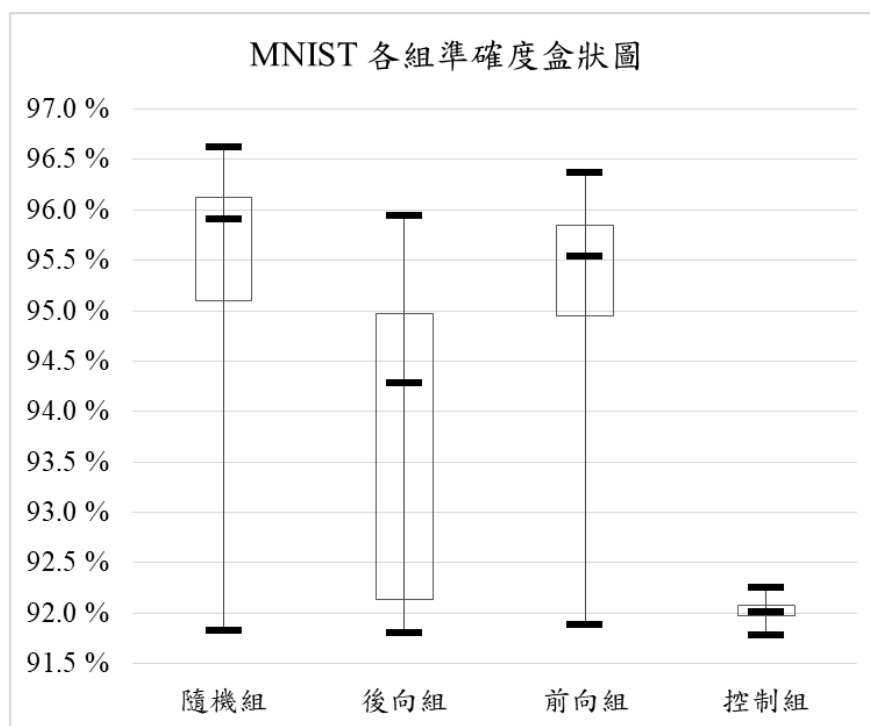


圖 25 MNIST 各組準確度盒狀圖

在各組綜合統計結果方面，參數數量平均最高組別為後向組的 13549.68，最低組別為前向組的 7191.44。控制組具最低參數數量標準差，同上圖顯示分布相對其餘組別密集，前三組準確度平均較控制組提升了 3.28 %、1.81 %、2.92 %，變化最高的組別為隨機組。前三組準確度的標準差則相近，皆大於 1.3。證實頻譜轉換層能夠有效提升準確度，示於表 4。

表 4 MNIST 各組綜合統計結果

	隨機組		後向組		前向組		控制組	
	平均值	標準差	平均值	標準差	平均值	標準差	平均值	標準差
參數數量	7904.88	5427.14	13549.68	5636.85	7191.44	4853.05	7850.00	0.00
準確度	95.30	1.37	93.83	1.33	94.94	1.39	92.02	0.08

此外，在取得神經網路模型的同時，由於自動構築的過程中，選擇上具隨機性，單憑表 4 和圖 25 就決定產生模型尚缺完整性，畢竟大家都希望能在最低的成本下以最少的時間找到最高的準確度作為最佳結果。然而對於分布結果的訊息掌握固然也就相當重要，因此彙整本資料集四組 100 個模型的參數數量分布和佔比於表 5，方便協助決策。畢竟針對實際情形，有時必須選擇盡可能少的參數數量，減少計算成本；有時必須最大化準確度，提升效能。兩者之間由於呈現似邊際成本效應的曲線而非線性關係，因此未能夠持續的相輔相成。

隨機組的部分，參數數量占比最大宗為 4714 (62 %)、其次是 18826 (19 %)，具最多種類。後向組最高占比就落在 18826 (53 %)、其次是 7850 (32 %)，有三種可能的參數數量。前向組參數數量也和後向組相同，與前兩組相較，相對容易碰上參數數量 4714，佔比高達 7 成。而控制組參數數量則幾乎完全落在 7850 (100 %)，意味著要取得相異參數數量可能不是那麼容易。

表 5 MNIST 各組參數數量統計

隨機組		後向組		前向組		控制組	
參數數量	百分比	參數數量	百分比	參數數量	百分比	參數數量	百分比
4714	62%	7066	15%	4714	70%	7850	100%
5106	2%	7850	32%	7850	16%		
7066	4%	18826	53%	18826	14%		
7850	13%						
18826	19%						

接著觀察最佳模型的構築過程，見下表 6，從初始模型（代數 start）開始，歷經 8 次成長，如圖 26 所示，期間在代數 15 到 17 之間，準確度的增幅（2.23 %）較為陡峭，其中經歷將激勵層取代成卷積層，括號內的數字表示為通道數。看來離散傅立葉轉換層之後似乎緊接卷積層的話對於整體卷積神經網路具相當大的助益。不僅如此，該模型在本資料集不但參數數量最少且準確度也最高，符合理想的最佳模型要求。

表 6 MNIST 最佳模型構築過程

MNIST Best Model of CNN	
代數	模型
start	Flatten
1	(Con+ Pool) (16) + Flatten
11	Dft + (Con+ Pool) (16) + Flatten
15	Dft + ReLU + (Con+ Pool) (16) + Flatten
17	Dft + Con (16) + (Con+ Pool) (16) + Flatten
44	Pool + Dft + Con (16) + (Con+ Pool) (16) + Flatten
53	Pool + (Con+ Pool) (32) + Dft + Con (16) + (Con+ Pool) (16) + Flatten
66	Pool + (Con+ Pool) (32) + Dft + Con (16) + (Con+ Pool) (16) + ReLU + Flatten
73	Pool + (Con+ Pool) (32) + Dft + Con (16) + Con (16) + (Con+ Pool) (16) + ReLU + Flatten

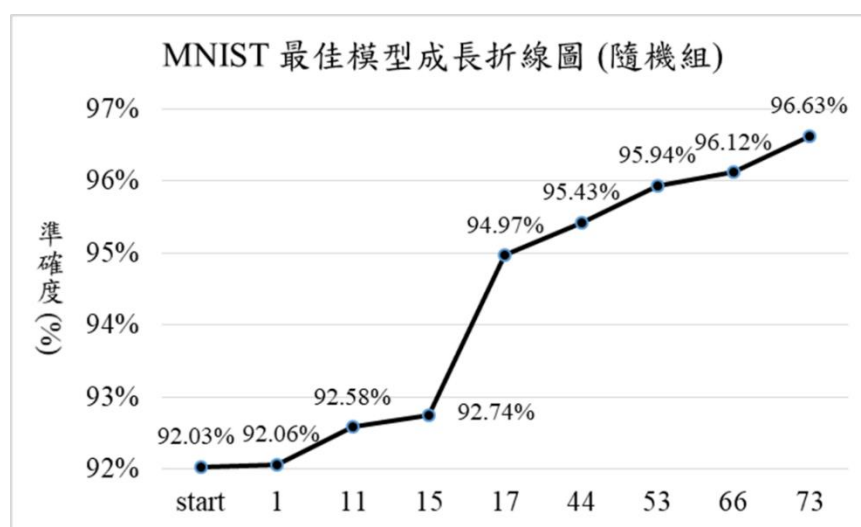


圖 26 MNIST 最佳準確度模型成長折線圖

4.3 QMNIST

如圖 27，額外添加網路層的前三組在整體準確度上高於控制組，綜合前三組的準確度範圍，和前資料集的前三組分布趨近，最大差異在後向組的部分，後向組盒子範圍的分布上相對較為集中。準確度分布上，隨機組和前向組的盒子區域依舊較為靠前，分別佔據第一和第二。最後再次由隨機組產出的卷積神經網路有最佳表現，以最低參數數量 4714 且高達 96.26 % 的準確度。

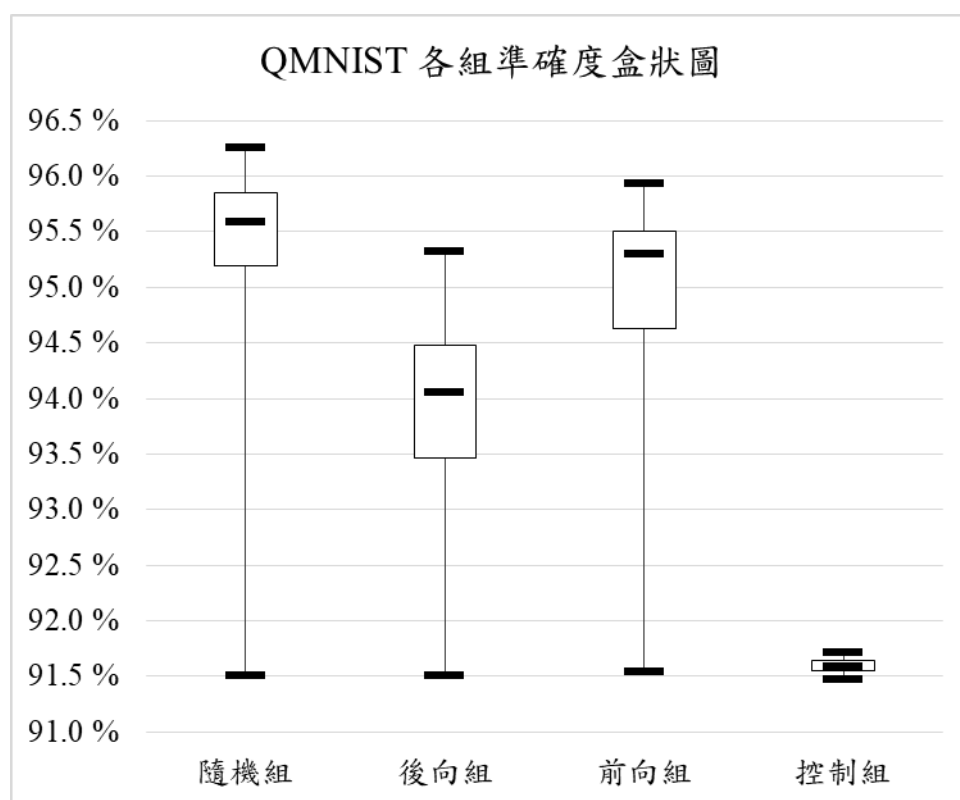


圖 27 QMNIST 各組準確度盒狀圖

在各組綜合統計結果方面，如下表 7，參數數量平均最高組別為後向組的 15219.6，最低組別為前向組的 6909.2。控制組仍同上圖 27 顯示分布相對其他前三組緊縮，平均參數數量 7850，在本資料集位列第二高。前三組準確度平均較控制組提升了 3.6 %、2.1 % 以及 3.06 %，前向組為準確度標準差最高的組別，雖然該組在平均參數數量上最少，準確度標準差卻高達 1.41，而隨機組和後向組準確度的標準差則僅差 0.04，都在 1.1 以上，兩組相當接近。說明頻譜轉換層能夠有效提升準確度。

表 7 QMNIST 各組綜合統計結果

	隨機組		後向組		前向組		控制組	
	平均值	標準差	平均值	標準差	平均值	標準差	平均值	標準差
參數數量	7293.36	5140.78	15219.60	5288.08	6909.20	4567.39	7850.00	0.00
準確度	95.20	1.18	93.70	1.14	94.66	1.41	91.60	0.06

彙整本資料集四組共 400 個模型的參數數量分布和佔比於表 8。隨機組的部分，參數數量占比最大宗為 4714 (73%)、其次是 18826 (16%)，具四個種類。後向組最高占比就落在 18826 (68%)、其次是 7850 (20%)，有三種可能的參數數量。前向組亦同，與前兩組相較，最容易遇上的參數數量是 4714，佔比已經超越了 7 成。而控制組參數數量則依舊幾乎完全落在 7850 (100%)。

表 8 QMNIST 各組參數數量統計

隨機組		後向組		前向組		控制組	
參數數量	百分比	參數數量	百分比	參數數量	百分比	參數數量	百分比
4714	73%	7066	12%	4714	72%	7850	100%
7066	3%	7850	20%	7850	16%		
7850	8%	18826	68%	18826	12%		
18826	16%						

和前資料集相比，兩者其實具有部分相同的圖像，其中 QMNIST 較 MNIST 多五萬筆圖像，其原因在於過去 MNIST 產生圖像的預處理步驟已遺失，尚未有可靠的方式能夠將資料集的字符與作者的 ID 相連結，同時也面臨對於測試集資料過於稀少，以至於無法提供有效的信賴區間的評論，因此後來紐約大學的兩位研究人員憑藉再度從 NIST-19 取得圖像，造就 QMNIST 的誕生，並嘗試仿效 MNIST 的重建，最後額外延伸出了五萬筆的測試集圖像，讓歷史長遠的手寫數字辨識資料集更加完整。

接著檢視在資料集的最佳模型的構築過程，見下表 9，從初始模型（代數 start）開始，歷經 5 次變化，配合圖 28 檢視，期間在代數 1 到 17 之間，準確度有連續兩度幅度較陡的提升 (2.62% 和 1.46%)，其中經歷離散傅立葉轉換層的加入、池化層的介入。池化層緊接卷積層其後的子結構相當常見，如 17 號轉 38 號的階段，僅在通道數上有所差異，雖不及前幾次的巨幅增長，僅僅做了微小的進步，在最後加入激勵層，形成卷積神經網路常見的子架構。

表 9 QMNIST 最佳模型構建過程

QMNIST Best Model of CNN	
代數	模型
start	Flatten
1	Con (2) + Flatten
4	Con (2) + Dft + Flatten
17	Con (2) + Pool + Dft + Flatten
38	Con (2) + Pool + Dft + (Con + Pool) (16) + Flatten
77	Con (2) + Pool + Dft + (Con + Pool) (16) + ReLU + Flatten

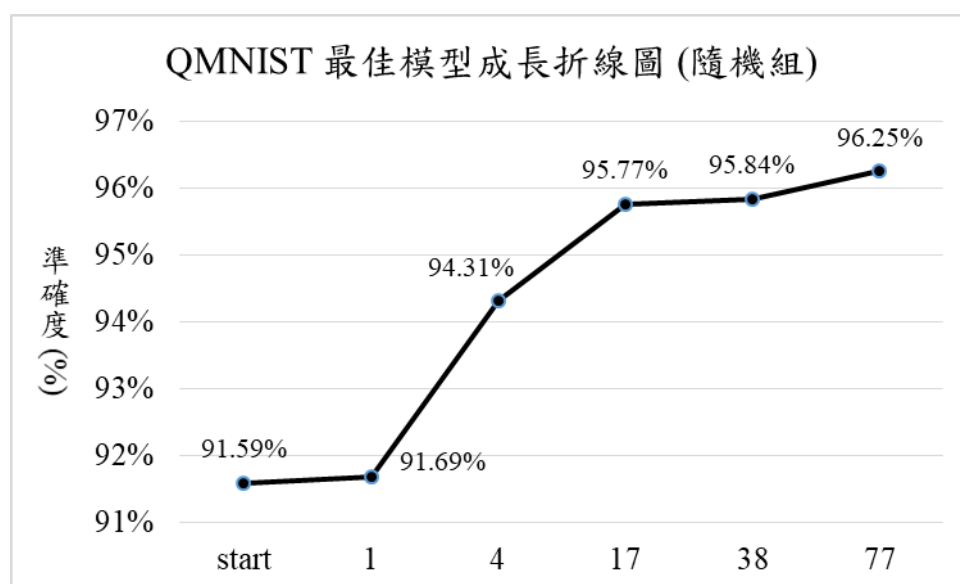


圖 28 QMNIST 最佳模型成長折線圖

4.4 KMNIST

如下圖 29 展示，在古日文的手寫文字辨識集，額外添加網路層的前三組整體準確度亦也高於控制組。前三組的準確度涵蓋範圍相當，分布上也未有甚大懸殊，中位數相當接近，並且以後向組居冠，該組連同的盒子範圍分布上相對最為緊密，區間差為 1.23 %。隨機組和前向組整體分布仍然維持密集且靠前，分別排在第二和第三。在本資料集下，雖然分數普遍下降了，由後向組產出的卷積神經網路有最佳表現，參數數量 18826 且高達 81.54 % 的準確度。

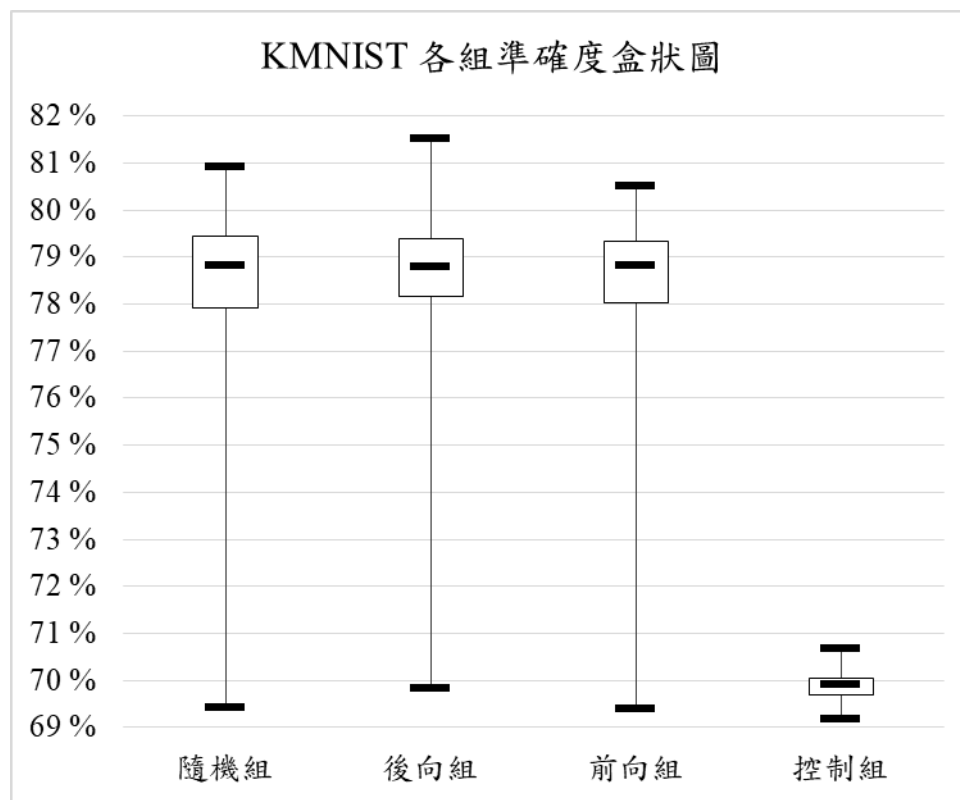


圖 29 KMNIST 各組準確度盒狀圖

KMNIST 在各組統計結果列於下表 10，控制組直到現在很有趣的是，其參數數量不斷維持在固定的數字（綜合前面資料集，準確度標準差分別為 0.08、0.06 和 0.25），推測若要探索出其餘種類的參數數量恐怕具挑戰性，幾乎完全是不斷嘗試以參數數量 7850 的神經網路去完成任何資料集的挑戰，如此單一的數值理所當然在準確度突破上更加難為。

況且回過頭檢視有額外加入網路層的前三組，整體上前三組的平均參數數量皆超過一萬，在參數數量類別數相似的前提下，顯示分布機率有所變化。隨機組具最低的平均參數數量與最高的參數數量標準差，搭配上圖 29 觀察，該組的盒子範圍有最大的區間，以結果回過頭去預估，凸顯出在該組產生的神經網路模型有高機率是落在盒子範圍內的。後向組盡管在平均參數數量排序上偏後，但在平均準確度上卻是領先四組。除此之外，在本資料集，最高準確度和最少參數數量的模型皆在該組，並且未合併為唯一模型。前向組有最高的平均參數數量，但是有最低的標準差，分布上最容易趨近平均值，平均準確度為前三組最低，與隨機組相近。準確度標準差卻是最大的，由於前三組的盒子範圍接近，因此在取得高準確度的卷積神經網路模型上亦相仿。

表 10 KMNIST 各組綜合統計結果

	隨機組		後向組		前向組		控制組	
	平均值	標準差	平均值	標準差	平均值	標準差	平均值	標準差
參數數量	10609.68	6793.28	11221.20	6946.03	13181.20	6719.32	7850.00	0.00
準確度	78.06	2.65	78.41	2.13	78.01	2.74	69.89	0.25

如表 11。首先在隨機組，參數數量占比先是 4714 (52 %)、其次是 18826 (40 %)、最後是 7850 (8 %)，相較前述資料集，4714 的比重下滑了近 2 成，前兩名占比更為均衡。後向組參數數量占比首先是 4714 (50 %)、其次是 18826 (45 %)、最後是 7850 (5 %)，分布上相仿，前兩名顯然也更加對立。前向組則最大參數數量 18826 (58 %) 佔 6 成的比重，卻未有更佳的單位參數的準確度。

表 11 KMNIST 各組參數數量統計

隨機組		後向組		前向組		控制組	
參數數量	百分比	參數數量	百分比	參數數量	百分比	參數數量	百分比
4714	52%	4714	50%	4714	33%	7850	100%
7850	8%	7850	5%	7850	9%		
18826	40%	18826	45%	18826	58%		

接著講述最高準確度模型的構築過程，展示於下表 12，參數數量 18826，精確度高達 81.54 %，從初始模型（代數 start）開始，歷經 5 次轉變，搭配圖 30，期間在代數 1 到 6 之間，因離散傅立葉轉換層的加入，準確度有幅度較陡的進展，在前期都能有效為準確度增幅 (6.27 %)。而後在 8 號轉 20 號的階段，卷積層 + 池化層的組合層代換為激勵層，準確度也有相當的進步 (3.93 %)。

表 12 KMNIST 最高準確度構建過程

KMNIST Highest Accuracy Model of CNN	
代數	模型
start	Flatten
1	(Con + Pool) (4) + Flatten
6	Dft + (Con + Pool) (4) + Flatten
7	Con (1) + Dft + (Con + Pool) (4) + Flatten
8	Con (1) + Dft + Con (4) + (Con + Pool) (4) + Flatten
20	Con (1) + Dft + Con (4) + ReLU + Flatten

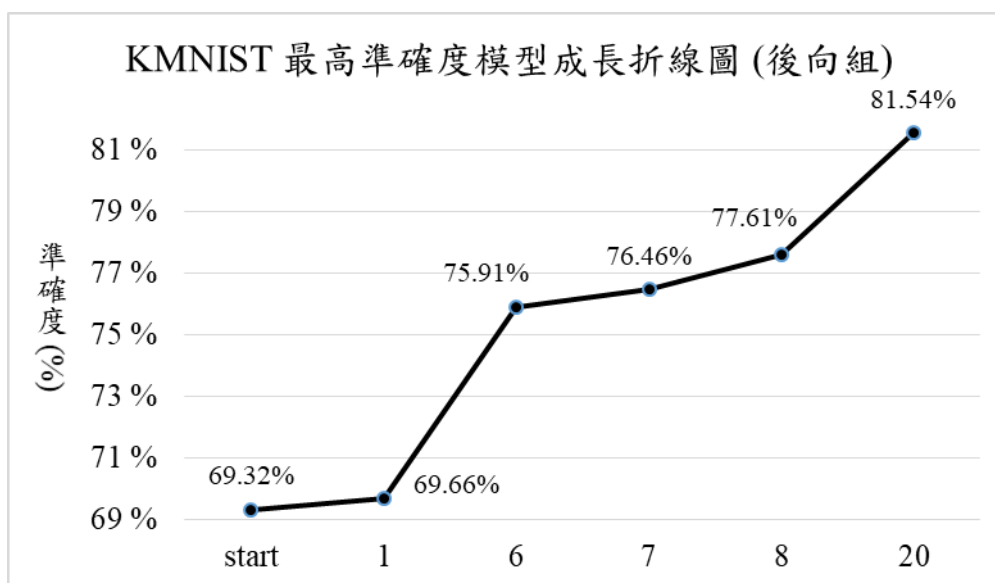


圖 30 KMNIST 最高準確度模型成長折線圖

最後展示最佳卷積神經網路模型的構築過程，見下表 13，參數數量 4714，精確度高達 80.59%，歷經了 5 次演變，配合圖 31 檢視，從初始模型（代數 start）到 1，在離散傅立葉轉換層剛加入，準確度直接提升 (6.49%)。之後連續兩度添加激勵層，但準確度並未有清楚的變化，後續在 29 到 31 階段，加入卷積層在兩激勵層中間，精準度就提升了不少，推測連續的激勵層組合在提升準確度上並不合適。最後在 31 到 71 階段，開頭加入了池化層，準確度提升亦也相當微小，估計池化層不適用於離散傅立葉轉換層之前。

表 13 KMNIST 最佳模型構建過程

KMNIST Best Model of CNN	
代數	模型
start	Flatten
1	Dft + Flatten
28	Dft + ReLU + Flatten
29	Dft + ReLU + ReLU + Flatten
31	Dft + ReLU + Con (32) + ReLU + Flatten
71	Pool + Dft + ReLU + Con (32) + ReLU + Flatten

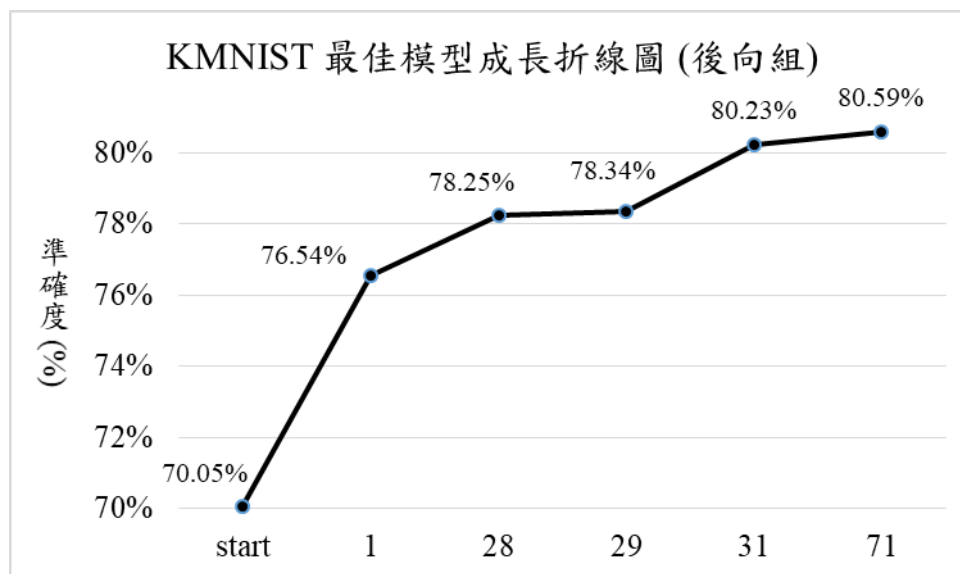


圖 31 KMNIST 最佳模型成長折線圖

4.5 Fashion-MNIST

前述的三種資料集，排除手寫文字辨識的特色外，圖像內容皆為平面，並且是在幾種有限的顏色下，幾乎由筆畫跟背景的面積占據幾近整張圖。而 Fashion-MNIST 資料集為立體內容，就是要延伸原來的轉換層圖像辨識效果。如圖 32，垂直軸間距為 0.2%，四組的盒子位置、範圍彼此相當接近，中位數趨近一致。整體涵蓋範圍以前向組最為集中，特別是控制組，不再是屈居於最底層的角落。

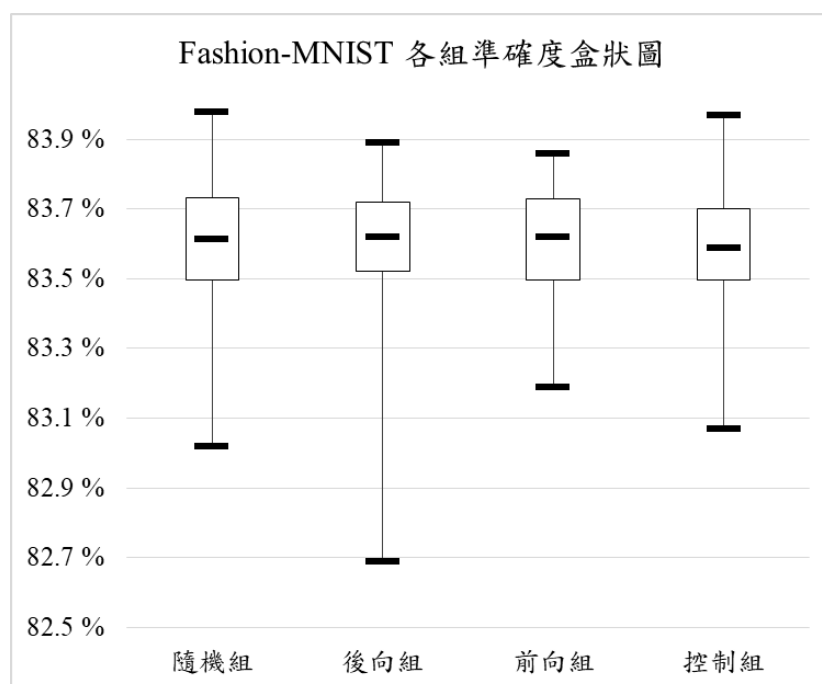


圖 32 Fashion-MNIST 各組準確度盒狀圖

依照表 14、15 所示，在本資料集，針對立體圖像內容，四組皆只有唯一的參數數量 7850，未有採納其餘參數數量，最大區間的後向組也不過 1.12 %，得返回檢視為何準確度尚未能有所突破，看起來應該是似乎在架構上因為過早完成訓練就輸出模型，並未花費更多的代數去探索出表現更佳的模型，而且也沒有見到更少或者更多的參數對於該資料集較差之類的表現，整體訊息量過少，如圖 32，四組的最小值組成的浮動範圍廣 (0.5 %)，表示彼此相當懸殊，然而卻在最大值彼此相近如同被天花板擋住般。於是乎推測兩個方向，一個為輸出模型的門檻過早，並未有足夠多的嘗試機會；另一方面則是假如提高輸出門檻但依然存在現況，意味著兩種轉換層對於立體內容的分辨受到影響，猜測是因為立體內容的透視感造成，而圖像終究只不過是圖中立體物件的特定角度投影，相較於平面圖像的輪廓面積，由於投影的關係，不清晰或者漸層的輪廓面積反而造成分析特徵上的障礙。

表 14 Fashion-MNIST 各組綜合統計結果

	隨機組		後向組		前向組		控制組	
	平均值	標準差	平均值	標準差	平均值	標準差	平均值	標準差
參數數量	7850.00	0.00	7850.00	0.00	7850.00	0.00	7850.00	0.00
準確度	83.598	0.182	83.591	0.184	83.601	0.159	83.582	0.172

表 15 Fashion-MNIST 各組參數數量統計

隨機組		後向組		前向組		控制組	
參數數量	百分比	參數數量	百分比	參數數量	百分比	參數數量	百分比
7850	100%	7850	100%	7850	100%	7850	100%

綜合表 16 以及圖 33 研判，門檻過低的情形再次浮現，初始模型歷經唯一一次改變後，僅加入卷積層和池化層，準確度上升 0.57 %，此後再也沒有突破過。同表 15，四組平均準確度都在 83.59 % 上下，尤其隨機組與前向組幾近相同。

表 16 Fashion-MNIST 最高準確度構建過程

Fashion-MNIST Best Model of CNN	
代數	模型
start	Flatten
5	(Conv + Pool) (4) + Flatten

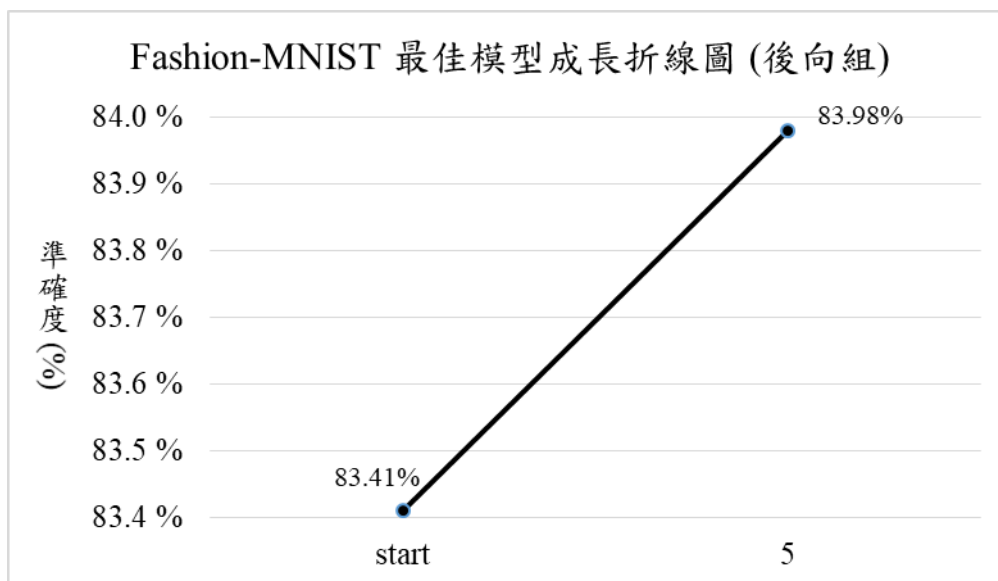


圖 33 Fashion-MNIST 最佳模型成長折線圖

與其他三個資料集綜合看下來，參數數量分布顯得過於單一，僅憑準確度最小值稍微的差異，也只能回推應該是本資料集大部分模型均未獲得充足的構築時間，如同揠苗助長，過早就輸出最終的卷積神經網路模型，未能從零碎的結果中找到更豐富的分析資訊。

4.6 本章總結

綜合前面 4.1 至 4.4 節，實驗設定為在 10000 代內，若有任一模型連續維持超過 50 代就視為成功的模型，並且提前輸出該卷積神經網路。以初始模型開始，輸入爬山演算法嘗試作多次更迭，在種數有限的網路層和通道數中隨機選擇參數配置，如此反覆搜索直至找到最終模型，依據三種開放手寫字符資料集 (MNIST、QMNIST、KMNIST) 與額外的 3D 圖像資料集 (Fashion-MNIST) 訓練並測試模型。每個資料集下，依據構築方式差異分為四組 (隨機組、後向組、前向組) 與控制組，每組分別執行 100 次後統計並分析，僅前三組分別在網路層的選擇上額外加入了兩種網路層，分別為離散傅立葉轉換層 (DFT) 和離散餘弦轉換層 (DCT)，皆取自開放原始碼，實驗全程以 Pytorch 實作。

四種資料集執行下來，三種開放的手寫字符資料集 MNIST、QMNIST、KMNIST 均在離散傅立葉轉換層 (DFT) 幫助下準確度突破不少，雖然未看到離散餘弦轉換層 (DCT) 於任一最佳模型的蹤跡，估計是在同組的其他模型中潛藏著，並且隱含於最後未能在準確度上取得領先的數個模型裡。但是在最後的立體圖像資料集 Fashion-MNIST，無論是否被選入，四組均為平手，無從直接性的判定各組優劣與否，視覺化結果顯示模型普遍均未獲得充足的探索代數，以至於未能完整檢視是否能有更多元的精確度、參數數量，最終不足以呈現出加入網路層與否的效能差異。

第五章 研究結論

本研究的實驗為在累計的極限代數內，若有任一代的模型連續佔據當前最高準確度模型，並超過門檻代數就視為最終輸出的卷積神經網路。自初始模型開始，以爬山演算法作多次訓練並測試，在種數有限的網路層和通道數中隨機選擇參數配置，如此反覆搜索直至找到最終模型，採用三種開放的手寫字符資料集 (MNIST、QMNIST、KMNIST) 與額外的 3D 圖像資料集 (Fashion-MNIST) 訓練並測試模型。每個資料集下依自動構成的差異分為三組 (隨機組、後向組、前向組) 外加控制組，每組分別執行 100 次後統計並分析，前三組分別在網路層的選擇上比控制組額外加入了兩種網路層，分別為離散傅立葉轉換層 (DFT) 和離散餘弦轉換層 (DCT)，皆取自開放原始碼，實驗全程以 Pytorch 實作。

5.1 結論

執行四種資料集後，三種開放的手寫字符資料集 MNIST、QMNIST、KMNIST 均在離散傅立葉轉換層 (DFT) 的幫助下，隨機組、後向組、前向組準確度超越控制組。平均準確度提升如下所示：

- MNIST：分別提升了 3.28 %、1.81 %、2.92 %。
- QMNIST：分別提升了 3.6 %、2.1 % 以及 3.06 %。
- KMNIST：分別提升了 1.17 %、1.52 %、8.12 %。

並且在資料集 MNIST 和 QMNIST 表現上，最高準確度模型還兼具參數數量最少的優勢，兩個模型在成長曲線上，皆是採納開始和最後走緩，但中間較陡的形式，具有相同的特性。顯示加入的轉換層對於整體準確度有所提升，並且可以在數個模型間找尋更佳的層次組合，協助人為設計。

然而在最後的延伸實驗，立體圖像資料集 Fashion-MNIST 就沒有如此佳績，四組均為平手，缺乏直接性的關鍵判定優劣與否，眾多視覺化結果普遍顯示模型均未獲得充足的探索代數，以至於架構限縮，導致未能得到更多元的精確度、參數數量，最終讓延伸實驗不足以呈現出加入網路層與否的效能差異。

5.2 未來研究建議

本研究的延伸實驗未能成功找到更多因素作分析，針對立體圖像內容學習上提高輸出門檻的窘境，假如提高輸出門檻但依然存在現況，意味著兩種轉換層對於立體內容的影像分辨有問題，是否因為立體內容的透視感影響，畢竟圖像終究只不過是圖中立體物件特定角度的投影，相較於平面內容圖像的輪廓，傅立葉轉換採用的是分解圖像頻率的原理，由於立體投影的關係，好比不清晰的輪廓或者漸層的區域而形成分解上有過多雜訊的影響，意義上如同過擬合的神經網路。未來研究也許能夠嘗試對於立體的文字辨識如也有相近的情形，嘗試將立體的圖像內容轉成平面內容，像是二值化、高通濾波處理等等。

檢視其餘三組成功組別，因實驗在多處選擇上均採隨機選擇，未來研究能嘗試不需要到 100 模型亦能找到最佳模型，透過一些最佳化組合的演算法，想必能夠以更少的總執行時間和代數就能找到近似解，同時降低對於機率的影響，讓搜索上更加便捷。

参考文献

- [1] D.E.Rumelhart, G.E.Hinton, and R.J.Williams, “Learning representations by back-propagating errors”, Nature, vol. 323, no. 6088, pp. 533-536, 1986.
- [2] J.Sun, J.C.Patra, and Y.Li, “Functional Link Artificial Neural Network-based disease gene prediction”, Proc.Int.Jt.Conf.Neural Networks, pp. 3003-3010, 2009.
- [3] C.Pedersen, J.Togelius, and G.N.Yannakakis, “Modeling player experience for content creation”, IEEE Trans.Comput.Intell.AI Games, vol. 2, no. 1, pp. 54-67, Mar 2010.
- [4] G.N.Yannakakis and J.Hallam, “Game and player feature selection for entertainment capture”, Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Games, pp. 244-251, 2007.
- [5] Y.LeCun, L.Bottou, Y.Bengio, and P.Haffner, “Gradient-based learning applied to document recognition”, Proc.IEEE, vol. 86, no. 11, pp. 2278-2323, 1998.
- [6] M.Raghul, K.Sharaj, S.Ragul Sankar, and S.Jayanthi Sree, “Deep Learning Technique for COVID 19 Prediction using CT Scan Images”, Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, pp. 1364-1371, Aug 2021.
- [7] K.Simonyan and A.Zisserman, “Very deep convolutional networks for large-scale image recognition”, 3rd International Conference on Learning Representations, 2015
- [8] K.He, X.Zhang, S.Ren, and J.Sun, “Deep residual learning for image recognition”, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-December, 2016.
- [9] B.Baker, O.Gupta, N.Naik, and R.Raskar, “Designing Neural Network Architectures using Reinforcement Learning”, 5th International Conference on Learning Representations, Nov 2016.
- [10] B.Zoph and Q.V.Le, “Neural Architecture Search with Reinforcement Learning”, 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, Nov 2016.
- [11] H.Pham, M.Y.Guan, B.Zoph, Q.V.Le, and J.Dean, “Efficient Neural Architecture Search via Parameter Sharing”, 35th International Conference on Machine Learning, vol. 9, Feb 2018.

- [12] C.Liu et al., “Progressive Neural Architecture Search”, 15th European Conference on Computer Vision, vol. 11205 LNCS, pp. 19-35, Dec 2017.
- [13] X.S.Yang, S.F.Chien, and T.O.Ting, “Bio-Inspired Computation and Optimization: An Overview”, Bio-Inspired Comput.Telecommun., pp. 1-21, Jan 2015.
- [14] C.-C.Chen, “A Hill-Climbing Algorithm to Construct Neural Network Automatically -- tested on MNIST”, open source paper, Oct 2021.
- [15] C.Yadav and L.Bottou, “Cold Case: the Lost MNIST Digits”, 33rd Conference on Neural Information Processing Systems, Canada, 2019.
- [16] T.Clanuwat, M.Bober-Irizar, A.Kitamoto, A.Lamb, K.Yamamoto, and D.Ha, “Deep Learning for Classical Japanese Literature”, Neural Information Processing Systems 2018 Workshop on Machine Learning for Creativity and Design, Canada, 2018.
- [17] M.Alberti, A.Botros, N.Schütz, R.Ingold, M.Liwicki, and M.Seuret, “Trainable Spectrally Initializable Matrix Transformations in Convolutional Neural Networks”, Proceedings - International Conference on Pattern Recognition, pp. 8204-8211, Nov 2019.
- [18] M.Alberti, V.Pondenk and ath, M.Wursch, R.Ingold, and M.Liwicki, “DeepDIVA: A highly-functional python framework for reproducible experiments”, Proceedings of International Conference on Frontiers in Hand writing Recognition,__vol. 2018-August, pp. 423-428, Dec 2018.
- [19] G.Kovács and L.Tóth, “The Joint Optimization of Spectro-Temporal Features and Neural Net Classifiers”, 16th international conference on Text, Speech and Dialogue, vol. 8082 LNAI, pp. 552-559, 2013.