東海大學

資訊工程研究所

碩士論文

指導教授：林祝興博士

利用卷積神經網路與平行運算過濾垃圾郵件之設計與實作

Filtering Spam Mails Using Convolutional Neural Networks and Parallel Computing

研究生：李鼎中

中華民國 107 年 7 月

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 李 鼎 中 所提之論文

利用卷積神經網路與平行運算過濾垃圾郵件之設計與實作

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

召 集 人 _____ 簽章

委 員 _____

_____

_____

指 導 教 授 _____ 簽章

中華民國 107 年 7 月 6 日

誌謝

首先要感謝指導老師林祝興教授，謝謝林老師在學業及論文上的教導，老師在教學及做學問的嚴謹態度，也是學生效法的榜樣。

也要感謝在進修在職專班這幾年，來自在職班同學、實驗室的學長、同學、學弟們的幫助，大家攜手在學業上共同精進。

在此也感謝國立中興大學資訊管理學系蔡垂雄教授、國立彰化師範大學資訊管理學系張隆池副教授、東海大學資訊工程學系劉榮村助理教授在論文口試中給予寶貴的意見，使學生論文更加完善。

最後要感謝我的家人，老婆及兒子、女兒，你們的鼓勵、包容是我在邊工作邊修習碩士學位的力量來源。

摘要

隨著資訊科技的發展以及資訊設備(如個人電腦、平板、手機)的普及，電子郵件已是工作或生活密不可分的溝通工具，並且因為政府業務及金融交易必須綁定能夠代表真實身分的聯絡工具，故電子郵件的重要性無法被取代。也正因為如此，電子郵件也成為廣告行銷、以及電腦駭客、商業間諜、國家間諜等有心人士用於散佈偽冒寄件者(引誘開啟郵件)、惡意程式(木馬程式、勒索病毒)、惡意連結(釣魚網站)、通知重新認證(騙取帳號、密碼)等之主要媒介，若不慎點擊郵件連結或遭受進一步感染，對個人及企業之資訊安全有重大危害。統計任職企業之垃圾郵件過濾系統，每月收到的郵件中，近 25%郵件被歸類為垃圾郵件，但仍有少量信件遭過濾機制漏攔(false negative)或誤攔(false positive)。加州大學爾灣分校(UCI)垃圾郵件開放性資料以及其使用郵件內文特徵(content-based)作為垃圾郵件過濾器之想法(約有 7%誤判率，93%正確率)。本論文以機器學習之類神經網路(neural network)及卷積神經網路(convolutional neural network)演算法實作 UCI 垃圾郵件過濾器之辨識率，也獲得良好的效果。實驗證明經過上述模型訓練後，卷積神經網路獲得更好的結果，可以達到 91%的正確率。此外，為了實務應用上效能的需求，我們加入了 GPU 平行運算，實驗顯示可以得到 4.17 倍的加速比。

關鍵字：垃圾郵件、類神經網路、卷積神經網路、GPU 平行化、加速比

ABSTRACT

Spam emails are the mails included with malicious attachments, such as Trojan horse, ransomware, or mails with malicious links, such as phishing websites. The main purpose of spam emails is to defraud money or steal secrets, which is a major threat to the security of information for individuals and businesses. In this thesis, we use University of California, Irvine's spam email dataset and its content-based spam email filter (with approximately 7% misclassification error, 93% correct rate). By using machine learning of neural network and convolutional neural network model, we compare the accuracy between NNs and CNNs and prove that CNNs has better performance. Finally, by using GPU parallel computing, we further obtain about 4.17 times of acceleration rate.

Keywords：spam emails, machine learning, neural network, convolutional neural network, GPU parallel computing, speedup

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

Chapter 1　簡介

1.1　研究動機

著資訊科技的發展以及資訊設備(如個人電腦、平板、手機)的普及，電子郵件已是工作或生活密不可分的溝通工具。在 2017 年每天發送和接收郵件的數量超過 2250 億封，每個帳號每天收到的 92 封信中，有 16 封是垃圾郵件(約佔 22%)；預估到 2019 年底，每天發送和接收郵件的數量超過 2460 億封，每個帳號每天收到的 96 封信中，有 19 封是垃圾郵件(約佔 20%)，並且全球有超過三分之一的人口使用電子郵件 [1]。

以商業應用而言，電子郵件最大的優點為在對方不需同步進行的情況下，與一個、二個人或多個人交換各種類型的資訊，其成本遠低於傳統會議或電話會議；並且也大量應用於行銷產品，可以在短時間寄送給大量的收件者。缺點則是垃圾郵件的侵害，企業會收到同一封信寄給大批企業內郵件用戶(unsolicited bulk email)，而郵件用戶往往會收到的不想要或不相關的信件(unsolicited commercial email)，降低生產力；商業間諜或電腦駭客常利用電子郵件散佈電腦病毒，藉以騙取帳號、密碼，竊取商業機密，勒索金錢，對個人及企業之資訊安全有重大危害。賽門鐵克公司之網路安全威脅報告指出，在 2016 年，超過 53%的電子郵件為垃圾郵件，且夾帶惡意軟體的郵件比例上升 [2]，而垃圾郵件的最大來源則仍是美國，佔 13.21%，第二到五名依序為中國 (11.25%)、越南 (9.85%)、印度 (7.02%)、德國 (5.66%) [3]。

由於垃圾郵件的氾濫著實降低員工生產力，並且對企業之資訊安全有重大危害，故垃圾郵件過濾系統是保障企業電子郵件不可或缺的一環。任職企業之垃圾郵件過濾系統包含多種過濾機制如 1.病毒、間諜程式過濾。2.應用程式及執行檔過濾。3.郵件內文關鍵字過濾。4.郵件內文連結信譽評等過濾。5.郵件來源 IP 信譽評等參考。6.商品促銷或美容減重等類別關鍵字過濾。7.DoS (denial of service)防禦機制。分析該系統每個月收到的外部郵件，近 25%的郵件被歸類為垃圾郵件，並且其中1%~2%的郵件屬於含有惡意附件、連結之高風險郵件，雖然過濾系統有多達 7 種過濾機制，每月仍有 0.25%~0.5%的郵件為誤攔 (false positive)，如 Figure 1.1、Figure 1.2，也必定會有漏攔 (false negative)的情況(漏攔郵件遭其他資安防護設備發現)。其實垃圾郵件過濾系統出現誤攔或漏攔的狀況是可以理解的，如果要達到零漏攔，則會有更高比例的信件會遭誤攔，因此，垃圾郵件過濾系統也必須在過濾速度及效果取得平衡點，過濾機制越複雜，則可能降低過濾速度。

類別統計 流量比例統計: 2017-01

類別	郵件封數
正常郵件 (Normal mail)	81289 (封)
垃圾郵件 (Spam mail)	23010 (封)
誤攔郵件 (Resend mail)	358 (封)
漏放郵件 (Missed spam)	0 (封)
威脅郵件 (Malicious mail)	620 (封)
其他	
丟棄郵件	1638 (封)
DoS 攻擊阻擋	2247 (封)
RBL 阻擋(連線層)	0 (封)
總量	109162 (封)

Figure 1.1: 垃圾郵件過濾系統之郵件類別信件量

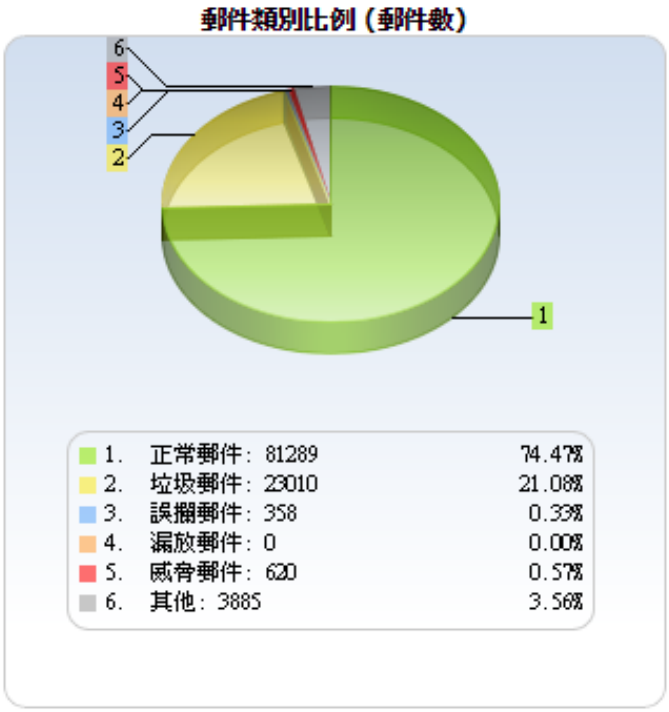


Figure 1.2: 垃圾郵件過濾系統之郵件類別統計

1.2 研究目標

本論文使用加州大學爾灣分校(UCI)之垃圾郵件開放性資料 [4]，共有 4601 筆郵件資料，依照郵件內容中最常出現的字詞、字符、及連續出現的大寫字母長度，是否為 SPAM 等，共分為 58 種屬性，過濾 4601 筆郵件資料各屬性及結果是否為垃圾郵件，結果有近 7%的誤判，正確率約 93%，但可歸納出信件中若出現 George 字眼及區號 650，則是正常郵件的指標，如廣泛收集正常郵件內文特徵的相關指標，即可產生通用性垃圾郵件過濾器 [4]。採用分類演算法如類神經網路(artificial neural network)，支援向量機(support vector machine)以及單純貝氏分類器(naive Bayes classifier)來判斷郵件內文與垃圾郵件的相關性，雖然也有不錯的效果，但若在分類器中加上句子或是公司領域的專有字詞，或是使用兩種或以上演算法搭配

過濾，可以進一步優化正確率 [5, 6]。

機器學習是人工智慧的一個子領域，它使用統計技術為電腦系統提供利用數據「學習」的能力，能夠藉由分析資料進而預測結果。有別於傳統的人工程式判斷，機器學習演算法能從資料中自動分析獲得規律，並利用規律對未知資料進行預測，使機器在經過分析大量的樣本後，輸出可靠的結果。近十多年來由於網際網路普及產生爆炸性的資料量，在圖像資料部分，如 2009 年 ImageNet Project 初期的 320 萬張經過標註的圖像，至 2017 年初，已收集到 1 千五百萬張經過標註的圖像，共有 2 萬 2 千種類別，加上硬體運算效能的大幅進步，更加速優化演算法，明顯地提高了機器學習演算法的預測的正確率。從 ImageNet Large Scale Visual Recognition Challenge(ILSVRC) 2010 到 2016 的競賽結果，機器學習演算法的分類錯誤率已從 2010 年的 28.2%一路下降至 2015 年的 3.57%，甚至優於人類的分類錯誤率 5.1% [7, 8]。機器學習已廣泛應用於資料探勘、電腦視覺[9]、自然語言解析[10]、生物特徵識別、手寫識別和機器人等領域，如自動駕駛車(電腦視覺)、機器翻譯(自然語言處理)、人臉辨識系統(生物特徵識別)、AlphaGo 人工智慧圍棋軟體等。

本論文運用兩種機器學習演算法 -- 類神經網路(neural network)及卷積神經網路(convolutional neural network)，為監督式學習之反向傳播型(backpropagation)類神經網路。透過梯度下降演算法(gradient descent)對神經網路中所有權重計算損失函數的梯度，透過梯度的反饋，自動更新各權重值以最小化損失函數 [11]。藉由多次的訓練及測試結果，模型會自動修正每個神經元的權重，進而提升正確率。其中卷積神經網路透過卷積層(convolutional layer)提取輸入資料特徵，再由池化層(pooling layer)保留特徵並縮小資料大小等架構特色，在圖像識別及自然語言處理之應用上，比其他演算法有更好的效果(正確率或分析速度)。

過度擬合(overfitting)是機器學習中常見的一種現象，主要原因來自於樣本數太少或是訓練次數過高，導致學習結果(分析函數扭曲)過於吻合訓練資料，而造成分辨新資料時出現更多誤判。為了避免這個情況，本論文加入了 dropout regularization [12, 13]，透過刻意忽略部分神經元的權重，可降低發生過度擬合的機會，提高預測正確率。

本論文使用以上述兩種模型分析、統計 UCI 垃圾郵件開放性資料，在使用 CPU 運算之類神經網路演算法訓練 3000 回合(約耗時 4824 秒)，正確率為 84%；卷積神經網路演算法訓練 3000 回(約耗時 1116 秒)，正確率為 96%。此外，考量實務上需求，我們利用 GPU(graphics processing unit)圖形處理器，GPU 過去主要用於處理影像資料運算，由於 GPU 有大量的運算核心，藉由每個核心的獨立運算處理，可大幅加快計算速度。機器學習具有大量的神經元及權重，這些存放在矩陣內的資料計算過程特別適合使用 GPU 平行運算。實驗結果顯示，以類神經網路演算法訓練 3000 回合(耗時僅須 517 秒)，正確率不變，卷積神經網路演算法訓練 3000 回(耗時僅須 319 秒)，正確率亦相同，訓練 30000 回(耗時僅須 3069 秒)，正確率可達 98%以上。由上述實驗證明，採用卷積神經網路演算法來分析郵件資料屬性及是否為垃圾郵件是更好的選擇，並且加入 GPU 平行化運算，可以獲得近 3.13 倍的加速比。

本論文共分為五個章節，第一章簡述研究動機、背景知識及研究結果；在第二章背景知識與相關文獻中詳細說明使用到的技術；第三章介紹本論文的研究方法，及各項方法的運作流程；第四章為研究結果，經統計、分析各項實驗數據，證明研究成效；最後於第五章討論本方法的優劣及未來展望。

Chapter 2　背景知識與相關文獻

2.1　機器學習(Machine Learning)

「機器學習」一詞最早於 1952 年由 IBM 的 Arthur Samuel 提出，Frank Rosenblatt 於 1957 年提出感知機(perceptron)演算法 [14]，但由於此演算法有本質上的缺陷，並無法處理線性不可分的問題。直到 1980 年代開始，新的機器學習演算法陸續被發表，如多層感知機 [15]、加入反向傳播功能的神經網路 [16, 17]、決策樹 [18]、支持向量機 [19]、卷積神經網路 [20]、深度神經網路 [21]。

機器學習是實現人工智慧的一個途徑，即以機器學習為手段解決人工智慧中的問題，依照任務性質可分為監督式學習(supervised learning)及非監督式學習(unsupervised learning)兩大類。隨著電腦的運算能力與儲存能力的快速進展，機器學習在近 30 多年已發展為一門多領域交叉學科，涉及機率論、統計學、逼近論、凸分析、計算複雜性理論等多門學科。機器學習理論主要是設計和分析一些讓電腦可以自動「學習」的演算法。機器學習演算法是一種從資料中自動分析獲得規律，並利用規律對未知資料進行預測的演算法，如決策樹、類神經網路及卷積神經網路等。

近十多年來由於網際網路普及產生爆炸性的資料量，滿足了發展機器學習的資料需求，加上硬體運算效能的大幅進步，更驅使加速優化演算法，明顯地提高了機器學習演算法的預測率，進而被廣泛應用於電腦視覺 [9]、自然語言解析 [10]、生物特徵識別、DNA 序列比對、語音和手寫識別，以及智慧機器人等領域。

2.2 類神經網路(Neural Network)

人工神經網路(artificial neural network)簡稱神經網路(neural network)或類神經網路，是一種模仿生物神經網路的結構和功能所產生的數學模型。可藉由程式設計及電腦的快速計算能力實作，產生經過訓練、學習後具有推論結果能力的人工智慧機器。Frank Rosenblatt 於 1957 年即提出可以模擬人類感知能力的機器，稱之為感知機(perceptron) [14]，但由於此演算法有本質上的缺陷，無法處理線性不可分的問題。

類神經網路通常由多組神經元所組成，共可分為輸入層、隱藏層以及輸出層，輸入層負責接收輸入的數值，這些數值會和神經元內的參數進行運算，參數通常包含權重(weight)與偏量(bias)，權重可代表該神經元的價值，權重越高則表示該神經元越能影響整個模型的結果。隱藏層是整個類神經網路的核心所在，隱藏層的功能主要是增加類神經網路的複雜性，以能夠模擬複雜的非線性關係，隱藏層可以是一個或數個，隱藏層過多會增加運算的複雜度，且提升的效果有限，本論文中採用三個隱藏層。輸出層是整個模型運算的結果，輸出形式通常為線性回歸或分類，其中線性回歸可以用於趨勢分析，而分類則應用於圖像識別。

本實驗模型為監督式學習之反向傳播型(backpropagation)類神經網路，透過梯度下降演算法(gradient descent)對神經網路中所有權重計算損失函數的梯度，透過調整學習率(learning rate)數值以控制權重更新的速度，各權重值藉自動更新機制以最小化損失函數 [11]。藉由多次的訓練及測試結果，模型自動修正每個神經元的權重，進而提升準確度，運作流程如 Figure 2.1。

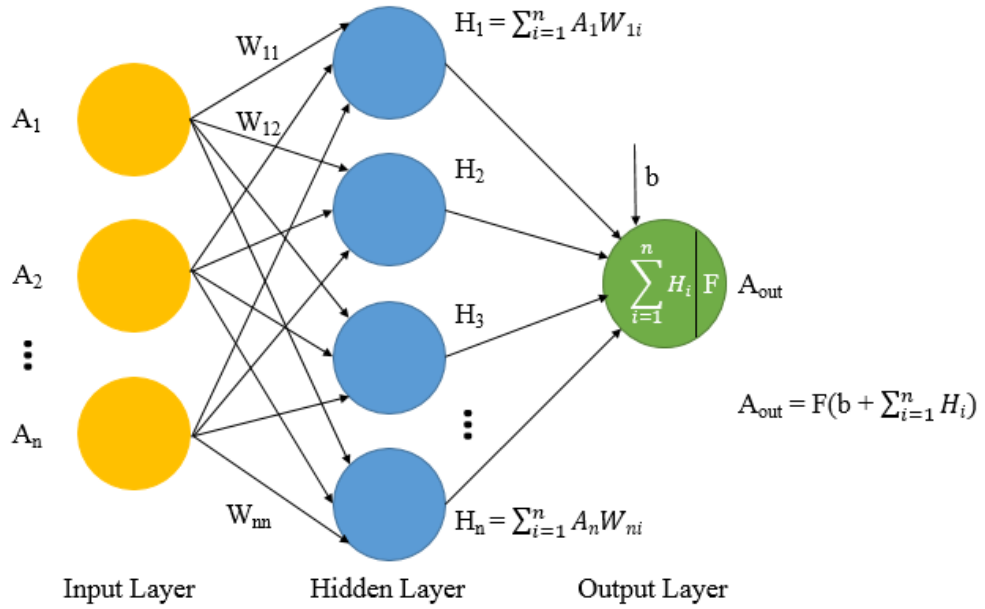


Figure 2.1: 類神經網路運作流程示意圖

輸入資料(神經元)共有 n 個，$A_{1,\ldots,}A_n$，每個輸入資料與權重之乘積和與偏量(bias)相加(結果暫定為 S)，F 為激勵函數(activation function)，利用線性或非線性的函數將 S 轉換成所需要的結果(Aout)。

激勵函數是神經網路的重要特點，主要功能在於將原本前一層輸入到下一層輸出(矩陣相乘)的線性關係，透過激勵函數得到非線性的預測結果，因此能解決早期類神經網路無法解決的預測 XOR 問題。本論文所採用的三個激勵函數，其定義如 Eq.(1)、Eq.(2)、Eq.(3)：

$$f(x) = \frac{x}{(1 + |x|)} \tag{1}$$

$$f(x) = \max(x, 0) \tag{2}$$

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad (3)$$

Eq.(1)是 Softsign 函數，用途為將神經元運算結果轉為-1 到 1 之間的值，避免資料特徵快速流失。

Eq.(2)是線性整流函數(Rectified Linear Unit, ReLU)，又稱修正線性單元，用途為將神經元運算結果為負數之值化為 0，大於 0 之值則保持原本結果，藉此能更有效淬煉出資料特徵。

Eq.(3)是 Softmax 函數，此函數將神經元運算結果透過自然對數轉換成 0 到 1 之間的值，藉此能比較出各神經元的價值。

2.3 卷積神經網路(Convolutional Neural Network)

揚·勒丘恩(Yann LeCun)於 1998 年首先提出卷積神經網路(CNN)架構，用來識別手寫圖像 [22]，受限於輸入資料的規模小及硬體運算效能不足，調校模型參數及驗證須必須花費數天的時間，難以廣泛被應用。隨著網際網路普及產生爆炸性的資料量，及硬體運算效能的提升，深度卷積神經網路(AlexNet)以 16.4%的誤判率獲得了 ILSVRC 2012 年冠軍，較 2011 年冠軍之誤判率(25.8%)大幅進步 9.4%，並且接續幾屆的冠軍 2013(Zeiler and Fergus)、2014(VGG)、2014(GoogLeNet)、2015(ResNet)皆使用 CNN 架構 [23-26]，而誤判率也從 2010 年的 28.2%一路下降至 2015 年的 3.57%，甚至已經優於人類的分類錯誤率 5.1%。

卷積神經網路是一種以類神經網路為基礎的模型，其架構與類神經網路類似，主要差別在於多了一個或多個的卷積層(convolutional layer)及池化層(pooling layer)。透過卷積層提取輸入資料特徵，池化層保留特徵，使卷積神經網路在圖像和語音識別方面能比原本類神經網路或其他深度學習模型有更高的正確率 [9]。卷積神經網路架構示意圖如 Figure 2.2 [27]。

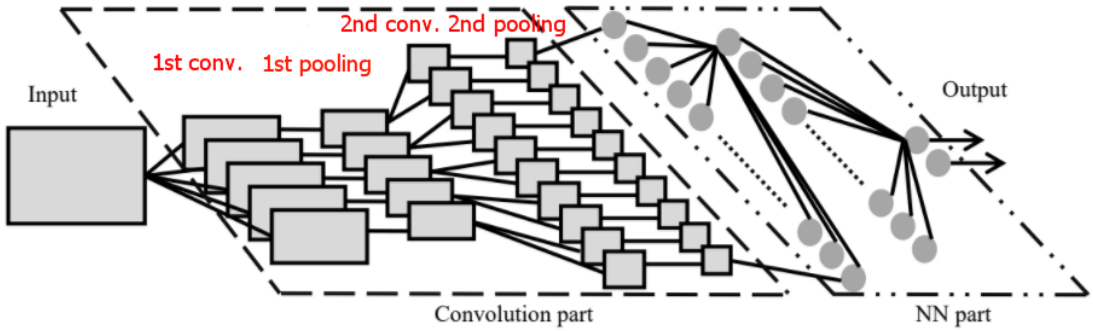


Figure 2.2: 卷積神經網路架構示意圖

2.3.1 卷積層

卷積層目的為取得輸入資料的局部特徵，作法為透過輸入資料中的特定資料特徵(Kernel)，與輸入資料各局部特徵進行卷積得到卷積單元(Feature Map)，若定義多種 kernel，則可提取越多的輸入資料特徵，對識別資料的能力也就更好。

卷積單元運作如 Figure 2.3，圖中的輸入資料為 3x3 二維矩陣，選定的 Kernel 範圍是 2x2，Kernel 的數字代表卷積單元的權重，卷積層運作時會由輸入矩陣左上角(藍框範圍)依序紅框、黃框、綠框提取值，各提取值和權重相乘，依序輸出而得到第一層卷積單元，利用多種 kernel，則可提取越多的輸入資料特徵 [28]。

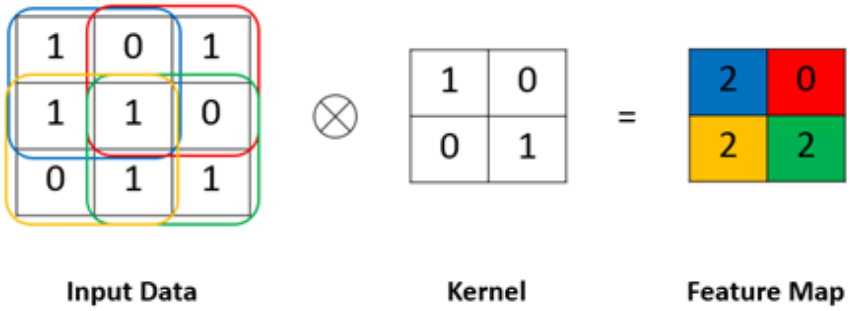


Figure 2.3: 卷積單元運作示意圖

2.3.2 池化層

池化層功能為保留卷積層得到的重要特徵，並壓縮資料大小，可加快模型運算效果。

輸入資料經過池化層池化以後，假設採用常見的最大池化(max pooling)，跨步(stride)為 2，則池化後的資料量會降為原本的四分之一，由於池化後的資料包含了原資料中各個範圍的最大值，所以還是保留了每個範圍最重要的特徵。透過 max pooling 依序處理藍、紅、黃、綠框範圍資料，stride 為 2，如 Figure 2.4。

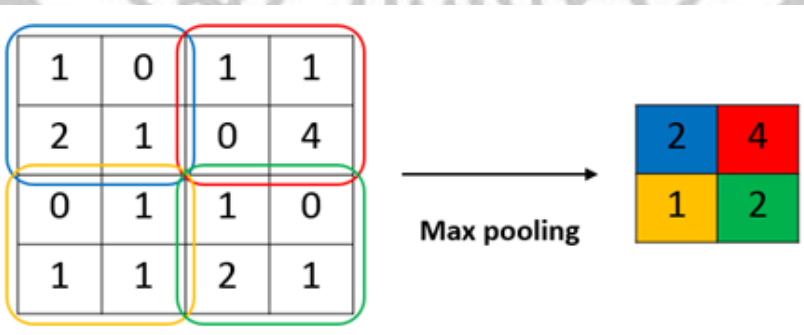


Figure 2.4: Max pooling 運作示意圖

2.4 GPU 平行運算

GPU(graphics processing unit)為圖形處理器，又稱顯示晶片或繪圖處理單元，是輝達公司(NVIDIA)在 1999 年 8 月首先提出的概念，GPU 是一種專門在個人電腦、工作站、遊戲機，或平板電腦、智慧型手機等行動裝置上執行繪圖運算工作的微處理器，由於 GPU 有大量的運算核心，藉由每個核心的獨立運算處理，可大幅加快計算速度。

現今高階的個人電腦 CPU 通常有四核心，伺服器等級 CPU 有二十四核心，但是普通等級的 GPU 就包含了數百至數千個更小型且更高效率的核心，如 Figure 2.5(取自 NVIDIA 官網)。本論文採用 Google 開發的開源軟體 Tensorflow 來建構類神經網路，搭配 NVIDIA 的 CUDA 平行運算架構，可發揮運用 GPU 強大和高效率的平行計算能力，有效改善類神經網路效能 [29, 30]。但 GPU 也有其限制，主要如 GPU 記憶體常是運算的瓶頸，必須依運算需求妥善分配，另外，並非所有的數學運算適合平行執行。

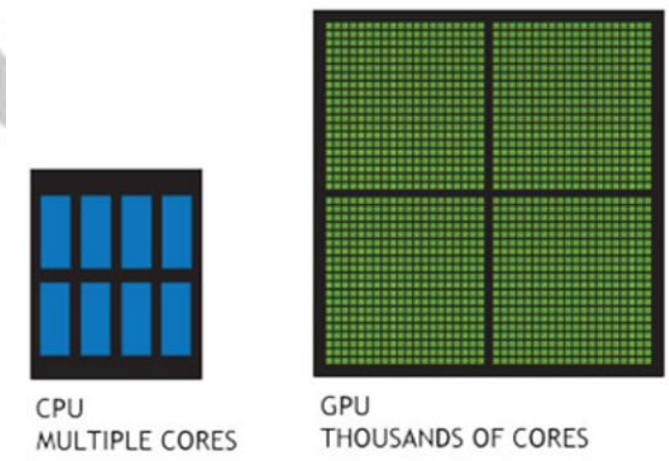


Figure 2.5: CPU 及 GPU 核心數差異示意圖

GPU 由多個 streaming multiprocessor(SMX)及一個 global memory 組成，每個 SM 包含 shared memory、Quad Warp Scheduler、Texture units、Read-Only Data Cache、指令提取分派 (instruction fetch/dispatch)、雙倍精度浮點數 (double precision unit) 等元件，以及大量的串流處理器(streaming processor，SP，CUDA core)。SP 在讀取記憶體數值進行運算時，相同 SM 中的 SP 可藉由 shared memory 交換資料，但在不同 SM 中的 SP 則必須透過 global memory 交換資料。因此，依照記憶體特性適當地存放神經元權重、偏量、數值、激勵函數、計算梯度值等，有助於優化運算效能[10,11]。本實驗環境使用的 GPU 為 Nvidia Kepler GTX 650 Ti(1GB memory 768 CUDA cores)，每個 SMX 包含 192 個 streaming processor，64 個 double-precision units，32 個 special function units (SFU)，以及 32 load/store units(LD/ST)。SMX 架構如 Figure 2.6、Figure 2.7 [31]。

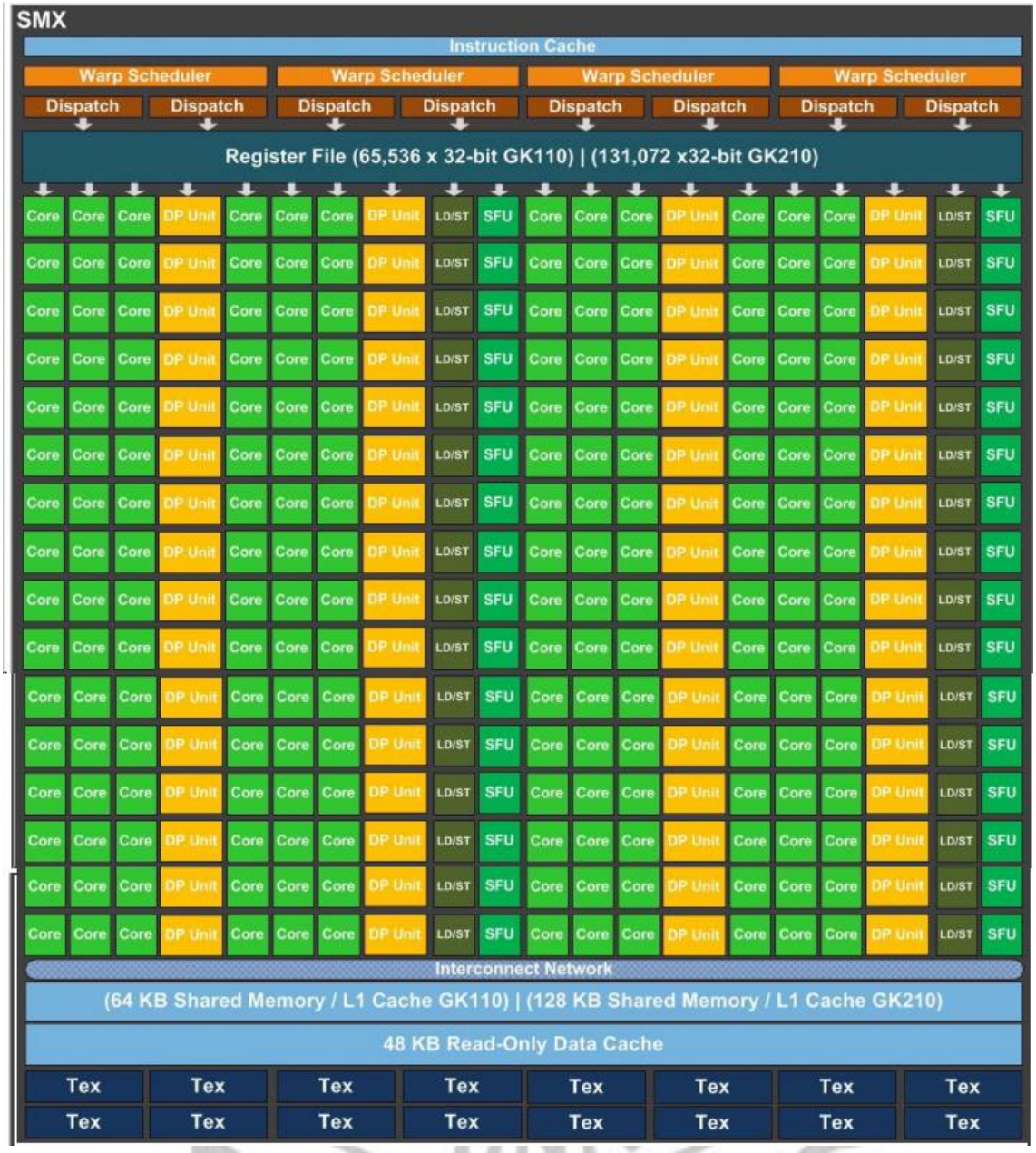


Figure 2.6: NVIDIA SMX 架構

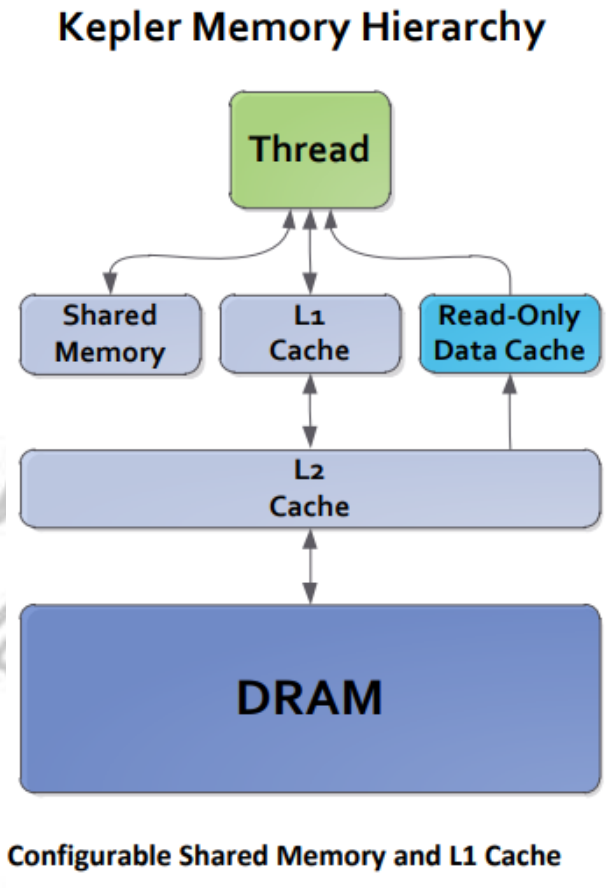


Figure 2.7: NVIDIA SMX 記憶體階層

Chapter 3　研究方法

3.1　實驗介紹

本論文使用加州大學爾灣分校(UCI)之垃圾郵件開放性資料，郵件資料為惠普公司(**HP**)內部技術報告，共有 4601 筆郵件資料，依照郵件內容中最常出現的字詞、字符、及連續出現的大寫字母長度，是否為 SPAM 等，共分為 58 種屬性，屬性說明如 Table 3.1。

Table 3.1: 郵件內容屬性說明

屬性	說明	數量(個)	數值
word_freq_WORD	特定字詞如 internet 或 money 等在信件中出現的百分比	48	0~100(%)
char_freq_CHAR	特定字符如;或$在信件中出現的百分比	6	0~100(%)
capital_run_length_average	大寫字母不間斷序列的平均長度	1	[1,...]
capital_run_length_longest	最長的不間斷大寫字母序列的長度	1	[1,...]
capital_run_length_total	電子郵件中的大寫字母總數	1	[1,...]
spam	是否為垃圾郵件	1	Spam (1) or not (0)

為了使各屬性特徵關聯度最大化，並且預留未來擴充新屬性的空間，我們將 SPAM 屬性以外的 57 種屬性擴充為 64 種，新增屬性之值先補上 0，機器學習時，

電腦會發現這些補 0 的屬性特徵不會影響結果，而自動忽略該特徵。

本實驗模型為監督式學習之反向傳播型類神經網路，我們使用其中 4140 筆資料作為訓練資料，此訓練資料中有 2557 筆為正常郵件，1583 筆為垃圾郵件；另取用剩下的 460 筆資料做為測試資料，此訓練資料中正常郵件及垃圾郵件各 230 筆。

實驗模型以 Python 3.5.2 程式語言撰寫，機器學習工具使用 Google tensorflow with GPU support 開源軟體函式庫，程式主要使用到的函式有 tf.device(使用 CPU 或加入 GPU 運算)、tf.nn.relu(使用激勵函數 ReLu)、tf.nn.softmax(使用激勵函數 Softmax)、tf.train.AdamOptimizer(使用梯度下降法優化)、tf.nn.dropout(加入 dropout 功能)等。使用 tensorflow with GPU support 版本時，模型會優先由 GPU 執行運算。

我們設計了以下三種實驗，類神經網路和卷積神經網路的分析能力比較，卷積神經網路的錯誤接受率和錯誤拒絕率，以及透過 GPU 效能優化，實驗環境如 Table 3.2。正確率代表正確辨識垃圾郵件的機率，正確率運算公式如 Eq.(4)。

Table 3.2: 實驗環境

CPU	Intel Q8200 2.33 GHz * 4
Memory	3.9 GB
OS	Ubuntu 16.04 64-bit
GPU	NVIDIA GTX 650 Ti 1GB memory 768 CUDA cores
機器學習工具	Google tensorflow with GPU support
程式語言	Python 3.5.2

$$\text{正確率} = \frac{\text{正確辨識出垃圾郵件數量}}{\text{測試資料筆數量}} \tag{4}$$

3.2 類神經網路實驗運作說明

本論文採用監督式學習之反向傳播型(backpropagation)類神經網路，並藉由調整模型之學習率(learning rate)數值可以控制權重更新的速度，經測試 0.001 及 0.00001 之間多個數值，learning rate 設定為 0.0001 有較佳的正確率。類神經網路詳細運作流程說明如 Figure 3.1。

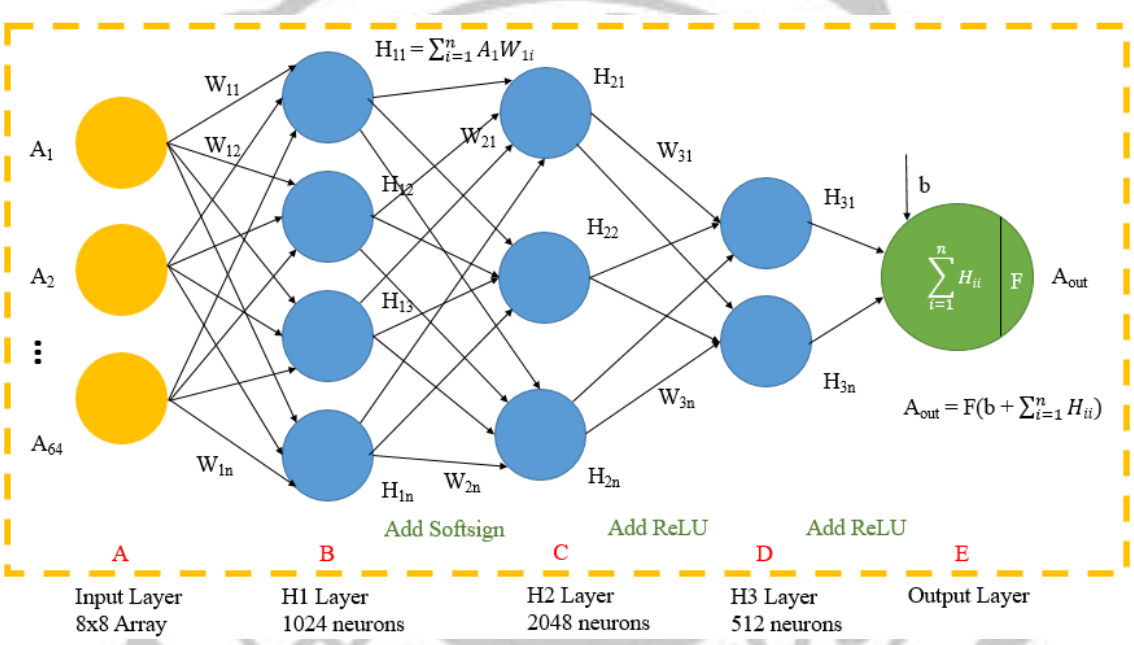


Figure 3.1: 類神經網路架構圖

3.2.1 資料輸入 (階段 A)

本階段是資料輸入，資料來源為 4140 筆郵件資料，每筆資料有分為 57 種屬性，並將 57 種屬性擴充為 64 種，新增屬性之值先補上 0，故每筆資料為 8*8 的二維矩陣。

3.2.2 隱藏層—1 (階段 B)

本階段為隱藏層，是本模型中的第一個隱藏層，神經元數量是 1024，藉多個神經元增加模型的複雜性以提高預測正確率，每個神經元有隨機的權重(特徵)及偏量，每個輸入資料與權重之乘積和與偏量相加之結果，透過本層的激勵函數(Softsign)轉換為下一層的輸入資料。透過 Softsign 函數將神經元運算結果轉為-1 到 1 之間的值，避免資料特徵快速流失。神經元數量主要依實驗結果調整，增加模型的複雜性也會增加模型訓練時間。

3.2.3 隱藏層—2 (階段 C)

本階段為隱藏層，是本模型中的第二個隱藏層，神經元數量是 2048，再次藉多個神經元增加模型的複雜性以提高預測正確率，每個神經元同樣有隨機的權重(特徵)及偏量，每個輸入資料與權重之乘積和與偏量相加之結果，透過本層的激勵函數(ReLU)轉換為下一層的輸入資料。透過 ReLU 函數將神經元運算結果之負數值化為 0，大於 0 之值則保持原本結果。

3.2.4 隱藏層—3 (階段 D)

本階段為隱藏層，是本模型中的第三個隱藏層，神經元數量是 512，再次藉多個神經元增加模型的複雜性以提高預測正確率，每個神經元同樣有隨機的權重(特徵)及偏量，每個輸入資料與權重之乘積和與偏量相加之結果，本層使用的激勵

函數同樣為 ReLu。

3.2.5 輸出層(階段 E)

本階段為輸出層，也是本模型的最後一個階段，用於呈現模型分類結果。

3.3 卷積神經網路實驗運作說明

本論文採用監督式學習之反向傳播型(backpropagation)卷積神經網路，並藉由調整模型之學習率(learning rate)數值可以控制權重更新的速度，經測試 0.01 至 0.000001 之間多個數值，learning rate 設定為 0.001 有最好的正確率。卷積神經網路詳細運作流程說明如 Figure 3.2、Figure 3.3。

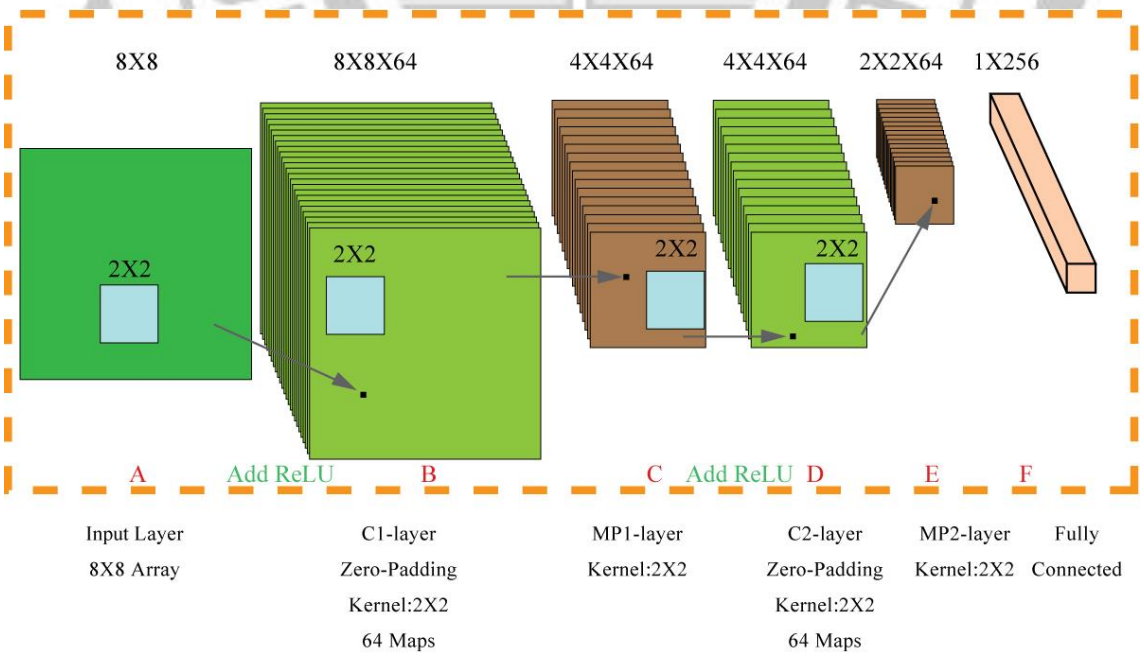


Figure 3.2: 卷積神經網路架構圖(convolution part)

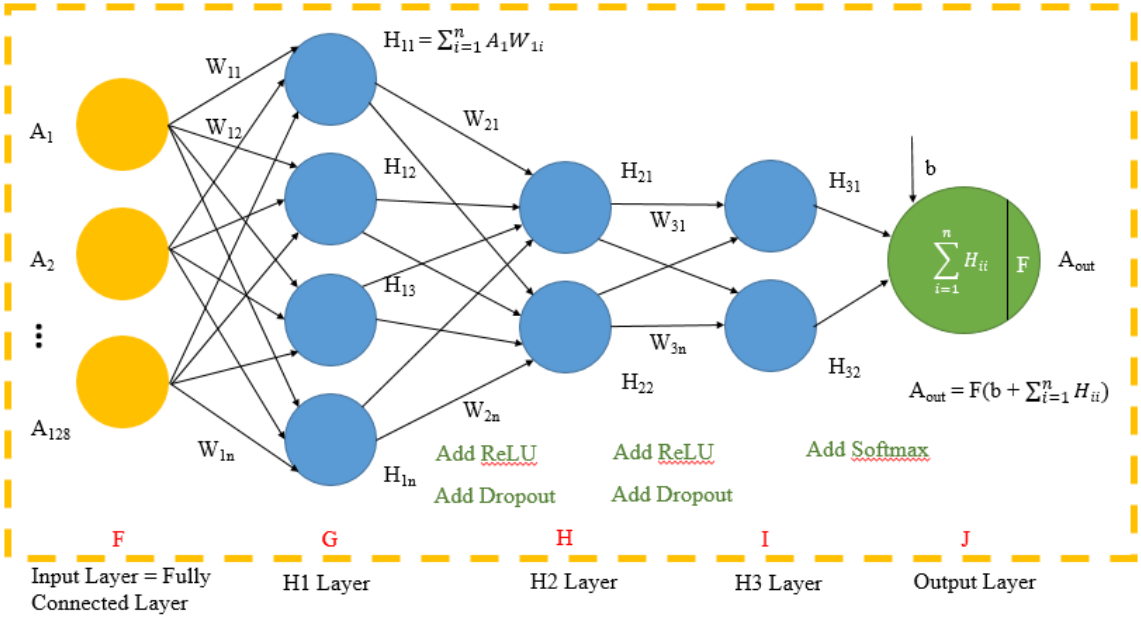


Figure 3.3: 卷積神經網路架構圖(NN part)

3.3.1 輸入層(階段 A)

本階段是資料輸入，資料來源為 4140 筆郵件資料，每筆資料有分為 57 種屬性，並將 57 種屬性擴充為 64 種，新增屬性之值先補上 0，故每筆資料為 8*8 的二維矩陣。

3.3.2 卷積層—1(階段 B)

本階段為卷積層，是本模型中的第一個卷積層，用於提取輸入資料特徵，本層提取的特徵(kernel)是 2*2 的二維矩陣，受限於硬體效能並考量模型訓練次數之正確率及花費時間，選用 64 個特徵，每個特徵從左上至右下掃過整個輸入矩陣，同時與輸入矩陣進行卷積得到卷積單元(feature map)，共有 64 層卷積單元。透過激勵

函數 ReLU 運算，使卷積結果為負數之值化為 0，大於 0 之值則保持原本結果，藉此保留資料特徵。特徵每次掃過輸入矩陣的距離稱為跨步(stride)，本次設定的跨步為 1，跨步越小可保存較多輸入矩陣資訊，避免資訊快速流失。並且在輸入矩陣周圍補上 0(zero padding)，藉以獲得輸入矩陣更多的邊界資訊。

3.3.3 池化層—1(階段 C)

本階段為池化層，是本模型中的第一個池化層，功能為保留卷積層得到的重要特徵，並壓縮資料大小，可加快模型運算效果。

池化的範圍是 2*2 的矩陣，透過最大值池化(max pooling)，跨步為 2，於由左上至右下掃過 64 層卷積單元，取得每層卷積單元中各個範圍的最大值，保留了每個範圍最重要的特徵，卷積單元經過池化層池化以後，其所包含的資料量會降為原本的四分之一。

3.3.4 卷積層—2(階段 D)

本階段為卷積層，是本模型中的第二個卷積層，本層提取的特徵(kernel)同樣是 2*2 的二維矩陣，受限於硬體效能並考量模型訓練次數之正確率及花費時間，選用 64 個特徵，跨步設定為 1，卷積動作與階段 B 相同，共有 64 層卷積單元。

3.3.5 池化層—2(階段 E)

本階段為池化層，是本模型中的第二個池化層，池化的範圍是 2*2 的矩陣，透過最大值池化(max pooling)，跨步為 2，池化動作與階段 C 相同。

3.3.6 全連接層(階段 F)

本階段為全連接層(fully connected layer)，作用是將階段 E 運算結果(2*2*64)攤平(flattening)為 256*1，做為類神經網路的輸入資料。

3.3.7 隱藏層—1(階段 G)

本階段為類神經網路的第一個隱藏層，使用的激勵函數同樣為 ReLU。本階段特別加入了 dropout 功能，此功能可以使機器學習過程中，刻意忽略掉一部分的神經元權重，來避免過度擬合(overfitting)的狀況。過度擬合是一種機器學習中常見的現象，主要原因來自於樣本數太少或是訓練次數過高，導致學習結果過於吻合訓練資料，而造成分辨新資料時出現更多誤判，解決過度擬合的方法有很多，這裡使用的是忽略權重，訓練模型時嘗試忽略權重比例 20%、30%、50%，以忽略權重比例 20%出現過度擬合的次數高於 30%、50%，且不同的忽略權重比例並不影響模型訓練時間，為了盡量取得較多訓練樣本，最後選定忽略掉的權重比例為 30%。神經元數量為 1024，藉多個神經元增加模型的複雜性以提高預測正確率，神經元數量主要依實驗結果調整，增加模型的複雜性也會增加模型訓練時間。

3.3.8 隱藏層—2(階段 H)

本階段為類神經網路的第二個隱藏層，使用的激勵函數同樣為 ReLU，在本階段同樣加入了 dropout 功能，來避免過度擬合(overfitting)的狀況，忽略掉的權重比例為 30%，神經元數量為 2。

3.3.9 隱藏層—3(階段 I)

本階段為類神經網路的第三個隱藏層，使用的激勵函數為 Softmax，此函數將階段 H 運算結果透過自然對數轉換成 0 到 1 之間的值，藉此能比較出各神經元的價值，輸出神經元數量為 2。

3.3.10 輸出層(階段 J)

本階段為輸出層，也是本模型的最後一個階段。階段 I 的輸出結果共分為兩類，較大的機率值被轉換為 1，較小的機率被轉換為 0，藉此達到分類效果。

3.4 卷積神經網路的錯誤接受率和錯誤拒絕率

本論文亦記錄卷積神經網路實驗結果，分析其錯誤接受率(false acceptance rate，FAR)和錯誤拒絕率(false rejection rate，FRR)，作為模型評價參考。錯誤接受率代表垃圾郵件判為正常郵件的機率；錯誤拒絕率代表正常郵件判為垃圾郵件的機率。FAR 運算公式如 Eq.(5)，FRR 運算公式如 Eq.(6)。

$$\text{FAR} = \frac{\text{垃圾郵件判為正常郵件數量}}{\text{測試資料筆數量}} \tag{5}$$

$$\text{FRR} = \frac{\text{正常郵件判為垃圾郵件數量}}{\text{測試資料筆數量}} \tag{6}$$

3.5 過度擬合

過度擬合(overfitting)是機器學習中常見的一種現象，主要原因來自於樣本數太少或是訓練次數過高，導致學習結果過於吻合訓練資料，而造成分辨新資料時出現更多誤判，如 Figure 3.4。

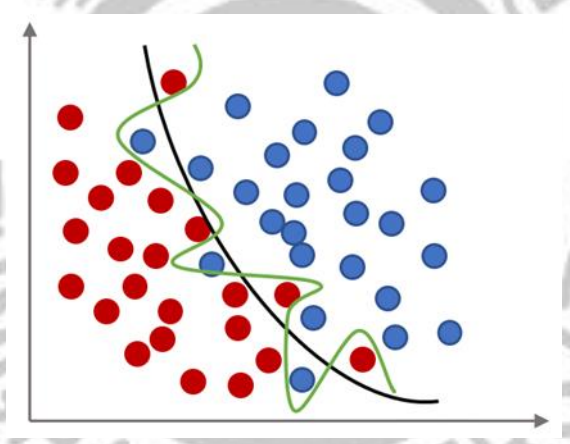


Figure 3.4: 過度擬合示意圖

本論文在輸入層及隱藏層加入 dropout 的功能，dropout 是一種機器學習中正規化的方法，此功能可使機器學習隨機忽略一部份的神經元權重，而避免本實驗中訓練次數過高時出現過度擬合現象 [9]。

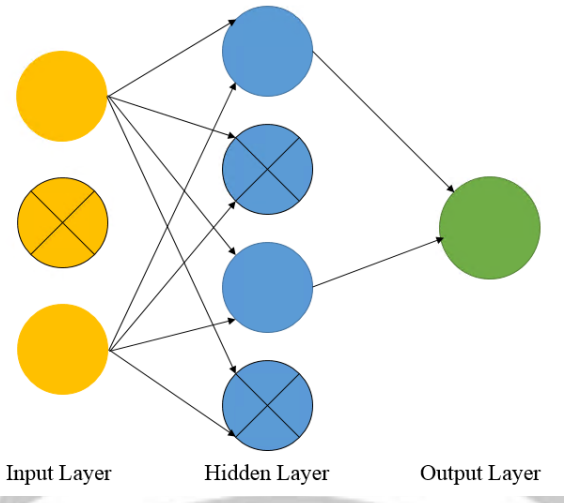


Figure 3.5: Dropout 功能運作示意圖

Chapter 4　研究結果

4.1　類神經網路實驗結果

本次實驗測試類神經網路的分析正確率，訓練次數由 500 次到 20000 次，實驗結果如 Table 4.1，顯示以類神經網路訓練次數之正確率及花費時間。

Table 4.1: 類神經網路之正確率及花費時間

訓練次數(次)	準確率(%)	CPU(秒)
500	73.8	873.0
1000	76.8	1861.6
3000	76.2	6205.3
5000	77.6	10242.1
7000	72.7	14340.7
10000	75.4	20734.5
15000	80	33182.2
20000	76.7	43830.0

由本次實驗結果得知，類神經網路訓練次數在 1000 次以上，大部分都有 75% 以上的正確率，20000 次的訓練時間則需要花費 43830 秒(約 12.2 小時)，相當耗時，且正確率並未隨訓練次數增加而提升。

4.2 卷積神經網路實驗結果

本次實驗改為使用卷積類神經網路，透過卷積層提取輸入資料之特徵，池化層保留特徵，確實提高了卷積神經網路的分析正確率，訓練次數由 500 次到 20000 次，實驗結果如 Table 4.2，顯示卷積神經網路執行各訓練次數之正確率及花費時間。

Table 4.2: 卷積神經網路之正確率及花費時間

訓練次數(次)	準確率(%)	CPU(秒)
500	87.2	205.9
1000	88.6	434.8
3000	90.3	1500.6
5000	90.5	2660.6
7000	90.6	3605.2
10000	90.7	6144.3
15000	91.3	10202
20000	91.4	13792.8

由本次實驗結果得知，卷積神經網路訓練次數達到 3000 次以上時，可以穩定達到 90%以上的正確率，訓練時間花費為 1500.6 秒(約 0.42 小時)，訓練次數達到 20000 次以上時，可以達到 91%以上的正確率，20000 次訓練時間花費為 1500.6 秒(約 0.42 小時)，各回合訓練正確率及時間皆優於類神經網路。

4.3 卷積神經網路的錯誤接受率和錯誤拒絕率實驗結果

由本次卷積神經網路實驗結果，分析其錯誤接受率(false acceptance rate，FAR)和錯誤拒絕率(false rejection rate，FRR)，作為模型評價參考。錯誤接受率代表垃圾郵件判為正常郵件的機率；錯誤拒絕率代表正常郵件判為垃圾郵件的機率，實驗結果如 Table 4.3，顯示卷積神經網路執行各訓練次數之正確率、FAR、FRR。此外，Figure 4.1 顯示卷積神經網路之正確率及 FAR、FRR 成長曲線圖。

Table 4.3: 卷積神經網路之正確率及 FAR、FRR

訓練次數(次)	準確率(%)	FAR(%)	FRR(%)
500	87.2	6.4	6.4
1000	88.6	6.8	4.6
3000	90.3	6.3	3.4
5000	90.5	5.8	3.7
7000	90.6	5.3	4.1
10000	90.7	4.8	4.5
15000	91.3	3.3	5.4
20000	91.4	2.8	5.8

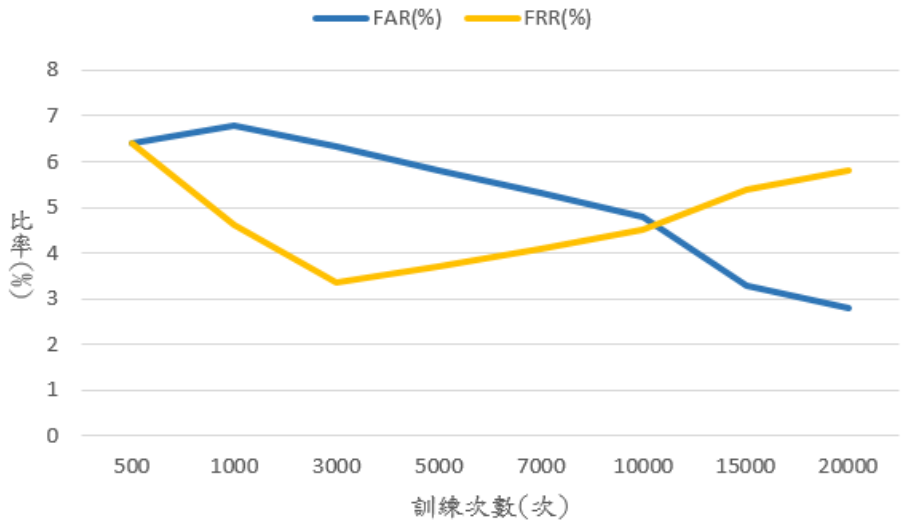


Figure 4.1: 顯示卷積神經網路之正確率及 FAR、FRR 成長曲線圖

由本次實驗結果得知，卷積神經網路的正確率隨訓練次數提高，訓練次數在 10000 次(含)以下時，FAR 會高於 FRR 或與 FRR 相當，但訓練次數達 10000 次以上，FAR 會低於 FRR，也印證郵件過濾系統要達到零漏攔，則會有更高比例的信件會遭誤攔。

4.4 GPU 平行運算

類神經網路模型包含大量的神經元數值、權重、偏量、激勵函數、計算梯度值等矩陣運算，因此，每個神經元可以交由 GPU 的串流處理器運算，特徵資料存放於共享記憶體內，權重及偏量存放在全域記憶體內，激勵函數則存放在 Read-Only Data Cache，依照 GPU 架構及各類記憶體特性存放資料，才能發揮充分發揮 GPU 平行運算能力。

本論文考量未來實務應用上會有大量訓練資料及更高的訓練次數，因此採用

GPU 平行化類神經網路及卷積神經網路模型運算，透過 GPU 平行運算，類神經網路分析速度超過 14 倍的加速效果，卷積神經網路分析速度也有超過 4 倍的加速效果。加速比公式如 Eq.(7)

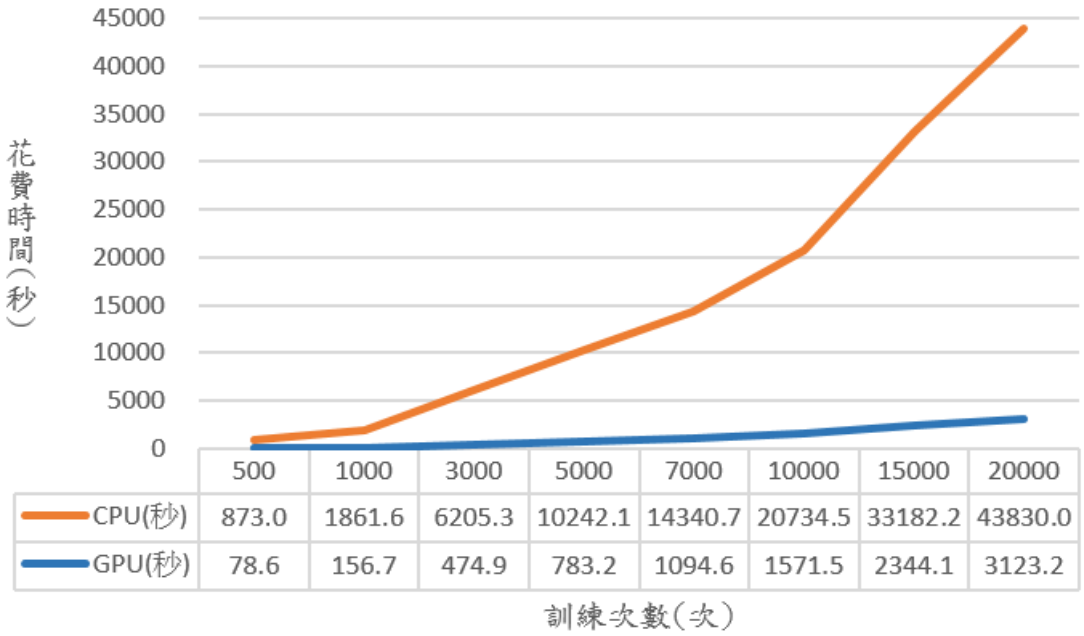
$$\text{加速比} = \frac{\text{CPU 執行時間}}{\text{GPU 執行時間}} \tag{7}$$

4.4.1 類神經網路實驗結果(增加 GPU 運算)

類神經網路訓練次數同樣由 500 次到 20000 次，實驗結果如 Table 4.4，顯示類神經網路使用 CPU 與加入 GPU 之各訓練次數時間比較。此外，Figure 4.2 顯示 CPU 和加入 GPU 之時間成長曲線圖(NN)。

Table 4.4: 類神經網路使用 CPU 和加入 GPU 之時間比較

訓練次數(次)	CPU(秒)	GPU(秒)	加速比(倍)
500	873.0	78.6	11.11
1000	1861.6	156.7	11.88
3000	6205.3	474.9	13.07
5000	10242.1	783.2	13.08
7000	14340.7	1094.6	13.10
10000	20734.5	1571.5	13.19
15000	33182.2	2344.1	14.16
20000	43830.0	3123.2	14.03



	500	1000	3000	5000	7000	10000	15000	20000
CPU(秒)	873.0	1861.6	6205.3	10242.1	14340.7	20734.5	33182.2	43830.0
GPU(秒)	78.6	156.7	474.9	783.2	1094.6	1571.5	2344.1	3123.2

Figure 4.2: CPU 和加入 GPU 之時間成長曲線圖(NN)

由本次實驗結果得知，類神經網路在未採用 GPU 平行運算時，3000 次的訓練時間需要花費 6205.3 秒(約 1.72 小時)，20000 次的訓練時間則需要花費 43830 秒(約 12.2 小時)。採用 GPU 平行運算後，3000 次的訓練時間僅需要花費 474.9 秒(約 0.13 小時)，20000 次的訓練時間僅需要花費 3123.2 秒(約 0.87 小時)，加速比達 14 倍。

4.4.2 卷積神經網路實驗結果(增加 GPU 運算)

卷積神經網路訓練次數同樣由 500 次到 20000 次，實驗結果如下 Table 4.5，顯示卷積神經網路使用 CPU 和加入 GPU 之各訓練次數時間比較。此外，Figure 4.3 顯示 CPU 和加入 GPU 之運算時間成長曲線圖(CNN)。

Table 4.5: 卷積神經網路使用 CPU 和加入 GPU 之時間比較

訓練次數(次)	CPU(秒)	GPU(秒)	加速比(倍)
500	205.9	83.9	2.45
1000	434.8	167.7	2.59
3000	1500.6	503.2	2.98
5000	2660.6	833	3.19
7000	3605.2	1161.5	3.10
10000	6144.3	1662.7	3.70
15000	10202	2524.8	4.04
20000	13792.8	3304.3	4.17

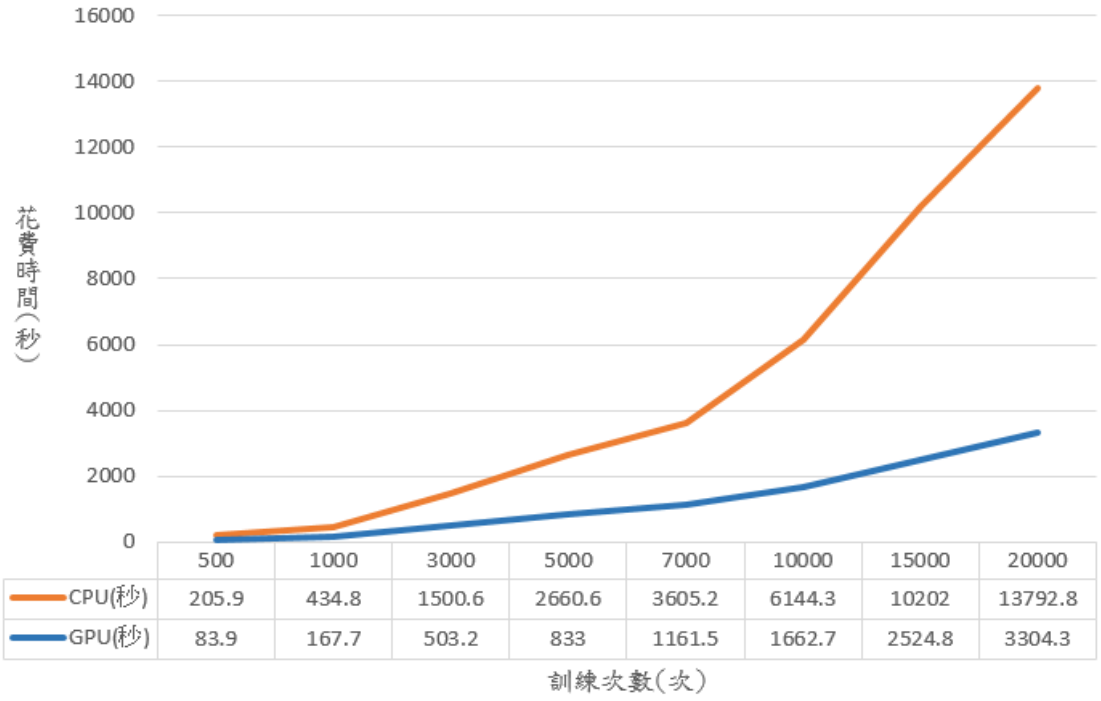


Figure 4.3: CPU 和加入 GPU 之時間成長曲線圖(CNN)

由本次實驗結果得知，卷積神經網路在未採用 GPU 平行運算時，3000 次的訓

練時間需要花費 1500.6 秒(約 0.42 小時)，20000 次訓練時間花費 13792.8 秒(3.83 小時)。採用 GPU 平行運算後，3000 次的訓練時間僅需要花費 503.2 秒(約 0.14 小時)，20000 次的訓練時間僅需要花費 3304.3 秒(0.92 小時)，加速比達 4.17 倍。

4.5 討論

本論文以機器學習之類神經網路(neural network)及卷積神經網路(convolutional neural network)演算法嘗試改善 UCI 垃圾郵件過濾器(content-based)之辨識正確率。實驗過程需要調校許多參數，包括類神經網路的隱藏層數量、各隱藏層的神經元數量，以及各隱藏層使用的激勵函數等；卷積神經網路的卷積層 Kernel 大小、數量，池化層的池化方式，以及全連接層後的類神經網路的隱藏層數量、各隱藏層的神經元數量，以及各隱藏層使用的激勵函數，dropout 比例等，都有可能影響模型正確率。

實驗結果顯示，在使用 CPU 運算之類神經網路演算法訓練 3000 回合(約耗時 6205.3 秒)，正確率為 76.2%，但是正確率並未隨訓練次數增加而提升。卷積神經網路演算法訓練 3000 回(約耗時 1500.6 秒)，正確率可達 90%。此外，考量實務上需求，我們利用 GPU(graphics processing unit)圖形處理器及機器學習工具 Google tensorflow with GPU support 開源軟體函式庫輔助運算，GPU 過去主要用於處理影像資料運算，由於 GPU 有大量的運算核心，藉由每個核心的獨立運算處理，可大幅加快計算速度。機器學習具有大量的神經元及權重，這些存放在矩陣內的資料計算過程特別適合使用 GPU 平行運算。實驗結果顯示，以類神經網路演算法訓練 3000 回(耗時僅須 474.9 秒)，訓練 20000 回(耗時僅須 3123.2 秒)；卷積神經網路演算法

訓練 3000 回(耗時僅須 503.2 秒)，訓練 20000 回(耗時僅須 3304.3 秒)，訓練時間低於一小時，且正確率可達 91%以上。由上述實驗證明，與類神經網路演算法相比，採用卷積神經網路演算法來分析郵件資料屬性及是否為垃圾郵件是更好的選擇，並且加入 GPU 平行化運算，可以獲得近 4.17 倍的加速比。

雖然卷積神經網路正確率(91%)未優於 UCI 垃圾郵件過濾器正確率(93%)，但可以預見的是，若能夠增加有效樣本數量，搭配深入優化模型參數，卷積神經網路加上 GPU 平行運算，以本實驗結果在訓練時間 10 分鐘內(3000 回)，即可達到很高的正確率，且不到一小時已可訓練達 20000 回，正確率有望再向上提升。

Chapter 5　結論與未來方向

隨著資訊科技的發展以及電腦、行動裝置設備的普及，電子郵件已是工作或生活密不可分的溝通工具，並且因為政府業務及金融交易必須綁定能夠代表真實身分的聯絡工具，故電子郵件的重要性尚無法取代。也正因為如此，電子郵件大量被運用於廣告行銷，或是電腦駭客、商業間諜、國家間諜等有心人士從事詐騙、勒索等不法行為。由於製作垃圾郵件成本低，傳播範圍大，回收效益相當可觀，故有心人士會更加「精進」垃圾郵件製作方法，所以也不難解釋，雖然公司大多數已建置垃圾郵件過濾系統，且過濾系統通常包含多層過濾機制，如防毒、附件掃描、郵件內文關鍵字過濾、郵件內文連結過濾、郵件來源 IP 黑名單、商品促銷或美容減重等類別關鍵字過濾等，但仍無法避免郵件遭過濾機制漏攔(false negative)或誤攔(false positive)。

近十多年來由於網路普及產生爆炸性的資料量，滿足了發展機器學習的資料需求，加上硬體效能的大幅進步，催生新演算法的提出，大幅改善的機器學習的預測率，機器學習已被廣泛應用於電腦視覺、自然語言解析、生物特徵識別、DNA 序列比對、語音和手寫識別，以及智慧機器人等領域。本論文以卷積神經網路(convolutional neural network)演算法嘗試改善 UCI 垃圾郵件過濾器(content-based)之辨識正確率，並且運用 GPU 平行運算增加運算效能，縮短運算時間。使用類神經網路演算法訓練 3000 次(約耗時 6205.3 秒)，正確率為 76.2%，但正確率並未隨訓練次數增加而提升；使用卷積神經網路演算法訓練次數 3000 次，耗時僅須 503.2 秒，可達到 90%的正確率，且訓練 20000 次花費時間低於一小時，正確率可超過 91%，證明卷積神經網路確實是很適合的選擇。未來若增加有效樣本數量，搭配深

入優化模型參數，相信能再提升正確率並降低訓練花費時間，也有機會實際整合為垃圾郵件過濾系統之其中一層過濾機制，協助強化過濾效果。

REFERENCE

[1] Team, R. (2015). Email Statistics Report, 2015-2019. The Radicati Group. 報告

[2] Chandrasekar, K., & Wueest, C. (2017). Symantec internet security threat report 2017. Symantec Corp., Mountain View, CA, USA, Tech. Rep. 報告

[3] Gudkova, D., & Demidova, N., (2018). Securelist spam and phishing in 2017, https://securelist.com/spam-and-phishing-in-2017/83833/

[4] Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[5] Youn, S., & McLeod, D. (2007). A comparative study for email classification. In Advances and innovations in systems, computing sciences and software engineering (pp. 387-391). Springer, Dordrecht.

[6] Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998, July). A Bayesian approach to filtering junk e-mail. In Learning for Text Categorization: Papers from the 1998 workshop (Vol. 62, pp. 98-105).

[7] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE.

[8] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211-252.

[9] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in neural information processing

systems (pp. 1097-1105).

[10] Socher, R., Bauer, J., & Manning, C. D. (2013). Parsing with compositional vector grammars. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 455-465).

[11] Qian, N. (1999). On the momentum term in gradient descent learning algorithms. Neural networks, 12(1), 145-151.

[12] Lemley, J., Bazrafkan, S., & Corcoran, P. (2017). Deep Learning for Consumer Devices and Services: Pushing the limits for machine learning, artificial intelligence, and computer vision. IEEE Consumer Electronics Magazine, 6(2), 48-56.

[13] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.

[14] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.

[15] Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, 32(14-15), 2627-2636.

[16] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533.

[17] Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In Neural networks for perception (pp. 65-93).

[18] Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

[19] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[20] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning

applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[21] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.

[22] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[23] Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham.

[24] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[25] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015, June). Going deeper with convolutions. Cvpr.

[26] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[27] Lin, C. H., Liu, J. C., & Lee, K. Y. (2018). On Neural Networks for Biometric Authentication Based on Keystroke Dynamics. Sensors and Materials, 30(3), 385-396.

[28] S. J. Nowlan and G. E. Hinton. Simplifying neural networks by soft weight-sharing. Neural Computation, 4(4), 1992.

[29] TensorFlow, https://www.tensorflow.org/

[30] NVIDIA. NVIDIA CUDA, https://www.nvidia.com/en-us/data-center/gpu-accelerated-applications/tensorflow/

[31] NVIDIA’s Next Generation CUDATM Compute Architecture: Kepler TM GK110/210 , http://international.download.nvidia.com/pdf/kepler/NVIDIA-

Kepler-GK110-GK210-Architecture-Whitepaper.pdf