

國立臺灣大學工學院應用力學研究所

碩士論文

Graduate Institute of Applied Mechanics

College of Engineering

National Taiwan University

Master Thesis



利用影像處理與類神經網路

進行人體坐姿判讀

Using Image Processing and Artificial Neural Networks

to Identify Human Sitting Postures

林 品 喬

Pin-Chiao Lin

指導教授：劉佩玲 博士

Advisor : Pei-Ling Liu, Ph.D.

中華民國 2013 年 7 月

July, 2013

國立臺灣大學碩士學位論文
口試委員會審定書

利用影像處理與類神經網路進行人體坐姿判讀

Using Image Processing and Artificial Neural
Networks to Identify Human Sitting Postures

本論文係林品喬君（學號：R00543018）在國立臺灣大學應用力學研究所完成之碩士學位論文，於民國 102 年 7 月 31 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

劉佩玲 劉佩玲
(指導教授)

毛慧芬 毛慧芬

梅興 梅興

所長 張家歐

致謝



本論文能夠順利完成，首先要感謝我的指導教授 劉佩玲教授，非常謝謝老師在這兩年期間所給予我最細心的指導，不厭其煩地一次又一次帶領我克服在研究上所遭遇到的各種難題，同時也讓我深刻學得做研究時對於每一個內容與步驟都必須要實事求是的精神以及自我解決問題的能力，在此獻上最真誠的敬意與感謝，謝謝老師的指導。

在論文定稿的過程中，承蒙輔仁大學資工系 梅興教授與台灣大學職能治療系 毛慧芬教授對於本文的指導與啟發，在百忙之中仍然抽空給予我許多寶貴的意見，使得本論文能夠更臻完備，在此亦表達最深的敬意給予以上兩位教授。

在研究所的這兩年中，謝謝葉柏涼學長總能夠在我遭遇問題的當下及時給予指導，謝謝阮智宇、吳宇盛同學陪同我在研究生涯中一同渡過各種階段，謝謝陳思妤學妹、曾柏融學弟、林立權學弟在實驗上獻身當模特兒以及給予各項幫忙與協助，也謝謝各位 227 研究室的助理姊姊們，總是親切的關懷我，謝謝所有 227 研究室的成員們，讓整個研究室增添不少笑語。

謝謝最親愛的父親大人與母親大人在背後的支持與鼓勵，謝謝奶奶總是在我北上之際給予我加油打氣，謝謝屁熊的經驗分享與英文文法修正與可愛的黑色擦子，謝謝牛同狸狸同老皮同小兔子同哈吉的歡

樂玩耍敲鍵盤與美白精華液，謝謝德先生所給的量子物理震撼教育，
謝謝無時無刻總能夠當合購夥伴的屁糖與她的方言，謝謝每天聽我吐
亂七八糟口水的偉誌哥哥，謝謝大半夜關心我身體健康叫我睡覺的郭
黑黑，謝謝胖大鄒的花蓮羊羹和伴手禮，謝謝台雁雁的加油簡訊，謝
謝當我爆肝還在線上可以聊天的小喬妹妹，謝謝給我影像處理好建議
的柏誠學長，謝謝所有三姑六婆社團的大家在一起三姑六婆的時光，
謝謝成大橋藝社的大家終於讓我在學生時代進了一次 BBO 轉播室，
謝謝 5/18 大土盃盃盃的夥伴們讓我重回大學時代打球的青春熱血，
辛苦阿磨頂著大太陽參加我的畢業典禮，謝謝你們讓我的研究所生涯
更加豐富與精采，我愛你們！將本文獻給所有曾經幫助我與關心我的
大家，謝謝！

摘要

本研究之目的在於使用人體影像進行坐姿的判讀。利用相機拍攝人體坐姿影像，以正規化與高斯濾波法消除背景，之後進行二元化處理。為避免坐姿的判讀會受到影響，嘗試將圖形中小腿刪除後之結果視為第一類目標圖型，將手與下半身刪除的結果視為第二類目標圖型。計算目標圖型的特徵參數—慣性矩 (moment of inertia) 與慣性積 (product of inertia)，並將參數輸入倒傳遞類神經網路 (Back Propagation Neural Networks, BPNN) 進行分析運算，最後再將網路輸出值透過分類器來判讀坐姿。

本研究中有10位受試者，共拍攝708張坐姿影像，其中屬於坐姿正的影像有317張，屬於坐姿不正的影像有391張。隨機挑選其中388張坐姿影像訓練類神經網路，剩餘的320張坐姿影像則進行後來的測試。其測試結果中第一類目標圖型之靈敏度 (sensitivity) 為80%，準確率 (accuracy) 為79.38%；第二類目標圖型之靈敏度 (sensitivity) 為78.13%，準確率 (accuracy) 為82.82%。因此由本研究的結果可知，以慣性矩與慣性積作為圖形特徵參數所建立起的坐姿判讀法具有辨識坐姿好壞的能力。

關鍵詞： 影像處理；類神經網路；慣性矩；慣性積；倒傳遞類神經網路

Abstract

The purpose of this study is to identify human sitting postures by using images. We use camera to capture the body's sitting images, remove the background by image normalization and Gaussian Smoothing filter, and then make images binaries. To avoid the identification of sitting postures being affected, we try to remove lower leg of posture images for first target pattern, remove hands and lower body for second target pattern. We calculate the target patterns characteristic parameters - moment of inertia and product of inertia, and then we make those parameters input back-propagation neural network (BPNN) for analysis and computing, finally, the output value of neural network through a classifier to identify the sitting postures.

In this study, there are 10 subjects and 708 sitting images, which belong to the good posture were 317 images, belong to the bad posture were 391 images. Randomly selected 388 sitting images to train the neural network, the remaining 320 sitting images do test later. The test results of the images of first target pattern, the sensitivity is 80%, and the accuracy is 79.38%; the sensitivity of second target pattern is 78.13%, and the accuracy is 82.82%. From the results of this study, the posture identify system which we set up by using moment of inertia and product of inertia has the ability to recognize good and bad posture.

Key words : image processing ; neural network ; moment of inertia ; product of inertia ; Back Propagation Neural Networks

目錄



口試委員會審定書.....	I
致謝.....	II
中文摘要.....	IV
英文摘要.....	V
目錄.....	VI
圖目錄.....	VII
表目錄.....	X
第一章 前言.....	1
1-1 研究動機.....	1
1-2 文獻回顧.....	2
1-3 研究內容.....	4
第二章 影像處理.....	7
2-1 去除背景.....	7
2-2 高斯平滑濾波.....	9
2-3 擷取目標圖型.....	12
第三章 類神經網路.....	32
3-1 生物神經網路.....	33
3-2 類神經網路學習演算法.....	38
3-1-1 類神經網路系統架構.....	41
3-1-2 類神經網路學習演算法.....	43
3-3 倒傳遞類神經網路.....	68
3-3-1 權重修正公式.....	68
3-3-2 圖形特徵參數.....	74
3-3-2 隱藏層層數與神經元數量.....	77
第四章 判斷分類器.....	104
第五章 坐姿判定之實驗分析與討論.....	114
5-1 受試者與受試環境介紹.....	114
5-2 實驗流程與結果.....	114
5-3 實驗討論.....	116
第六章 結論與未來展望.....	138
參考文獻.....	141

圖目錄



圖 1-3-1	人體坐姿判斷系統流程圖	6
圖 2-1-1	影像座標示意圖	19
圖 2-1-2	光度校正圖	19
圖 2-2-1	幕罩運算範圍示意圖	21
圖 2-2-2	一維的高斯函數圖	21
圖 2-2-3	二維的高斯函數圖	22
圖 2-3-1	高斯濾波後之圖形	22
圖 2-3-2	高斯濾波後圖形之灰階分佈	23
圖 2-3-3	經閥值判斷後之二元圖	23
圖 2-3-4	正姿圖形一	24
圖 2-3-5	正姿圖形二	25
圖 2-3-6	標準坐姿二元圖	26
圖 2-3-7	九宮格劃分預想圖	27
圖 2-3-8	九宮格實際架構圖	28
圖 2-3-9	九宮格實際完成圖	28
圖 2-3-10	刪除小腿之上半身圖	29
圖 2-3-11	刪除小腿、手與大腿之上半身圖	29
圖 2-3-12	非標準坐姿劃分九宮格圖	30
圖 2-3-13	非標準坐姿刪除小腿之上半身圖	31
圖 2-3-14	非標準坐姿刪除小腿、手與大腿之上半身圖	31
圖 3-1-1	神經系統對於外界刺激與回應圖	81
圖 3-1-2	神經元構造圖	81
圖 3-1-3	神經元傳遞訊息圖	82
圖 3-1-4	膜電位示意圖	82
圖 3-1-5	動作電位圖	83
圖 3-2-1	人工神經元模型圖	83
圖 3-2-2	活化函數圖	84
圖 3-2-3	前饋式類神經網路之架構	84
圖 3-2-4	單層前饋式類神經網路圖	85
圖 3-2-5	多層前饋式類神經網路圖	85
圖 3-2-6	回饋式類神經網路之架構圖	85
圖 3-2-7	降兩量回傳時間稽延圖	86
圖 3-2-8	學習演算法分類	86
圖 3-2-9	赫賓學習法示意圖	87
圖 3-2-10	贏者全拿學習法示意圖	87

圖 3-2-11 神經元權重學習示意圖	88
圖 3-2-12 最小均方演算法示意圖	88
圖 3-2-13 最陡坡降法示意圖	89
圖 3-2-14 單層感知器學習法示意圖	89
圖 3-2-15 多層感知器學習法示意圖	90
圖 3-2-16 Delta 學習法示意圖	90
圖 3-2-17 Grossberg 學習法示意圖	91
圖 3-3-1 倒傳遞類神經網路架構圖-前饋式網路	91
圖 3-3-2 倒傳遞類神經網路正向傳播與負向傳播圖	92
圖 3-3-3 倒傳遞類神經網路演算之流程圖	92
圖 3-3-4 任意形狀平面積	93
圖 3-3-5 矩形的慣性矩	93
圖 3-3-6 在對稱軸線上慣性積等於零	94
圖 3-3-7 慣性矩與慣性積與圖形關係圖	94
圖 3-3-8 不同坐姿圖	95
圖 3-3-9 ROC 曲線圖	98
圖 3-3-10 判斷結果分佈圖	98
圖 3-3-11 閥值移動對照 ROC 曲線圖	99
圖 3-3-12 具有高判斷力的 ROC 曲線圖	99
圖 3-3-13 不具有高判斷力的 ROC 曲線圖	100
圖 3-3-14 完全不具判斷力的 ROC 曲線圖	100
圖 3-3-15 刪除小腿之上半身神經元個數與 AUC 關係圖	101
圖 3-3-16 刪除小腿之上半身之 ROC 曲線面積圖	101
圖 3-3-17 刪除小腿、手與大腿之上半身神經元個數與 AUC 關係圖	102
圖 3-3-18 刪除小腿、手與大腿之上半身神經元個數與 ROC 曲線面積圖	102
圖 4-1 刪除小腿之上半身之坐姿正輸出統計長條圖	109
圖 4-2 刪除小腿之上半身坐姿不正輸出統計長條圖	109
圖 4-3 刪除小腿、手與大腿之上半身坐姿正時輸出統計長條圖	110
圖 4-4 刪除小腿、手與大腿之上半身坐姿不正時輸出統計長條圖	110
圖 4-5 刪除小腿之上半身機率密度函數圖	111
圖 4-6 刪除小腿、手與大腿之上半身機率密度函數圖	111
圖 4-7 刪除小腿之上半身判斷閥值與 κ 圖	112
圖 4-8 刪除小腿、手與大腿之上半身判斷閥值與 κ 圖	112
圖 5-2-1 實驗判斷流程圖	121
圖 5-2-2 訓練資料在刪除小腿時參數正規化圖	122
圖 5-2-3 訓練資料在刪除小腿、大腿與手時參數正規化圖	123
圖 5-2-4 測試資料在刪除小腿時參數正規化圖	125
圖 5-2-5 測試資料在刪除小腿、大腿與手時參數正規化圖	126

圖 5-2-6 測試資料在刪除小腿時 ROC 曲線圖	128
圖 5-2-7 測試資料在刪除小腿時不同閥值與 κ 值關係圖	128
圖 5-2-8 測試資料在刪除小腿、大腿與手時 ROC 曲線圖	129
圖 5-2-9 測試資料在刪除小腿、大腿與手時不同閥值與 κ 值關係圖	129
圖 5-3-1 誤判姿勢類型 1	130
圖 5-3-2 誤判姿勢類型 2	130
圖 5-3-3 誤判姿勢類型 3	131
圖 5-3-4 誤判姿勢類型 4	132
圖 5-3-5 誤判姿勢類型 5	133
圖 5-3-6 誤判姿勢類型 6	133
圖 5-3-7 誤判姿勢類型 7	134
圖 5-3-8 誤判姿勢類型 8	134
圖 5-3-9 誤判姿勢類型 9	135

表目錄



表 3-3-1 參數正規化	103
表 3-3-2 Confuse Matrix	103
表 3-3-3 ROC 量表	103
表 4-1 κ 量表	113
表 4-2 兩位醫生判斷結果之對照表.....	113
表 4-3 類神經網路目標與實際輸出結果之對照表.....	113
表 5-2-1 測試資料在刪除小腿時各判斷值	137
表 5-2-2 測試資料在刪除小腿、大腿與手時各判斷值	137



第一章 前言

1-1 研究動機

在一天 24 小時之中，人們自起床到下次上床睡覺之中的時間約莫有 16 小時。而在這 16 小時之中，人們又有將近八成的時間都是採取「坐著」的姿勢。不論是辦公室人員、司機、工程師、教授、作業員、會計師、顧問等職業的工作者，以及任何年齡層的學生，甚至是已經退休閒賦在家與高齡的老人人們，每個人每天都離不開「椅子」！

尤其現今是資訊爆炸的時代，越來越多的知識只需要透過網路就可以得知，更使得人們不需要出門使需要坐在電腦桌前就能解決問題，相對起過去的年代，也就大大增加了接觸椅子的時間；甚至是娛樂方面的線上遊戲等也都非常蓬勃發展，更使得平日已經坐了一整天椅子的學生，在假日或是空閒時也仍然長時間的持續坐在電腦桌前。因此，如果不去注意坐姿的好壞，長期這樣下去會使得人體產生腰酸背痛或是一些脊椎方面的問題。一旦出現這樣的問題後要再去解決，基本上會花上非常多的金錢與時間成本，如此一來可以說是得不償失。

因此如果能夠發展出一種方式具有可以有效辨識坐姿好壞的能力，就可預防很多因坐姿不良而引起的病痛，尤其是正值發育期的學

生，在身體的骨骼尚未完全成熟之時，坐姿的好壞對於身體的發展非常最重要。

本研究之目的即在建立坐姿判讀的準則與方法，藉由讀取影像的方式，來對於自己本人的坐姿做出正與不正的判讀。由於自己是無法判斷自身坐姿的，需要外人對於自己的姿勢進行監測或糾正，故本研究最終希望能夠達到自我監測之效果，同時亦能用此準則結合任何具有影像拍攝功能的系統或是工具，可監測所有需要監測坐姿的使用者，以達到預防與保健之功用。

1-2 文獻回顧

利用非侵入式感測器量測人體訊號是醫學上非常重大的發展，K. K. Kim, Y. K. Lim and K. S. Park (2005) 發展出一張能夠測量人體ECG(心電圖)訊號的智慧椅。隨著社會發展，越來越多文明病的出現，尤其是因久坐與姿勢不良產生的腰酸背痛現象，因此有許多關於人體坐姿這方面的議題與研究內容，然而一直以來的研究重心幾乎都是圍繞著壓力感測器。H. Z. Tan, L. A. Slivovsky and A. Pentland (2001) 發展出壓力椅，藉由將壓力感測器裝置於辦公室椅上，觀察壓力分佈的情形來判讀坐姿。G. Chen, Q. Wu, and Z. Yang (2008) 將壓力感測器裝置在椅子上，利用量測人坐在椅子上不同坐姿時的壓力分佈為何，去發展出一套舒適度最高的輔助椅。G. A. Vos *et al.*

(2006) 在探討不同坐姿與不同椅子設計時，對於椅子上壓力分布的情形。H. Nakane and J. Toyama (2011) 認為人在疲勞時坐姿會與精神好時的坐姿有所差異，藉由監控壓力感測椅中的壓力變化可以判讀出疲勞與否，進而給予提醒。M. Yamada *et al.* (2009) 藉由在椅墊與椅背上裝上壓力感測器，透過感測器量測到的訊息達到判讀此時椅子上的人在做什麼樣的動作。Y. Li and R. Aissaoui 亦是藉由將壓力感測器裝置在椅子上的方式，來判斷人的坐姿為何。

有別於利用椅子上的壓力感測器來判讀坐姿，S. Murphy , P. Buckle and D. Stubbs (2002) 則是利用攝影與在身上穿戴感測器的方式監測學生上課時的坐姿。而C. C. Li and Y. Y. Chen (2006) 則是利用攝影的方式拍下人體影像，並透過人體中的一些特徵參數，例如身高或體寬等等，藉由這些參數的處理來發展出一套規則辨別人體站姿的系統，以此達到居家看護系統對於小孩或老年人的監控。在醫學上，一般是採用兩種方式來對人體姿勢做出判讀，一種是以多台攝影機拍攝的方式，或是藉由X光的拍攝，來找出人體轉動的關節處為何，進而對人體姿勢做出判讀，如Jun Ohya and Fumio Kishino (1994) 就是藉由架設多台相機拍攝人體姿勢影像的方式，來建立起人體姿勢三維圖型，Juergen Gall *et al.* (2009) 亦是；另一種則是在人體關節處貼上標記 (marks)，藉由觀察標記處位移的變化來對於人體關節處

的運動做記錄，如Kazutaka Kurihara *et al.* (2002)。

過往研究中判讀人體坐姿大多是以壓力感測器為主，但是一來壓力感測器的價格非常昂貴，無法普及化，二來其判斷的方式也不若人眼來的直接。而醫學上判斷姿勢的影像或是亮點追蹤的兩種方式雖然很精準，但是所需要耗費的成本太大，基於多方考量，因此本研究利用拍照的方式記錄坐姿，類比人眼觀看他人坐姿的視野，且只採用一台相機，之後再將這些照片進行一連串的處理，最後對於坐姿做出判斷，並給予警告。由於不需要在身上穿戴儀器，故不會影響到做事的效率，而影像的來源可以從個人手機照相得來，所以人人得以對自己的坐姿進行監控，也可以與居家看護的監測系統做結合，如此一來，就能達到非常普及化之效果。

1-3 研究內容

本研究將發展一套坐姿判讀的方式。處理過程為先利用相機將人體各坐姿影像攝入，結合影像處理的方式消除背景，只餘下人體姿勢的圖形，再將圖形做二值化處理，以計算圖形的特徵參數值，之後將各參數值輸入至類神經網路中，模擬人腦在做判斷時對於多方因素比較與考量的情境。最後，設定一閥值來當做劃分人體坐姿正與不正之標準。如果判斷結果為正的姿勢，則不給予任何警告；反之，如果被判斷為不正的姿勢，則立即給予警告。本研究之人體判斷系統流程圖

如 1-3-1 所示。

本論文共分六章，各章節內容安排如下：

第一章 前言

第二章 為本研究中人體坐姿判讀流程的第一步驟：將圖形進行影像處理之過程。先介紹如何如何將影像去除背景，之後再將圖形透過平滑濾波的處理消除雜訊以及將圖形內部做一平均的動作，以利最後做影像二元化與擷取目標圖型之處理

第三章 為本研究中人體坐姿判讀流程的第二步驟：類神經網路之析。一開始先介紹生物神經網路與類神經網路結構，之後介紹本研究所使用的倒傳遞類神經網路，以及如何將圖形特徵轉化成類神經網路的輸入值

第四章 為本研究中人體坐姿判讀流程的第三步驟：坐姿分類器。介紹如何將類神經網路之演算結果透過閥值的方式來分類出坐姿正與不正，以及閥值該如何選取之過程

第五章 利用上述所介紹之方法來對於實驗受試者進行坐姿的判斷與分析，並且討論其結果

第六章 總結全文並對未來研究方向提出建議



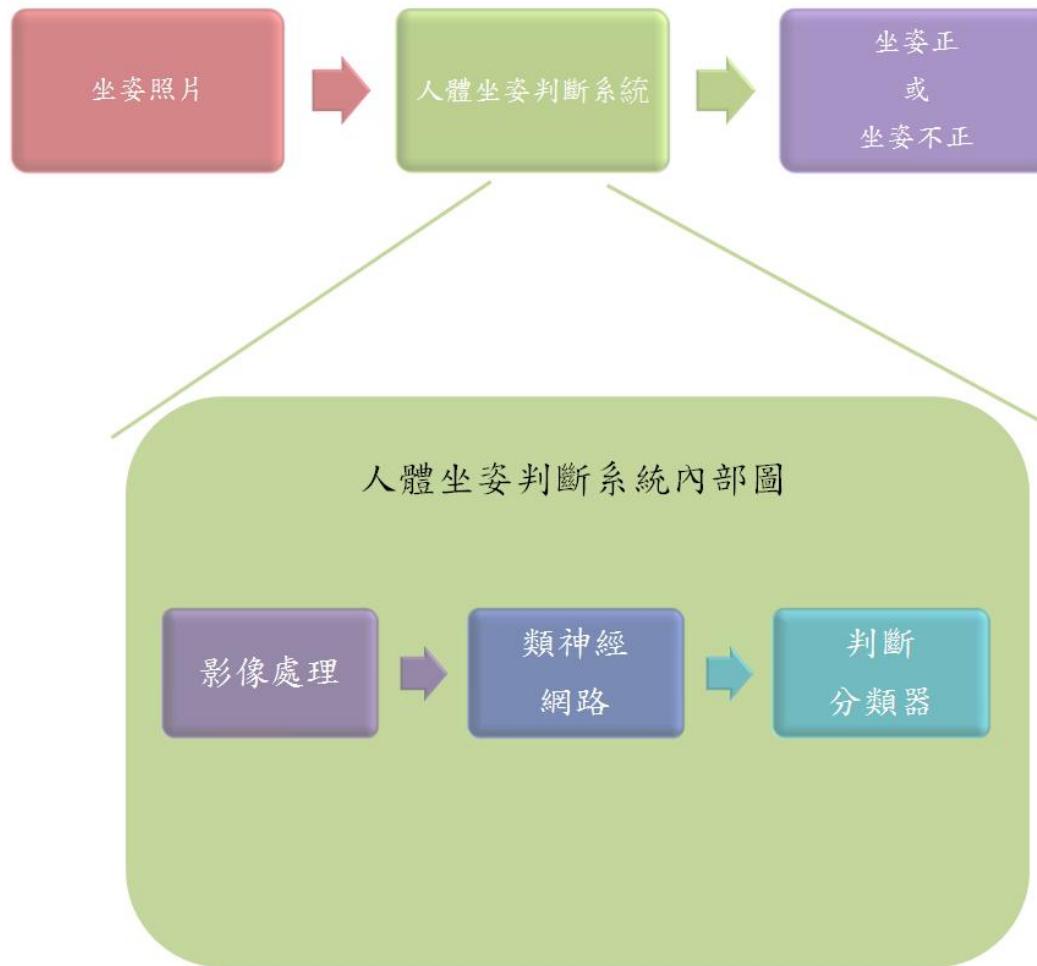


圖 1-3-1 人體坐姿判斷系統流程圖

第二章 影像處理



此章節是人體坐姿判讀流程的第一步驟，在此我們將會介紹如何把拍攝到的原始圖檔處理成我們所需要的人體坐姿圖形，首先介紹如何將原始圖檔去除背景，之後引入一平滑濾波器做一些處理，最後則是將圖片中的人體姿勢部分擷取出來，以利下一步驟類神經網路分析的處理。

2-1 去除背景

在進行影像處理之前，我們必須將原始的圖片從彩色的 RGB 圖檔轉換成灰階圖檔。之所以要將圖檔轉換成灰階圖檔，一來是因為灰階圖檔是二維矩陣的表示方式，且每個像素 (pixel) 由 8 個位元來表示，故灰階值的變化可以從 0 到 255，而彩色圖檔為三維矩陣且每一個像素需要用 24 位元來表示，相較之下用灰階圖能夠較省空間；二來是因為若將圖檔用灰階表示之，則其物件的輪廓會比用彩色表示更加明顯。因此，我們每次拍攝到的坐姿圖，都會先將其轉成灰階圖檔後才能做後續的處理。

圖檔的座標格式與我們常用的座標格式略有不同，如圖 2-1-1 所示，由圖 2-1-1 我們可以發現圖檔的原點是在左上角，且圖檔座標的表示法是先行 (row) 後列 (column)，與我們所用的在卡氏座標中描述一

點的座標時， x 座標為橫坐標 y 座標為縱座標的方式不同。

在本研究採取的背景消除方法是採用相減法，其處理過程分成兩步驟，第一步驟是先校正光度，第二部份則是將校正光度後的圖片與背景圖做相減的動作。在此我們將椅子視為不會移動的背景之一，意即背景圖是指椅子上尚未坐人時的圖檔，人體坐姿圖則是人坐在椅子上的圖檔，將兩張圖相減之後，相同的部分就會變成零，不同的地方就是人體姿勢取代原本背景圖的地方，因此這兩張圖有灰階值變化的地點就是整個人體姿勢圖。因為在實際過程中，拍攝背景圖與人體坐姿圖時光度可能會產生了改變，所以在將兩張圖檔做相減的動作之前需要先做光度校正的處理。

在本研究中光度校正的過程為：先從背景圖的左上角取一小塊像素的背景灰階值，再從人體坐姿圖中相同位置的地方取一塊像素的背景灰階值，之後用從人體坐姿圖中取出的這一塊像素的灰階值各自去除上背景圖中相同位置的灰階值，用以判斷此時人體坐姿圖與純背景圖的光度變化了多少，之後再將得到的灰階值變化取平均，最後用人體坐姿圖去除上這個灰階變化值的平均，就完成光度校正的工作了。我們用圖檔座標的格式來描述背景圖與人體坐姿圖的灰階值，令背景圖中各畫素的灰階值為 $B(r,c)$ ，坐姿圖中各畫素的灰階值為 $P(r,c)$ ，座標中的 r 代表第幾行， c 代表第幾列，則光度校正的計算過程如下

式：

$$s = \frac{\sum_{r=1}^m \sum_{c=1}^n P(r, c)}{m \times n}$$
$$\hat{P}(r, c) = \frac{P(r, c)}{s} - B(r, c) \quad (2-1-2)$$



其中 s 為人體坐姿圖中光度校正的倍率， $\hat{P}(r, c)$ 則是去除掉背景後的人體姿勢圖，如圖 2-1-2 所示，在本研究中我們 m 與 n 都是取 100。由圖 2-1-2(c) 可以大致看出坐姿的圖樣，然而單單只是相減之後的圖檔其背景的雜訊仍然存在，雖然光度經過校正但並不是每個屬於背景的畫素在經過相減之後的灰階值都一定會是零，因此我們需要將背景殘餘的雜訊去掉，故我們在此引入一平滑濾波器來做後續的處理。

2-2 高斯平滑濾波 (Gaussian Smoothing filter)

空間濾波器一般又稱為幕罩 (mask)，使用幕罩對於影像處理的方式就稱為幕罩法，藉由移動一固定大小的幕罩，將此幕罩所涵蓋的區域的灰階值與幕罩做疊積運算 (convolution)。一般常使用的幕罩大小為 3×3 的矩陣，如以下的格式：

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

假設 $\hat{P}(r, c)$ 為一張影像中的任意一點 (r, c) 所具有的灰階值，利用幕罩

法將一個 3×3 矩陣的幕罩與以 (r, c) 點為中心、半徑為 1 所形成的

3×3 區塊(如圖 2-2-1 所示)的灰階值做疊積運算，運算過程如下：

$$\begin{aligned}\tilde{P}(r, c) = & w_1 \hat{P}(r_{i-1}, c_{j-1}) + w_2 \hat{P}(r_i, c_{j-1}) + w_3 \hat{P}(r_{i+1}, c_{j-1}) \\ & + w_4 \hat{P}(r_{i-1}, c_j) + w_5 \hat{P}(r_i, c_j) + w_6 \hat{P}(r_{i+1}, c_j) \\ & + w_7 \hat{P}(r_{i-1}, c_{j+1}) + w_8 \hat{P}(r_i, c_{j+1}) + w_9 \hat{P}(r_{i+1}, c_{j+1})\end{aligned}\quad (2-2-1)$$

其中 $\tilde{P}(r, c)$ 代表灰階值 $\hat{P}(r, c)$ 經過幕罩運算後所得到的新的灰階值。

其意思也就是說針對任意一點 (r, c) 之灰階值，會被以此點為中心、半徑為 1 的周圍 9 個點之灰階值分別乘上一權重後加總的灰階值取代之。

最常用的平滑濾波器為高斯平滑濾波，一維的高斯函數即是我們常說的高斯分佈曲線，也就是常態分布曲線，如圖 2-2-2 所示，其式子如下：

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2-2-2)$$

其中 σ 代表的是標準差， μ 代表的是平均數（一般來說為零）。

二維的高斯函數則可寫成：

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-\mu)^2 + (y-\mu)^2}{2\sigma^2}} \quad (2-2-3)$$

其圖形示意圖為圖 2-2-3。

由圖形 2-2-3 可看出，二維的高斯函數具有以下特性：



1. 在二維的平面中，二維高斯函數是一個環狀對稱，對於任何方向處理的平滑程度都是相等的。
2. 高斯二維函數是一單峰值的函數，距離原點越遠則峰值越小，也就是說對於每一個要運算灰階值的點，若是週圍的點與其距離越遠，則影響的權重就越小，因此能夠保留局部特性而消除其他不必要之雜訊。
3. 高斯平滑濾波器之濾波平滑程度是藉由 σ 來控制的，當 σ 越大則越平滑，能夠濾掉越多雜訊，但是相對的影像也會越模糊；反之， σ 越小則越不平滑，濾掉的雜訊較少，但是相對的影像也會越清晰。

由於我們的目的是要消除雜訊，但又不能夠使得人體姿勢圖形太過模糊，因此我們選用的 σ 為 1，當 $\sigma=1$ 時，對於一 3×3 高斯平滑濾波器可以得到以下的濾波器係數：

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

而高斯平滑濾波除了可以消除掉背景的雜訊之外，對於人體姿勢圖的內部灰階值亦有平均化的功能，可以使得裡面各點的灰階值與周圍的灰階值變得平滑，如此一來，則對於下一步要進行目標圖型的擷取會有很大的幫助。



2-3 擷取目標圖型

為了要將姿勢圖與背景圖完全的做出區隔，在此我們要對影像進行二元化的處理。

所謂的二元化處理乃是將影像變成只用 0 跟 1 這兩種值來表示之，值為 0 呈現黑色，值為 1 則呈現白色。一來可以使圖上只留下需要的人體坐姿圖（將其變成 1），完整消除不要的人體坐姿圖之背景（將其變成 0），二來是為了簡化後續圖形參數的計算，將姿勢圖中各畫素的灰階值都變成相同的值（同為 1），就可以將人體姿勢圖上的各分點都視為均值（uniform）分布的狀態。

要想將影像做二值化處理，最簡單的方式就是設定一個閥值（threshold），藉由邏輯二元分類的概念，將各畫素灰階值大於此閥值的灰階值令為 1（白色），將小於此閥值的灰階值令為 0（黑色），如此一來就可以將人體坐姿圖與背景單調化，造成完全分離的黑白對比現象。

至於閥值的大小應該要如何設立呢？如果閥值選擇的過小，則會造成除了人體坐姿圖點外還有其他外在背景圖像的干擾；但如果閥值選擇的過大，則會造成人體坐姿圖中有部分的人體圖象消失，使得圖形失真。我們先觀察經過高斯平滑濾波之後的坐姿圖形與其灰階分布圖，由圖 2-3-1 與圖 2-3-2 所示，由於高斯平滑濾波會對於背景的雜



訊有消除的能力，所以由圖 2-3-1 可看出背景（黑色）的部分佔了整張圖很大的比例，對照圖 2-3-2 可看到在灰階值很小的時候，數量非常的多，此即為背景的部分；此外，高斯平滑濾波對於人體姿勢圖內部的灰階值會做平均化的動作，所以我們亦可以從圖 2-3-2 的灰階分佈圖發現，過了某個灰階值之後，其餘灰階值的分佈數量幾乎都差不多，此時就是非背景的部分，故閥值就應設立在灰階值分布陡降的時候。因此我們就去計算何時為灰階值數量變化最大的時候，則閥值就訂為此灰階值，利用這個值來當作劃分影像二元化的門檻，大於此門檻就定義為邏輯 1（即白色），小於此門檻就定義為邏輯 0（即黑色），如此一來，就能夠確實的將經過邊緣檢測後的圖檔轉換成只具有黑或白之二元形態圖檔。而圖 2-3-1 在經過閥值做二元轉化之後，其結果為圖 2-3-3，判斷式可以寫成：

$$\text{If } \tilde{P}(r,c) > \text{threshold} \text{, then } \tilde{P}(r,c) = 1 \quad (2-3-1)$$

$$\text{If } \tilde{P}(r,c) < \text{threshold} \text{, then } \tilde{P}(r,c) = 0 \quad (2-3-2)$$

在本研究中，對於坐姿正的定義是以人體的上半身與下半身呈現垂直的狀態（上半身是筆直且腿部是平放）與放鬆往後靠椅背當作是坐姿正的狀態，往後對於姿勢的判讀都是以這兩類的姿勢做為坐姿正與不正判斷的基準。在當二元化圖檔產生之後，整個人體姿勢圖就完全與背景圖做分割，然而此時我們需要思考一個問題：判斷人體坐姿



的時候，對於腳的部分需要考慮嗎？例如圖 2-3-4 與 2-3-5，若是不看椅子以下小腿的部分，只看椅子以上的部分，兩張圖的姿勢可以說是一模一樣的！且人眼的直觀判斷上，我們會將這兩張圖視為相同的「正」的姿勢，畢竟，當我們在判斷一個人的坐姿為正或不正的時候，其實看的部分也只專注在人體的「上半身」是否有歪斜。因此我們可以說：要判斷人體姿勢正與不正，只需要看上半身的部分即可。然而對於上半身的說法，我們在此分成兩種類型的擷取方式：一類是較寬鬆的上半身擷取，只刪掉小腿以下的部分，保留椅子以上的人體部份；另一類是嚴格的上半身擷取，只看人體的軀幹部份，也就是刪掉小腿、大腿以及手的部分，只剩下頭與身軀的部份。

要想從完整的人體姿勢二元圖型中去擷取出我們所要的目標上半身圖型，必須要先知道小腿、大腿與手會分別分佈在什麼樣的區域中。因此就先以一九宮格的框架來框出人體上半身的區域，之後再從九宮格去對應所要刪除的區域為何，進而達到擷取圖形的動作。然而九宮格要怎麼建立呢？在此我們建立九宮格的方式為從「標準坐姿」下手。所謂的標準坐姿，就是用來判斷坐姿正與不正的依據坐姿。一個判斷系統必須要先有一個標準所在，之後才能夠對於後續資料進行判斷，因此本研究的坐姿判斷系統也必須要先建立起標準坐姿，才能判斷現在所拍攝到的姿勢是正或不正。而標準坐姿的建立方式非常簡

單，就是在一開始的時候請受試者擺出一個正的坐姿，之後將此坐姿進行去背景、高斯平滑濾波與二元化處理之後，結果就會如圖 2-3-6 所示。



當已經有了受試者的標準坐姿之後，接著就是對這個標準坐姿去做九宮格的劃分。由於我們的目標圖型是人體的上半身，因此人體圖形的頭部就令為九宮格的頂端，所坐的椅子的椅面就令為九宮格的底端，這樣就能夠繪出整個九宮格的全高；而寬度部份，是將人體的胸部到背部的距離(也就是人體軀幹的寬度)當作是一格的格寬，之後各往左右延伸相同的距離，就能夠繪出整個九宮格的全寬，圖形如圖 2-3-7 所示。

由圖 2-3-7 可以看出要劃分的位置所在，接下來就是設定劃分的條件。在進入九宮格之前，都是背景的部分，進入九宮格之後才開始進入到人體的姿勢圖裡，此時橫向的畫素值會增多，因此我們設立一條件當作判斷是否進入到人體姿勢圖的頭頂之中：當圖檔的每一行 (row) 中累計的畫素值大於 0.1 最大寬度 ($0.1w_{\max}$) 時，最小的行座標即為頭頂處。因為已經經過二元化處理，所以背景部分的值為 0，唯有在人體圖形內部才會有值，因此判斷式可寫成：

For $r=1:m$

If $\sum_{c=1}^n \tilde{P}(r, c) > 0.1w_{\max}$, then $\min r$ is 九宮格頂部 (2-3-3)



其中 n 為圖形的列數(column number) , m 為圖形的行數(row number)

而在九宮格的底部時，小於此底部為椅子以上的部份，大於此底部為椅子以下小腿的部份，在從椅子接觸面進入到小腿時，此時人體姿勢圖的畫素寬度會驟降很多，因此設立一條件當作判斷是否出去到九宮格底部以外：當圖檔的每一行 (row) 中累計的畫素值大於 0.6 最大寬度($0.6w_{\max}$)時，最大的行座標則為九宮格底部。判斷式可寫成：

For $r=1:m$

If $\sum_{c=1}^n \tilde{P}(r, c) > 0.6w_{\max}$, then $\max r$ is 九宮格底部 (2-3-4)

同樣的，當從背景處進到背部時，此時的縱向畫素會驟增，而從人體的圖形離開胸腹部的部分，此時縱向畫素會驟降，因此設立一條件當作判斷判斷人體的背部與胸腹部：

For $c=1:n$

If $\sum_{r=1}^m \tilde{P}(r, c) > 0.5h_{\max}$, then $\min c$ 人體背部 (2-3-5)

If $\sum_{r=1}^m \tilde{P}(r, c) > 0.5h_{\max}$, then $\max c$ 人體胸腹部 (2-3-6)

其中 n 為圖形的列數(column number) , m 為圖形的行數(row number) 劃分出以上的四條線之後，就如圖 2-3-8 所示，之後再平分高度、延伸寬度，最後九宮格就會變成圖 2-3-9 的樣子。如此一來，九宮格

的劃分動作就完成了。

根據九宮格的分布狀態可以看出大腿就出現在格 9 處，手會出現在格 6 處，小腿為九宮格之下。寬鬆的上半身擷取就只需要刪掉小腿，也就是把九宮格以下的部分的畫素都變成 0(黑色)，如圖 2-3-10 所示。至於刪除椅子上的大腿與手的部分應該要如何刪除呢？由於一般人的上臂圍約為大腿圍的一半，而大腿圍約為腰圍再少 10 毫米，因此我們將大腿寬視為腰的寬度，手的寬度視為大腿寬的一半，而腰的寬度即是我剛剛畫分九宮格的格寬，也就是人體軀幹從背部到腹部之間的距離。因此，如果格九中出現的人體圖形部分之行高開始小於的小於九宮格格寬時，即視為進入大腿的部分；若格六中出現的人體圖形部分的行高開始小於 $\frac{1}{2}$ 九宮格格寬時，即視為進入手臂的部分，當找出手臂與大腿之後，便將其畫素值視為 0 (黑色)，經過這樣的刪除結果，最後人體的姿勢圖就只會剩下軀幹的部分而已，如圖 2-3-11 所示。

由標準坐姿所畫分出九宮格的大小，之後便將此九宮格定型，當受試者陸續產生不同的坐姿時，我們則將完成九宮格產生左右移動來框出人體的坐姿分布。因為椅子視為背景的部分所以不會改變，故高度方面不需要做任何調整，至於左右要移多少呢？在標準坐姿時畫分背部與胸腹部是藉由人體坐姿圖形的縱向畫素值大於最大高度的一





半當做劃分處，因此我們也利用同樣的方式先找出受試者之後的坐姿在縱向畫素大於最大高度 $\frac{1}{2}$ 處的左右格線，然後求這兩條線列座標的中間值當做是中心九宮格新的中心點。如果是正的坐姿，則左右格線切割出來的分佈圖會和標準坐姿圖是一樣的，會切在人體的背部與胸腹部處，如果不是正的坐姿，則切出來的圖形就不會剛好是背部與胸腹部處，由圖 2-3-12 可以看出坐姿不正時所切出來之結果，圖 2-3-13 為此坐姿刪除小腿之結果，圖 2-3-14 則為此坐姿刪除格 6 手的部分與格 9 大腿部分之結果。

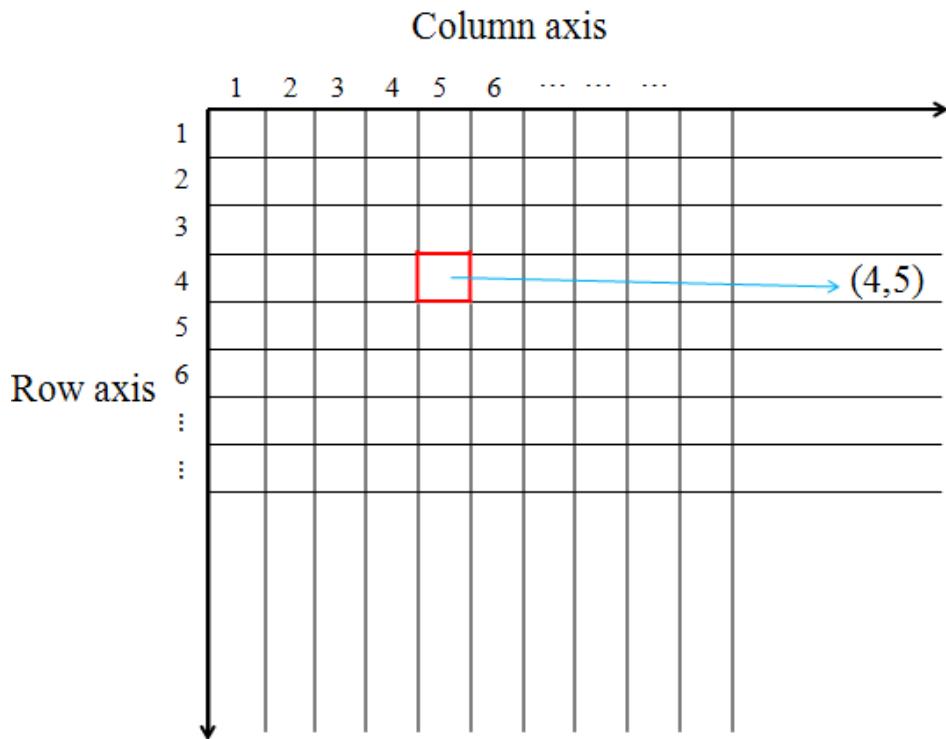
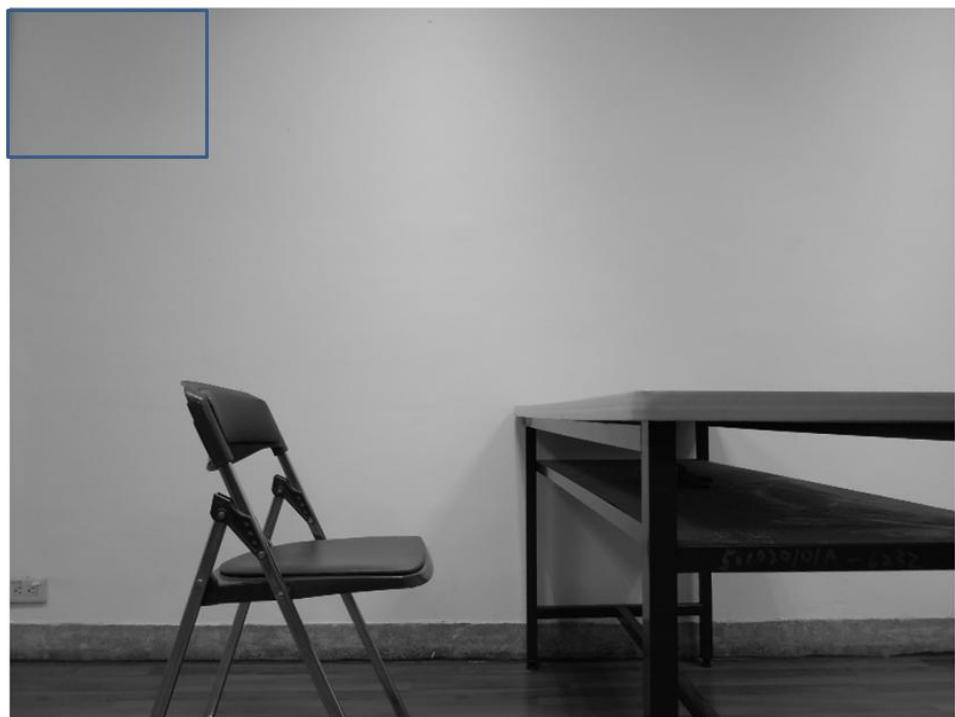


圖 2-1-1 影像座標示意圖



(a) 背景校正光線選取區塊



(b) 人體姿勢圖校正光線選取區塊



(c) 校正後相減去背景圖

圖 2-1-2 光度校正圖

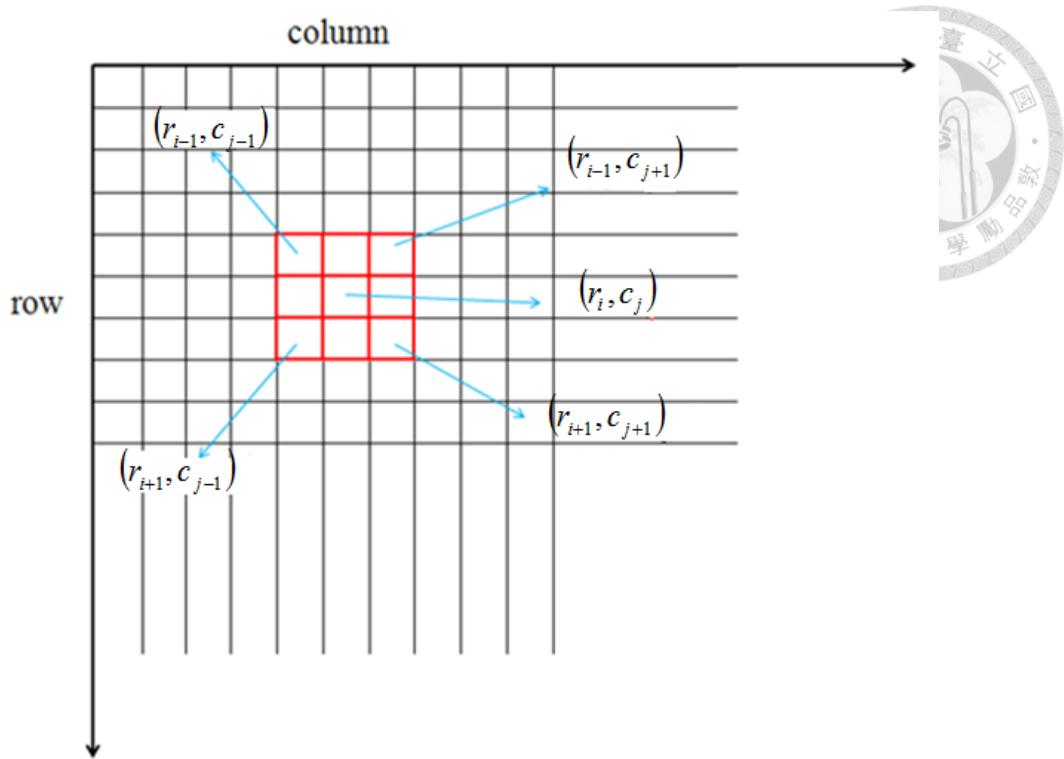


圖 2-2-1 幕罩運算範圍示意圖

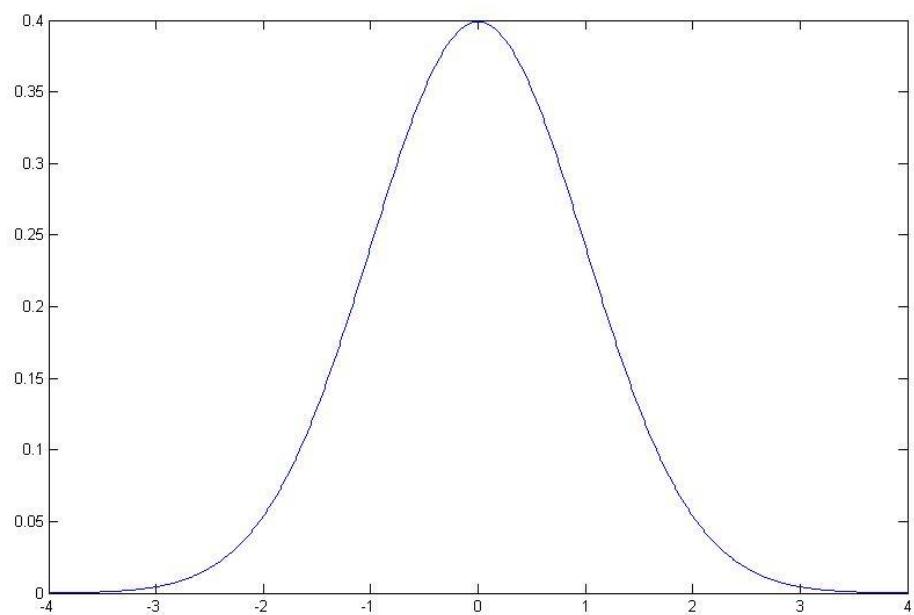


圖 2-2-2 一維的高斯函數圖

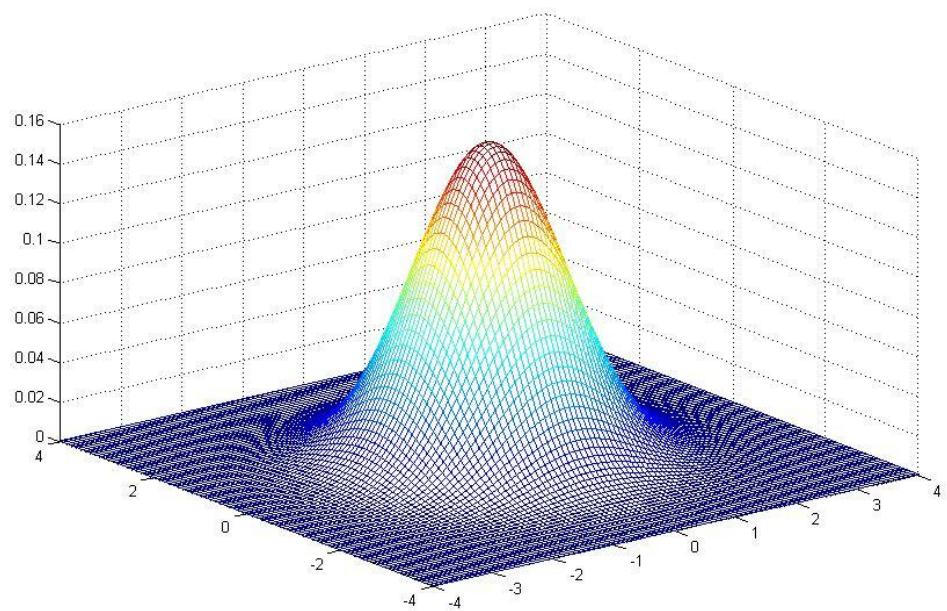


圖 2-2-3 二維的高斯函數圖



圖 2-3-1 高斯濾波後之圖形

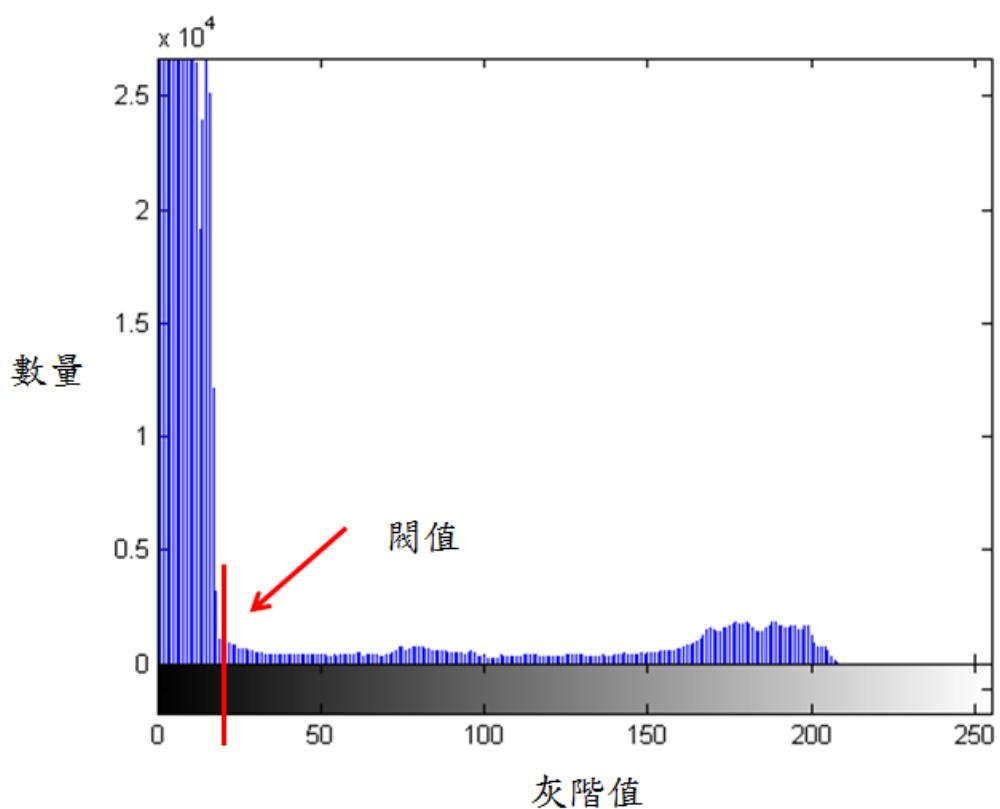


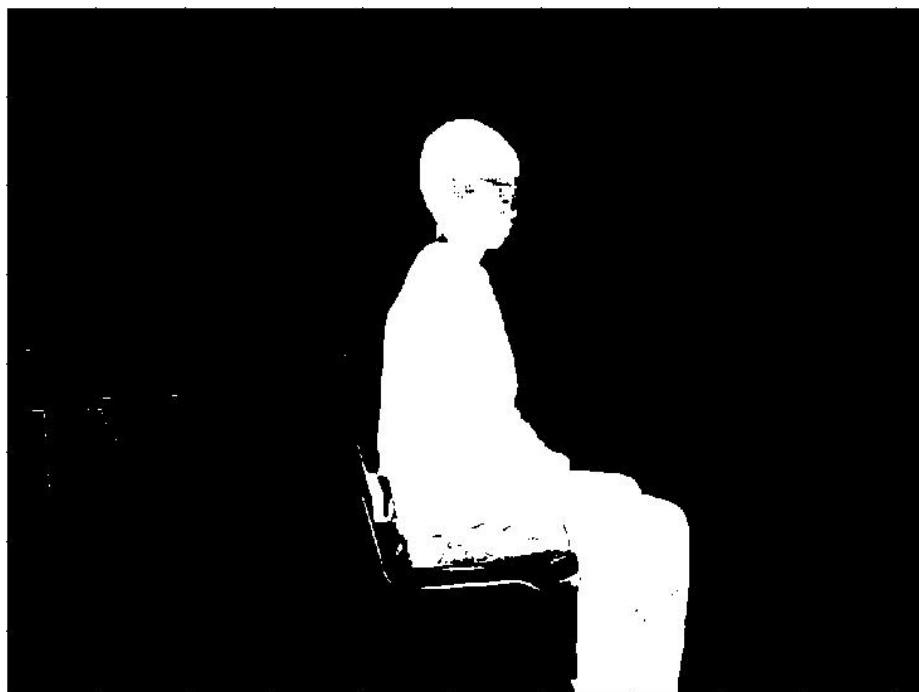
圖 2-3-2 高斯濾波後圖形之灰階分佈



圖 2-3-3 經閾值判斷後之二元圖



(a) 原始圖形

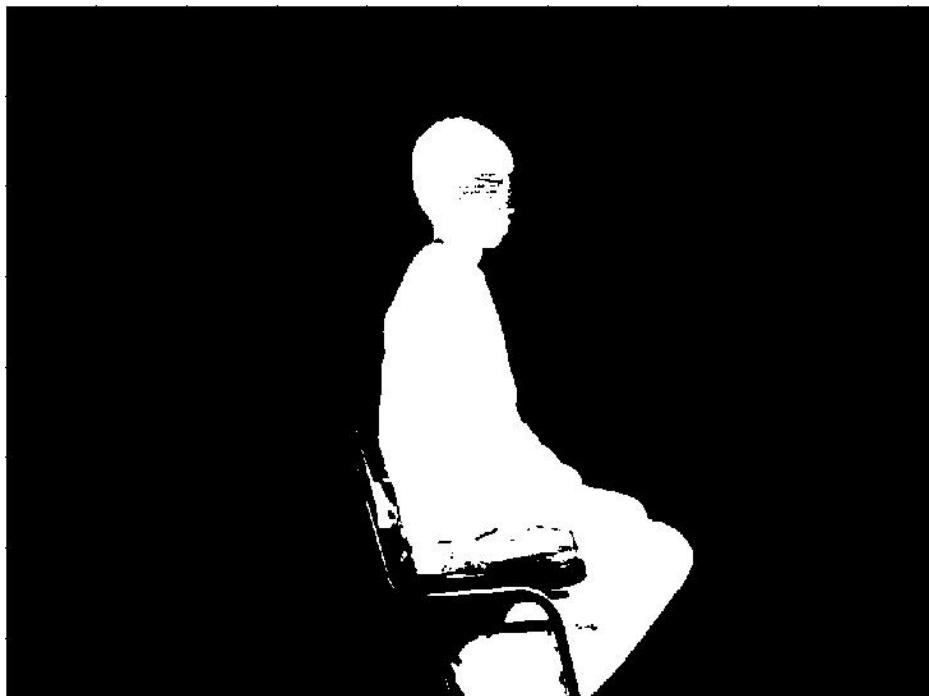


(b) 二元圖形

圖 2-3-4 正姿圖形一



(a) 原始圖形

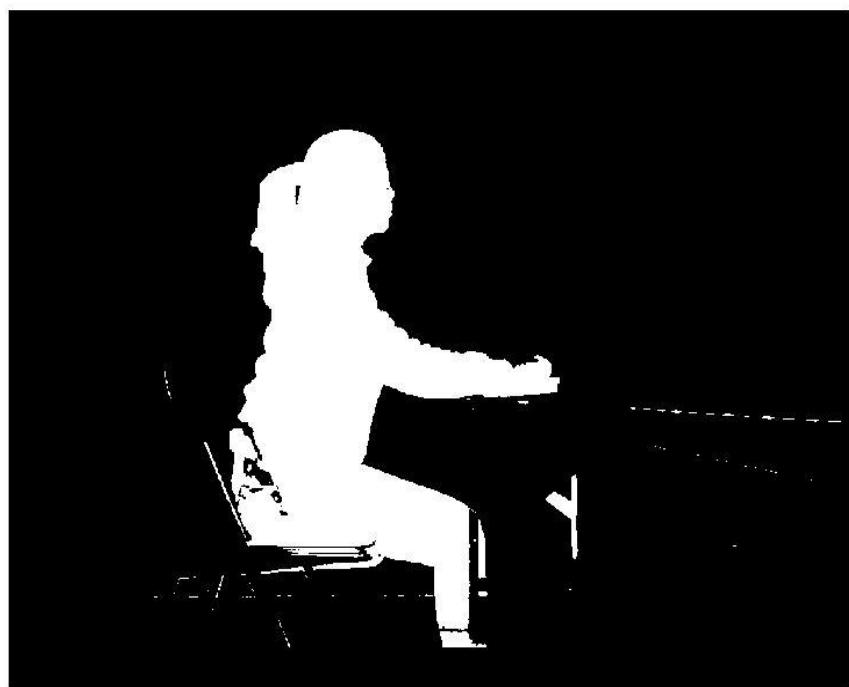


(b) 二元圖形

圖 2-3-5 正姿圖形二

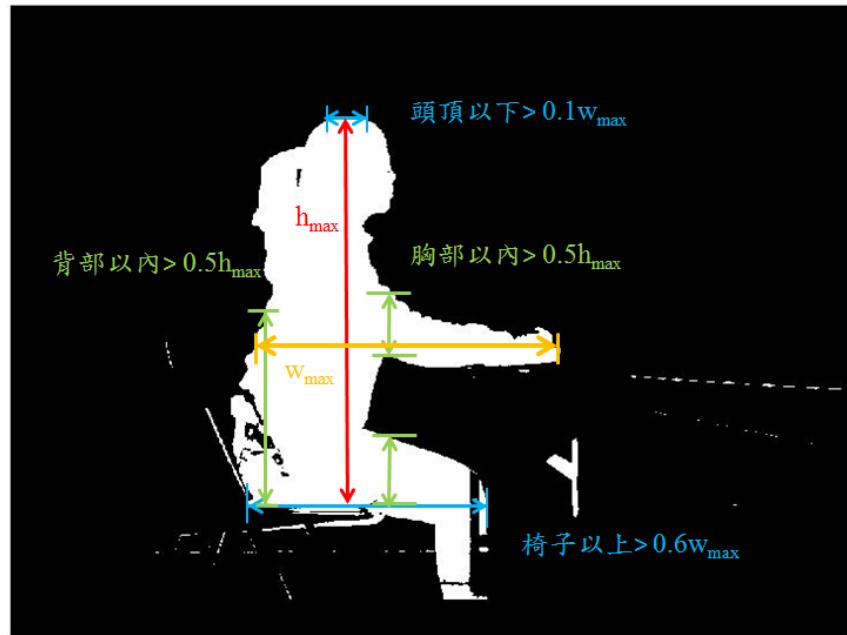


(a) 原始圖形

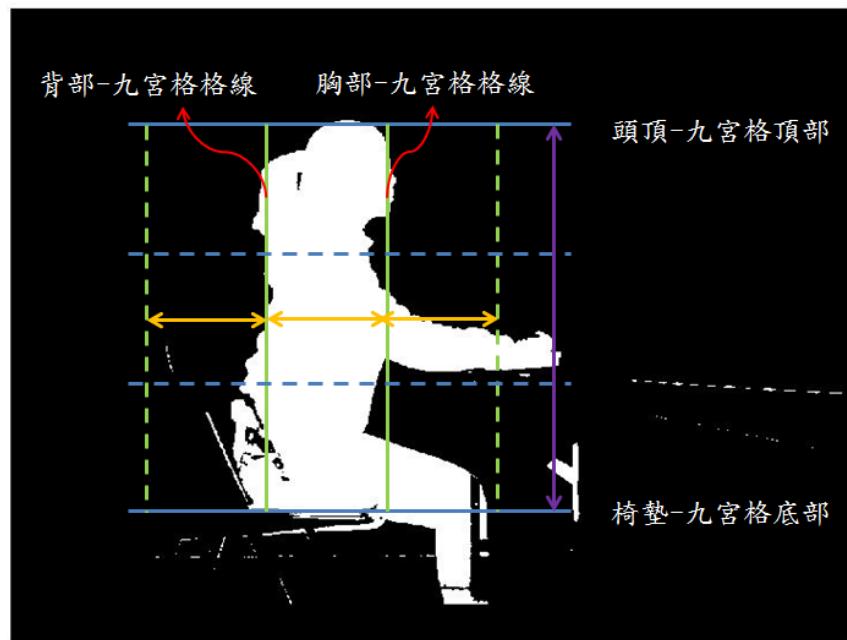


(b) 二元圖形

圖 2-3-6 標準坐姿二元圖



(a) 九宮格判斷條件示意圖



(b) 九宮格劃分圖

圖 2-3-7 九宮格劃分預想圖

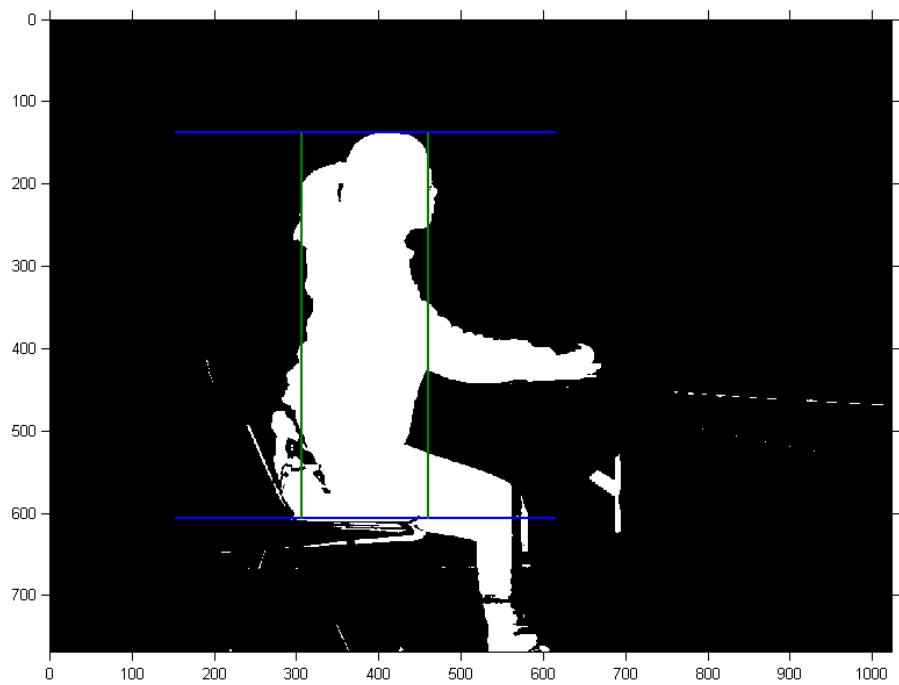


圖 2-3-8 九宮格實際架構圖

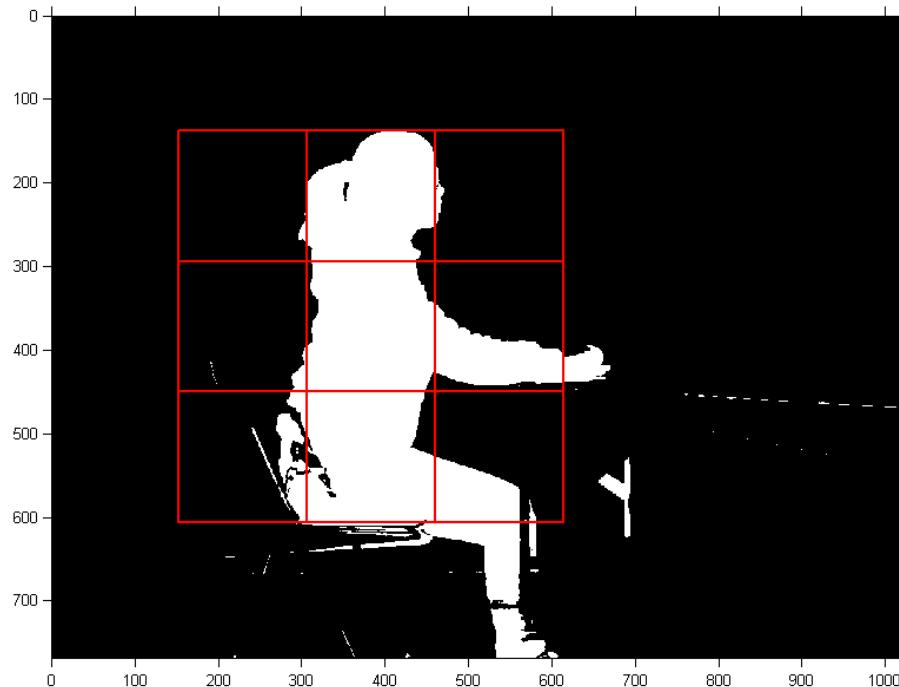


圖 2-3-9 九宮格實際完成圖

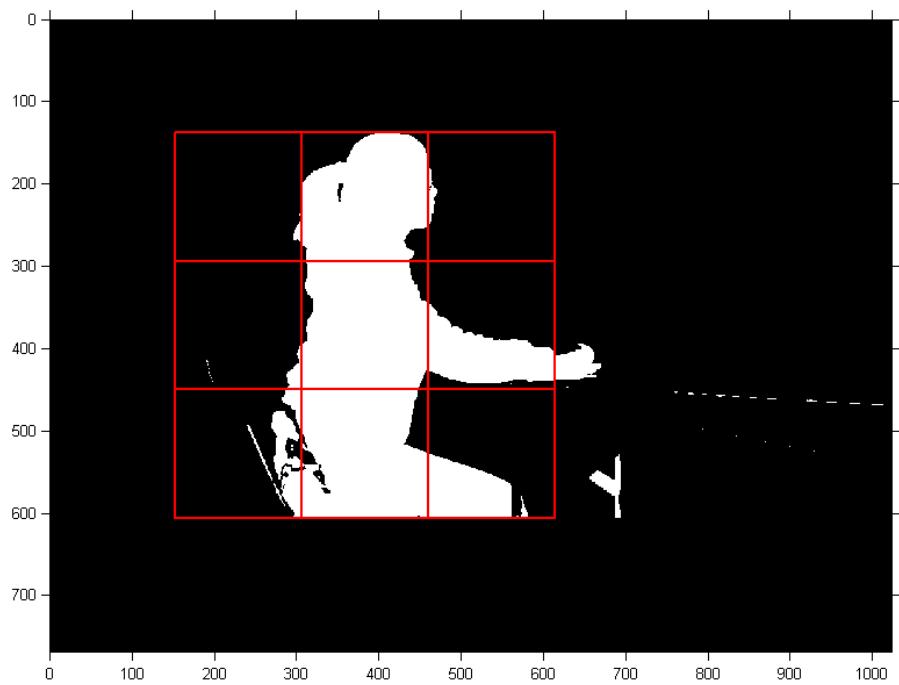
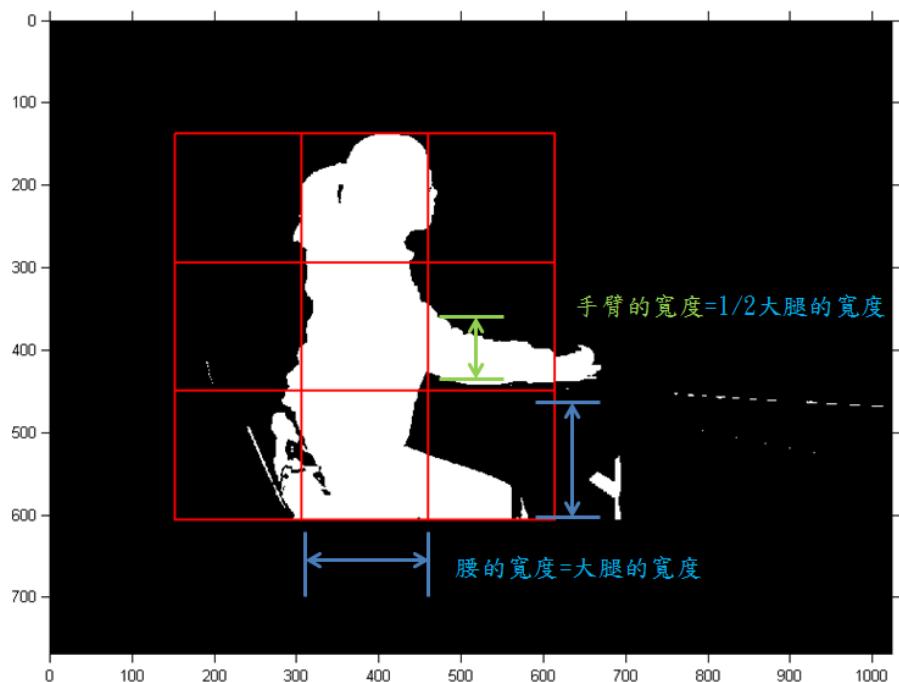
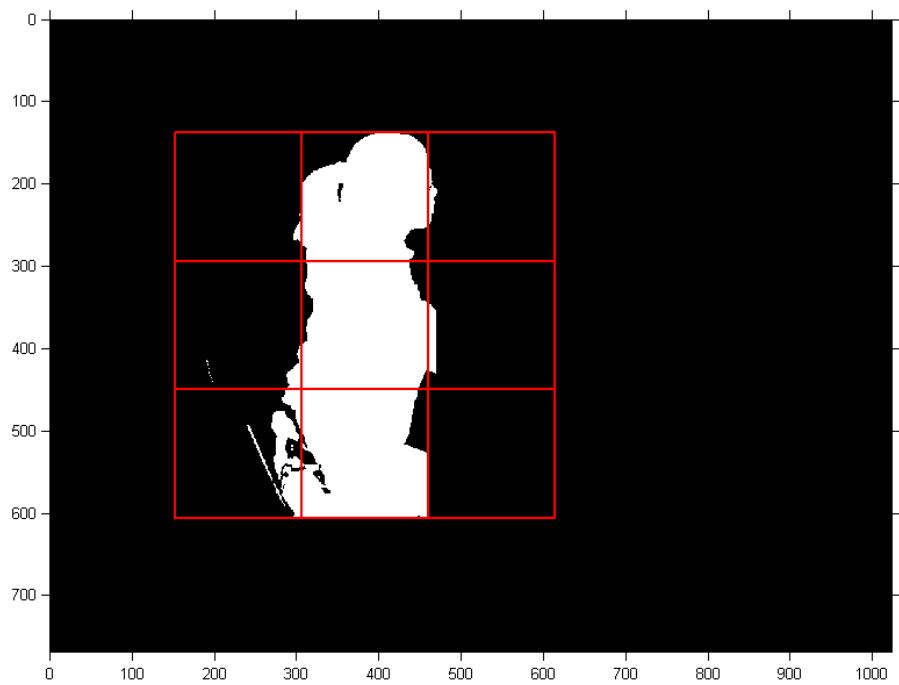


圖 2-3-10 刪除小腿之上半身圖



(a) 欲刪除之部分



(b) 實際刪除之結果

圖 2-3-11 刪除小腿、手與大腿之上半身圖

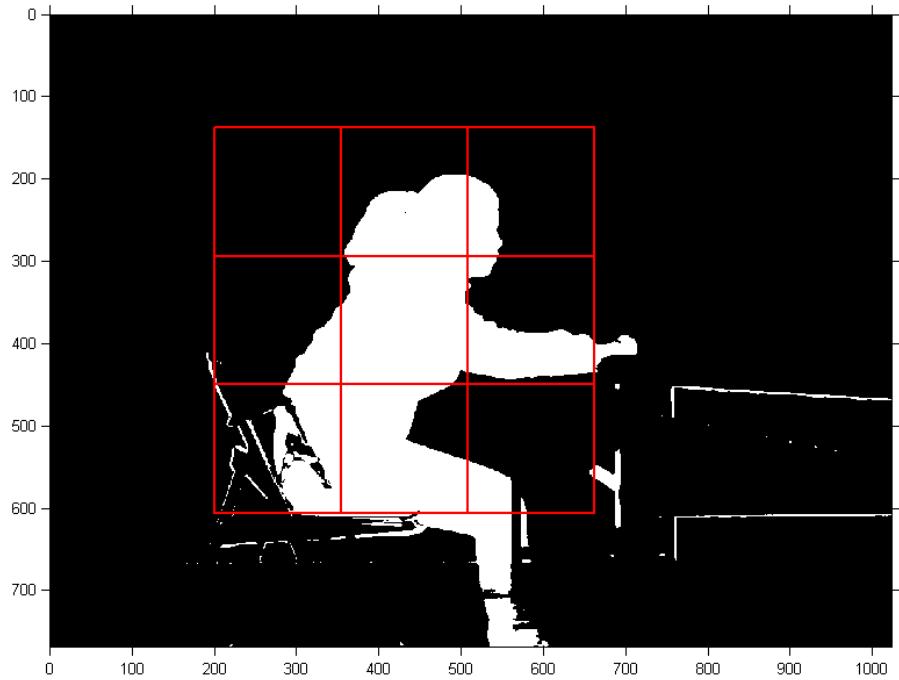


圖 2-3-12 非標準坐姿劃分九宮格圖

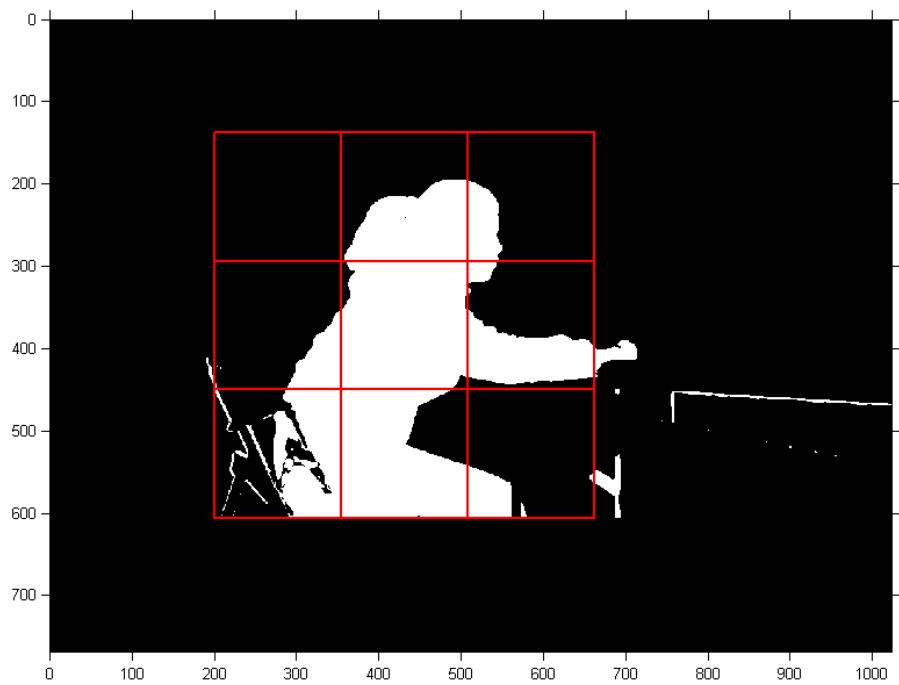


圖 2-3-13 非標準坐姿刪除小腿之上半身圖

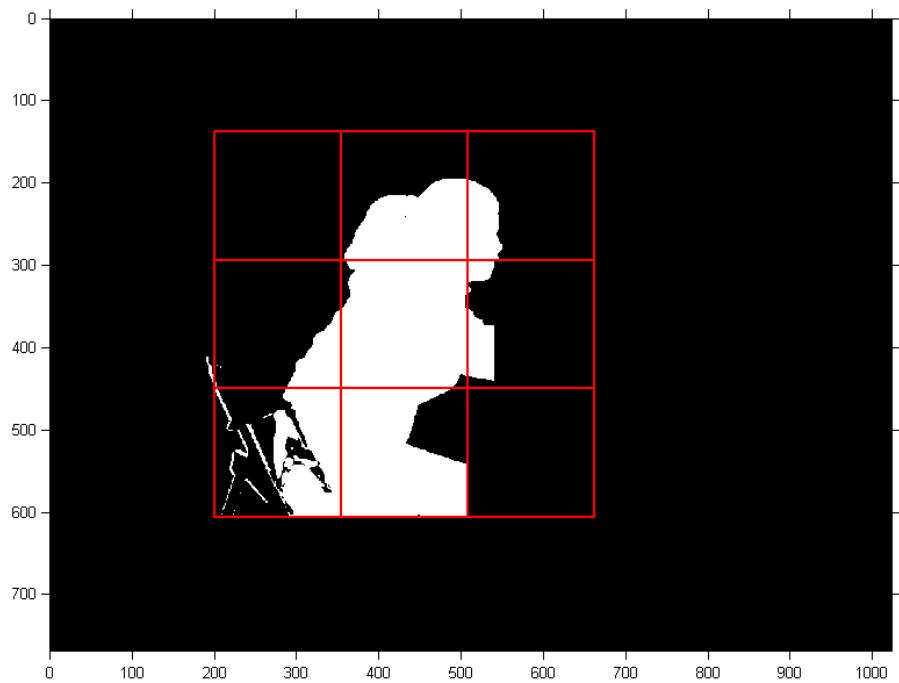


圖 2-3-14 非標準坐姿刪除小腿、手與大腿之上半身圖

第三章 類神經網路



此章節是人體坐姿判讀流程的第二步驟，本章節會先介紹生物神經網路，之後介紹組成類神經網路的結構與機制，最後則是介紹本研究判斷系統所使用的倒傳遞類神經網路系統與其內部構造。

類神經網路是現今非常重要且也廣為人使用的一項技術，屬於人工智慧（Artificial Intelligence，AI）的一個分支。人工智慧的發展最早可以追溯到第二次世界大戰時期，在當時為了解決軍事情報上的問題，科學家們於是就研究發展出具有智慧的機器。早期的人工智慧研究形式主要在於模仿人類逐步思考與推演的過程，像玩棋盤遊戲一樣推衍如何走下一步的思考與判斷過程。而到了 1980 與 1990 年代之後，統合機率與經濟學的概念，更進一步發展出能夠解決具有不確定性資訊的方法。而現今所謂的人工智慧，是指將電腦科學、心理學、生物學、工程學、數學等知識作為基礎的學科，把人的智能加以電腦化的過程，使電腦如同人一般也能夠同時具有思考、學習與解決問題之能力。常見的應用在於感測器、專家系統、語音辨識、人臉辨識系統、機器手臂、指紋辨識等等。類神經網路就是在人們了解腦的思考與學習模式之後，所提出的一套計算模式。其目的在於希望能夠藉由具有大量計算能力的電腦，來模擬出人類神經細胞網路運作過程的一

套科學運算方法。



3-1 生物神經網路

類神經網路源自於模擬人類腦神經系統思考與判斷方式，因此若要了解類神經網路，則必須先探討與了解生物神經系統的運作方式。

在人體中，神經系統掌管著人體所有的行為：從起床時的刷牙、洗臉，動手拿起早餐、吃早餐，搭車時的站立、走路，中午坐在椅子上休息，到晚上上床睡覺等任何動作，全都離不開神經系統。神經系統就是人體中的一個管制中心，同時監督、調整與管理我們身體的各個部分。若是依照功能的角度來看，神經系統主要提供三大項功能：

1. 受器接收訊號 (sensory input)

受器接受訊號即是指身體的感受器將對於外界所感受到的刺激與資訊傳入神經系統中。

2. 整合 (integration)

整合即是對於傳入的訊號做出分析與處理，進而判斷出對於此時外界之刺激應該要做出怎樣的動作或防禦措施來回應。

3. 動器產生動作 (motor output)

動器產生動作則是指整個神經系統的輸出，由於在整合過程中已經決定好要做出什麼樣的動作，則神經系統就會將此判斷結果的訊息傳給動器，讓身體的動器對外界的刺激做出適當的回應。

如圖 3-1-1 所示，眼睛看到一杯水即是眼睛中光的感受器收到的訊號，並將此訊號傳給神經系統；整合即是當大腦接受到此訊號時，做出要伸手拿起杯子喝水的決定；而當神經系統將要拿起杯子喝水的訊息傳遞給手部的肌肉，則手就會伸出去拿起杯子來。主宰像上述這樣一連串由受到刺激到產生相對回應的過程，就是神經系統作用的範疇。

而構成神經系統的基本單元為神經元 (neuron)，又稱神經細胞。每一個神經元都包含了三個主要的部分：細胞本體 (cell body)、樹突 (dendrite) 和軸突 (axon)，如圖 3-1-2 所示。神經元中間為細胞本體，內含有細胞核 (nucleus) 和細胞質 (cytoplasm) 等等構造，主要目的在於維持神經細胞的基本所需，如蛋白質的合成和能量的製造。樹突和軸突則是神經細胞特化的突起結構，此兩者的差別在於對訊號傳遞的方向與外形。就傳遞方向來說，樹突的作用是蒐集從其他神經元傳來的訊息並將其由樹突突起的末端傳入細胞本體中；軸突則是將細胞本體送出來的訊息傳送至其他神經元中。就外形來說，樹突由於為了要增加從外界接受各種刺激的能力，導致在細胞本體的表面上產生了許許多多的樹突；相反的，在大部分的神經元中，只會有一支長長的軸突，但軸突在末端會分成許多分支，目的在於將訊號能夠有效的傳遞給下一個神經元。

而在軸突末端特化的許多分支，稱之為突觸末梢 (synaptic terminal)，突觸末梢會藉由釋放神經傳導物質 (neurotransmitter) 的方式來將神經訊號傳給另一個神經元。將神經訊號傳出的軸突末梢稱為前突觸部位 (presynaptic site)，在下一個神經元接收神經訊號的部位稱為後突觸部位 (postsynaptic site)，此兩者合起來統稱為突觸 (synapse)，此結構如圖 3-1-3 所示。在人體內有約有 $10^9 \sim 10^{10}$ 個神經元，神經元間之突起相互連接形成非常複雜的網路，而這樣複雜的神經網路即是人體內所有訊息傳遞和處理的系統。

神經元傳遞訊息的方式主要是藉由化學物質的傳遞來達成，其中傳遞的內容是一種電的信號。當生物體沒有傳遞訊號時，會說神經元是處於休止狀態 (resting state)，此時細胞膜內電荷總和為負電，細胞膜外電荷總和為正電，細胞膜內膜外的電壓差稱為膜電位 (membrane potential)，如圖 3-1-4 所示，而細胞在休止狀態下時膜電位為 -70mV 。

而當神經元接受外界刺激，或者是從其他神經元傳來的訊號，都會使得神經元表面的離子產生流動。若在某個時刻，當一神經元所受到的刺激使膜電位值超過了此神經元的門檻值 (threshold，膜電位的門檻值為 -55mV)，則該神經元就會產生動作電位 (action potential)，如圖 3-1-5 所示，進入一種特定形式的改變。

由圖 3-1-5 可看出，所謂的動作電位指的就是膜電位急速上升後又下降至略低於休止狀態膜電位的一種過程，在該神經元膜電位急速上升的期間，可以引發附近其他神經元的膜電位超過門檻值，因此訊號就會達到傳遞出去的效果。其生成原因是主要是因為 Na^+ 離子和 K^+ 離子的流通。細胞膜上有許多種通道，其中讓離子通過的就稱為離子通道 (ion channel)。人體細胞中所需的 K^+ 離子量大於 Na^+ 離子量，因此 K^+ 離子濃度是細胞內大於細胞外， Na^+ 離子則是細胞內小於細胞外。

當受到刺激時，會有一部分的 Na^+ 離子通道開啟，讓 Na^+ 離子通過 Na^+ 離子通道進入細胞內，提升細胞內電壓值，使得膜電位值會上升。而有另外一部分的 Na^+ 離子通道是要在膜電位值上升時才會被打開的，因此在有一部分的 Na^+ 離子進入細胞內時，膜電位值會上升，之後就會被啟動這一部分的 Na^+ 離子通道，進而又會再去開啟更多因膜電位值上升才會打開的 Na^+ 離子通道，於是就變成一種正向的回饋作用。是故當外界刺激讓膜電位上升到門檻值時，就會引發後續一連串正向回饋現象，反之，則不會。

若外界刺激能讓膜電位上升到門檻值，此時 Na^+ 離子通道的正向回饋會使得 Na^+ 離子通道大量打開，引發細胞外的 Na^+ 離子順著濃度差大量進入細胞內，當細胞內外 Na^+ 離子濃度幾乎一致時，則 Na^+ 離



子通道就會關閉，此時膜電位會上升到約為+50mV。在膜電位上升至約為+30mV 時， K^+ 離子通道會被開啟，由於在細胞內 K^+ 離子濃度是大於細胞外的，所以 K^+ 離子通道開啟主要目的是將 K^+ 離子由細胞內流出到細胞外，藉由讓細胞內的正離子離開到細胞外的方式，來將膜電位回復到休止狀態的-70mV。然而，由於 K^+ 離子通道開啟所需的反應時間比 Na^+ 離子通道多得多，因此開啟與關閉的速度皆比 Na^+ 離子慢的多。故從 K^+ 離子通道在膜電位為+30mV 時接到訊號到真正完全開啟時，膜電位已經上升到約為+50mV；而當膜電位回復至原本休止狀態（膜電位為-70mV）， K^+ 離子通道從接收到訊號至真正完全關閉時，膜電位已經降至-75mV。因為 Na^+ 離子通道與 K^+ 離子通道皆已關閉，故最後細胞需要利用細胞膜上的鈉鉀幫浦（sodium/potassium pump），藉由消耗能量的方式傳遞細胞內外的 Na^+ 與 K^+ 離子，將膜電位調回至休止狀態。

整個動作電位之產生符合了全有全無律（all or none event），只要膜電位到達門檻值，不論是剛好到達或是超出許多，所產生的動作電位都是一樣的，若沒有到達門檻值，則完全沒有動作電位之產生。

而生物對新事物的學習方式，基本上是藉由神經元間連結強弱的改變或者是讓原本沒有連結的神經元間產生新的連結，以此來達到學習新事物的能力。例如說：在原本平坦的房間地板中凹陷了一個洞，



由於每天行走時往往都會不小心踏到此凹洞，自然就會將腳接受到的訊息轉成訊號傳進大腦中，不斷學習，大腦神經元就會產生新的連結，於是將使得人從原本的不習慣踏到凹洞到慢慢習慣於腳下的這種觸感。因此，要是之後凹洞被填平了，又變回了最初的平地，本該是習慣平地的自己與雙腳，反而會變得不習慣平地給雙腳的一切觸感，那是因為平地的刺激太久沒有存在，會使得記憶消退了，也就是大腦神經元彼此間的連結變弱了，而凹洞的連結卻被建立了起來。

生物對於環境的改變有非常良好的適應性，且不需要用到任何的數學程式計算，只需要給予數次上的訓練，當神經元間的連結產生之後，則目標就完成了。

3-2 類神經網路

類神經網路是一模仿人體神經系統傳導方式的電腦網路系統。由於神經元是神經系統中最小的傳遞以及處理訊息單位，故在類神經網路中，也採用了神經元的概念，人工神經元就如同神經系統中的神經元一般，為運算訊息的基本單元體。圖 3-2-1 為一個人工神經元之模型。人工神經元主要分成三個部分：

1. 一組權重（對應至生物神經元的突觸部分）：

權重 w_{jn} 主要是模擬不同神經元間的連結程度。由於各輸入訊號

x_1 、 x_2 、 \dots 、 x_n ，進入到神經元中時，會因為彼此的重要程度不

同，則各自就會對應到不同的權重值，當權重 w_{jn} 為正值且越大，

就代表這兩個神經元間的連結與增益效果都越強；當權重為負值

且越大，則代表這兩個神經元間的連結與抑制效果都越強；當權

重 w_{jn} 值越接近 0，則代表這兩個神經元間的連結越弱。

2. 輸入訊號疊加器（對應至生物神經元的細胞本體部分）：

在於模擬當生物神經元接受到自外界傳來的刺激時，膜電位的總

變化量，也就是指當輸入訊號經過不同的權重（連結）加權之後，

以疊加器將所有訊號做線性加總之結果。

3. 活化函數（對應至生物神經元的門檻值部分）：

原本是用來模擬神經元的門檻值，但現今不只是用來模擬門檻

值，也被用來限定人工神經元輸出值的大小範圍，因此活化函數

又可以稱為壓制函數。通常一個神經元的輸出活化函數是將輸出

值做正規化的動作，輸出範圍為 $[0, 1]$ 或者是 $[-1, 1]$ 這兩種區間。

此外，人工神經元除了以上的三個部份之外，通常會再額外給定

一個偏權值，寫成 b_j 的形式。偏權值主要的目的在於控制活化函數的

網絡輸入，當偏權值為正的，則就代表增益輸入值，若偏權值為負的，

就代表抑制輸入值。就數學上而言，一個神經元 j 對於訊息的處理可

以用以下公式表示之：

$$net_j = \sum_{i=1}^n w_{ji} x_i + b_j \quad (3-2-1)$$

$$y_j = f(net_j)$$

(3-2-2)

其中 y_j 為第 j 個神經元輸出訊號， x_i 為輸入訊號， w_{ji} 為神經元 j 對於第 i 個輸入值的突觸加權值， net_j 為輸入訊號後訊號線性疊加器的輸出， b_j 為偏權值， $f(\cdot)$ 為活化函數， $f(net_j)$ 為經過活化函數作用之後的輸出值，也等於 y_j 。

常見的活化函數有下列四種形式，如圖 3-2-2 所示：

1. 門檻值函數

$$f(net_j) = \begin{cases} 1 & \text{if } net_j \geq 0 \\ 0 & \text{if } net_j < 0 \end{cases} \quad (3-2-3)$$

2. 片段線性函數

$$f(net_j) = \begin{cases} 1 & \text{if } net_j \geq +\frac{1}{2} \\ net_j + \frac{1}{2} & \text{if } -\frac{1}{2} < net_j < +\frac{1}{2} \\ 0 & \text{if } net_j \leq -\frac{1}{2} \end{cases} \quad (3-2-4)$$

3. S 形函數

(3-2-5)

$$f(net_j) = \frac{1}{1 + e^{-\alpha net_j}}$$

4. 雙曲線函數

$$f(net_j) = \tanh(net_j) \quad (3-2-6)$$

活化函數一般都不為線性函數，目的是為了保留非線性特性，因

為若是輸入資料有非線性特性而活化函數為線性函數時，則會嚴重影響到後續類神經網路模擬與輸出時的結果。

利用人工神經元所組成的類神經網路，其運算能力和網路架構以及訓練的學習演算法都有非常大的關係。網路架構的型態會影響到系統的學習能力，而學習演算法則是藉由類似經驗法則的方式來調整各個神經元間彼此的連結（權重）強弱關係。因此，一個好的類神經網路的建立，必須從網路架構與學習演算法這兩方面下手。

3-2-1 類神經網路系統架構

類神經網路的架構型態是描述整個類神經對於外來訊息如何處理的過程，當一類神經網路中的神經元數目太少，則會無法解決較複雜的問題，若神經元太多，除了需花費較多時間之外，還有可能會產生過度敘述（over-fitting）的現象。因此，神經元數目的多寡，一般是根據問題的複雜程度來決定，而若要知道確切的神經元數目，除了參考過去學者們的經驗累積之相關研究結果之外，至目前為止所能採取的方式是不斷的利用試誤法（try and error）才能得知。

建構一類神經網路除了神經元數目需要掌握之外，再來是需要決定網路間的連結形式為何。現今類神經網路的連結型態主要可分為前饋式類神經網路（feedforward network）與回饋式類神經網路（feedback network）這兩大類。

前饋式類神經網路之架構如圖 3-2-3 所示，是一種只往前傳的單一連結方向之型態，因此所有的神經元都不會將資訊往後傳。而前饋式類神經網路有可以細分為兩個類別：一是單層前饋式類神經網路 (single-layer feedforward networks)，另一種是多層前饋式類神經網路 (multilayer feedforward networks)。

1. 單層前饋式類神經網路

這是形式最簡單的類神經網路，架構如圖 3-2-4 所示。所有的神經元都平行的排成一列，訊息會先經過一個權重的處理再傳入神經元內，接著訊息會在神經元內做加總，最後再搭配著活化函數後將訊息輸出。所謂的「單層」表示訊息只經過一層的神經元處理，由於輸入層並沒有對訊息做任何處理，所以單層指的是輸出層。

2. 多層前饋式類神經網路

多層前饋式類神積網路和單層前饋式類神經網路的最大不同即是在於，多層前饋式類神經網路中含有不只一層的神經元，也就是除了最後一層的「輸出層」之外，中間還多了至少一層的「隱藏層」，架構如圖 3-2-5 所示。

隱藏層的功能主要為調節的作用，目的在於解決單層神經元所無法解決的問題，例如高維度的非線性問題或是邏輯電路中的 XOR 與 NXOR 計算等，因為多了隱藏層的處理，將神經元做了更多額外的連

結，從圖 3-2-5 可以看出：前一層的神經元會和所有下一層中每一個神經元都有連結，我們稱之為完全連結 (fully connected)，這樣的連結會使得類神經網路具有較全面性的觀點。

然而，不管是單層前饋式類神經網路或是多層前饋式類神經網路，同一層的神經元是不會彼此有所連結的，也就是連結方向是層層遞進且只會往下一層的方向傳遞。

回饋式類神經網路和上述前饋式類神經網路的不同處在於傳遞的方向，如圖 3-2-6 所示，回饋式類神經網路的傳遞方向除了可以將訊息往下一層傳之外，也可以同一層間的神經元彼此互傳，更甚至可以是下一層的神經元將訊息傳到上一層神經元。回饋式類神經網路很常被用來處理動態平衡或者是具有時間性質的訊息，例如降雨量的累積，因為在神經元彼此間傳遞訊息時會產生一時間延宕，如圖 3-2-7 所示，不同時間所產生的訊息既是這一刻的輸出又是下一時刻的輸入值，因此回饋式類神經網路往往又被稱為遞迴式網路。回饋式類神經網路因為具有不斷回饋的能力，會加強網路的學習能力與表現效果，尤其是非線性動態系統，回饋式類神經網路的詮釋能力幾乎是最好的。

3-2-2 類神經網路學習演算法

由於類神經網路就是以模擬人的大腦目的，而人要學習一項新事



物憑藉的就是從經驗法則中去延伸，學習演算法就是要模擬這樣的過程。在上述章節 3-1 部分提過：生物對新事物的學習能力，是根據神經元間連結強弱的改變或者是讓原本沒有連結的神經元產生新的連結。因此學習演算法可以說是藉由調整權重的方式來模擬各神經元間彼此連結強度改變的現象。因此從改變權重值的角度來看，學習演算法主要可以分成兩大類：非監督式學習（unsupervised learning）與監督式學習（supervised learning），兩者差異如圖 3-2-8 所示。

非監督式學習是從觀察和發現中學習的一種過程。因為一開始並不知道標準答案為何，所以是藉由觀查到很多現象之後才發現其中的連結。例如牛頓是因為觀察到蘋果成熟會往下掉，之後才進而發現地心引力，然而在牛頓之前並沒有人知道這個定理的存在。因此非監督式網路學習演算法中也是一樣，一開始並沒有給予何謂「對」的答案，也就是沒有目標輸出值，因此就無法從誤差的角度去改善輸出值，只根據現有的資料中去找出內部的規律，故非監督式網路常被應用在尋找輸入資料的分類上。

常見的非監督式學習演算法有：赫賓學習法（Hebbian learning rule）、贏者全拿學習法（Winner-Take-All learning rule）等等。

1. 赫賓學習法（Hebbian learning rule）：

是最早發現且最有名的學習方式，此方法之學習示意圖如圖

3-2-9 所示。赫賓學習法是由心理學者赫賓 (Hebbian, 1949) 提出的概念所發展出來的：假設神經元 A 的軸突足以刺激鄰近神經元 B 時，若不斷的給予刺激會激發神經元 B 的新陳代謝，會使神經元 A 對於神經元 B 的效果增加。

此概念後來被 Stent (1973) 推導成兩個神經元傳送法則：

1. 若兩個連結的神經元同時被激發，則彼此間的連結強度會增

加。

2. 若兩個連結的神經元不是同時被激發，則連結強度會變弱或

者是消失。

因此我們根據上述法則可以知道：兩個神經元的連結強度與這兩個神經元的輸入與輸出值有所關係。假設輸入的值總共有 n 個，設 w_{ji} 表示第 j 個神經元與第 i 個輸入之連結權重，此時輸入的值為 x_i ，而第 j 個神經元輸出值為 y_j ，權重對於第 k 次的迭代修正值為：

$$\Delta w_{ji}(k) = \eta y_j(k) x_i(k) \quad (3-2-7)$$

其中 η 為一常數，稱為網路學習速率，目的用來調整每一次迭代修正值的大小，根據圖 3-2-9 中我們可以看出第 j 個神經元的輸出值是權重向量和輸入向量做線性相乘積加總的函數，即

$$net_j = \sum_{i=1}^n w_{ji} x_i = \mathbf{W}_j^T \mathbf{X} \quad (3-2-8)$$

$$y_j = f(net_j) \quad (3-2-9)$$



故權重修正值可以表示成：

$$\Delta w_{ji}(k) = \eta f(\mathbf{W}_j^T \mathbf{X}) x_i(k)$$

$$w_{ji}(k+1) = w_{ji}(k) + \Delta w_{ji}(k)$$

由上式可以看出如果輸出值 y_j 與輸入值 x_i 相乘為正則權重修正量 Δw_{ji} 亦為正，表示新權重比舊權重大，也就意味著神經元的連結增強；反之，若輸出值 y_j 與輸入值 x_i 相乘為負，則就表示神經元的連結減弱。在初始狀態時會將權重值先設為 0，這樣才能最直接看出權重修正值的變化。

2. 贏者全拿學習法 (Winner-Take-All learning rule)：

目的在於從眾多競爭的神經元中找出優勝者，並且只對於連結該優勝神經元的權重做修正。整個運作的形式如圖 3-2-10 所示，在同一層的神經元中找出對於輸入訊息有最大反應的神經元，則此神經元就稱為優勝神經元 (winner)，只有該神經元的權重值能夠去做調整，而其他神經元則維持原本的權重值。

假設優勝神經元的權重向量為 \mathbf{W}_m

$$\mathbf{W}_m = [w_{m1} w_{m2} \cdots w_{mi} \cdots w_{mn}]^T$$

則權重調整的量為輸入值與原本權重值的差值去乘上一學習速率

$$\Delta w_{mi} = \eta(x_i - w_{mi}) \text{ for } i = 1, 2, \dots, n \quad (3-2-12)$$

其中 η 為學習速率，在迭代過程中會逐漸縮小且 $\eta > 0$ ， x_i 為第 i 個輸入項， Δw_{mi} 為優勝神經元 m 連結第 i 個輸入項 x_i 的權重調整值， w_{mi} 為優勝神經元 m 和輸入項 x_i 的原本連結權重。

要如何從眾多神經元中找出優勝者呢？常用的方式為找出各神經元的連結向量與輸入向量最相近者，或者是尋找連結向量與輸入向量距離最短者亦可，則數學表示式如下：

$$\mathbf{d}(\mathbf{X}) = \max(\mathbf{W}_j^T \mathbf{X}) \quad \text{for } j = 1, 2, \dots, p \quad (3-2-13)$$

或

$$\mathbf{d}(\mathbf{X}) = \min\left(\|\mathbf{W}_j^T - \mathbf{X}\|^2\right) \quad \text{for } j = 1, 2, \dots, p \quad (3-2-14)$$

其中 $\mathbf{d}(\mathbf{X})$ 表示優勝神經元之結果， p 則代表此層中有 p 個神經元在競爭， $\mathbf{W}_j^T \mathbf{X}$ 為第 j 個中神經元其權重向量和輸入向量做線性相乘後加總的結果，由式 3-2-8 而來，因此我們可以根據上述的式子來找出優勝神經元為哪一個神經元。

監督式學習簡單來說就是從範例中學習的一種過程。以人們學習知識的過程來說，就是藉由已經知道答案的題目，一次又一次的反覆練習，直到明白每一個參數在題目中所扮演的角色與定位所在，類神經網路亦是如此。在經過每一個確定輸入值與輸出值的範例訓練，不斷調整各個權重的大小，直到最後輸出值與我們已知的輸出值誤差小於一定的容許值為止，則此時類神經網路的學習才算完整。其每一神

經元權重學習的示意圖如圖 3-2-11 所示，藉由將回傳的學習訊號 r (目標輸出值與實際網路輸出值之誤差)、輸入向量 \mathbf{X} 與學習速率 η 做統合的方式來達到調整權重值之結果。

常見的監督式學習演算法有：最小均方演算法 (Least Mean Square algorithm, LMS)、感知器學習法 (perceptron learning rule)、Delta 學習法 (Delta learning rule)、Grossberg 學習法 (Grossberg learning rule) 等等。

1. 最小均方演算法 (Least Mean Square algorithm, LMS)：

由於最小均方演算法是 1960 年由史丹佛大學的 Widrow 與 Hoff 兩位教授所發展出來的，故又稱為 Widrow-Hoff 學習法。有許多種類的類神經網路都是使用此方法。如圖 3-2-12 所示，關於權重的數學式為：

$$net_j = \sum_{i=1}^n w_{ji} x_i = \mathbf{W}_j^T \mathbf{X} \quad (同 3-2-8)$$

$$y_j = f(net_j) \quad (同 3-2-9)$$

只是最小均方演算法並沒有所謂的活化函數來將輸入值轉換成輸出值，其輸出值就直接是各輸入向量與各權重向量相乘積後線性疊加值，因此第 j 個神經元之輸出為：

$$y_j = f(net_j) = \mathbf{W}_j^T \mathbf{X} \quad (3-2-15)$$

在最小均方演算法中調整權重的方式是依據計算坡降的方式而來



的，最陡坡降法 (steepest descent method) 是最一开始被采用的方式，最陡坡降法的示意圖如圖 3-2-13 所示。由於最小均方演算法是屬於監督式學習法，所以已能夠確知目標值為何，故我們假設 \mathbf{X} 為輸入向量， d 為目標輸出值， \mathbf{W} 為權重向量， n 為迭代次數，誤差值 e 就為目標輸出向量減去神經元的輸出值，而在經過 n 次迭代後的誤差如下式：

$$e(n) = d(n) - y(n) = d(n) - \mathbf{W}^T(\mathbf{n})\mathbf{X}(\mathbf{n}) \quad (3-2-16)$$

我們根據均方差 (Mean Square Error, MSE) 的計算方式來定義誤差函數 E ，均方差即為誤差值 e 平方和，而為了計算方便，誤差函數定為均方差的 $\frac{1}{2}$ ，寫法如下式：

$$E(\mathbf{W}) = \frac{1}{2} [e^2(n)] \quad (3-2-17)$$

因為目的在於尋找要如何調整權重值，因此誤差函數為權重值的函數，將誤差函數作展開，可得：

$$\begin{aligned} E(\mathbf{W}) &= \frac{1}{2} [e^2(n)] \\ &= \frac{1}{2} \left\{ [d(n) - \mathbf{W}^T(\mathbf{n})\mathbf{X}(\mathbf{n})]^2 \right\} \\ &= \frac{1}{2} [d^2(n)] - [d(n) \cdot \mathbf{X}(\mathbf{n})^T] \mathbf{W}(\mathbf{n}) + \frac{1}{2} [\mathbf{W}^T(\mathbf{n})\mathbf{X}(\mathbf{n})\mathbf{X}(\mathbf{n})^T] \mathbf{W}(\mathbf{n}) \end{aligned} \quad (3-2-18)$$

當此目標函數 $E(\mathbf{W})$ 值達到最小時，即表示此時神經元輸出與目標值的均方差為最小。然而在實務上要得知是否發生最小值，所根據的是微分原理，某一函數的最小值是在一次微分等於零的情形之下，

寫成數學式子即是 $\nabla E(\mathbf{W}) = 0$ ，過算過程如下所示：

$$\begin{aligned}
 \nabla E(\mathbf{W}) &= \frac{1}{2} \frac{\partial [e(n)^2]}{\partial \mathbf{W}} \Big|_{\mathbf{W}=\mathbf{W}(\mathbf{n})} \\
 &= \frac{1}{2} \frac{\partial}{\partial \mathbf{W}(n)} \left[d^2(n) - 2d(n)\mathbf{X}^T(\mathbf{n})\mathbf{W}(\mathbf{n}) + \mathbf{W}^T(\mathbf{n})\mathbf{X}(\mathbf{n})\mathbf{X}^T(\mathbf{n})\mathbf{W}(\mathbf{n}) \right] \\
 &= -d(n)\mathbf{X}(\mathbf{n}) + \mathbf{X}(\mathbf{n})\mathbf{X}^T(\mathbf{n})\mathbf{W}(\mathbf{n}) \\
 &= -d(n)\mathbf{X}(\mathbf{n}) + \mathbf{W}(\mathbf{n})^T \mathbf{X}(\mathbf{n})\mathbf{X}(\mathbf{n}) \\
 &= \left[-d(n) + \mathbf{W}(\mathbf{n})^T \mathbf{X}(\mathbf{n}) \right] \mathbf{X}(\mathbf{n}) \\
 &= -[d(n) - \mathbf{W}(\mathbf{n})^T \mathbf{X}(\mathbf{n})] \mathbf{X}(\mathbf{n}) \\
 &= -e(n)\mathbf{X}(\mathbf{n})
 \end{aligned} \tag{3-2-19}$$

根據上式，我們可以得知利用最陡坡降法推導得到的下一個權重應該調整的方向為：

$$\mathbf{W}(\mathbf{n}+1) = \mathbf{W}(\mathbf{n}) + \eta(-\nabla_{\mathbf{W}} E(\mathbf{W})) = \mathbf{W}(\mathbf{n}) + \eta e(n)\mathbf{X}(\mathbf{n}) \tag{3-2-20}$$

因為調整權重的目的是使得誤差越來越小，所以調整權重的方向會與坡降方向差一個負號，式子中的 η 代表的是學習的速度：若 η 值太大則權重容易調整過大造成學習過程中出現震盪 (damping) 之現象；若 η 值太小則收斂速度慢，所需的迭代步數會多很多。為了解決這個問題，Haykin 在 1996 年提出 η 值的範圍最好選在

$$0 < \eta < \frac{2}{\lambda_{\max}} \tag{3-2-21}$$

其中 λ_{\max} 為 $\mathbf{X}(\mathbf{n})^T \mathbf{X}(\mathbf{n})$ 矩陣中特徵值的最大值；然而也有其他研究



建議 η 值不一定要是一個定值，是可以隨著迭代的步數增加而減少，因為迭代次數增加後，所需要調整的權重值理論上會越變越小，故可寫成以下式子：

$$\eta(n) = \frac{\eta_0}{1 + \frac{n}{\tau}} \quad (3-2-22)$$

其中 n 為迭代次數， $100 \leq \tau \leq 500$ ， η_0 為初始學習速率且數值大於零。

最陡坡降法在使用上非常簡便，其使用方式的是以目標函數的一階微分來尋找目標函數的最小值，雖然簡單但卻不見得是最好的方法，因為如果目標函數非常複雜且為高次函數，則會很容易就落入鄰近的局部解，且收斂速度較慢。因此就發展出了另一套坡降法：牛頓法。

牛頓法是採用目標函數的一階與二階微分的方式來尋找最小值，使用較高階的微分方式會比只用一階微分來的有效率。要尋找目標函數 E 在 $x=a$ 部分的近似值，根據泰勒展開式（Taylor series expansion）可寫成：

$$\begin{aligned} E(x) &= E(a) + \frac{E'(a)}{1!}(x-a) + \frac{E''(a)}{2!}(x-a)^2 + \dots + \frac{E^{(k)}(a)}{k!}(x-a)^k + \dots \\ &= \sum_{k=0}^{\infty} \frac{E^{(k)}(a)}{k!}(x-a)^k \end{aligned} \quad (3-2-23)$$

如果只看 E 在 $x=a$ 的二次近似值，則只需看一階與二階的微分項，就變成：



$$E(x) \approx E(a) + \frac{E'(a)}{1!}(x-a) + \frac{E''(a)}{2!}(x-a)^2 \quad (3-2-24)$$

同理，如果要尋找的是目標函數 E 在 $x = x_{(k)}$ 時的二次近似值，

則可以寫成：

$$E(x) \approx E(x_{(k)}) + \frac{E'(x_{(k)})}{1!}(x-x_{(k)}) + \frac{E''(x_{(k)})}{2!}(x-x_{(k)})^2 \quad (3-2-25)$$

在此引進一參數，稱為 Hessian matrix，假設有一函數為

$E(x_1, x_2, \dots, x_n)$ ，如果 E 所有的二階導數都存在，那麼 E 的 Hessian matrix 則為：

$$\mathbf{H}(E)_{ij}(\mathbf{x}) = D_i D_j E(x)$$

其中 $x = (x_1, x_2, \dots, x_n)$

$$\mathbf{H}(E) = \begin{bmatrix} \frac{\partial^2 E}{\partial x_1^2} & \frac{\partial^2 E}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 E}{\partial x_1 \partial x_n} \\ \frac{\partial^2 E}{\partial x_2 \partial x_1} & \frac{\partial^2 E}{\partial x_2^2} & \dots & \frac{\partial^2 E}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial x_n \partial x_1} & \frac{\partial^2 E}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 E}{\partial x_n^2} \end{bmatrix} \quad (3-2-26)$$

又 $E(x)$ 的一階微分 $E'(x_{(k)})$ 可視為 $\nabla E(x_{(k)})$ ，因此用矩陣形式來寫可寫成：

$$\mathbf{E}(\mathbf{x}) \approx \mathbf{E}(\mathbf{x}_{(k)}) + (\mathbf{x} - \mathbf{x}_{(k)})^T \nabla \mathbf{E}(\mathbf{x}_{(k)}) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_{(k)})^T \mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x} - \mathbf{x}_{(k)}) \quad (3-2-27)$$

當 $E(x)$ 位於最小值時，則此時 $\nabla \mathbf{E}(x)$ 會等於 0，因此：

$$\nabla \mathbf{E}(\mathbf{x}) \approx \nabla \left[\mathbf{E}(\mathbf{x}_{(k)}) + (\mathbf{x} - \mathbf{x}_{(k)})^T \nabla \mathbf{E}(\mathbf{x}_{(k)}) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_{(k)})^T \mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x} - \mathbf{x}_{(k)}) \right]$$

$$= \nabla E(x_{(k)}) + H(x_{(k)}) (x - x_{(k)}) \\ = 0$$



(3-2-28)

將式 3-2-28 移項可得：

$$x = x_{(k)} - H(x_{(k)})^{-1} \nabla E(x_{(k)}) \quad (3-2-29)$$

其中 k 從 0 開始，因此可得到：

$$\begin{aligned} x_1 &= x_0 - H(x_0)^{-1} \nabla E(x_0) \\ x_2 &= x_1 - H(x_1)^{-1} \nabla E(x_1) \\ &\vdots \\ x_{k+1} &= x_k - H(x_k)^{-1} \nabla E(x_k) \end{aligned} \quad (3-2-30)$$

再將步移速率加進去此式子中，最後就會變成：

$$x_{k+1} = x_k - \eta H(x_k)^{-1} \nabla E(x_k) \quad (3-2-31)$$

此即為牛頓法之學習演算法計算內容。

因為在類神經網路中我們所需要調整的為權重值，因此將 x 替換成權重 W ，迭代次數為 n ，則會變成：

$$W(n+1) = W(n) - \eta H(W(n))^{-1} \nabla E(W(n)) \quad (3-2-32)$$

即為所求。

然而牛頓法中的函數實際的 Hessian matrix 不是一個正定矩陣 (Positive Definite Matrices) 時，則此時使用牛頓法的搜尋到的方向 $H(x_{(k)})^{-1} \nabla E(x_{(k)})$ 就不一定會是函數值下降的方向，因此為了確保每一次的搜尋方向都是函數坡降的方向，就將牛頓法中的 Hessian

matrix 做了修正，而此修正法就稱為「Levenberg—Marquardt 修正模式」（Levenberg，1944，Marquardt，1963）。

此修正方式就是將 Hessian matrix ， $\mathbf{H}(\mathbf{x})$ 加上一修正矩陣 $\mu_k \mathbf{I}$ ，

因此就變成：

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[\mathbf{H}(\mathbf{x}_k)^{-1} + \mu_k \mathbf{I} \right] \nabla \mathbf{E}(\mathbf{x}_k) \quad (3-2-33)$$

其中 μ_k 為一常數且 $\mu_k \geq 0$ ， μ_k 會隨著迭代次數而變動，在演算前期時， μ_k 約等於零，因為在剛開始搜尋方向還不會有太大的問題，此時的運算過程與牛頓法幾乎一樣；然後演算到後來，為了確保搜尋方向正確，則會增加 μ_k 的值。再將步移速率加進去 Levenberg—

Marquardt 的修正式子中，最後就會變成：

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \left[\mathbf{H}(\mathbf{x}_k)^{-1} + \mu_k \mathbf{I} \right] \nabla \mathbf{E}(\mathbf{x}_k) \quad (3-2-34)$$

此即為 Levenberg—Marquardt 修正模式之學習演算法計算內容。

而將 \mathbf{x} 替換成權重 \mathbf{W} ，迭代次數為 n ，則權重的修正會變成：

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \eta \left[\mathbf{H}(\mathbf{W}(n))^{-1} + \mu_n \mathbf{I} \right] \nabla \mathbf{E}(\mathbf{W}(n)) \quad (3-2-35)$$

此外，除了 Levenberg—Marquardt 的修正模式之外，還有另一種修正牛頓法的型態，我們稱之為「擬牛頓法」，其方法內容如同名稱一般，是將牛頓法做了些微改正而來的。由牛頓法中可以知道其運算之結果為：

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}(\mathbf{x}_k)^{-1} \nabla \mathbf{E}(\mathbf{x}_k) \quad (\text{同 } 3-2-30)$$

由於可能出現 $E(\mathbf{x}_{k+1}) > E(\mathbf{x}_k)$ ，使得搜尋方向就不是坡降的方向，因此將牛頓法之結果改寫成：

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{H}(\mathbf{x}_k)^{-1} \nabla E(\mathbf{x}_k) \quad (3-2-36)$$

藉由決定常數 α_k 的大小來確保坡降方向正確性。

此外，牛頓法還有個計算上的缺點，就是對於 Hessian matrix $\mathbf{H}(\mathbf{x}_k)$ 的計算，而且還需要解出 $\mathbf{H}(\mathbf{x}_{(k)})^{-1} \nabla E(\mathbf{x}_{(k)})$ 去找出每一次的坡降方向，因此為了避免計算真實的 Hessian 反矩陣 $\mathbf{H}(\mathbf{x}_k)^{-1}$ ，擬牛頓法就是利用了數值近似的方式來取代 $\mathbf{H}(\mathbf{x}_k)^{-1}$ 。

現今最常用的擬牛頓法為 DFP 演算法與 BFGS 演算法。

DFP 演算法：

由 Davidon 在 1959 年提出，後來被 Fletcher 與 Powell 在 1963 修改而成，演算過程最初是從牛頓法步驟開始推導。

在牛頓法中可用泰勒展開式得目標函數 E 在 $x = x_{(k)}$ 時的二次近似值，而當 $E(x)$ 位於最小值時，則此時 $\nabla E(\mathbf{x})$ 會等於 0，則為：

$$\nabla E(\mathbf{x}) \approx \nabla E(\mathbf{x}_{(k)}) + \mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x} - \mathbf{x}_{(k)}) = 0 \quad (3-2-37)$$

移項可得：

$$-\mathbf{H}(\mathbf{x}_{(k)})^{-1} \nabla E(\mathbf{x}_{(k)}) = \mathbf{x} - \mathbf{x}_{(k)} \quad (3-2-38)$$

將 $-\mathbf{H}(\mathbf{x}_{(k)})^{-1} \nabla E(\mathbf{x}_{(k)})$ 令為坡降方向 g ，將式 3-2-37 同乘 g 後移項：

$$g \nabla E(\mathbf{x}_{(k)}) = g \left[-\mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x} - \mathbf{x}_{(k)}) \right] \quad (3-2-39)$$



又 $-\mathbf{H}(\mathbf{x}_{(k)})^{-1}\nabla E(\mathbf{x}_{(k)})$ 也等於 $\mathbf{x} - \mathbf{x}_{(k)}$ ，因此上式變成：

$$g\nabla E(\mathbf{x}_{(k)}) = (\mathbf{x} - \mathbf{x}_{(k)}) \left[-\mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x} - \mathbf{x}_{(k)}) \right] \quad (3-2-40)$$

如果說等號右邊 $-(\mathbf{x} - \mathbf{x}_{(k)})\mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x} - \mathbf{x}_{(k)})$ 是小於零的，則等號左邊

$g\nabla E(\mathbf{x}_{(k)})$ 的意思為「沿著方向 g ，則 $\nabla E(\mathbf{x}_{(k)})$ 函數值會下降」，在尋找

目標函數最小值的過程中，本來就是希望梯度值越來越小，因此可以

確定 $-(\mathbf{x} - \mathbf{x}_{(k)})\mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x} - \mathbf{x}_{(k)})$ 一定是小於零的。

而 $-(\mathbf{x} - \mathbf{x}_{(k)})\mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x} - \mathbf{x}_{(k)})$ 小於零，則 $(\mathbf{x} - \mathbf{x}_{(k)})\mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x} - \mathbf{x}_{(k)})$ 就大於 0，

以數學上的特性來說，此時的 $\mathbf{H}(\mathbf{x}_{(k)})$ 為一正定矩陣(即一個對稱矩陣

中的特徵值皆為正數，則稱為正定矩陣)。

原本目標函數的 Hessian 矩陣一般來說不一定能保證正定，而擬

牛頓法可使用數值方式創造出一 Hessian 矩陣，且能夠確保其一定正

定，由於創造 Hessian 矩陣的過程亦是從牛頓法而來，因此與真實的

Hessian 矩陣會「很像」，故擬牛頓法也具有牛頓法的二階收斂性。

且根據正定矩陣之性值：一正定矩陣的反矩陣也會是正定矩陣，因此

擬牛頓法中 Hessian 矩陣的反矩陣 $\mathbf{H}(\mathbf{x}_{(k)})^{-1}$ 也會是一正定矩陣。

同牛頓法的部分，若是取任一函數 E 在 $x_{(k)} = x_{(k+1)}$ 時的二次近似值，在要計算 $E(x_{(k)})$ 位於最小值時的 $\nabla E(\mathbf{x}_{(k)})$ 可推導成：

$$\nabla E(\mathbf{x}_{(k)}) \approx \nabla E(\mathbf{x}_{(k+1)}) + \mathbf{H}(\mathbf{x}_{(k+1)}) (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \quad (\text{同 3-2-37})$$



上式中左右同乘 $\mathbf{H}(\mathbf{x}_{(k+1)})^{-1}$ ，會變成：

$$\mathbf{H}(\mathbf{x}_{(k+1)})^{-1} \nabla \mathbf{E}(\mathbf{x}_{(k)}) \approx \mathbf{H}(\mathbf{x}_{(k+1)})^{-1} \nabla \mathbf{E}(\mathbf{x}_{(k+1)}) + (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \quad (3-2-41)$$

將 Hessian 矩陣的反矩陣 $\mathbf{H}(\mathbf{x}_{(k+1)})^{-1}$ 用矩陣 $\mathbf{A}(\mathbf{x}_{(k+1)})$ 來替換，並

將近似相等換成等號，則式子可寫成：

$$\mathbf{A}(\mathbf{x}_{(k+1)}) \nabla \mathbf{E}(\mathbf{x}_{(k)}) = \mathbf{A}(\mathbf{x}_{(k+1)}) \nabla \mathbf{E}(\mathbf{x}_{(k+1)}) + (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \quad (3-2-42)$$

上式就是擬牛頓法的方程式。

又每一個 $\mathbf{A}(\mathbf{x}_{(k+1)})$ 矩陣都是由前一個 $\mathbf{A}(\mathbf{x}_{(k)})$ 修正而來的，因此 $\mathbf{A}(\mathbf{x}_{(k+1)})$ 可以視為 $\mathbf{A}(\mathbf{x}_{(k)})$ 去加上一修正的矩陣 $\mathbf{D}(\mathbf{x}_{(k)})$ ，帶回式 3-2-40，則變成：

$$[\mathbf{A}(\mathbf{x}_{(k)}) + \mathbf{D}(\mathbf{x}_{(k)})] \nabla \mathbf{E}(\mathbf{x}_{(k)}) = [\mathbf{A}(\mathbf{x}_{(k)}) + \mathbf{D}(\mathbf{x}_{(k)})] \nabla \mathbf{E}(\mathbf{x}_{(k+1)}) + (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \quad (3-2-43)$$

移項可得：

$$[\mathbf{A}(\mathbf{x}_{(k)}) + \mathbf{D}(\mathbf{x}_{(k)})] [\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)})] = \mathbf{x}_{(k+1)} - \mathbf{x}_{(k)} \quad (3-2-44)$$

由於每個 $\mathbf{A}(\mathbf{x}_{(k)})$ 都是正定矩陣，且根據正定矩陣的定義，任意正定矩陣的都可以寫成：

$$\mathbf{A}(\mathbf{x}_{(k)}) = \mathbf{M}(\mathbf{x}_{(k)})^* \mathbf{M}(\mathbf{x}_{(k)}) \quad (3-2-45)$$

如果 $\mathbf{M}(\mathbf{x}_{(k)})$ 都是實數，則成為：

$$\mathbf{A}(\mathbf{x}_{(k)}) = \mathbf{M}(\mathbf{x}_{(k)})^T \mathbf{M}(\mathbf{x}_{(k)}) \quad (3-2-46)$$

同理， $\mathbf{A}(\mathbf{x}_{(k+1)})$ 可以寫成

$$\mathbf{A}(\mathbf{x}_{(k+1)}) = \mathbf{N}(\mathbf{x}_{(k)})^T \mathbf{N}(\mathbf{x}_{(k)})$$

(3-2-47)

因此修正矩陣 $\mathbf{D}(\mathbf{x}_{(k)})$ 可視為兩個正定矩陣 $\mathbf{A}(\mathbf{x}_{(k)})$ 與 $\mathbf{A}(\mathbf{x}_{(k+1)})$ 之

差值，故可以寫成：

$$\mathbf{D}(\mathbf{x}_{(k)}) = a\mathbf{M}(\mathbf{x}_{(k)})^T \mathbf{M}(\mathbf{x}_{(k)}) + b\mathbf{N}(\mathbf{x}_{(k)})^T \mathbf{N}(\mathbf{x}_{(k)}) \quad (3-2-48)$$

式子中 a 與 b 皆是常數，帶入式 3-2-42 中可得到：

$$\begin{aligned} & \left[\mathbf{A}(\mathbf{x}_{(k)}) + a\mathbf{M}(\mathbf{x}_{(k)})^T \mathbf{M}(\mathbf{x}_{(k)}) + b\mathbf{N}(\mathbf{x}_{(k)})^T \mathbf{N}(\mathbf{x}_{(k)}) \right] \left[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) \right] \\ &= \mathbf{A}(\mathbf{x}_{(k)}) \left[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) \right] + \left[a\mathbf{M}(\mathbf{x}_{(k)})^T \mathbf{M}(\mathbf{x}_{(k)}) \right] \left[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) \right] \\ &+ \left[b\mathbf{N}(\mathbf{x}_{(k)})^T \mathbf{N}(\mathbf{x}_{(k)}) \right] \left[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) \right] \\ &= \mathbf{x}_{(k+1)} - \mathbf{x}_{(k)} \end{aligned} \quad (3-2-49)$$

為了簡化式子，令 $\mathbf{x}_{(k+1)} - \mathbf{x}_{(k)} = \mathbf{s}$ ， $\left[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) \right] = \mathbf{q}$ ，則

式 3-2-47 就變成：

$$\begin{aligned} &= \mathbf{A}(\mathbf{x}_{(k)})\mathbf{q} + a\mathbf{M}(\mathbf{x}_{(k)})^T \mathbf{M}(\mathbf{x}_{(k)})\mathbf{q} + b\mathbf{N}(\mathbf{x}_{(k)})^T \mathbf{N}(\mathbf{x}_{(k)})\mathbf{q} \\ &= \mathbf{A}(\mathbf{x}_{(k)})\mathbf{q} + \mathbf{M}(\mathbf{x}_{(k)})^T \left[a\mathbf{M}(\mathbf{x}_{(k)})\mathbf{q} \right] + \mathbf{N}(\mathbf{x}_{(k)})^T \left[b\mathbf{N}(\mathbf{x}_{(k)})\mathbf{q} \right] \\ &= \mathbf{s} \end{aligned} \quad (3-2-50)$$

我們可以根據上式找出當 $\mathbf{M}(\mathbf{x}_{(k)})^T$ 等於 $\mathbf{A}(\mathbf{x}_{(k)})\mathbf{q}$ ， $a\mathbf{M}(\mathbf{x}_{(k)})\mathbf{q}$ 等於 -1 ， $\mathbf{N}(\mathbf{x}_{(k)})^T$ 等於 \mathbf{s} ， $b\mathbf{N}(\mathbf{x}_{(k)})\mathbf{q}$ 等於 1 時，可為一組解。

因此， $a = \frac{-1}{\mathbf{M}(\mathbf{x}_{(k)})\mathbf{q}}$ ， $b = \frac{1}{\mathbf{N}(\mathbf{x}_{(k)})\mathbf{q}}$ ，帶回修正矩陣 $\mathbf{D}(\mathbf{x}_{(k)})$ 中，

$\mathbf{A}(\mathbf{x}_{(k+1)}) = \mathbf{A}(\mathbf{x}_{(k)}) + \mathbf{D}(\mathbf{x}_{(k)})$ 就可以改寫成：

$$\mathbf{A}(\mathbf{x}_{(k+1)}) = \mathbf{A}(\mathbf{x}_{(k)}) + \mathbf{D}(\mathbf{x}_{(k)})$$



$$\begin{aligned}
&= \mathbf{A}(\mathbf{x}_{(k)}) + \left[a \mathbf{M}(\mathbf{x}_{(k)})^T \mathbf{M}(\mathbf{x}_{(k)}) + b \mathbf{N}(\mathbf{x}_{(k)})^T \mathbf{N}(\mathbf{x}_{(k)}) \right] \\
&= \mathbf{A}(\mathbf{x}_{(k)}) + \frac{-1}{\mathbf{M}(\mathbf{x}_{(k)}) \mathbf{q}} \mathbf{M}(\mathbf{x}_{(k)})^T \mathbf{M}(\mathbf{x}_{(k)}) + \frac{1}{\mathbf{N}(\mathbf{x}_{(k)}) \mathbf{q}} \mathbf{N}(\mathbf{x}_{(k)})^T \mathbf{N}(\mathbf{x}_{(k)}) \\
&= \mathbf{A}(\mathbf{x}_{(k)}) + \frac{-1}{[\mathbf{A}(\mathbf{x}_{(k)}) \mathbf{q}]^T \mathbf{q}} \mathbf{A}(\mathbf{x}_{(k)}) \mathbf{q} \left[\mathbf{A}(\mathbf{x}_{(k)}) \mathbf{q} \right]^T + \frac{1}{\mathbf{s}^T \mathbf{q}} \mathbf{s} \mathbf{s}^T \\
&= \mathbf{A}(\mathbf{x}_{(k)}) - \frac{\left\{ \mathbf{A}(\mathbf{x}_{(k)}) \left[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) \right] \right\} \left\{ \left[\mathbf{A}(\mathbf{x}_{(k)}) \nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) \right] \right\}^T}{\left\{ \mathbf{A}(\mathbf{x}_{(k)}) \left[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) \right] \right\}^T \left[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) \right]} \\
&\quad + \frac{\left[\mathbf{x}_{(k+1)} - \mathbf{x}_{(k)} \right] \left[\mathbf{x}_{(k+1)} - \mathbf{x}_{(k)} \right]^T}{\left[\mathbf{x}_{(k+1)} - \mathbf{x}_{(k)} \right]^T \left[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) \right]} \\
&\tag{3-2-51}
\end{aligned}$$

此即為 DFP 演算法之計算內容。

將 \mathbf{x} 替換成權重 \mathbf{W} ，則上式就會變成 DFP 演算法的權重修正公式。

BFGS 演算法：

此公式是由 Broyden、Fletch、Goldfarb 與 Shanno 在 1970 年所提出，是目前所公認最好的擬牛頓算法。其運算過程和 DFP 演算法很類似，若是取任一函數 E 在 $x_{(k)} = x_{(k+1)}$ 時的二次近似值，在要計算 $E(x_{(k)})$ 位於最小值時的 $\nabla \mathbf{E}(\mathbf{x}_{(k)})$ 可推導成：

$$\nabla \mathbf{E}(\mathbf{x}_{(k)}) \approx \nabla \mathbf{E}(\mathbf{x}_{(k+1)}) + \mathbf{H}(\mathbf{x}_{(k+1)}) (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \quad (\text{同 3-2-37})$$

移項可得：

$$\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) = \mathbf{H}(\mathbf{x}_{(k+1)}) (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \quad (3-2-52)$$

又根據 DFP 演算法中，構造出一 Hessian 矩陣使其確保一定正定，且每一個 $\mathbf{H}(\mathbf{x}_{(k+1)})$ 矩陣都是由前一個 $\mathbf{H}(\mathbf{x}_{(k)})$ 修正而來的，因此可寫成：

$$\mathbf{H}(\mathbf{x}_{(k+1)}) = \mathbf{H}(\mathbf{x}_{(k)}) + \mathbf{U}(\mathbf{x}_{(k)}) \quad (3-2-53)$$

其中 $\mathbf{U}(\mathbf{x}_{(k)})$ 為修改矩陣，且因為 $\mathbf{U}(\mathbf{x}_{(k)})$ 為兩個正定函數 $\mathbf{H}(\mathbf{x}_{(k)})$ 與 $\mathbf{H}(\mathbf{x}_{(k+1)})$ 之差值，同 DFP 演算法中的修改函數 $\mathbf{D}(\mathbf{x}_{(k)})$ ，故 $\mathbf{U}(\mathbf{x}_{(k)})$ 可以寫成：

$$\mathbf{U}(\mathbf{x}_{(k)}) = c\mathbf{K}(\mathbf{x}_{(k)})^T \mathbf{K}(\mathbf{x}_{(k)}) + d\mathbf{L}(\mathbf{x}_{(k)})^T \mathbf{L}(\mathbf{x}_{(k)}) \quad (3-2-54)$$

式子中 c 與 d 皆是常數，帶入式 3-2-50 中可得：

$$\begin{aligned} \nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)}) &= \mathbf{H}(\mathbf{x}_{(k+1)}) (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \\ &= [\mathbf{H}(\mathbf{x}_{(k)}) + \mathbf{U}(\mathbf{x}_{(k)})] (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \\ &= \mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) + \mathbf{U}(\mathbf{x}_{(k)}) (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \\ &= \mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) + [c\mathbf{K}(\mathbf{x}_{(k)})^T \mathbf{K}(\mathbf{x}_{(k)}) + d\mathbf{L}(\mathbf{x}_{(k)})^T \mathbf{L}(\mathbf{x}_{(k)})] (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \\ &= \mathbf{H}(\mathbf{x}_{(k)}) (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) + c\mathbf{K}(\mathbf{x}_{(k)})^T \mathbf{K}(\mathbf{x}_{(k)}) (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \\ &\quad + d\mathbf{L}(\mathbf{x}_{(k)})^T \mathbf{L}(\mathbf{x}_{(k)}) (\mathbf{x}_{(k)} - \mathbf{x}_{(k+1)}) \end{aligned} \quad (3-2-55)$$

為了簡化式子，令 $\mathbf{x}_{(k+1)} - \mathbf{x}_{(k)} = \mathbf{s}$ ， $[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)})] = \mathbf{q}$ ，則式

3-2-52 等於：

$$\begin{aligned} \mathbf{q} &= \mathbf{H}(\mathbf{x}_{(k)}) \mathbf{s} + c\mathbf{K}(\mathbf{x}_{(k)})^T \mathbf{K}(\mathbf{x}_{(k)}) \mathbf{s} + d\mathbf{L}(\mathbf{x}_{(k)})^T \mathbf{L}(\mathbf{x}_{(k)}) \mathbf{s} \\ &= \mathbf{H}(\mathbf{x}_{(k)}) \mathbf{s} + \mathbf{K}(\mathbf{x}_{(k)})^T [c\mathbf{K}(\mathbf{x}_{(k)}) \mathbf{s}] + \mathbf{L}(\mathbf{x}_{(k)})^T [d\mathbf{L}(\mathbf{x}_{(k)}) \mathbf{s}] \end{aligned}$$

們可以根據上式找出當 $\mathbf{K}(\mathbf{x}_{(k)})^T$ 等於 $\mathbf{H}(\mathbf{x}_{(k)})\mathbf{s}$ ， $c\mathbf{K}(\mathbf{x}_{(k)})\mathbf{s}$ 等於 -1，
 $\mathbf{L}(\mathbf{x}_{(k)})^T$ 等於 \mathbf{q} ， $d\mathbf{L}(\mathbf{x}_{(k)})\mathbf{s}$ 等於 1 時，為一組解。

因此， $c = \frac{-1}{\mathbf{K}(\mathbf{x}_{(k)})\mathbf{s}}$ ， $d = \frac{1}{\mathbf{L}(\mathbf{x}_{(k)})\mathbf{s}}$ ，帶回修改函數 $\mathbf{U}(\mathbf{x}_{(k)})$ 中，

則式 3-2-51 就可以改寫成：

$$\begin{aligned}
 \mathbf{H}(\mathbf{x}_{(k+1)}) &= \mathbf{H}(\mathbf{x}_{(k)}) + \mathbf{U}(\mathbf{x}_{(k)}) \\
 &= \mathbf{H}(\mathbf{x}_{(k)}) + c\mathbf{K}(\mathbf{x}_{(k)})^T \mathbf{K}(\mathbf{x}_{(k)}) + d\mathbf{L}(\mathbf{x}_{(k)})^T \mathbf{L}(\mathbf{x}_{(k)}) \\
 &= \mathbf{H}(\mathbf{x}_{(k)}) + \frac{-1}{\mathbf{K}(\mathbf{x}_{(k)})\mathbf{s}} \mathbf{K}(\mathbf{x}_{(k)})^T \mathbf{K}(\mathbf{x}_{(k)}) + \frac{1}{\mathbf{L}(\mathbf{x}_{(k)})\mathbf{s}} \mathbf{L}(\mathbf{x}_{(k)})^T \mathbf{L}(\mathbf{x}_{(k)}) \\
 &= \mathbf{H}(\mathbf{x}_{(k)}) + \frac{-1}{[\mathbf{H}(\mathbf{x}_{(k)})\mathbf{s}]^T \mathbf{s}} [\mathbf{H}(\mathbf{x}_{(k)})\mathbf{s}] [\mathbf{H}(\mathbf{x}_{(k)})\mathbf{s}]^T + \frac{1}{\mathbf{q}^T \mathbf{s}} \mathbf{q} \mathbf{q}^T \\
 &= \mathbf{H}(\mathbf{x}_{(k)}) - \frac{[\mathbf{H}(\mathbf{x}_{(k)})(\mathbf{x}_{(k+1)} - \mathbf{x}_{(k)})][\mathbf{H}(\mathbf{x}_{(k)})(\mathbf{x}_{(k+1)} - \mathbf{x}_{(k)})]^T}{[\mathbf{H}(\mathbf{x}_{(k)})(\mathbf{x}_{(k+1)} - \mathbf{x}_{(k)})]^T (\mathbf{x}_{(k+1)} - \mathbf{x}_{(k)})} \\
 &\quad + \frac{[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)})][\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)})]^T}{[\nabla \mathbf{E}(\mathbf{x}_{(k+1)}) - \nabla \mathbf{E}(\mathbf{x}_{(k)})]^T \mathbf{s} \quad (3-2-56)}
 \end{aligned}$$

此即為 BFGS 演算法之計算內容。

將 \mathbf{x} 替換成權重 \mathbf{W} ，則上式就會變成 BFGS 演算法的權重修正公式。

2. 感知器學習法 (perceptron learning rule)：

在整個類神經網路的發展過程中，單層感知器 (single layer perceptron) 對後續發展所造成的影响最為深遠。其最初的概念是由

Rosenblatt 在 1958 年所提出的，網路架構的屬性上是屬於單層前饋式
類神經網路。早期的活化函數為二值的門檻函數，也就是非 1 即 0 (或
-1) 的形式，當神經元的累加值超過所設定的門檻值 S 時，神經元輸
出即為 1，反之則為 0 (或-1)，數學式為

$$f(net) = \begin{cases} 1 & \text{if } net \geq S \\ 0(\text{或}-1) & \text{if } net < S \end{cases} \quad (3-2-57)$$

而其圖形如圖 3-2-2 (a) 所示，式子中的 net 為神經元的累加值，同
式 3-2-8。

由於為監督式學習方法，所以目標函數為已知，在此網路中的第
 j 個神經元學習信號 r 為目標函數 d_j 與類神經網路輸出值 y_j 之差值：

$$r = d_j - y_j \quad (3-2-58)$$

第 j 個神經元的 net 則可以寫成

$$net_j = \sum_{i=1}^n w_{ji} x_i = \mathbf{W}_j^T \mathbf{X} \quad (\text{同 } 3-2-8)$$

類神經網路輸出值 y_j 為

$$y_j = sign(\mathbf{W}_j^T \mathbf{X} - S) \quad (3-2-59)$$

又可寫成

$$y_j = sign(\mathbf{W}_j^T \mathbf{X} - S) = \begin{cases} 1 & \text{if } \mathbf{W}_j^T \mathbf{X} \geq S \\ 0(\text{或}-1) & \text{if } \mathbf{W}_j^T \mathbf{X} < S \end{cases} \quad (3-2-60)$$

在此學習法中之權重調整值為：

$$\begin{aligned} \Delta \mathbf{W}_j &= \eta [d_j - sign(\mathbf{W}_j^T \mathbf{X} - S)] \mathbf{X} \\ &= \eta r \mathbf{X} \end{aligned} \quad (3-2-61)$$

寫成純量看各自權重調整值則為：

$$\Delta w_{ji} = \eta [d_j - \text{sign}(\mathbf{W}_j^T \mathbf{X} - S)] x_i \quad \text{for } i = 1, 2, \dots, n \quad (3-2-62)$$

示意圖如圖 3-2-13 所示。

然而單層感知器只能夠做線性處理的問題，並不能處理非線性的問題，例如邏輯電路中的 XOR 是無法用單層感知器去分類的，因此就將單層感知器學習法做出延伸發展出多層感知器（Multilayer Perceptron，MLP）學習法。

為了讓多層感知器能夠有效處理非線性問題，於是將多層感知器學習法做了兩個不同於單層感知器的修改方向：一是活化函數的部分，不再選用二值函數當活化函數，而是選用 Sigmoid 函數當活化函數，Sigmoid 函數則如圖 3-2-2 (c) 所示；二是調整權重的方法改用坡降法去找出誤差值最小的部分。也就是說，多層感知器學習法其實就是結合最小均方演算法與 Sigmoid 函數當活化函數之方法。而網路的輸出值 y_j 就會變成：

$$net_j = \sum_{i=1}^n w_{ji} x_i = \mathbf{W}_j^T \mathbf{X} \quad (\text{同 3-2-8})$$

$$y_j = f(net_j) = f(\mathbf{W}_j^T \mathbf{X}) \quad (\text{同 3-2-9})$$

而 Sigmoid 函數的表示式為：

$$f(net_j) = \frac{1}{1 + e^{-\alpha net_j}} \quad (\text{同 3-2-5})$$



其中 α 為一可調整的常數。

引述最小均方演算法的部分：

設立一誤差函數 E ，誤差函數定為學習信號 r 平方的 $\frac{1}{2}$ ，則可得到：

$$E = \frac{1}{2} [r^2] \quad (3-2-63)$$

因為目的在於尋找要如何調整權重值，因此誤差函數為權重值的函數，將誤差函數作展開，可得：

$$\begin{aligned} E(\mathbf{W}) &= \frac{1}{2} [r^2] = \frac{1}{2} [(d_j - y_j)^2] \\ &= \frac{1}{2} \left\{ [d_j - f(\mathbf{W}_j^T \mathbf{X})]^2 \right\} \end{aligned} \quad (3-2-64)$$

當此目標函數 $E(\mathbf{W})$ 值達到最小時，即表示此時神經元輸出與目標值的均方差為最小。取一次微分等於零的情形之下，則上式可以寫成：

$$\begin{aligned} \nabla E(\mathbf{W}) &= \frac{1}{2} \frac{\partial [r^2]}{\partial \mathbf{W}} \\ &= (d_j - y_j) [-f'(\mathbf{W}_j^T \mathbf{X}) \mathbf{X}] \\ &= (d_j - y_j) [-f'(\text{net}_j) \mathbf{X}] \\ &= -r f'(\text{net}_j) \mathbf{X} \end{aligned} \quad (3-2-65)$$

因為權重的調整即是為了要使誤差減小，因此我們可以說權重的修正量會與誤差梯度差了一個負號，故可以寫成：



因此權重調整值為：

$$\Delta \mathbf{W}_j = \eta f'(net_j) \mathbf{X}$$

乘上學習速率 η 後就變成：

$$\Delta \mathbf{W}_j = \eta r f'(net_j) \mathbf{X} \quad (3-2-67)$$

示意圖如圖 3-2-14 所示。

將每一個神經元配上各自調整過後的權重值，最後就能得到誤差最小之結果。

3. Delta 學習法 (Delta learning rule) :

是一種適用於當神經元的活化函數為連續函數的演算法，由於為監督式學習法，所以目標值是已知的，因此定義誤差為目標函數 d 與類神經網路輸出值 y 之差值，而第 j 個神經元之誤差為：

$$e = d_j - y_j \quad (3-2-68)$$

其中 d_j 為第 j 個神經元之目標值， y_j 是第 j 個神經元之輸出值，而在此網路中的輸出值 y_j 為：

$$net_j = \sum_{i=1}^n w_{ji} x_i = \mathbf{W}_j^T \mathbf{X} \quad (\text{同 3-2-8})$$

$$y_j = f(net_j) = f(\mathbf{W}_j^T \mathbf{X}) \quad (\text{同 3-2-9})$$

誤差函數 E 定義為誤差值的平方再乘上 $\frac{1}{2}$ ，因此可以寫成：

$$E = \frac{1}{2} [e^2] \quad (3-2-69)$$

因為目的在於尋找要如何調整權重值，因此誤差函數為權重值的函數，將誤差函數作展開，可得：

$$\begin{aligned} E(\mathbf{W}) &= \frac{1}{2} [e^2] = \frac{1}{2} [(d_j - y_j)^2] \\ &= \frac{1}{2} [(d_j - f(\mathbf{W}_j^T \mathbf{X}))^2] \end{aligned} \quad (3-2-70)$$

欲找誤差最小值，即為誤差函數一次微分等於 0 的時候，因此

$$\begin{aligned} \nabla E(\mathbf{W}) &= \frac{1}{2} \frac{\partial [e^2]}{\partial \mathbf{W}} \\ &= (d_j - f(\mathbf{W}_j^T \mathbf{X})) [-f'(\mathbf{W}_j^T \mathbf{X}) \mathbf{X}] \end{aligned} \quad (3-2-71)$$

權重的調整即是為了要使誤差減小，因此我們可以說權重的修正量會與誤差梯度差了一個負號，再乘上學習速率 η 則可以寫成：

$$\Delta \mathbf{W}_j = \eta (d_j - f(\mathbf{W}_j^T \mathbf{X})) [-f'(\mathbf{W}_j^T \mathbf{X}) \mathbf{X}] \quad (3-2-72)$$

為了化簡上式，我們將上式中的 $(d_j - f(\mathbf{W}_j^T \mathbf{X})) f'(\mathbf{W}_j^T \mathbf{X})$ 部分令為學習信號 k ，因此權重調整值 $\Delta \mathbf{W}_j$ 就可以寫成：

$$\Delta \mathbf{W}_j = \eta k \mathbf{X} \quad (3-2-73)$$

Delta 學習法的初始權重可以為任意值，Delta 學習法與單層感知器學習法不同點在於 Delta 學習法中的神經元是具有連續性質的，而單層感知器學習法則適用於離散性質之神經元。此外，Delta 學習法在上述的數學運算過程中，可以發現也是利用坡降法來計算誤差的部分，此方式與最小均方演算法 (LMS) 使用的過程是一致的，只是最

小均方演算法中並沒有活化函數的作用，而 Delta 學習法則適用於任一具有連續性值的活化函數。

4. Grossberg 學習法 (Grossberg learning rule) :

此方法與非監督式學習中的贏者全拿學習法有許多相似之處，都是在同一層的眾多神經元中選擇出優勝者，並且只調整優勝神經元的權重值。這兩者的主要差別在於贏者全拿學習法是屬於非監督式的學習演算法，因為並沒有目標輸出值所以是採取調整輸入向量到各神經元間的權重值，而 Grossberg 學習法則是屬於監督式的學習方法，因有其特定的目標輸出值，故是採要調整到輸出神經元的權重向量方式去使得輸出值與目標值趨於一致，其示意圖如圖 3-2-14 所示。

當前一層的優勝神經元為第 m 個神經元，連結到下一層第 k 個神經元時，其連結權重為 w_{km} ，則權重調整的方式為目標輸出值與原本權重值的差值乘上一學習速率：

$$\Delta w_{km} = \eta(d_k - w_{km}) \quad \text{for } k = 1, 2, \dots, p \quad (3-2-74)$$

其中 η 為學習速率，在迭代過程中會逐漸縮小且 $\eta > 0$ ， d_k 為第 k 個神經元目標輸出值。

以上各學習演算法皆為從改變權重角度所切入的類神經網路學習演算法，可以發現很多學習法用的都是利用坡降的方法來計算誤差值，因為迭代速度快且又穩定，且能夠快速的尋找到誤差最低點。



3-3 倒傳遞類神經網路

類神經網路的種類非常的多，而本研究的坐姿判斷系統所採用的類神經網路型態為倒傳遞類神經網路（Back Propagation Neural Networks，BPNN），就網路架構來看是屬於多層前饋型網路，整個結構如圖 3-3-1，而就學習方法的角度來看是屬於監督式學習網路，誤差修正的過程如圖 3-3-2 所示。會選用倒傳遞類神經網路之目的在於類神經網路的主要功能為用來處理輸出與輸入之非線性映射關係，利用增加隱藏層的概念，來詮釋非線性之特性，因此能解決的問題範圍非常的廣泛。且學習方法為監督式學習，能夠藉由回傳誤差的方式不斷的來調整權重，達到非常準確的結果。

3-3-1 權重修正公式

倒傳遞類神經網路的學習方法是使用監督式學習法中的多層感知器學習法，其活化函數是 Sigmoid 函數，調整權重的演算法是利用最陡梯度演算法。在倒傳遞網路的學習過程中，可以看成是由一正向傳播訊息過程與一負向傳播訊息過程所構築的：正向傳播訊息是指訊號從輸入層輸入經隱藏層內部權重加權後，再透過活化函數轉換後將訊息傳給輸出層計算輸出值的過程，每一層神經元只會影響下一層神經元並不會越級傳遞，且傳遞方向皆往前；負向傳遞訊息則是當輸出

層不能得到目標輸出值的時候，會將誤差沿原路徑回傳到上一層神經元之中，或是再回傳到上上層……等等，藉由修改各層神經元的權重值，直到最後輸出結果與目標輸出值之誤差在容許範圍之內，整體過程可如圖 3-3-2 所示。

在倒傳遞類神經網路中，第 n 層第 j 個神經元的輸入值為第 $n-1$ 層神經元輸出的非線性函數值：

$$y_j^n = f\left(\sum w_{ji}^n y_i^{n-1}\right) = f\left(\text{net}_j^n\right) \quad (3-3-1)$$

$$\text{net}_j^n = \sum w_{ji}^n y_i^{n-1} + b_j^n \quad (3-3-2)$$

y_j^n 為第 n 層第 j 個神經元的輸出值， f 為活化函數， net_j^n 為第 $n-1$ 層輸出值的權重累加值， w_{ji}^n 為第 n 層第 j 個神經元與第 $n-1$ 層第 i 個神經元的連結權重值， b_j^n 為第 n 層第 j 個神經元的偏權值。

由於倒傳遞類神經為監督式學習，其目標輸出值是已知的，學習之目的在於降低網路輸出值與目標輸出值之誤差，我們定義誤差函數 E 為目標輸出值與實際輸出值之誤差平方和：

$$E = \frac{1}{2} \sum_k (d_k - y_k)^2 \quad (3-3-3)$$

其中 d_k 為第 k 個神經元之目標輸出值， y_k 為第 k 個神經元之類神經網路輸出值。由於採用最陡坡降法的方式來尋找誤差函數的最小值，誤差最小值會出現在一次微分等於零之情形。又因為目的在於要如何調整內部的每一個權重值，因此將誤差函數對各權重微分即可知道各



個權重該如何調整：

$$\nabla E = \frac{\partial E}{\partial w_{ji}}$$

又因為微調權重值的目的在於使誤差減小，都會朝向誤差梯度的反方向進行，因此每次權重的調整值可寫成：

$$\Delta w_{ji} = -\eta \nabla_{w_{ji}} E \quad (3-3-5)$$

其中 η 為一正的學習速率常數，其值的大小會決定所修正權重值的步幅。而 $\frac{\partial E}{\partial w_{ji}}$ 可以利用微積分中的鏈微法則 (chain rule) 得到下面的式子：

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial net_j^n} \frac{\partial net_j^n}{\partial w_{ji}} = \frac{\partial E}{\partial y_j^n} \frac{\partial y_j^n}{\partial net_j^n} \frac{\partial net_j^n}{\partial w_{ji}} \quad (3-3-6)$$

此式子可以分成三個部分來看： $\frac{\partial y_j^n}{\partial net_j^n}$ 、 $\frac{\partial net_j^n}{\partial w_{ji}}$ 與 $\frac{\partial E}{\partial y_j^n}$

1. 由於 $y_j^n = f(net_j^n)$ ，因此 $\frac{\partial y_j^n}{\partial net_j^n}$ 就等於 $f'(net_j^n)$

2. 而 $net_j^n = \sum w_{ji}^n y_i^{n-1} + b_j^n$ ，則 $\frac{\partial net_j^n}{\partial w_{ji}}$ 就會等於下式：

$$\frac{\partial net_j^n}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}^n} (\sum w_{ji}^n y_i^{n-1} + b_j^n) = y_i^{n-1} \quad (3-3-7)$$

式子中的 n 是代表第 n 層，因此在一層一層獨立計算的時候並不會影響其微分過程。



3. $\frac{\partial E}{\partial y_j^n}$ 可分成兩種情形：

1. 若第 n 層為輸出層時，則此時的誤差函數中的誤差就是來自目標輸出值與輸出層的網路輸出值，因此可寫成下式：

$$E = \frac{1}{2} \sum_k (d_k - y_k)^2 \quad (同 3-3-3)$$

故就以 $j=k$ 帶入式子中：

$$\begin{aligned} \frac{\partial E}{\partial y_j^n} &= \frac{\partial}{\partial y_j^n} \left[\frac{1}{2} \sum_k (d_k - y_k^n)^2 \right] \\ &= \frac{\partial}{\partial y_k^n} \left[\frac{1}{2} \sum_k (d_k - y_k^n)^2 \right] \\ &= -(d_k - y_k^n) \end{aligned} \quad (3-3-8)$$

2. 若第 n 層不是輸出層時，也就是網路的隱藏層之一，此時可用鏈微法則得：

$$\frac{\partial E}{\partial y_j^n} = \sum_k \left(\frac{\partial E}{\partial net_k^{n+1}} \frac{\partial net_k^{n+1}}{\partial y_j^n} \right) \quad (3-3-9)$$

將 $net_j^n = \sum_i w_{ji}^n y_i^{n-1} + b_j^n$ 帶入 $\frac{\partial net_k^{n+1}}{\partial y_j^n}$ 中可得：

$$\frac{\partial net_k^{n+1}}{\partial y_j^n} = \frac{\partial}{\partial y_j^n} \left(\sum_i w_{ki}^n y_j^n + b_k \right) = w_{kj}^n \quad (3-3-10)$$

此時引入一代數 σ ，令 $\frac{\partial E}{\partial net_k^n} = -\sigma_k^n$ ，故 $\frac{\partial E}{\partial y_j^n}$ 為：



因此，可以總結當連結權重是處於輸出層與隱藏層間時：

$$\begin{aligned}
 \frac{\partial E}{\partial w_{ji}} &= \frac{\partial E}{\partial net_j^n} \frac{\partial net_j^n}{\partial w_{ji}} = \frac{\partial E}{\partial y_j^n} \frac{\partial y_j^n}{\partial net_j^n} \frac{\partial net_j^n}{\partial w_{ji}} \\
 &= -\delta_j^n y_i^{n-1} \\
 &= -(d_j - y_j^n) f'(u_j^k) y_i^{n-1}
 \end{aligned} \tag{3-3-12}$$

當連結權重是處於隱藏層與隱藏層間時：

$$\begin{aligned}
 \frac{\partial E}{\partial w_{ji}} &= \frac{\partial E}{\partial net_j^n} \frac{\partial net_j^n}{\partial w_{ji}} = \frac{\partial E}{\partial y_j^n} \frac{\partial y_j^n}{\partial net_j^n} \frac{\partial net_j^n}{\partial w_{ji}} \\
 &= -\delta_j^n y_i^{n-1} \\
 &= \left[-\sum_k \delta_k^{n+1} w_{kj} \right] f'(u_j^k) y_i^{n-1}
 \end{aligned} \tag{3-3-13}$$

由以上式子可看出不論連結權重是處在隱藏層與隱藏層之間，

或者是隱藏層與輸出層之間，則 $\frac{\partial E}{\partial w_{ji}}$ 都可以寫成以下通式：

$$\frac{\partial E}{\partial w_{ji}} = -\delta_j^n y_i^{n-1} \tag{3-3-14}$$

又 $\nabla E = \frac{\partial E}{\partial w_{ji}}$ ，因此可得：

$$\begin{aligned}
 \Delta w_{ji} &= -\eta \nabla_{w_{ji}} E \\
 &= \eta \delta_j^n y_i^{n-1}
 \end{aligned} \tag{3-3-15}$$

此一式子即為倒傳遞類神經網路之權重修改公式，當每輸入一組訓練

範例時，各個連結權重便可依照上述之公式加以調整訓練之。

然而在進行網路訓練之前，一般都會先將輸入資料做正規化之處理 (normalize)，這樣能夠使得所有權重調整的速率相近，使得收斂過程較為平穩。

整個倒傳遞類神經網路演算之流程圖如圖 3-3-3 所示(藍色步驟表示前傳遞，紫色部分表示後傳遞)，程序則如下所述：

前傳遞部分：

1. 設定學習速率 η 、迭代步數與容忍誤差 (即目標輸出值與網路輸出值之差值)
2. 各層神經元之初始權重以給定隨機亂數的方式處理
3. 將訓練資料從輸入層輸入網路中
4. 計算出隱藏層之輸出值與輸出層之輸出值
5. 計算誤差函數 $E = \frac{1}{2} \sum_k (d_k - y_k)^2$

後傳遞部分：

1. 分別計算出輸出層與隱藏層的 σ_k^n
2. 分別修正輸出層權重與隱藏層權重值
3. 回到前傳遞部分的第 3 步驟，再次進行重複計算與修正工作，直到誤差值小於容許範圍，或者是達到設定之最大訓練次數為止

學習速率 η 又稱為權重的步移大小，一般而言，學習速率要夠小，才能夠找到越小的誤差函數。但若 η 值太小，則收斂速度慢，迭代步數會需要很大的數目，會花去非常多的時間；若 η 值太大，則會使網路產生不穩定之現象。通常會設定較大的 η 初始值，並隨著迭代次數增加後降低 η 值；或者是採取變動的 η 值來增加學習效率，當經過前一次迭代結果之後誤差值下降，則可以提高 η 值，因為代表經過這樣的步移大小後還沒有到達最小梯度點，若是經過前一次迭代結果之後誤差值上升，就表示這樣的步移大小過大，導致無法繼續朝向梯度小的地方前進，故需要降低 η 值。

3-3-2 圖形特徵參數

選定好類神經網路的類型之後，接著便是要將資料輸入做分析，然而，要用什麼樣的參數內容才能夠完整的表示出一張坐姿圖的特徵為何？在此我們引入慣性矩 (moment of inertia) 與慣性積 (product of inertia) (J. M. Gere, 2005) 來作為表示坐姿圖形的特徵參數。

何謂慣性矩呢？在一任意形狀的平面積中，如圖 3-3-4 所示。此平面內各個微小的截面積乘上與各轉軸間之距離平方總和，即為此平面對各轉軸之慣性矩，分別以下列積分式子定義：

$$I_x = \int y^2 dA \quad (3-3-16)$$

$$I_y = \int x^2 dA \quad (3-3-17)$$

由於上述式子中的慣性矩即是用各個微小面積 dA 乘上其到各軸線距離的平方，故慣性矩又稱為二次面積矩 (second moment of area)，且因為距離平方的關係，所以慣性矩一定為正值。以圖 3-3-5 以一矩形為例來計算同一圖形中慣性矩 I_x 與 I_y 之差別，在此圖形中之慣性矩分別為：

$$I_x = \int y^2 dA = \int_{-\frac{h}{2}}^{\frac{h}{2}} y^2 b dy = \frac{bh^3}{12}$$

$$I_y = \int x^2 dA = \int_{-\frac{b}{2}}^{\frac{b}{2}} x^2 h dx = \frac{hb^3}{12}$$

此外，對於一任意形狀的平面積中，若任取平面上的一組垂直軸系，則平面積內各個微小面積 dA 乘上對於此垂直軸系兩軸各自的座標值，定義為慣性積，其積分式如下：

$$I_{xy} = \int xy dA \quad (3-3-18)$$

由於垂直坐標軸系的座標值是會根據不同象限而有正負之分的，因此若面積全部落在一、三象限時，則慣性積為正；若面積全部落在二、四象限時，則慣性積為負；若面積不只落在一個象限時，則慣性積之正負取決於面積在各象限的分布狀態之總和。

慣性矩代表的物理意義即是此一物體對旋轉運動的慣性，而所謂的慣性是指物體持續維持不改變的行為。也就是說若當一物體目前是



呈現靜止狀態時，慣性矩越大則抵抗扭轉的能力也越大；若是物體已經呈現轉動狀態，慣性矩越大則表示維持像目前這樣轉動狀態的能力也愈大。慣性積主要則是被用在判斷圖形對稱。當圖形為一對稱圖形時，就會存在對稱軸，而在當對稱軸為計算慣性積時所選取的垂直軸線時則慣性積一定等於 0。如圖 3-3-6 所示，在對稱軸 y 軸上時，其慣性積 I_{xy} 為零。

在不同的圖形之下，慣性矩與慣性積會有一些不一樣的現象，如圖 3-3-7 所示：在圖 3-3-7(a)與圖 3-3-7(b)中，圖 3-3-7(a)的 I_x 大於圖 3-3-7(b)的 I_x 值，圖 3-3-7(a)的 I_y 會小於圖 3-3-7(b)的 I_y 值，而不管在圖 3-3-7(a)或圖 3-3-7(b)中，慣性積 I_{xy} 皆等於 0；在圖 3-3-7(a)與圖 3-3-7(c)中，圖 3-3-7(a)的 I_x 大於圖 3-3-7(c)的 I_x 值，圖 3-3-7(a)的 I_y 會小於圖 3-3-7(c)的 I_y 值，圖 3-3-7(a)的 I_{xy} 會小於圖 3-3-7(c)的 I_{xy} 值。

我們將圖 3-3-7 的圖形對比成人體坐姿圖形的樣子：圖 3-3-7(a)為坐姿正的時候，圖 3-3-7(c)為坐姿不正的時候，圖 3-3-7(b)則為人體平躺的時候，可以得知以下現象：

1. I_x : 在姿勢正的時候越大，在姿勢不正的時候越小。
2. I_y : 在姿勢不正的時候越大，在姿勢正的時候越小。
3. I_{xy} : 在姿勢不正的時候越大，在姿勢正的時候越小。

因此我們就用已經擷取目標圖型的二元化人體圖檔，拿來

計算其慣性矩與慣性積的值，因為二元化之處理使得有人體姿勢分布的地方值為 1，背景值為 0，因此可知每個點的微小面積都是 1。求出每個姿勢的慣性矩與慣性積之後，將其除上標準坐姿的慣性矩與慣性積，以達到正規化之結果。其過程如表 3-3-1 與圖 3-3-8 所示。將所有的坐姿圖形的參數都做了正規化處理之後，這些參數即是類神經網路的輸入訊號。

3-3-3 隱藏層層數與神經元數量

一般而言，隱藏層內的神經元數目以及隱藏層的數量往往都需要透過試誤法才能夠決定。然而在過去許多理論研究與工程應用上都顯示隱藏層的層數並不需要超過兩層以上，至於是一層隱藏層比較好或是兩層隱藏層比較好，在不同研究與問題上則是各有論述，不過目前大部分的研究都是採用一層隱藏層，除非是要解決非常高維度的問題。由於本研究中每一張坐姿圖中的參數來源只有三個，分別為 I_x 、 I_y 、與 I_{xy} ，故在本研究中是採用一層隱藏層的架構來進行運算。

故此時網路的大小是由一層輸入層、一層隱藏層與一層輸出層所構成，輸入層的神經元數量會根據輸入的參數有多少而跟著變動，故輸入層神經元為三個，而輸出層則只會輸出一個值，故輸出層的神經元只會有一個，而隱藏層的神經元數量是可以隨意變動的，然則過少的神經元會使得網路描述系統的能力不足，過多的神經元為使得網路

過度描述系統，因此我們必須決定隱藏層的神經元需要多少個，此時的類神經網路才會有最佳結果。

目前為止並沒有一套方式可以馬上判斷出一個類神經網路系統會需要多少神經元，故尋找隱藏層的神經元數目是採用試誤法。將隱藏層神經元數量一個一個慢慢的增加，觀察在多少數量的神經元之下類神經網路會最具有判斷能力。

在此我們引進了一套能夠鑑定判斷能力的分析方式：Receiver Operating Characteristic Curve，又稱為 ROC 曲線。ROC 曲線的目的在於判斷檢驗方法的鑑別能力。以醫學上的例子來說，假設用某一醫學檢驗方法去判斷檢驗者是否有患病，所謂的 TP (True positive，真陽性) 就是指有患病的人被診斷出有患病，TN (True negative，真陰性) 就是指沒有患病的人被診斷出沒有患病，FP (False positive，假陽性) 就是指沒有患病的人被診斷出有患病，FN (False negative，假陰性) 就是指有患病的人被診斷出沒有患病，一般會以表格的形式表示之，如表 3-3-2 所示。TP rate 就是有患病且被診斷出有患病的人，佔全部有患病的人的比例，對照表 3-3-2 寫成的式子為 $\frac{TP}{TP + FN}$ ；TN rate 就是沒有患病且被診斷出沒有患病的人，佔全部沒有患病的人的比例，對照表 3-3-2 寫成的式子為 $\frac{TN}{TN + FP}$ ；FP rate 就是指沒有患病的人卻被診斷成有患病的人，佔全部沒有患病的人的比例，對照表



3-3-2 寫成的式子為 $\frac{FP}{TN + FP}$ ；FN rate 就是指有患病的人卻被診斷出沒有患病，佔全部有患病的人的比例，對照表 3-3-2 寫成的式子為 $\frac{FN}{TP + FN}$ 。而 ROC 曲線為利用 TP rate 和 FP rate 之交集點，所畫出之曲線，如圖 3-3-9 所示。而當曲線下的面積越大，則此檢驗方法的判斷力就越好（越能準確的判斷出是否有患病），因此圖 3-3-9 中，這四條曲線中判斷能力的好壞排序為：4>3>2>1。

而 ROC 曲線下面積（Area Under the ROC Curve，簡稱 AUC）的判定能力之準則如表 3-3-3 所述。如果說曲線為一 45 度的對角直線，則代表此檢驗方法沒有任何的判斷能力，如圖 3-3-9 中的曲線 1。而我們可以從圖 3-3-10 與圖 3-3-11 來明白 ROC 曲線該如何繪製，如果有一檢驗方法對於某疾病越具有判斷能力，則患者跟沒有患者是能夠分的越開的兩個集合，隨著門檻值的移動，畫出來的 ROC 曲線會越接近外圍的邊界，曲線下的面積會很大，如圖 3-3-12 所示；反之，如果檢驗方法對於疾病越沒有判斷能力的，則會發現患者跟沒有患者是具有高度重合的兩個集合，隨著門檻值的移動，會發現 TP rate 與 FP rate 的值會較接近，如圖 3-3-13 所示，我們可看出圖 3-3-13 曲面下的面積比圖 3-3-12 來的小，如果當患者跟沒有患者是完全重合時，則此時畫出來的 ROC 曲線就會是一條斜率為 1 的 45 度斜直線，如圖 3-3-14 所示，故可以說是完全沒有判斷能力的。

為了檢驗最佳網路，我們需要先利用訓練資料去訓練出一類神經網路，儲存此類神經網路之後，將訓練資料再次丟入到方才訓練出來的類神經網路中去判斷，進而繪出其 ROC 曲線。在這裡的訓練資料為 10 個受試者，拍攝出總共 388 張照片，不同受試者用自己的標準坐姿去正規化自己坐姿時的各個參數值。

當 ROC 曲線下之面積越大時，則表示此類神經網路的判斷能力越好，故去找尋哪一個數量下之隱藏層神經元會是 ROC 曲線下之面積最大值，此一網路即為我們所求之最佳網路。

在目標圖型為擷取寬鬆定義的上半身時的類神經網路，其隱藏層神經元數量與 AUC 做圖為圖 3-3-15，從圖 3-3-15 可以看出，當隱藏層神經元數量大於 14 之後，AUC 趨於一穩定值，此時的 AUC 值幾乎都為 0.955，因此我們就取神經元數量為 14 個；在目標圖型為擷取嚴格定義的上半身時，其隱藏層神經元數量與 AUC 做圖為圖 3-3-16，從圖 3-3-16 可以看出，當隱藏層神經元數量大於 14 之後，AUC 趨於一穩定值，此時的 AUC 值幾乎都為 0.941，因此我們就取神經元數量為 14 個。而從表 3-3-3 可知：當 $AUC \leq 0.9$ 時，代表此系統具有非常好的判斷能力。

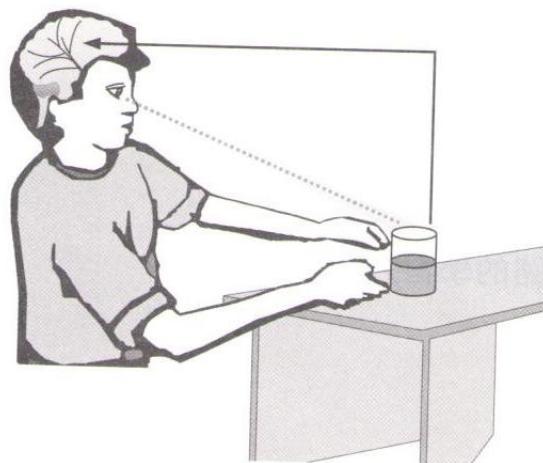


圖 3-1-1 神經系統對於外界刺激與回應圖
(圖片來源為 類神經網路導論，張斐章，張麗秋)

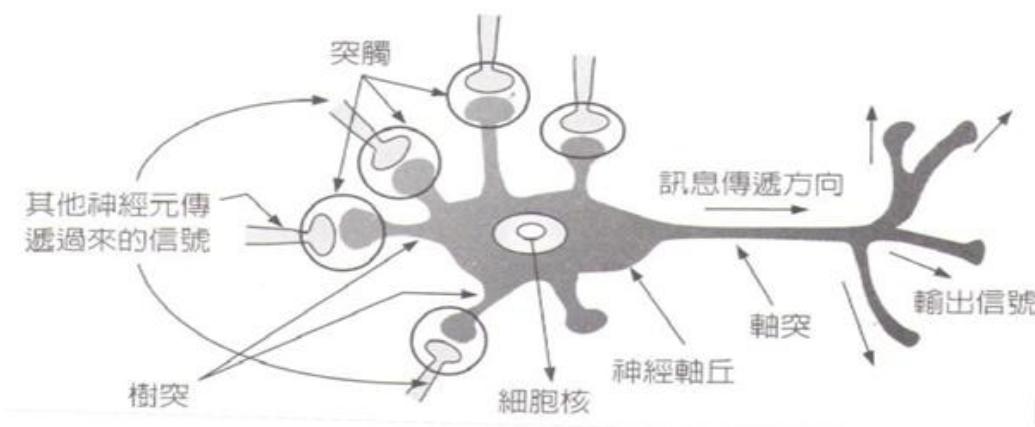


圖 3-1-2 神經元構造圖
(圖片來源為 類神經網路導論，張斐章，張麗秋)



圖 3-1-3 神經元傳遞訊息圖
(圖片來源為 類神經網路導論，張斐章，張麗秋)

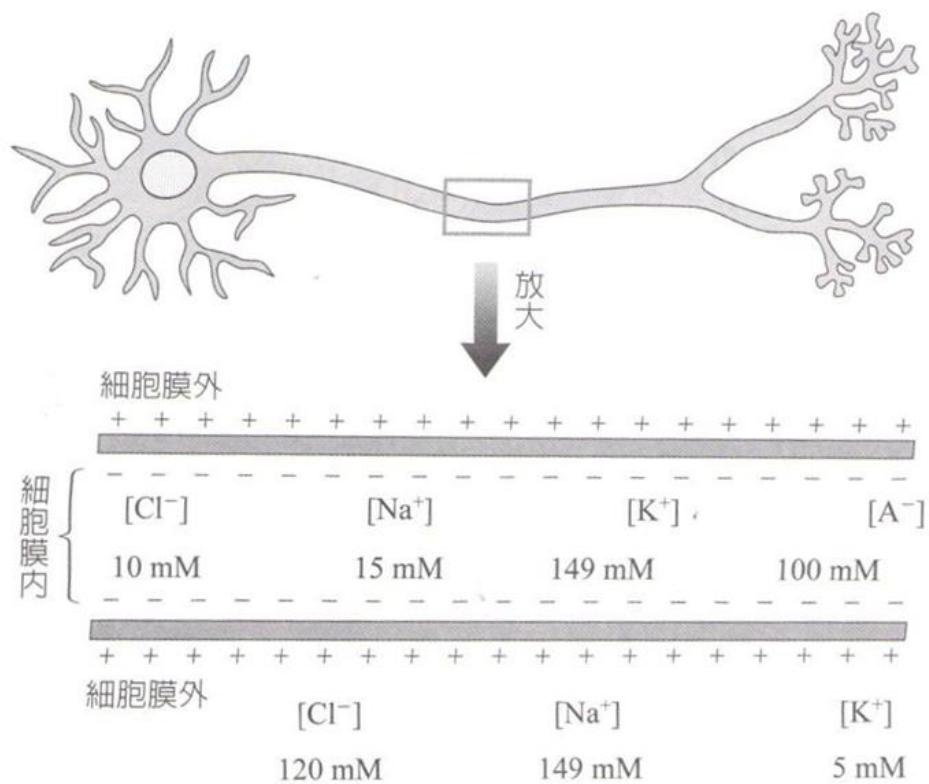


圖 3-1-4 膜電位圖
(圖片來源為 類神經網路導論，張斐章，張麗秋)

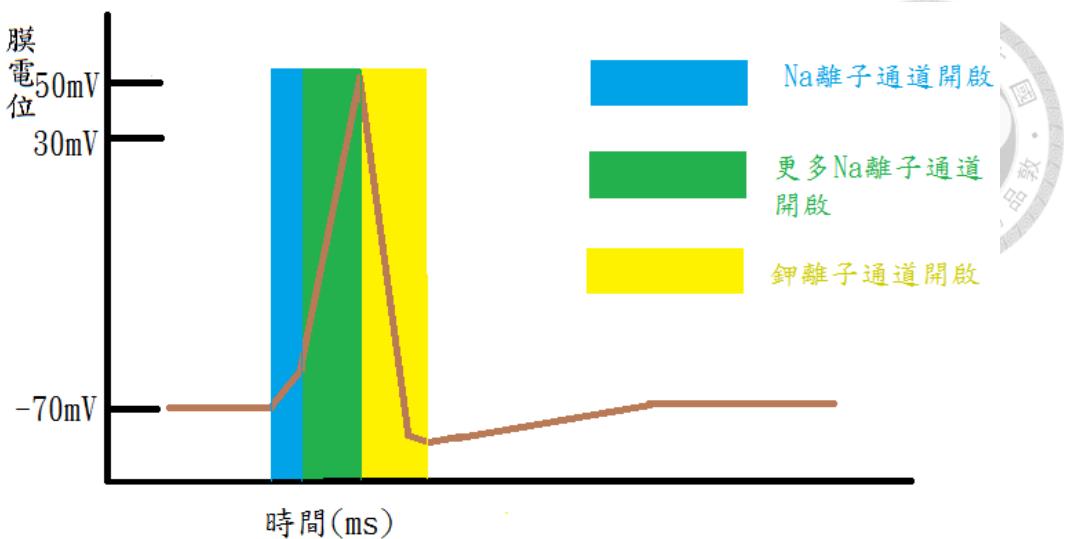


圖 3-1-5 動作電位圖

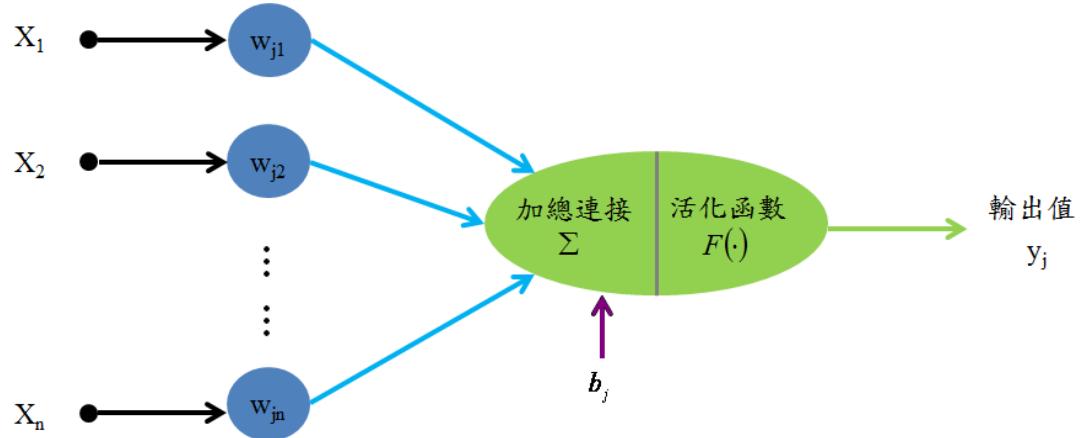


圖 3-2-1 人工神經元模型

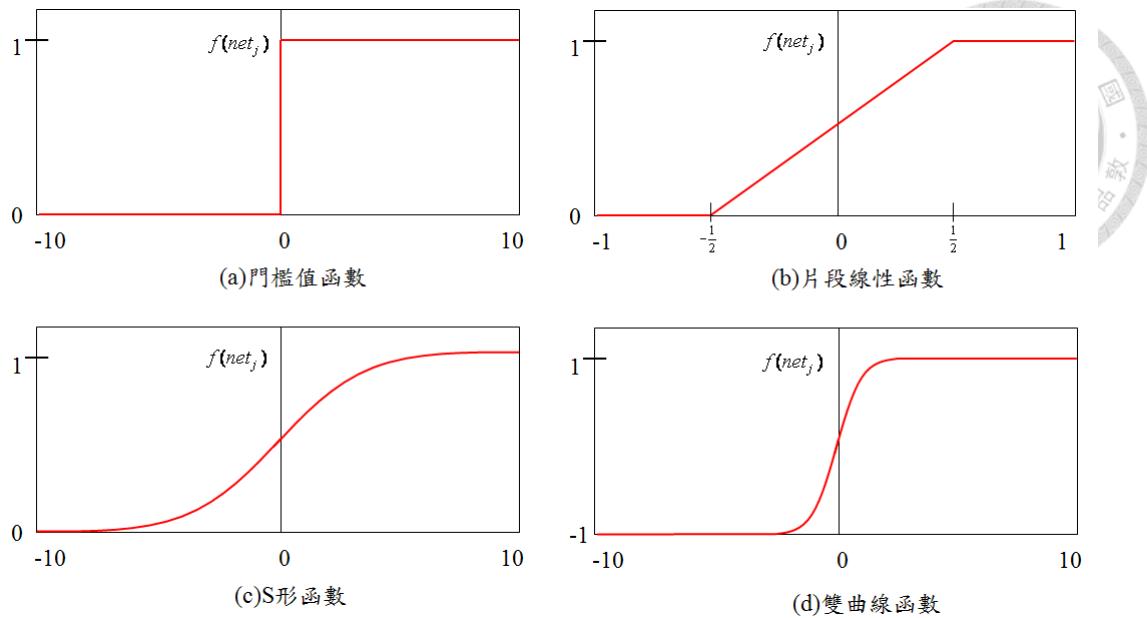


圖 3-2-2 活化函數圖

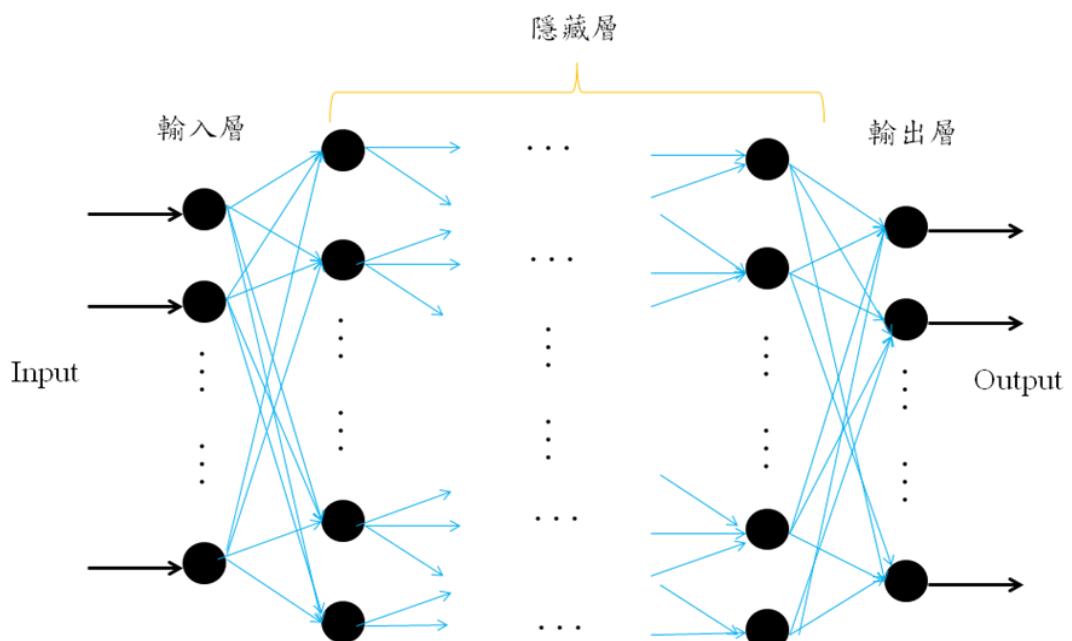


圖 3-2-3 前饋式類神經網路之架構

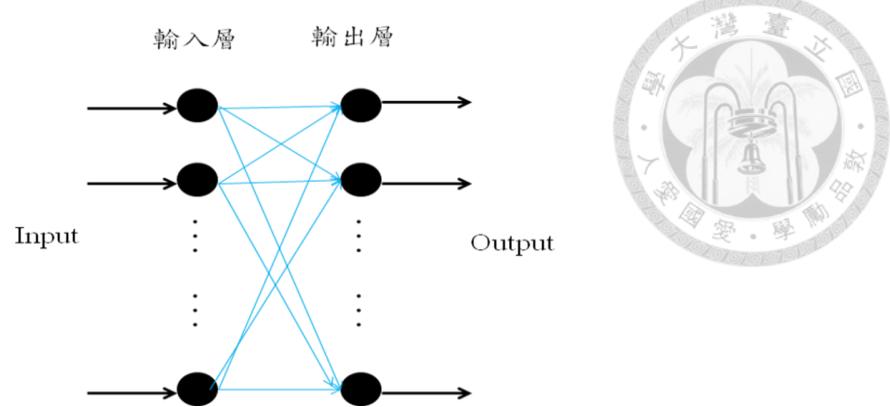


圖 3-2-4 單層前饋式類神經網路圖

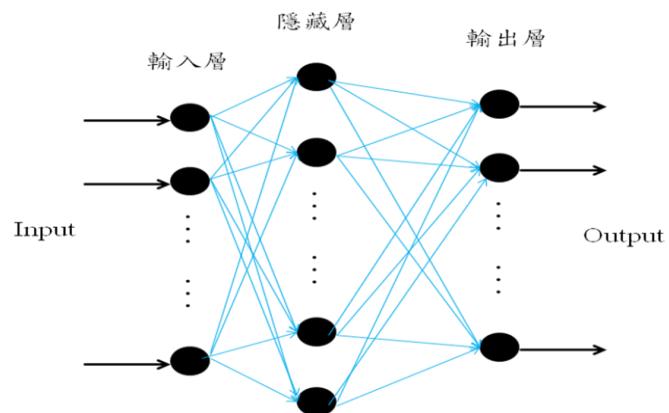


圖 3-2-5 多層前饋式類神經網路圖（隱藏層為一層時）

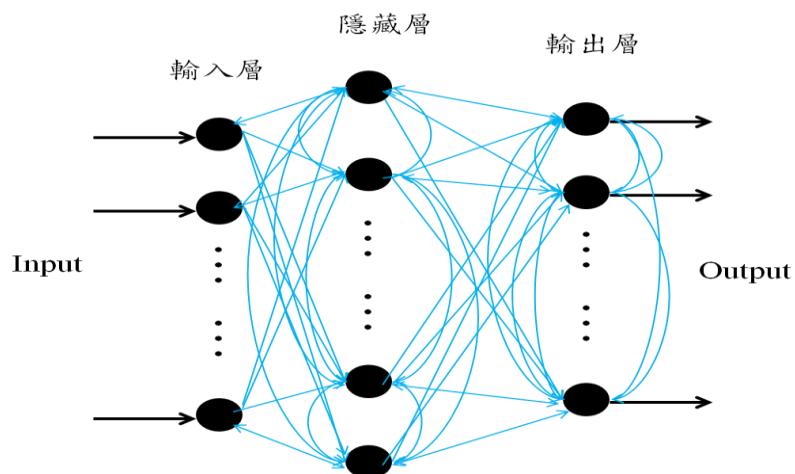


圖 3-2-6 回饋式類神經網路之架構圖（隱藏層為一層時）

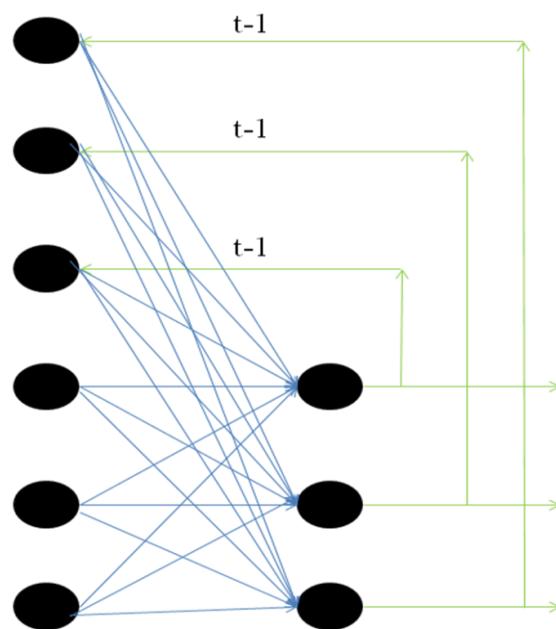
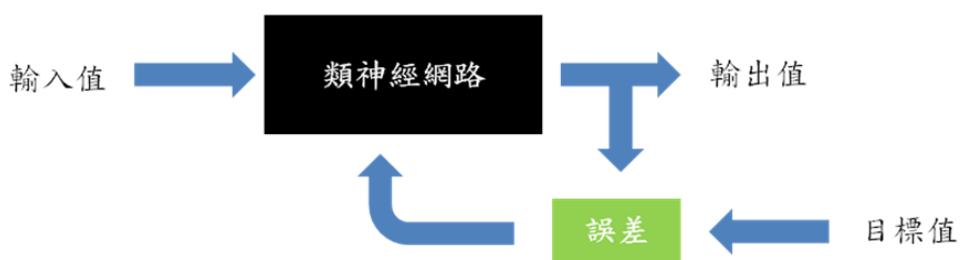


圖 3-2-7 降兩量回傳時間稽延圖



(a)非監督式學習演算法



(b)監督式學習演算法

圖 3-2-8 學習演算法分類

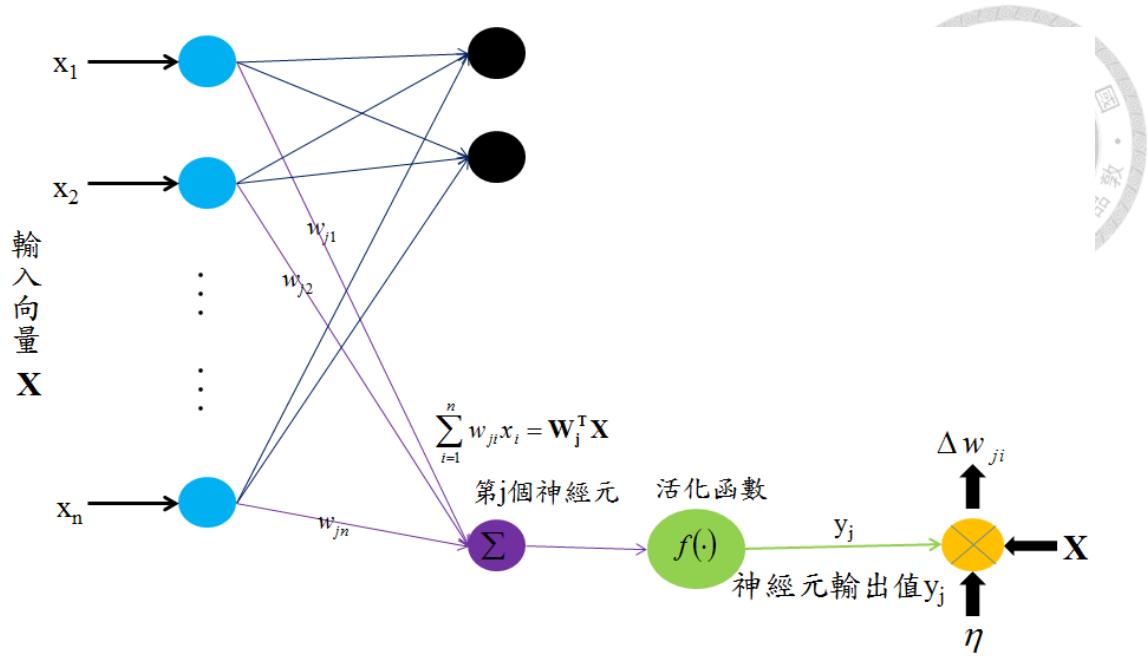


圖 3-2-9 赫賓學習法示意圖

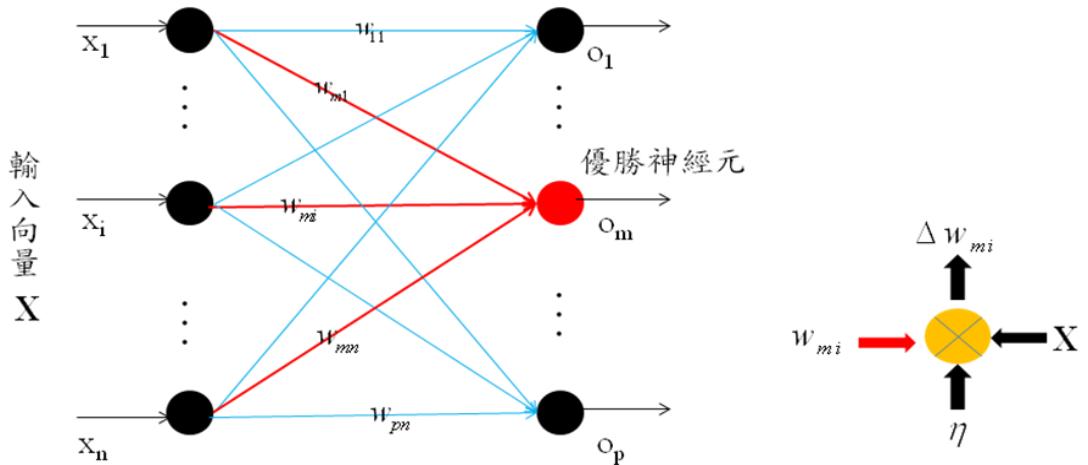


圖 3-2-10 贏者全拿學習法示意圖

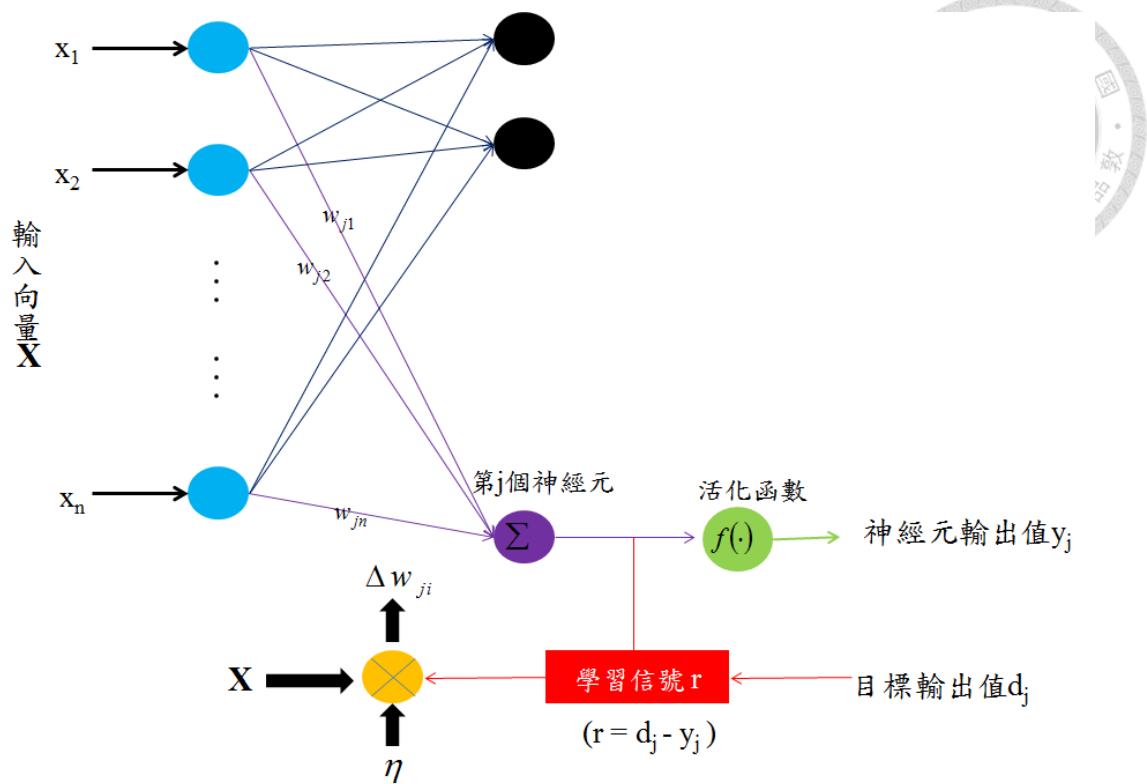


圖 3-2-11 神經元權重學習示意圖

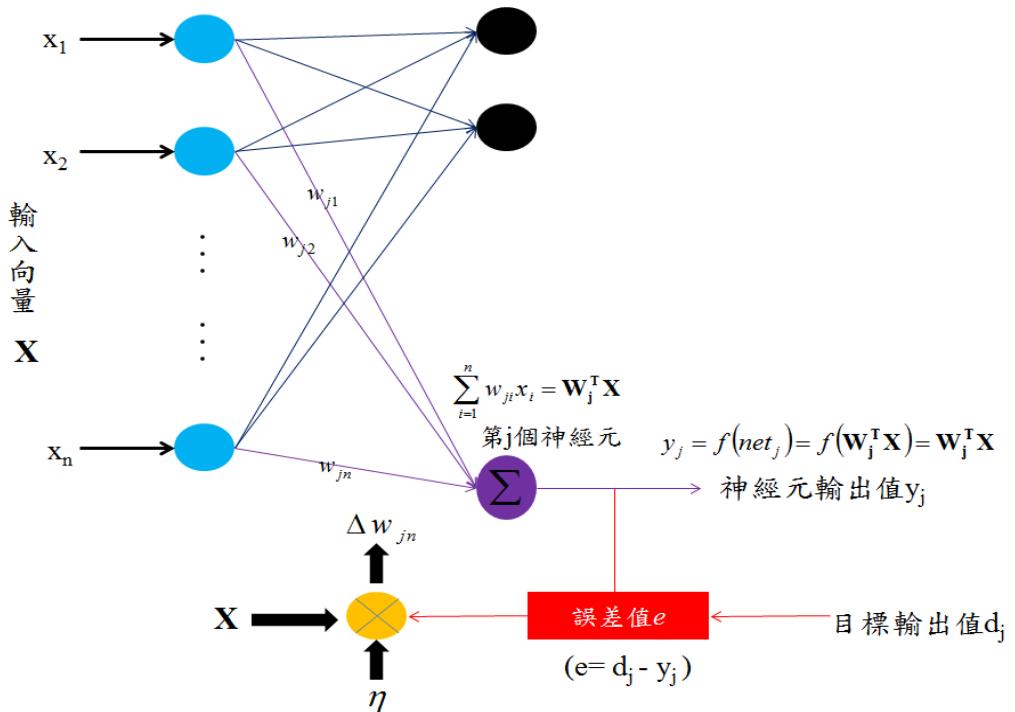


圖 3-2-12 最小均方演算法示意圖

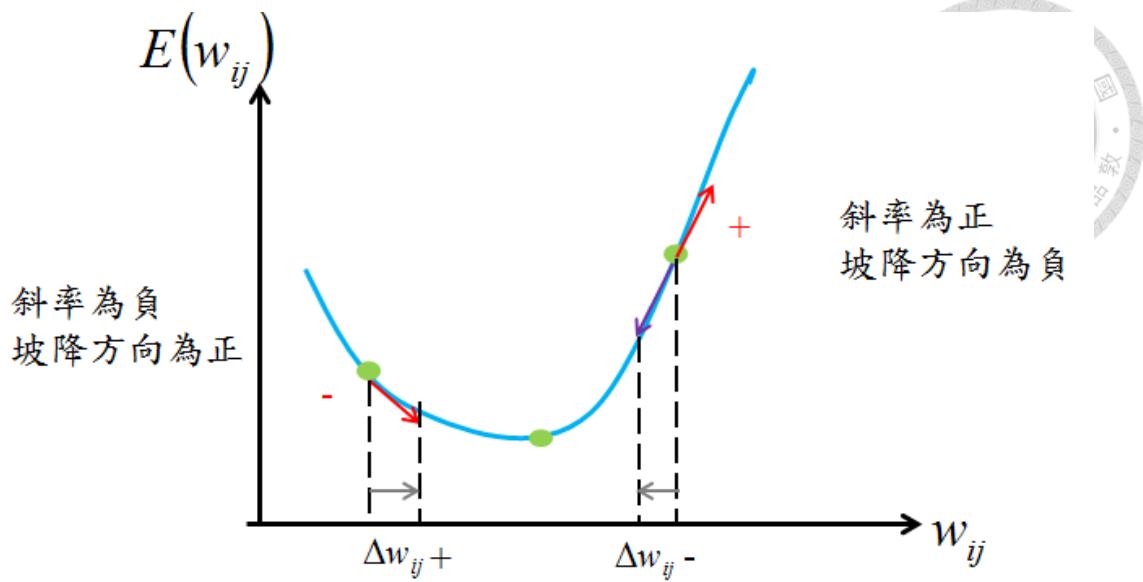


圖 3-2-13 最陡坡降法示意圖

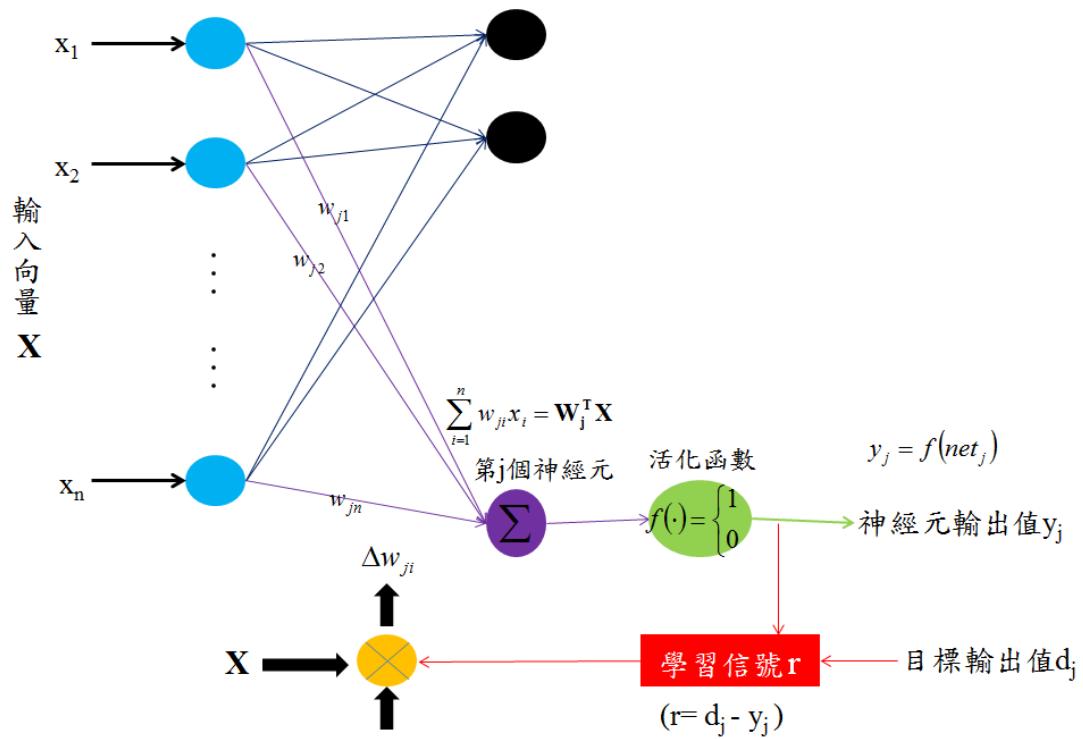


圖 3-2-14 單層感知器學習法示意圖

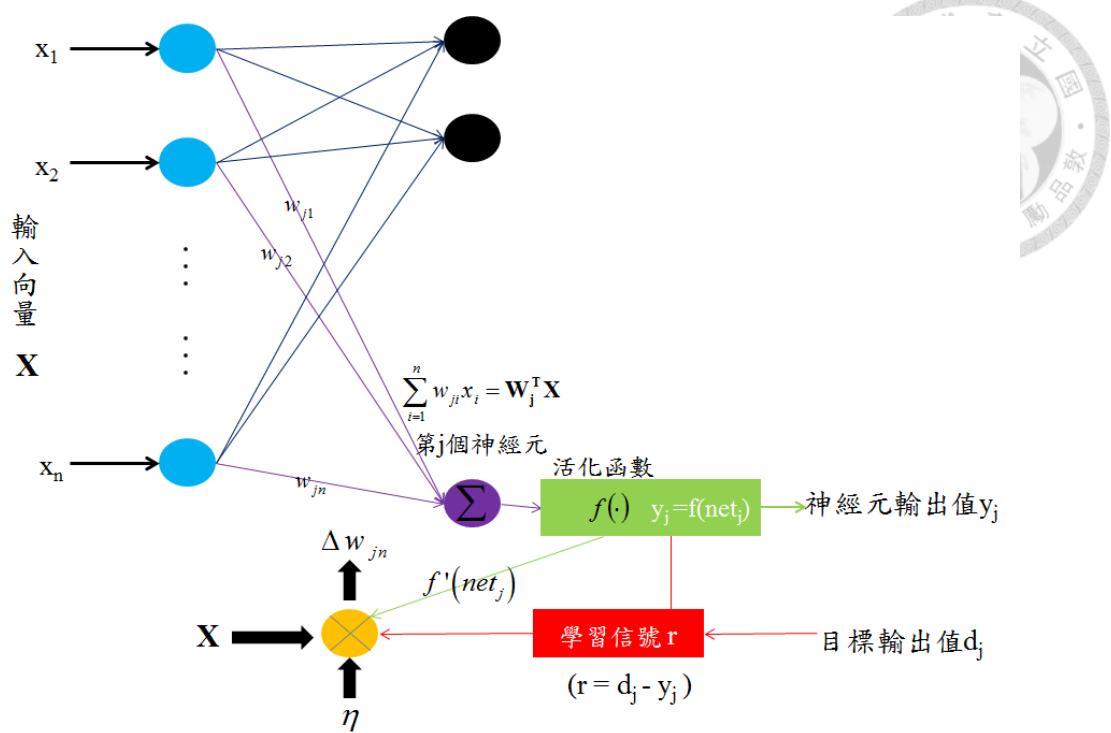


圖 3-2-15 多層感知器學習法示意圖

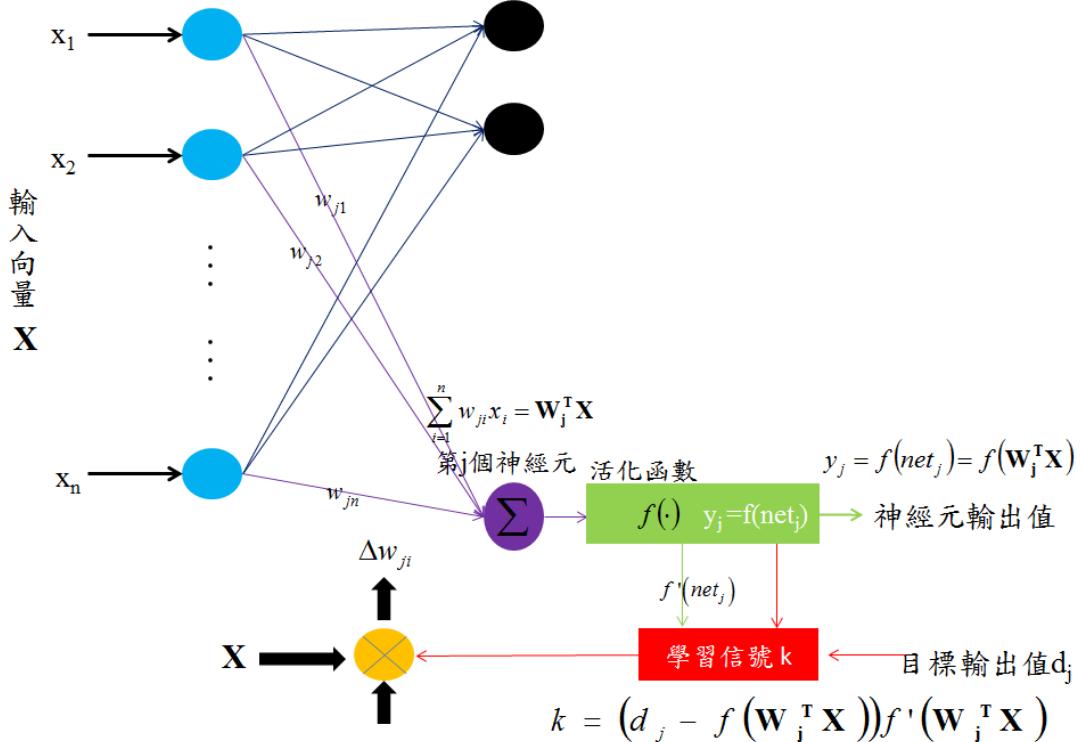


圖 3-2-16 Delta 學習法示意圖

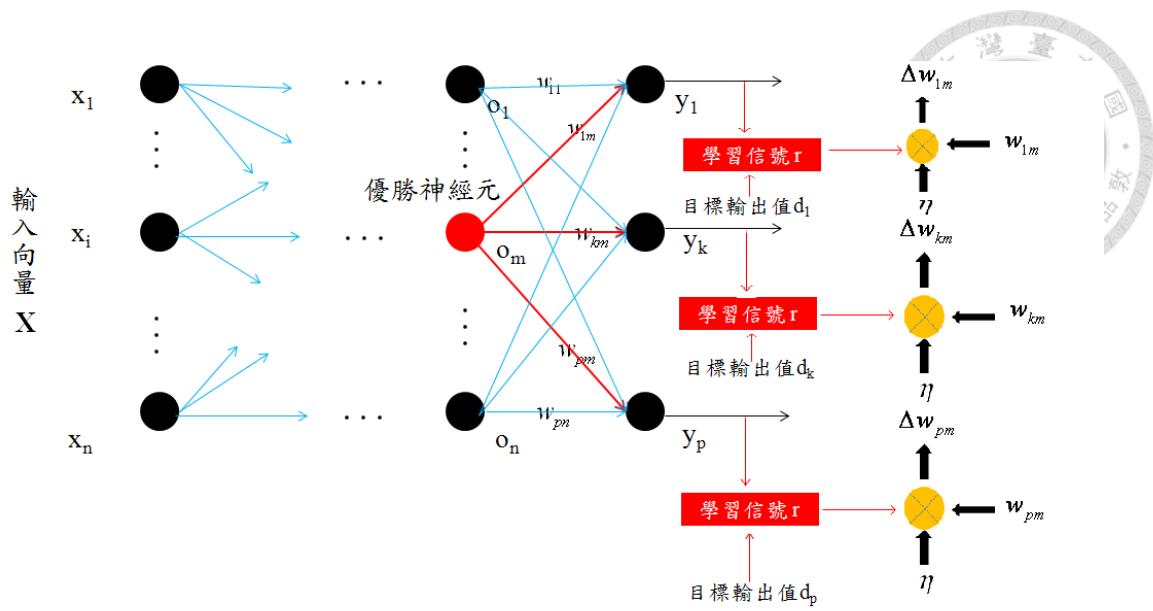


圖 3-2-17 Grossberg 學習法示意圖

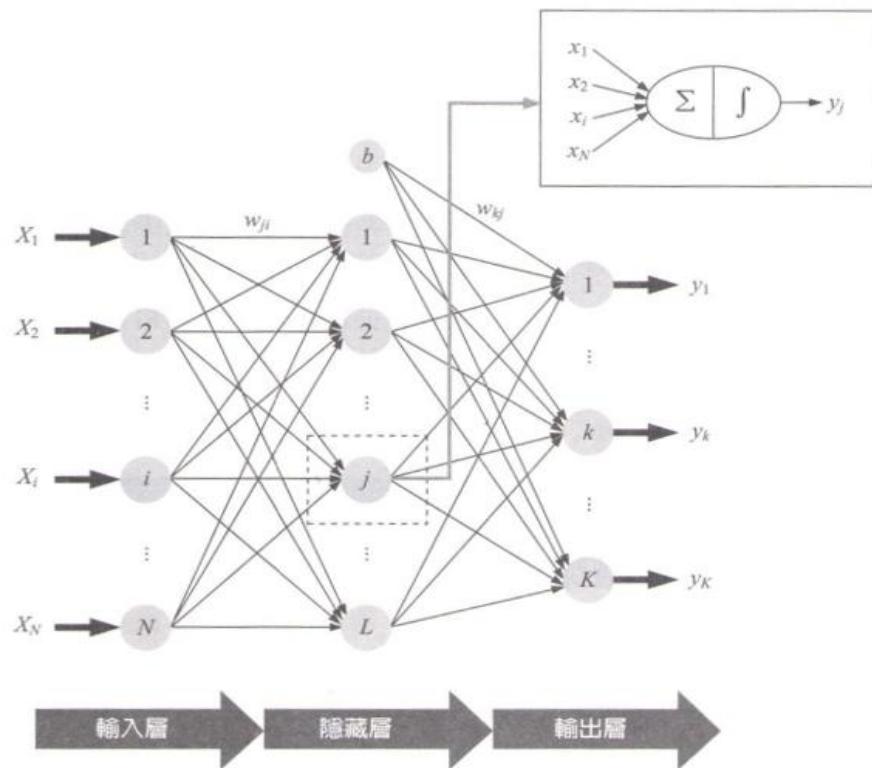


圖 3-3-1 倒傳遞類神經網路架構圖-前饋式網路
(圖片來源為 類神經網路導論，張斐章，張麗秋)

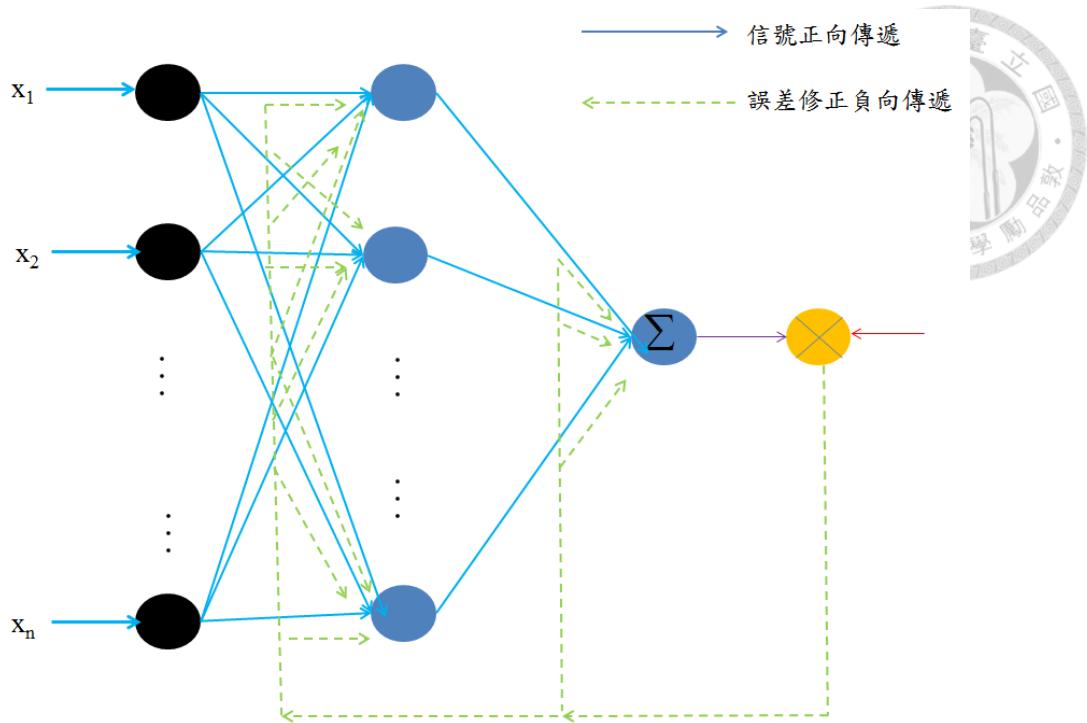


圖 3-3-2 倒傳遞類神經網路正向傳播與負向傳播圖

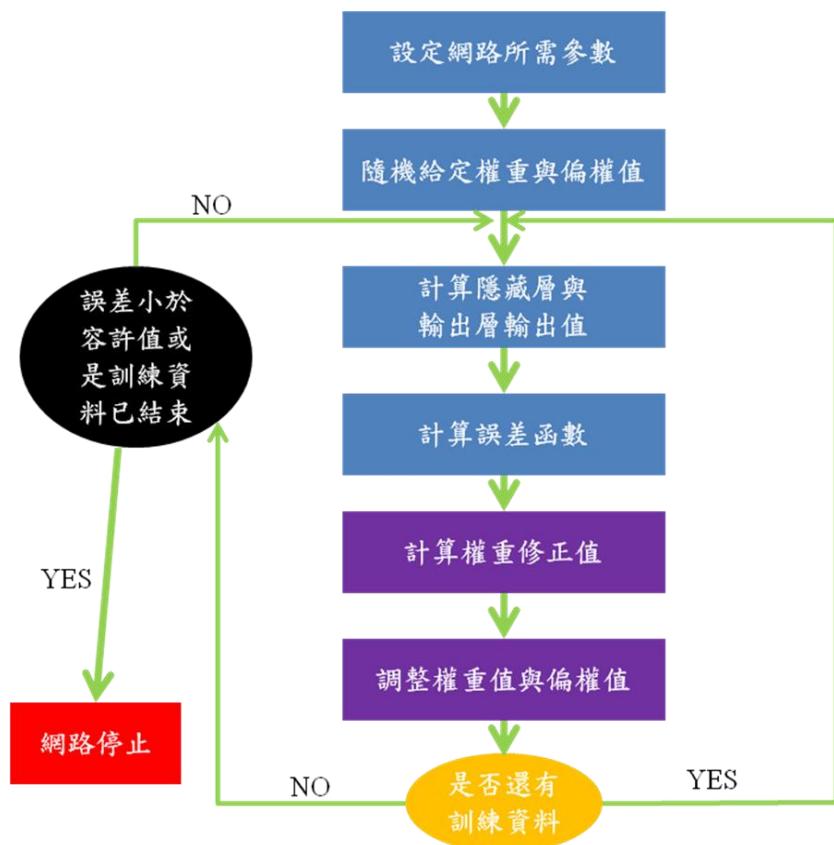


圖 3-3-3 倒傳遞類神經網路演算之流程圖

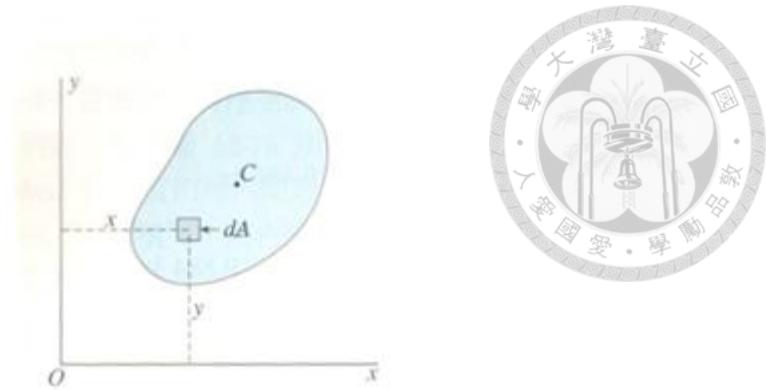


圖 3-3-4 任意形狀平面積

(圖片來源為 材料力學, James M. Gere 著, 林震, 劉興華, 蘇金佳譯,)

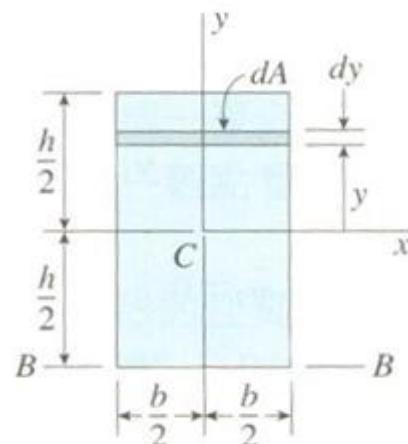


圖 3-3-5 矩形的慣性矩

(圖片來源為 材料力學, James M. Gere 著, 林震, 劉興華, 蘇金佳譯,)

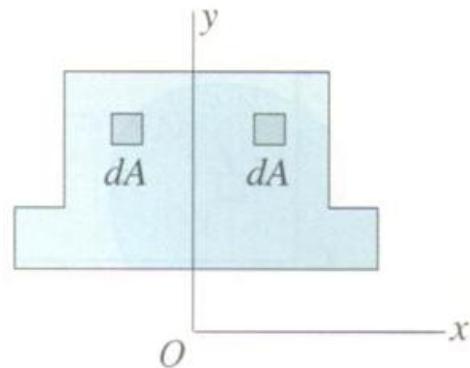


圖 3-3-6 在對稱軸線上慣性積等於零

(圖片來源為 材料力學，James M. Gere 著，林震，劉興華，蘇金佳譯，)

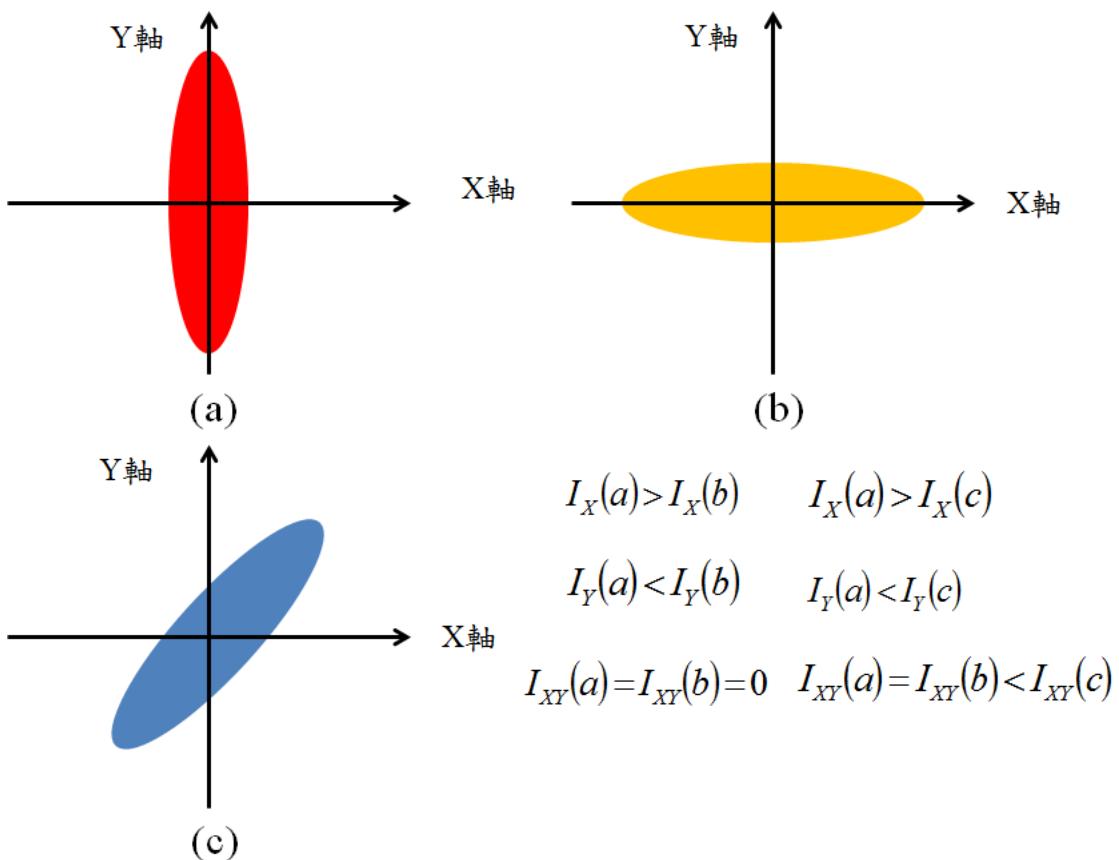


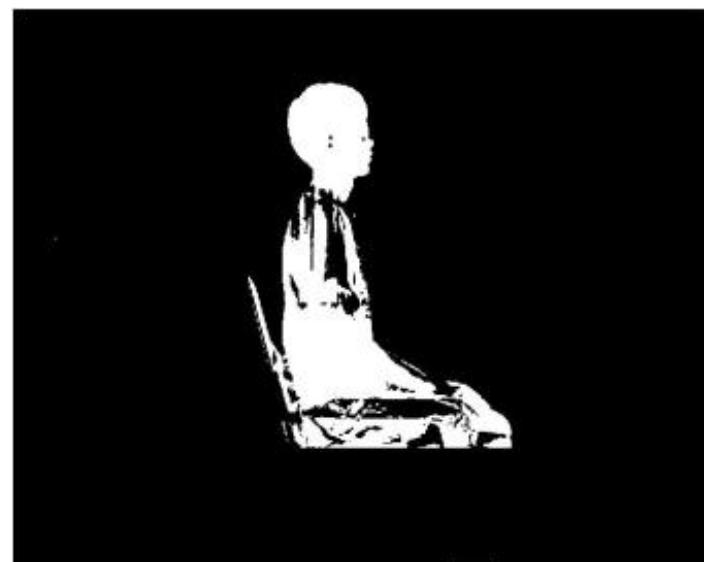
圖 3-3-7 慣性矩與慣性積與圖形關係圖



(a) 標準姿勢



(b) 坐姿一



(c) 坐姿二

圖 3-3-8 不同坐姿圖

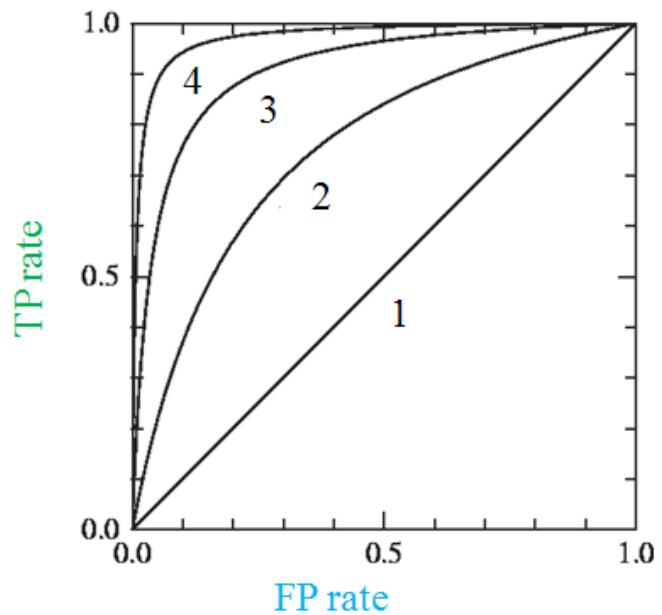


圖 3-3-9 ROC 曲線圖

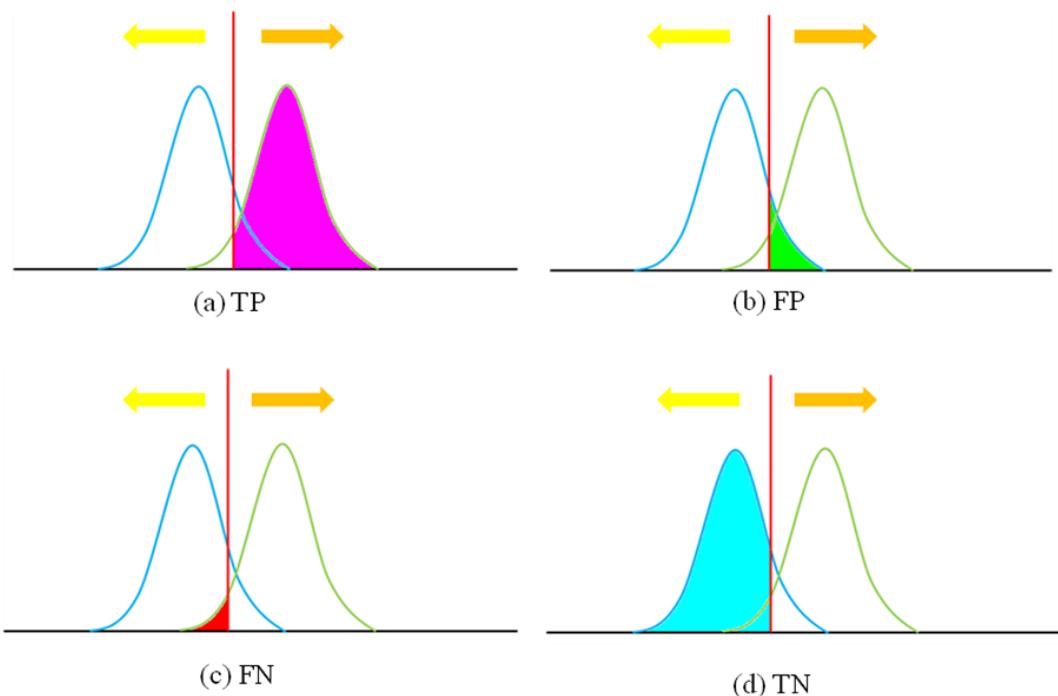


圖 3-3-10 判斷結果分佈圖

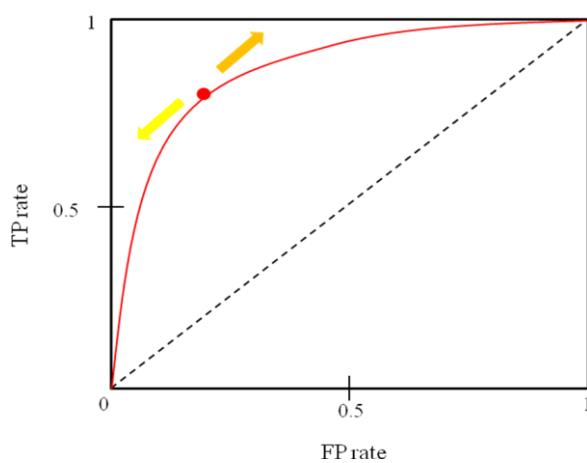
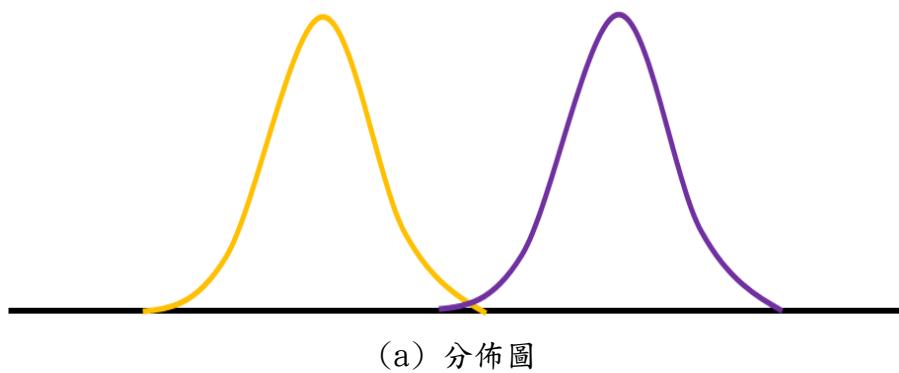
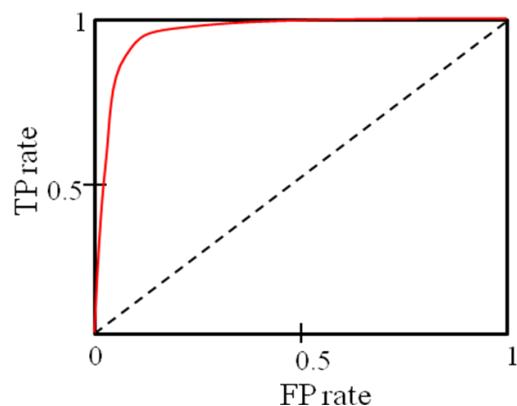


圖 3-3-11 閾值移動對照 ROC 曲線圖



(a) 分佈圖

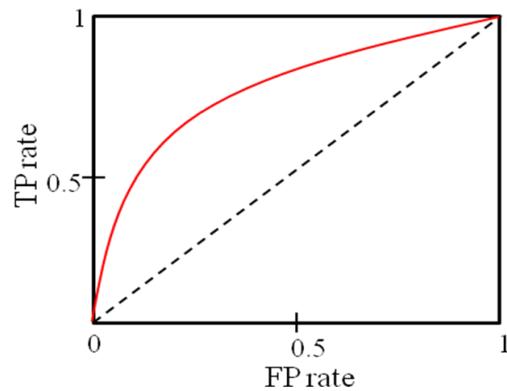


(b) ROC 曲線圖

圖 3-3-12 具有高判斷力的 ROC 曲線圖



(a) 分佈圖

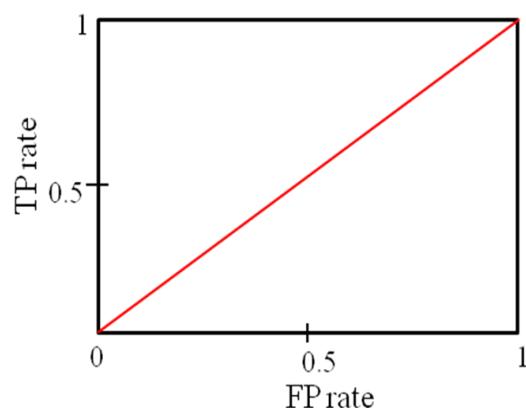


(b) ROC 曲線圖

圖 3-3-13 不具有高判斷力的 ROC 曲線圖



(a) 分佈圖



(b) ROC 曲線圖

圖 3-3-14 完全不具判斷力的 ROC 曲線圖

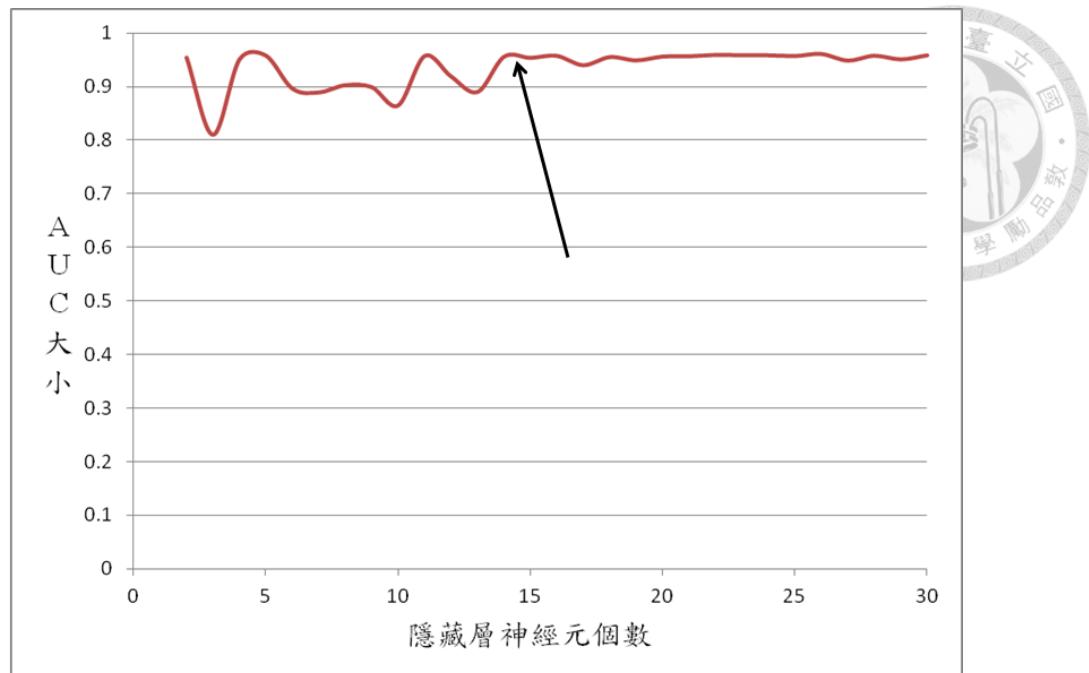


圖 3-3-15 刪除小腿之上半身神經元個數與 AUC 關係圖

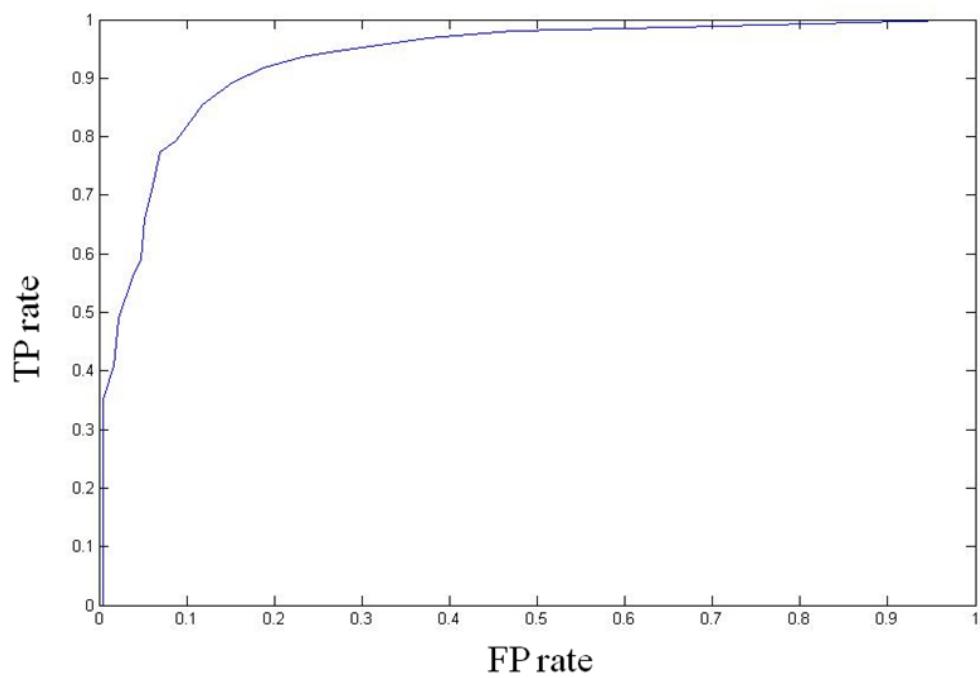


圖 3-3-16 刪除小腿之上半身之 ROC 曲線面積圖

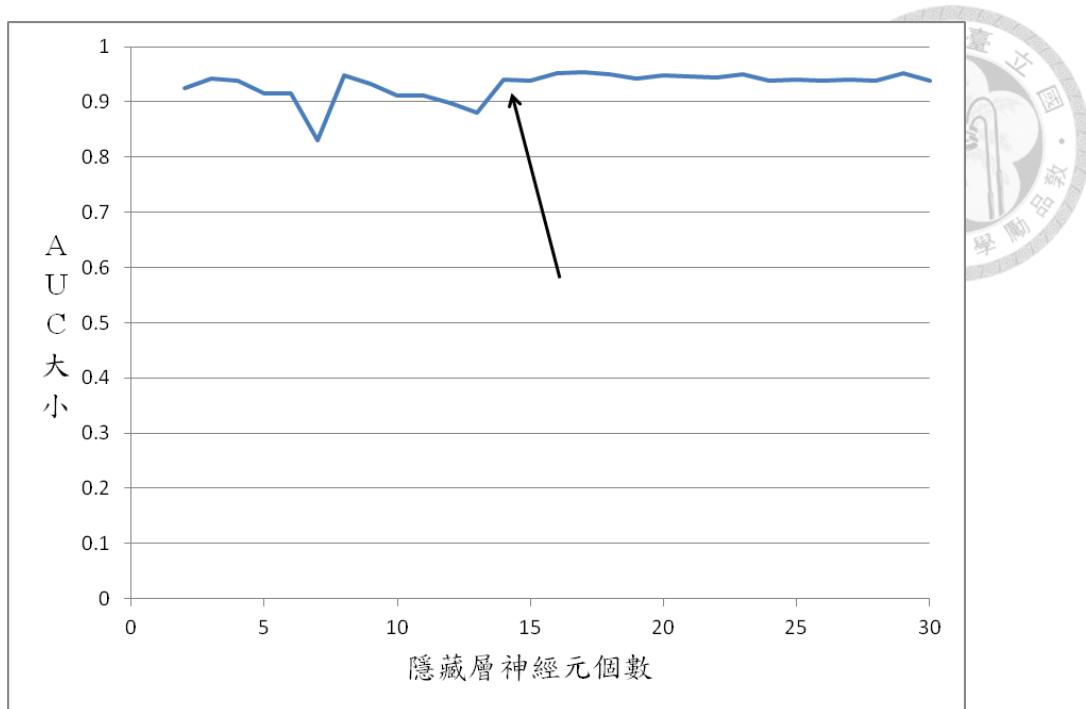


圖 3-3-17 刪除小腿、手與大腿之上半身神經元個數與 AUC 關係圖

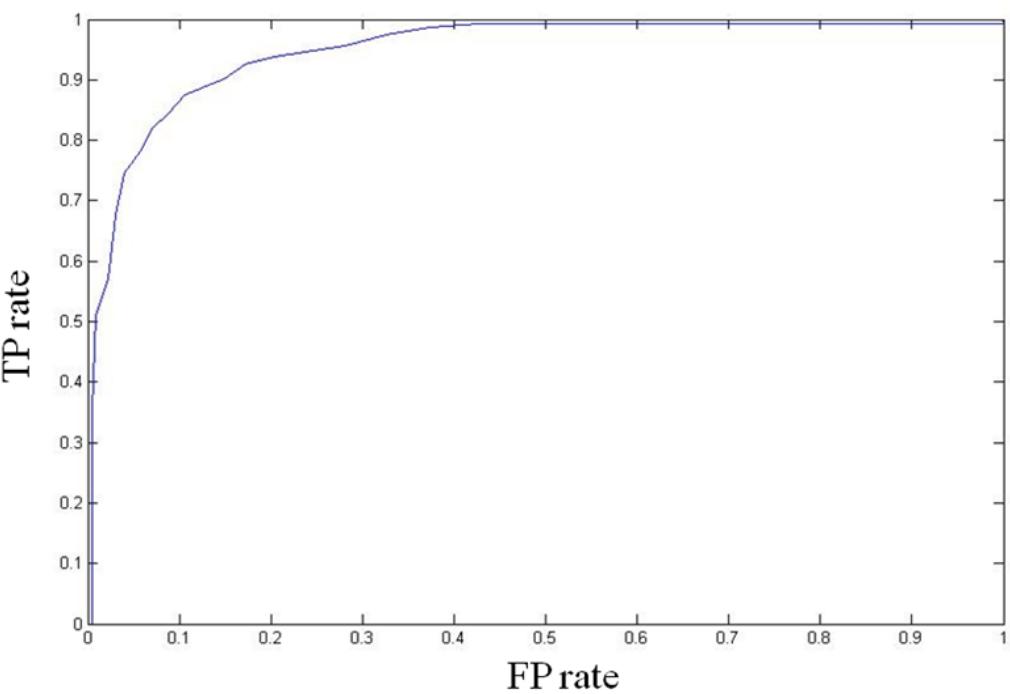
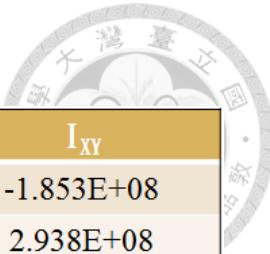


圖 3-3-18 刪除小腿、手與大腿之上半身之 ROC 曲線面積圖

表 3-3-1 姿勢正規化表格



	I_X	I_Y	I_{XY}
標準坐姿	1.290E+09	1.750E+08	-1.853E+08
坐姿1	9.310E+08	4.385E+08	2.938E+08
坐姿2	1.266E+08	1.738E+08	-1.655E+08
標準坐姿	1	1	1
坐姿1(正規化)	0.722	2.506	1.586
坐姿2(正規化)	0.981	0.993	0.893

表 3-3-2 Confuse Matrix

		實際情形	
		患病	沒有患病
判斷情形	患病	TP	FP
	沒有患病	FN	TN

表 3-3-3 ROC 量表

ROC 量表
$AUC = 0.5$ 無任何判斷能力
$0.7 \leq AUC < 0.8$ 可接受的判斷能力
$0.8 \leq AUC < 0.9$ 好的判斷能力
$AUC \leq 0.9$ 非常好的判斷能力

第四章 判斷分類器



為本研究中人體坐姿判讀流程的第三步驟：判斷分類器。介紹如何將類神經網路之演算結果透過閾值的方式來分類出坐姿正與不正，以及閾值該如何選取之過程。

當已經用 ROC 曲線下面積來選定類神經網路之後，再來就是要設立判斷的依據。我們藉由觀查訓練資料在經過類神經網路處理後其輸出值的分佈情形，來判定坐姿正與不正的閾值該為多少。

雖然目標輸出值僅為 0 或 1，但是類神經網路的輸出不見得會是零誤差，因此輸出值在經過活化函數的處理之後會坐落在 $[0, 1]$ 之間。而將訓練資料透過類神經網路處理後的輸出值，分別依照為其屬於坐姿正或坐姿不正，將輸出值出現的次數做統計結果，畫出長條圖，圖 4-1 與圖 4-2 為切除人體下半身時的統計長條圖，圖 4-3 與圖 4-4 為人體軀幹部分的統計長條圖。其中長條圖的高度，代表的就是類神經網路輸出的這個值屬於坐姿正或坐姿不正的次數，將長條圖的高度除上所有正的坐姿或是所有不正的坐姿，就會分別變成正的坐姿或不正的坐姿的機率密度函數，不同影像處理之下的機率密度分佈圖分別如圖 4-5 與圖 4-6 所示。

由圖 4-5 與圖 4-6 我們可以看出坐姿正與坐姿不正其實是分得很



開的兩大群，兩者重複的部分其實並不多，但是，我們也同樣可以從圖 4-5 與圖 4-6 中發現到，坐姿正與坐姿不正的分布雖然分別是屬於右偏型與左偏型，但是中間部分之曲線較於平緩，實在很難直接就能找出分割坐姿正與坐姿不正的閥值為何。因此，我們在此介紹可供我們找出坐姿正與坐姿不正判定閥值之參數—Precision（準確率）與 Recall（召回率）、Kappa coefficient（A. Agresti，1996）這兩大類。

Precision 與 Recall 專門用在信息檢索中，用來對信息檢索能力做出好壞，由表 3-3-2 中可以知道，將對的資訊判斷成對的，稱為 TP 值，將對的資訊判斷成不對的，稱為 FN 值，將不對的資訊判斷成對的，稱為 FP 值，將不對的資訊判斷成不對的，稱為 TN 值。而所謂的 Precision，指的就是判斷成對的資訊中，有多少是實際正確的，因此又可以寫成

$$P = \frac{TP}{TP + FP} \quad (4-1)$$

而 Recall 是指所有正確的資訊中，有多少是判斷對的資訊，因此又可以寫成

$$R = \frac{TP}{TP + FN} \quad (4-2)$$

如果說某個方法越具有判斷力的話，則 Precision 值與 Recall 值就要越大。

κ 值則是一種用來描述一致性的指標，其判斷的種類可分為人際

信度(inter-rater reliability)與人內信度(intra-rater reliability)這兩種。例如說同一位 x 光閱片醫生兩次閱片（隔一段時間後閱同一片子），要判斷這一位醫生對於自己兩次的判斷結果是否一致的現象，就稱為人內信度；如果說是同一張 x 光片有兩個醫生閱片，來探討這兩位醫生判斷的一致性，就稱為人際信度。在這裡，我們可以將人際信度的判斷對比成：對同一個姿勢（同一張 x 光），類神經網路的目標輸出與實際結果輸出（不同醫生判斷之結果），對比其一致性。如果一致性越高，則代表此時的閥值能夠越精準的劃分坐姿正與做姿不正。

κ 值本身的定義為：在考慮隨機一致性比例的情形下，修正原本的觀測一致性比例，其算式可寫成：

$$\kappa = \frac{P_0 - P_C}{1 - P_C} \quad (4-3)$$

其中 P_0 稱為觀測一致性，代表的是前後兩種判斷結果確實一致的機率； P_C 稱為期望一致性，代表的是機率上的一致性，也就是假設當這兩種判斷方式為完全獨立 (Independent) 時，仍會同時判斷正確的機率。因此 κ 值計算的目的就是在於排除機率上的一致性之下，這兩種判斷方式仍為一致的程度為何，表 4-1 代表不同 κ 值之下，這兩種判斷方法是否一致的程度，因為在統計上正確判斷發生機率的大小稱為可靠度，所以表 4-3-1 是以可靠度的說法來呈現。然而 P_0 與 P_C 在實際計算上該如何運算呢？我們舉一個例子來說明之，假設有兩位醫生

觀察 45 張受試者的胸部 x 片，其中第一位醫生判定有 26 位受試者罹患肺結核，19 位受試者沒有罹患肺結核；第二位醫生判定有 29 位受試者罹患肺結核，16 位受試者沒有罹患肺結核，而兩位醫生共同判定得病的受試者有 21 位，共同判定沒有得病的受試者有 11 位，則我們將兩位醫生的判斷結果以表 4-2 表示之。由表 4-2 可看出：

$$P_0 = \frac{a+d}{n} = \frac{21+11}{45} = 0.7111$$

$$P_C = \frac{f_1}{n} \times \frac{g_1}{n} + \frac{f_2}{n} \times \frac{g_2}{n} = \frac{26}{45} \times \frac{29}{45} + \frac{19}{45} \times \frac{16}{45} = 0.5225$$

$$\kappa = \frac{P_0 - P_C}{1 - P_C} = \frac{0.7111 - 0.5225}{1 - 0.5225} = 0.3950$$

如果將兩位醫生的判斷結果替換成類神經網路的目標輸出值與結果輸出值之結果，則表 4-2 就會被替換成表 4-3，表 4-3 的 a 就是我們所謂的 TP 值， b 就是 FP 值， c 就是 FN 值， d 就是 TN 值，則利用這些值就能夠計算出 κ 值的大小。

不論是計算 κ 值或是 Precision 與 Recall 值，其背後原理都是來自於表 3-3-2 的 Confuse Matrix，然而本研究所採用的方式是計算 κ 值，原因是因為在計算 κ 值時有進行排除因機率而產生一致性的可能性在裡頭，因此其計算出來的值的準確性會較嚴格，故本研究採取 κ 值來當作選取閾值的方式。

確立好閾值選取的方式之後，我們將類神經網路輸出值的統計圖從 0 到 1 每隔 0.05 累計一次，畫出其機率密度函數，並藉由機率密

度函數乘上總數量去求出從 0 到 1 每隔 0.05 取一次閥值時，TP、FP、FN 與 TN 值各為何，進而求出此時判斷結果的 κ 值。而當 κ 值越大，就說明此時類神經網路的目標輸出值與實際輸出值越吻合！因此 κ 值最大的時候所對應的閥值，就是我們所要的類神經網路之最佳閥值。

由圖 4-7 可以看出，在寬鬆的上半身定義下圖形擷取的判斷閥值為 0.55，此時的 κ 值為 0.771；由圖 4-8 可以看出，在嚴格的上半身定義下圖形擷取的判斷閥值為 0.55，此時的 κ 值為 0.792，從表 4-1 表示 κ 值判斷能力的表格中我們可以知道：當 κ 值介在 0.60 與 0.80 之間的時候，此時的可靠度程度為可靠度優。

因此，總結本研究中所要發展的坐姿判斷系統架構內容：在寬鬆的上半身定義下圖形擷取時，類神經網路隱藏層神經元數量選用 14 個，判斷閥值則設為 0.55，大於此閥值視為正姿，小於此閥值則視為不正姿；在嚴格的上半身定義下圖形擷取時，類神經網路隱藏層神經元數量為 14 個，判斷閥值設為 0.55，大於此閥值視為正姿，小於此閥值則視為不正姿。

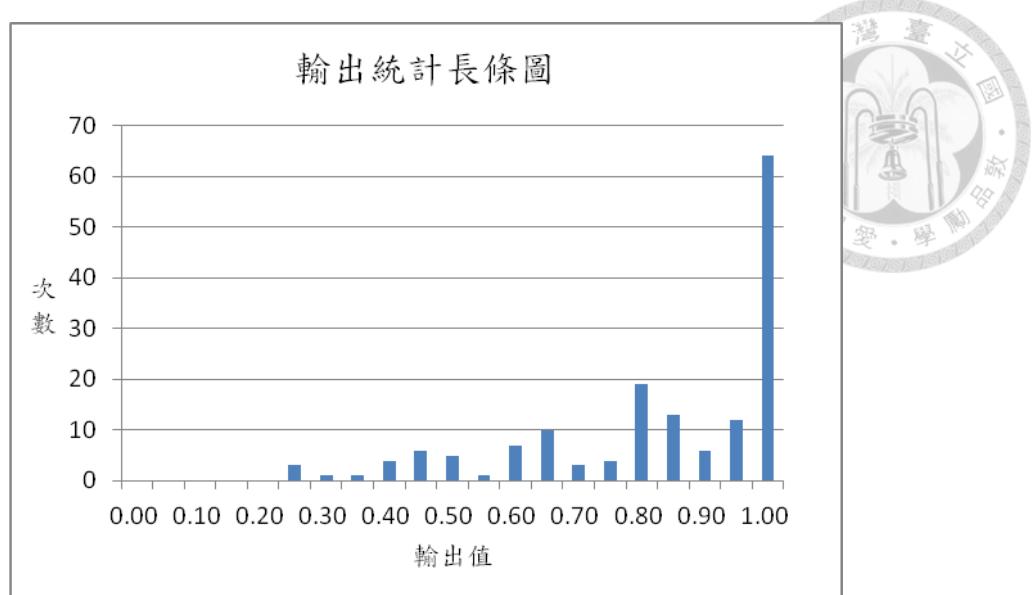


圖 4-1 刪除小腿之上半身之坐姿正輸出統計長條圖

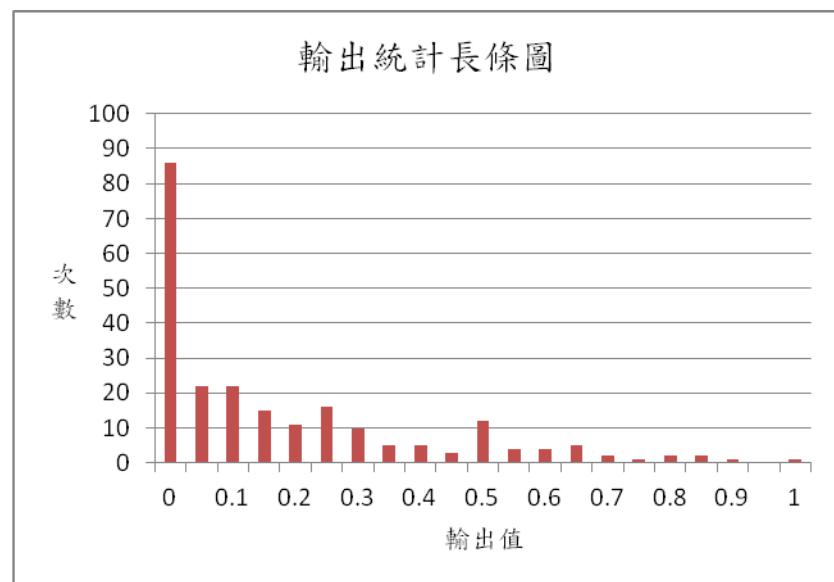


圖 4-2 刪除小腿之上半身坐姿不正輸出統計長條圖

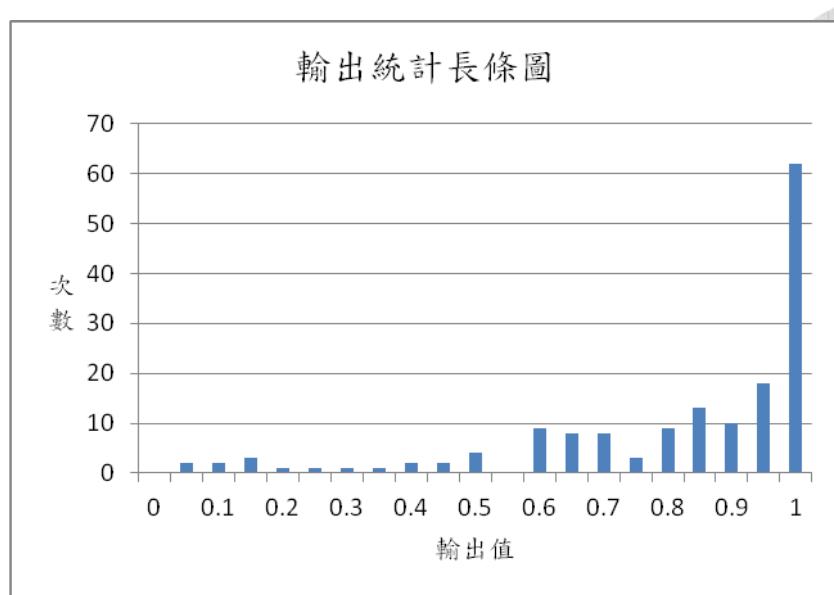


圖 4-3 刪除小腿、手與大腿之上半身坐姿正時輸出統計長條圖

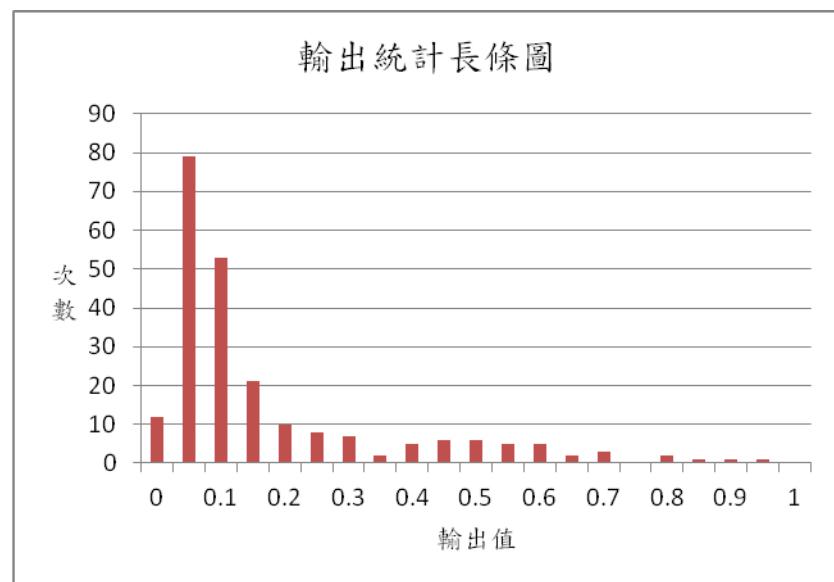


圖 4-4 刪除小腿、手與大腿之上半身坐姿不正時輸出統計長條圖

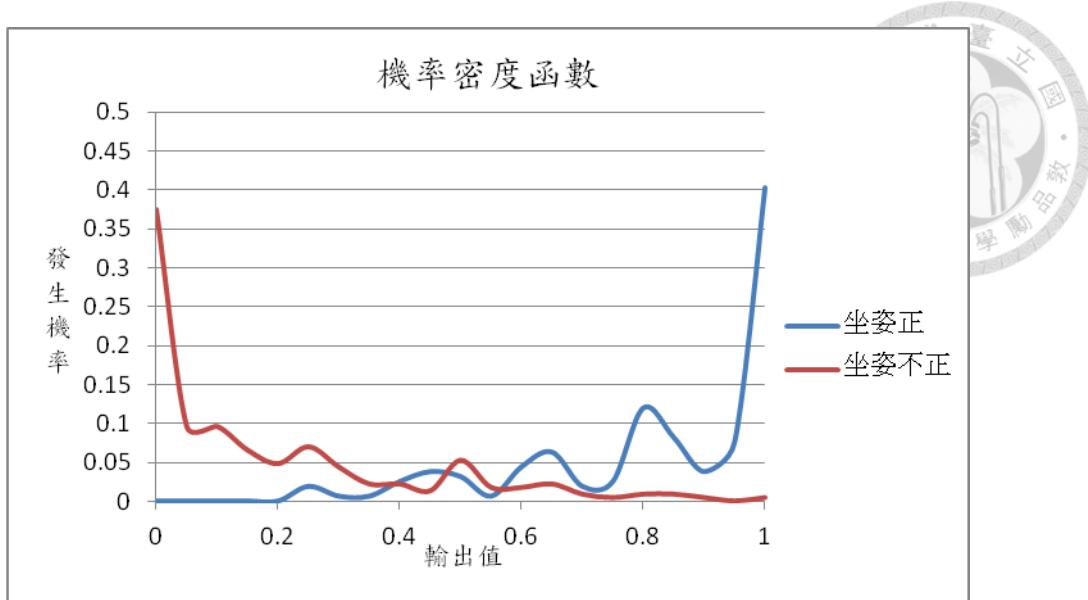


圖 4-5 刪除小腿之上半身機率密度函數圖

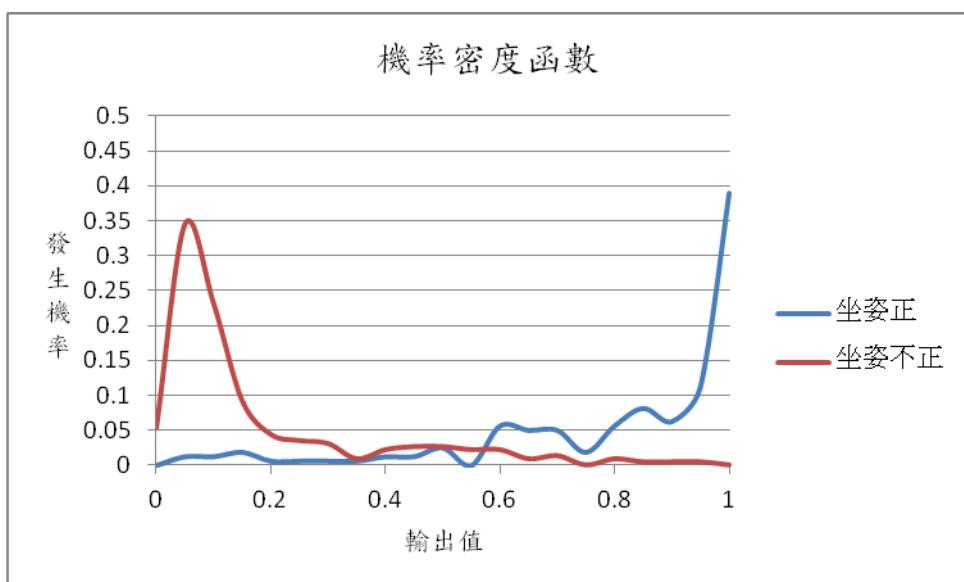


圖 4-6 刪除小腿、手與大腿之上半身機率密度函數圖

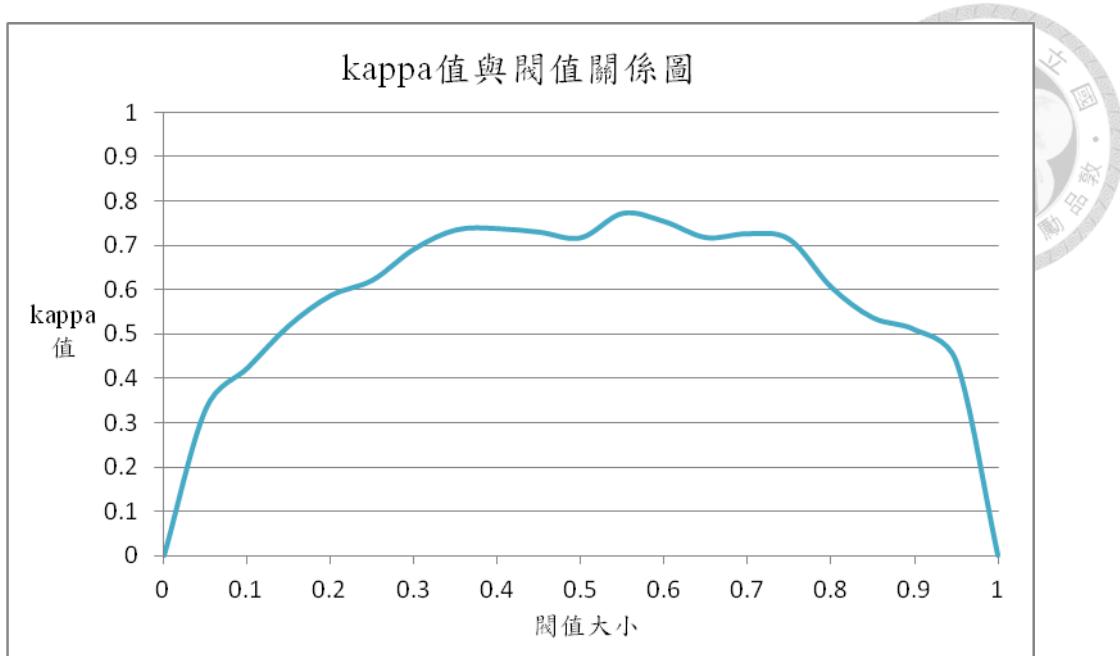


圖 4-7 刪除小腿之上半身判斷閥值與 κ 圖

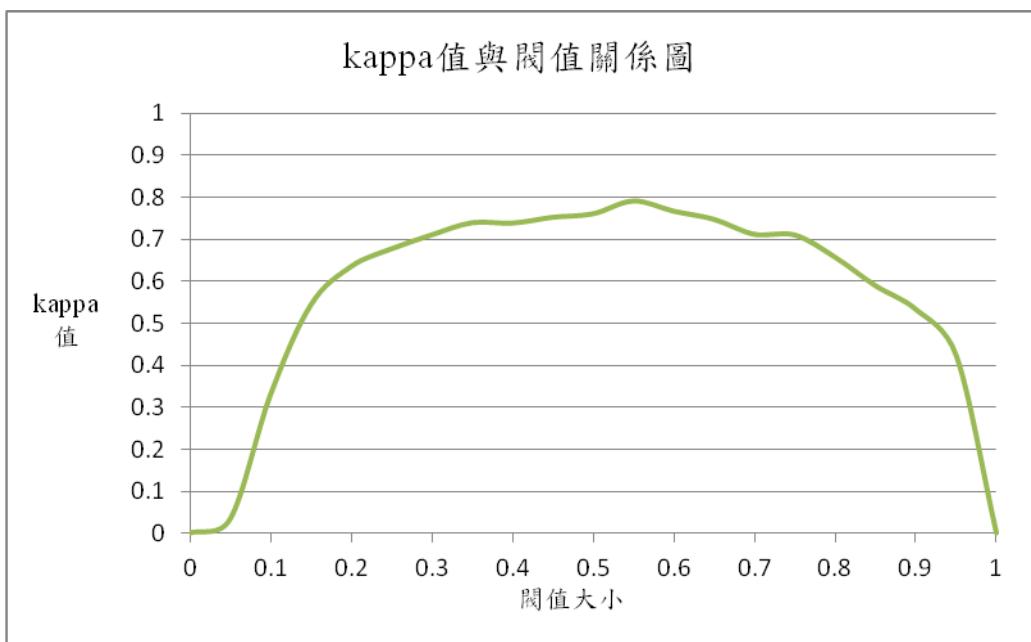
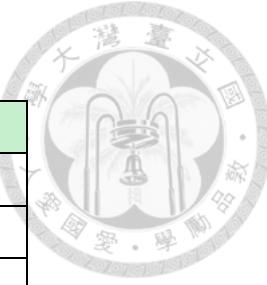


圖 4-8 刪除小腿、手與大腿之上半身判斷閥值與 κ 圖

表 4-1 κ 量表

κ 量表	
$\kappa \leq 0.40$ 無可靠度	
$0.40 < \kappa \leq 0.60$ 可靠度中等	
$0.60 < \kappa \leq 0.80$ 可靠度優	
$0.80 < \kappa \leq 1$ 完全可靠	

表 4-2 兩位醫生判斷結果之對照表

		第一位醫生判斷		
		患病	沒有患病	
第二位醫 生判斷	患病	21(a)	8(b)	29(g1)
	沒有患病	5(c)	11(d)	16(g2)
		26(f1)	19(f2)	45(n)

表 4-3 類神經網路目標與實際輸出結果之對照表

		目標輸出值		
		坐姿正	坐姿不正	
類神經網 路判斷	坐姿正	TP(a)	FP(b)	TP+FP(g1)
	坐姿不正	FN(c)	TN(d)	FN+TN(g2)
		TP+FN(f1)	FP+TN(f2)	TP+FP+FN+TN(n)

第五章 坐姿判斷之實驗分析與討論



本章將介紹坐姿判斷實驗分析內容，包括實驗受試者、實驗流程、實驗數據及分析其實驗結果。

5-1 受試者與受試環境介紹

實驗受試者為台大應用力學研究所 227 結構安全評估實驗室的 3 名碩士一年級生、3 名碩士二年級生、3 名行政助理以及 1 名博士後研究員，總共 10 名受試者。

為了模擬與考慮到不同室內燈光的真實環境，受試者的受試時間白天與晚上都有，地點選用在國立台灣大學應用力學研究所二樓的 231 教室與 233 教室，並且也直接利用教室裡面的桌椅，以驗證在不同桌椅下此判斷坐姿方法之普遍性。

5-2 實驗流程與結果

整個實驗判斷流程如圖 5-2-1 所示，一開始利用相機拍攝空的桌椅與環境當做背景部分，之後受試者坐在椅子上做出一標準姿勢，之後將姿勢圖與背景圖校正光線後相減，再經過高斯平滑濾波與二元化處理，擷取目標圖型，之後計算慣性積與慣性舉等參數。爾後受試者陸續做出不同的坐姿，這些坐姿也經過上述標準坐姿的處理過程，然

後把非標準坐姿計算出來的參數值除上標準坐姿的參數值進行正規化，之後將各姿勢正規化的參數帶入到利用訓練資料訓練好的類神經網路中去計算，之後再將類神經網路的輸出值直接套入分類器判斷系統判斷出此姿勢為正姿或是不正姿。

本實驗蒐集了十位受試者共 708 張姿勢圖，選取其中 388 張姿勢圖當作訓練資料，訓練出本研究中的所需的類神經網路架構與分類器判斷系統規則，剩餘的 320 張姿勢圖則當作測試資料，測試其效果如何。圖 5-2-2 為訓練資料在刪除小腿時各參數經正規化後的分布圖，圖 5-2-3 為訓練資料在刪除小腿、大腿與手時各參數經正規化後的分布圖，圖 5-2-4 為測試資料在刪除小腿時各參數經正規化後的分布圖，圖 5-2-5 為測試資料在刪除小腿、大腿與手時各參數經正規化後的分布圖。由這些圖可以看出當坐姿正的時候，參數 I_x 普遍會比坐姿不正來的大，參數 I_y 與 I_{xy} 普遍會比坐姿不正來的小。

測試資料刪除小腿時的類神經網路之 ROC 曲線下如圖 5-2-6 所示，面積為 0.8235；在此類神經網路下不同閥值與 κ 值關係如圖 5-2-7 所示，在分類器判斷閥值為 0.580 時， κ 值為 0.545，套用表 4-1 之結果可知此時的 κ 值代表可靠性中等；而此時的 TP 等值的四個分布如表 5-2-1 所示，可以看出原本姿勢正被判斷成姿勢不正的姿勢有 32 個，原本姿勢不正卻被判斷成姿勢正的姿勢有 34 個。

測試資料計算在刪除小腿、大腿與手時的類神經網路之 ROC 曲

線下如圖 5-2-8 所示，面積為 0.8894；在此類神經網路下不同閥值與

κ 值關係如圖 5-2-9 所示，在分類器判斷閥值為 0.55 時， κ 值為 0.653，

套用表 4-1 之結果可知此時的 κ 值代表考靠性優；而此時的 TP 等值

的四個分布如表 5-2-2 所示，可以看出原本姿勢正被判斷成姿勢不正

的姿勢有 35 個，原本姿勢不正卻被判斷成姿勢正的姿勢有 20 個。

5-3 實驗討論

不管是只刪除小腿的下半身，或者是刪掉所有腿部與手的目標圖形所計算出的參數值，由圖 5-2-2 至圖 5-2-5 可以看出在參數 I_x 的部份，坐姿正會比坐姿不正來的大，而在參數 I_y 與 I_{xy} 部分，坐姿不正比坐姿正來的小。然而因為每個人一開始給定的標準坐姿不見得一定會非常的「正」，因此在正規化之後，同樣都是 0.6 的正規化值，對於第一位受試者來說是坐姿「不正」，對第二位受試者來說是坐姿「正」，因此在圖 5-2-2 至圖 5-2-5 中，會發現坐姿正與坐姿不正在中間會有交界的地帶出現。

由圖 5-2-6 與圖 5-2-8 中可以看出，在刪掉所有腿部與手的情況下，判斷的結果會比較好；由表 5-2-1 與表 5-2-2 也可以看出，在 TP 值的部分這兩種目標圖型擷取下判斷出來的結果差不多，但是在 TN 值的部分，刪除掉所有腿與手的圖形擷取之結果會比只刪除小腿之結

果好很多，TP 值代表的意思是坐姿正判斷成坐姿正，如果 TP 值下降則代表此系統對於坐姿正的誤判率高，也就是判斷較為「保守」的意思；而 TN 值代表的意思是坐姿不正判斷成坐姿不正，如果 TN 值下降則代表此系統對於坐姿不正的誤判率高，也就是「無法糾正坐姿不正」之結果，因此對於本坐姿判斷系統而言，TN 值的重要性比 TP 值還要高。

本研究的坐姿判斷系統在只刪除小腿部分的情形下，TP rate (又稱為靈敏度，sensitivity)為 80%，TN rate 為 78.75%，準確率為 79.38%；在刪除腿部與手的情形下，TP rate 為 78.13%，TN rate 為 87.5%，準確率為 82.82%。由實驗的結果可以得知，利用慣性矩與慣性積當作圖形特徵參數時，是能夠作為判斷坐姿好壞的依據，然而仍有一些姿勢被誤判，以下為探討誤判坐姿的部分。

1. 將正姿誤判成分正姿的部分：在只刪除小腿部分的情形下，會很容易對於一些明明是正的坐姿，但只是將手放置上桌上等姿勢判斷成不正，如圖 5-3-1 所示，這類型的姿勢在只刪除小腿的圖形擷取的部分，幾乎都被誤判，因為在沒有刪除手的情形下，手的影像會使的慣性矩矩中 I_y 變大，因此系統會將這樣的姿勢判成坐姿不正。而在刪除所有腿部與手的情形下，會對於一些影像模糊(聚焦沒對好)或者是影像處理的不夠乾淨的圖形產生誤差，如圖 5-3-2 與圖 5-3-3，因為所

選取的區塊變小了，一些沒有完全清除掉的背景對於參數值的計算會造成較大的影響，進而產生誤判。而這兩種圖形擷取同時都可能產生誤判的坐姿，則為身軀往後靠在椅背上的坐姿，如圖 5-3-4 所示，雖然不是每一張靠椅背的坐姿都會被誤判，但卻是這兩種圖形擷取唯一同時產生誤判的坐姿，其因素可能是因為當身軀往後靠在椅背上的時候，會因為椅背具有斜度，而使得此時的慣性矩 I_y 與慣性積 I_{xy} 變大，高度降低，使慣性矩 I_x 的變小，如此一來就會被判斷成不正的姿勢。此外，還有一些特殊的姿勢也被誤判，如圖 5-3-5 與圖 5-3-6 所示，在圖 5-3-5 中，雖然上半身是挺直的(因此視為正姿)，但是手擺放的位置也會影響到參數的計算，故被系統判成坐姿不正；而在圖 5-3-6 中，如果只看上半身挺直的，但是此時的雙腳是翹腳的姿勢，因此也很難斷定坐姿的型態(在本系統中視為正姿，然而被只刪除小腿部分的判斷系統判定成正姿，被刪除腿部與手的系統判斷成非正姿)。

2. 在將非正姿判斷成正姿的部分：在只刪除小腿部分的情形下，略微往後仰的姿勢大多會產生誤判，如圖 5-3-7 中的姿勢，刪除所有腳與手的系統對略微往後仰的姿勢偶爾也會產生誤判；而不管是只刪除小腿部分或是腿與手都刪除的系統時，身軀略微往前傾的坐姿很容易被誤判，誤判姿勢如圖 5-3-8 所示；而只刪除小腿部分的情形下，也會誤判一些翹腳或是盤腳的姿勢，如圖 5-3-9 所示，然而刪除所有

腿部與手的判斷系統則不會誤判翹腳與盤腳的姿勢。

對於系統產生誤判的部分，也有可能是因為數據不夠多，所以使得類神經網路的判斷能力還不夠好；此外，也有可能是因為參數靈敏度稍嫌不足，在身體略微往前傾與略微往後仰的這種參數值變化不是很多的姿勢下，往往很容易產生誤判的結果。且目前的判斷系統對於靠椅背的姿勢很容易會產生誤判，因此可能要再尋找其他的參數輔助，來使得此判斷系統能夠更加準確。而如果能將上述的條件再做修改，對於準確率的提升應該會很有幫助，則此一坐姿判斷系統就能夠更趨於成熟與穩定。

此外，對於一些特殊坐姿：如翹腿但是上半身是正的，此姿勢該定義為正姿或是不正姿本身有所爭議，因此對於坐姿正或不正的定義可能還需要從醫生或是復健師等專業人士來訂定之。而對於一些用力在腳上或者是用力靠椅背等人體肌肉施力時的姿勢與不施力放鬆時的姿勢，在本研究影像法的判斷下會被視為同一種姿勢，無法進一步判斷彼此的差異，因此身體部分用力與否的坐姿亦為本研究的侷限性之一。

而本研究中目前的受試對象大多為 20 歲左右的年輕人組群，對於年紀較年長的或是年紀更輕的族群，或是純男性或純女性的族群尚未有更仔細的受試者與結果分析，也有可能會因為骨骼結構有所差異而



使姿勢會有不同的結果，老年人坐姿正的標準是有可能會異於年輕人。而本研究的目的在於對姿勢整體做出判讀，並不在於如果醫學檢驗般需要百分之百的找出各個關節的變動量，因此與醫學上在關節上貼標記的方式相較起來，有其限制性存在。且在醫學上，同一種姿勢其實是很難維持太久，因此本研究的判讀方式未來若能與時間做結合，則更能夠達到提醒使用者的作用。

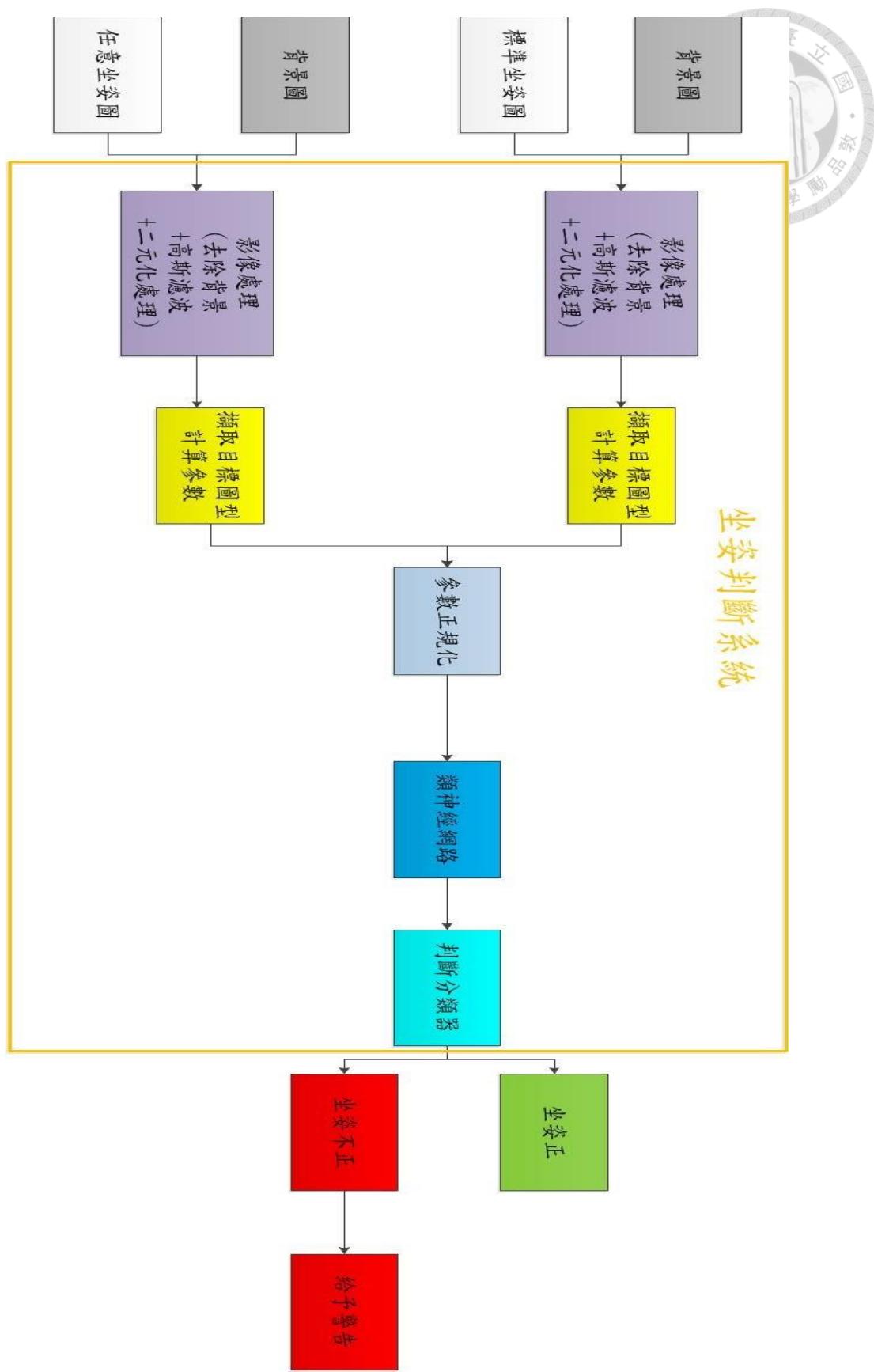
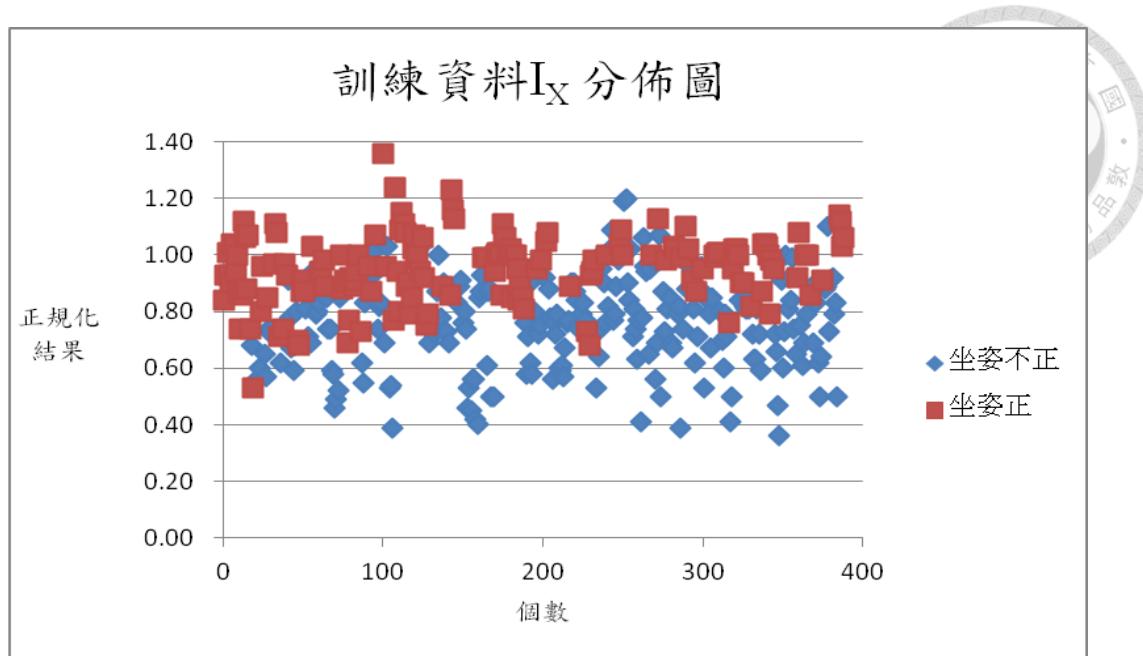
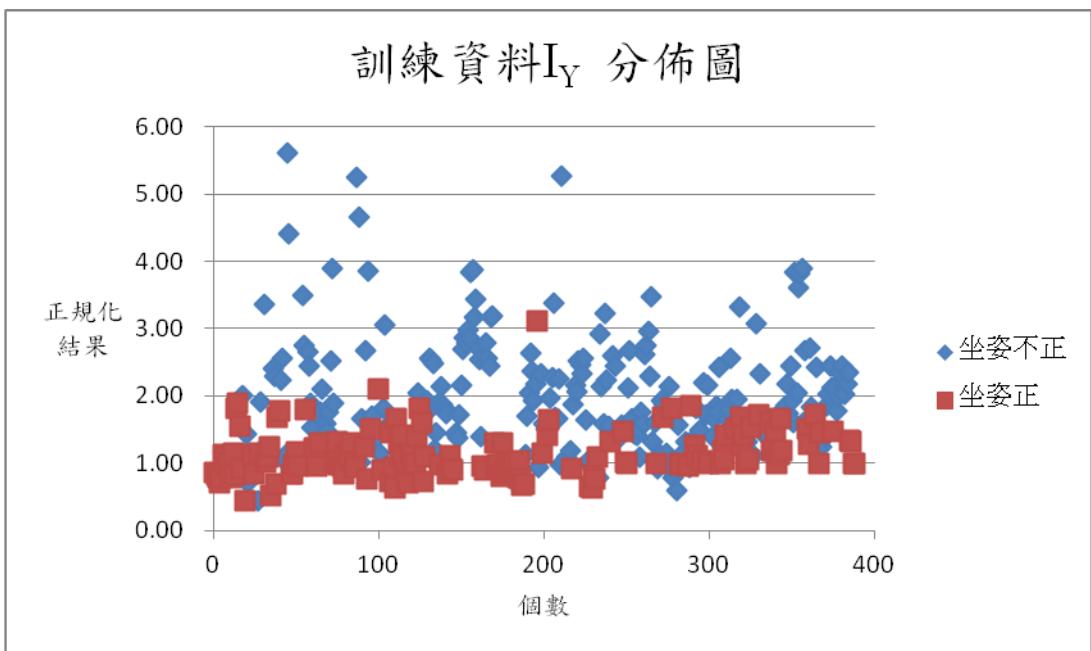


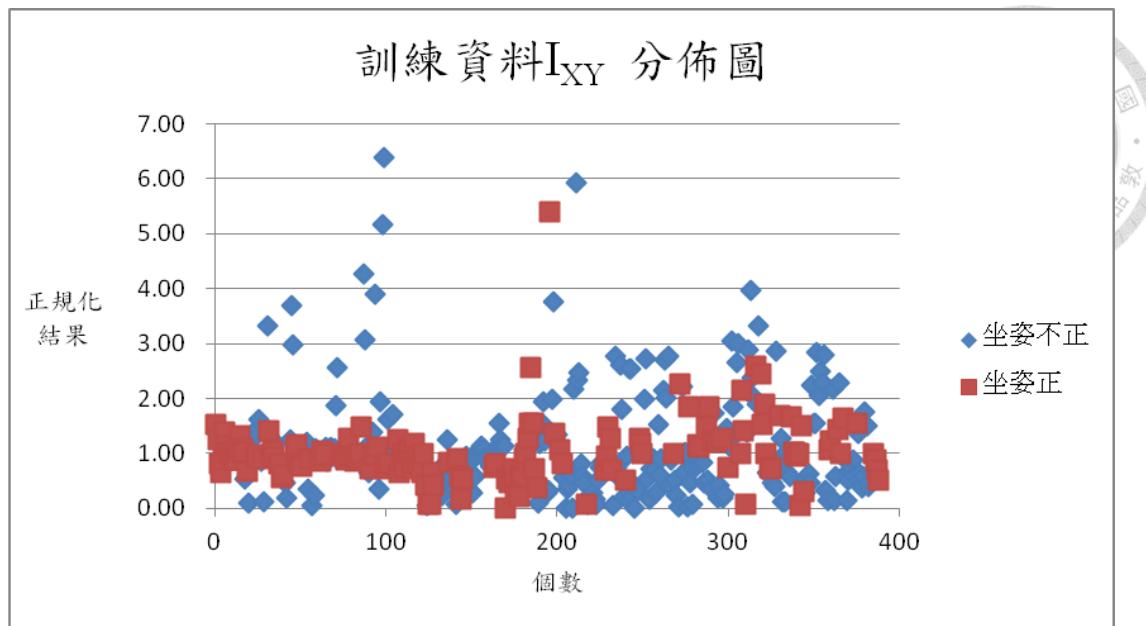
圖 5-2-1 實驗判斷流程圖



(a) 訓練資料 I_X 圖

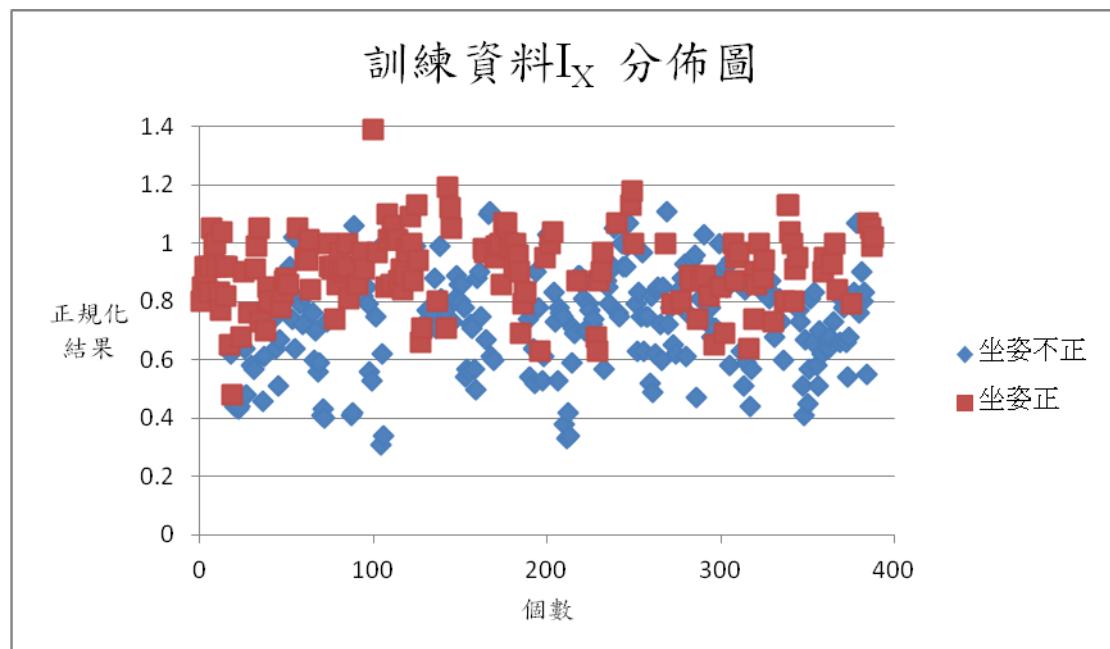


(b) 訓練資料 I_Y 圖

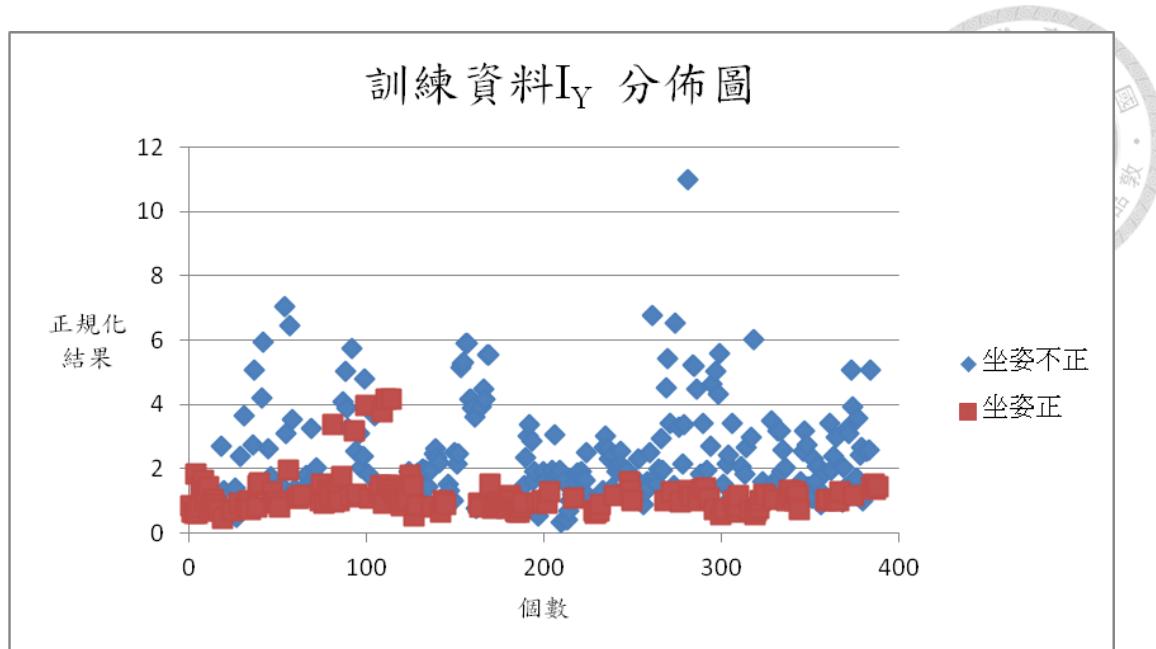


(c) 訓練資料 I_{XY} 圖

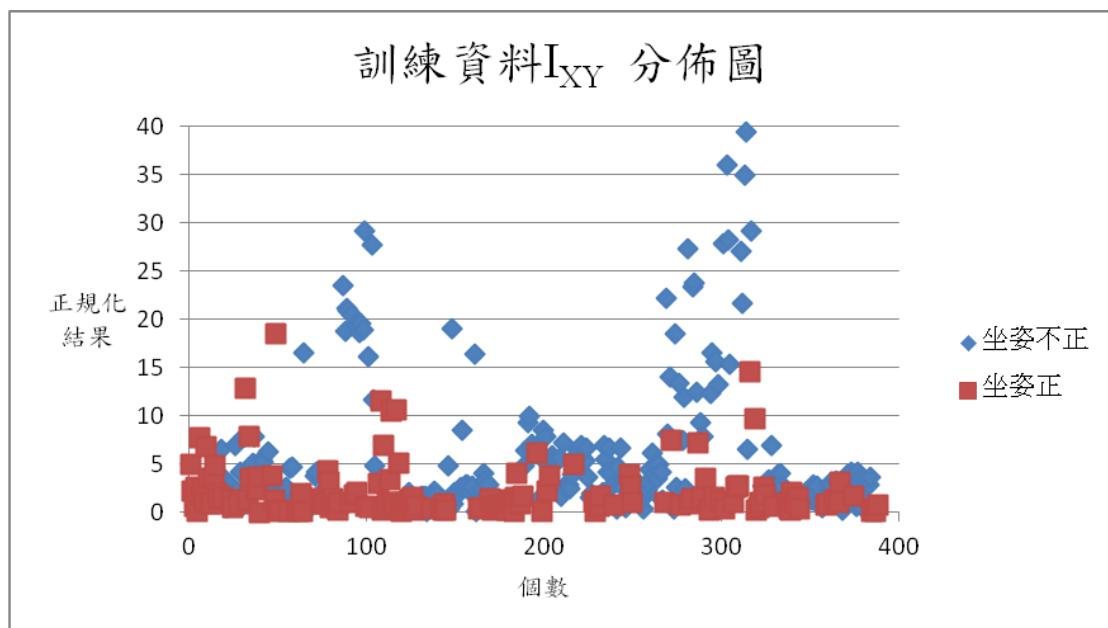
圖 5-2-2 訓練資料在刪除小腿時參數正規化圖



(a) 訓練資料 I_X 圖

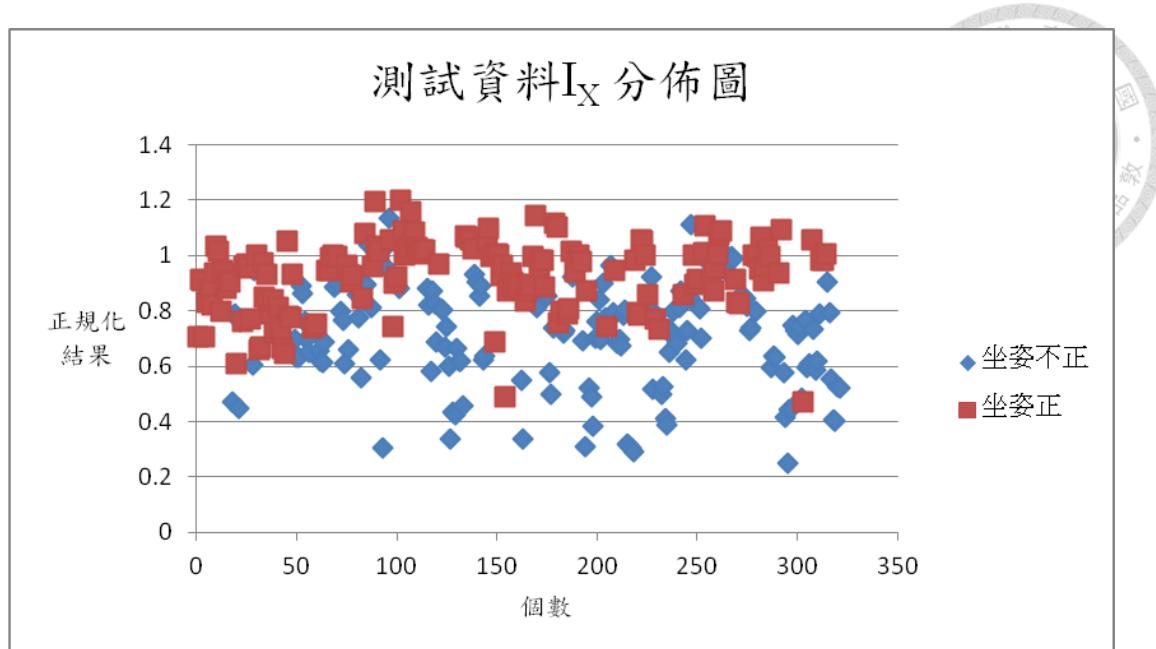


(b) 訓練資料 I_Y 圖

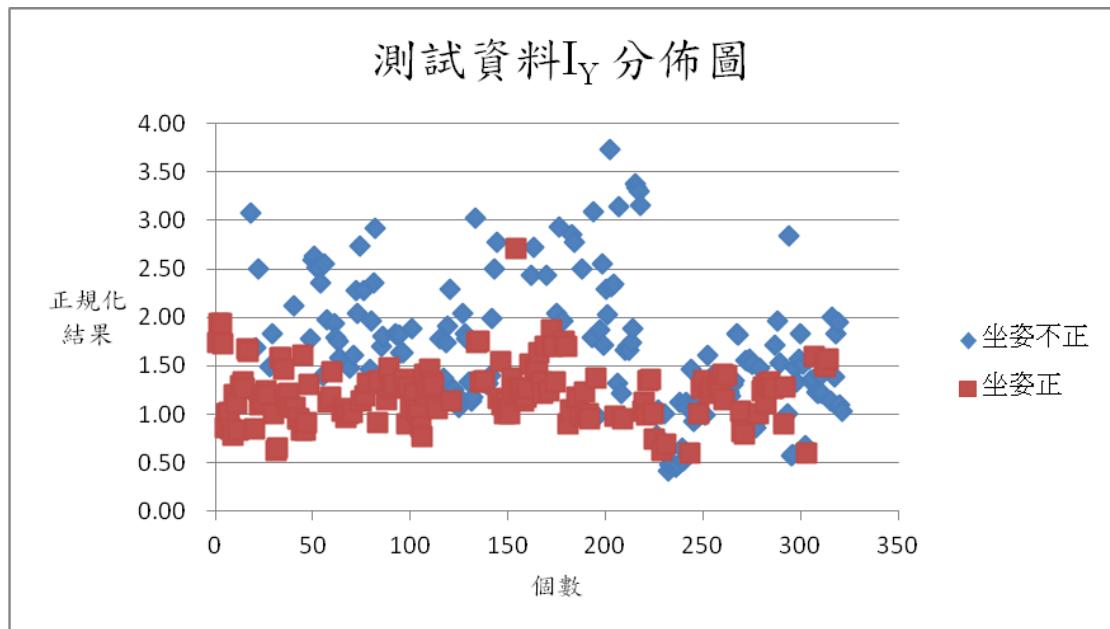


(c) 訓練資料 I_{XY} 圖

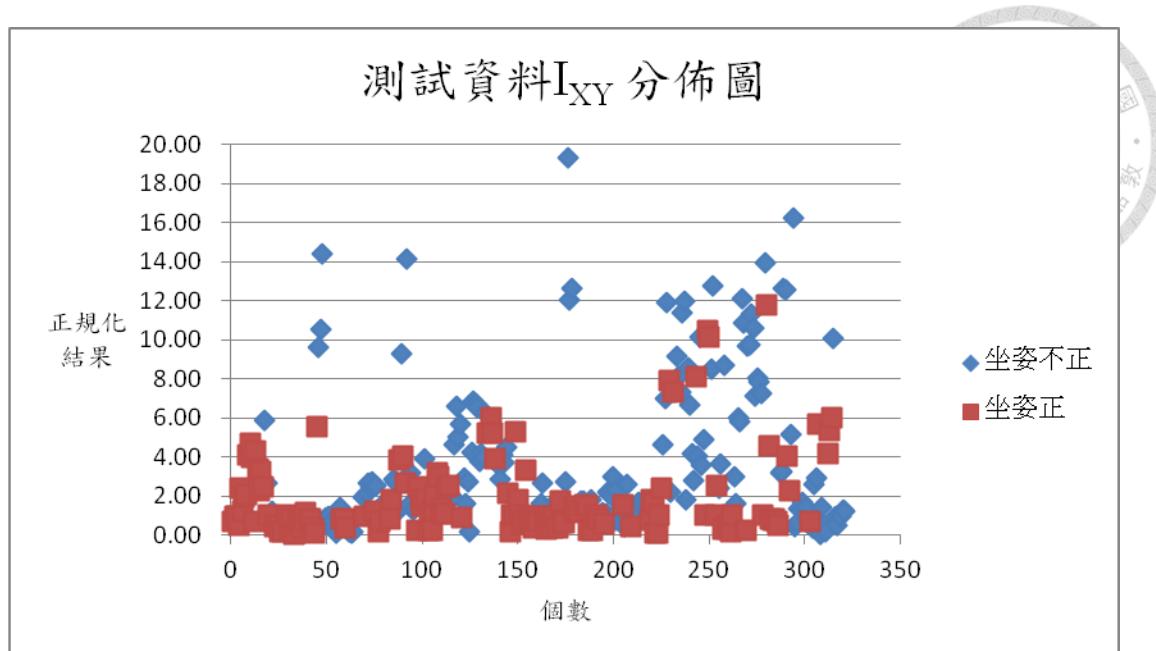
圖 5-2-3 訓練資料在刪除小腿、大腿與手時參數正規化圖



(a) 測試資料 I_X 圖

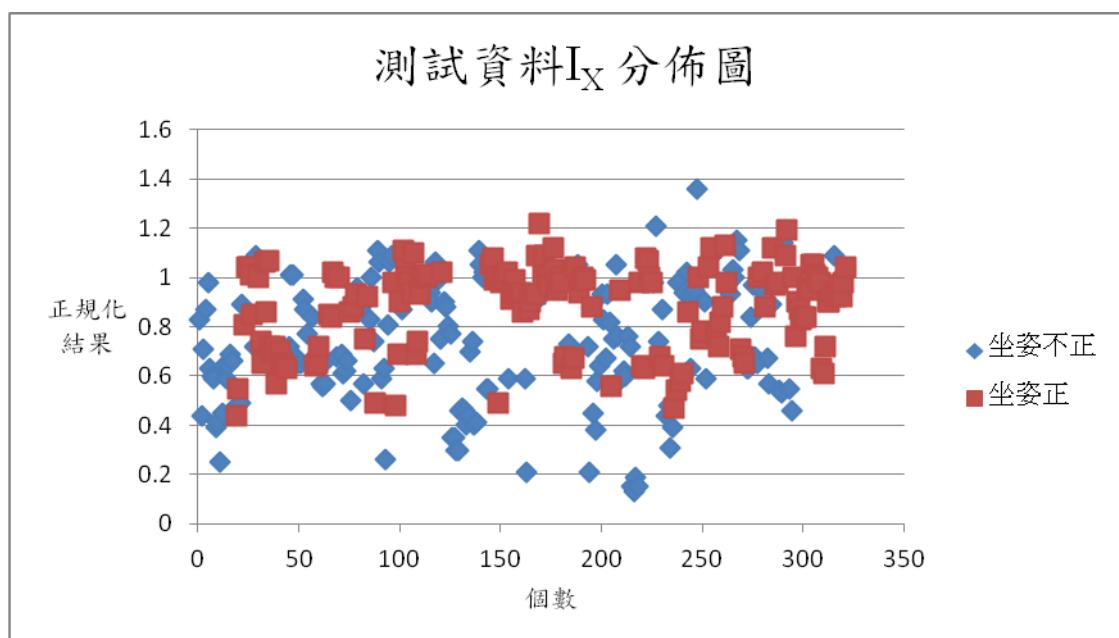


(b) 測試資料 I_Y 圖

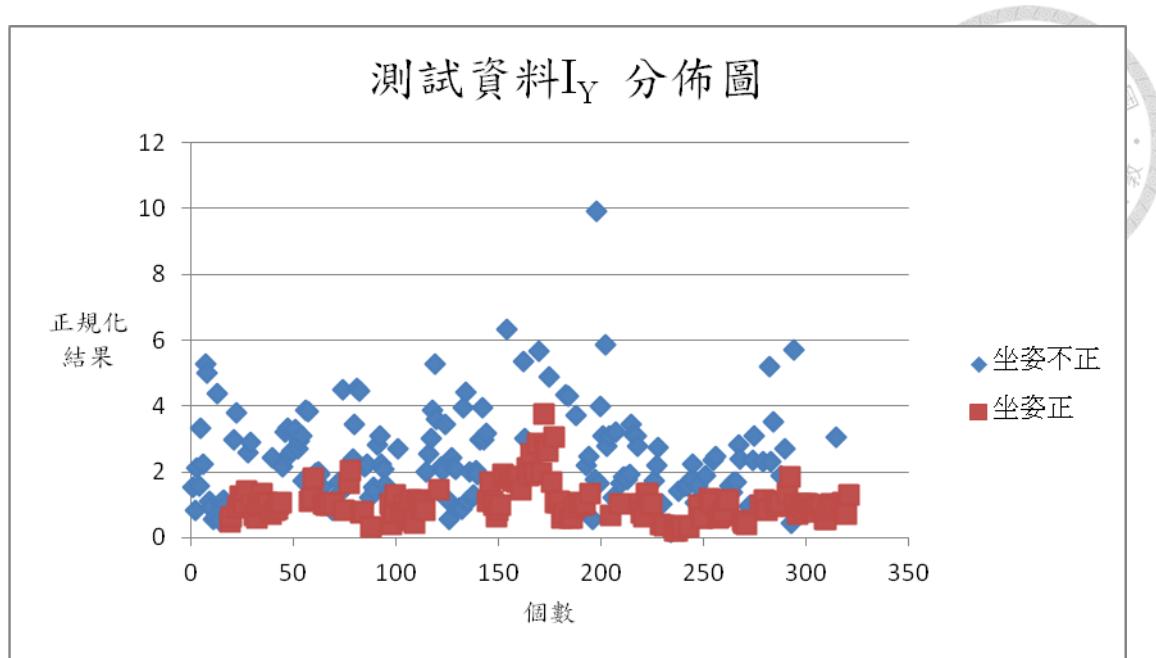


(c) 測試資料 I_{XY} 圖

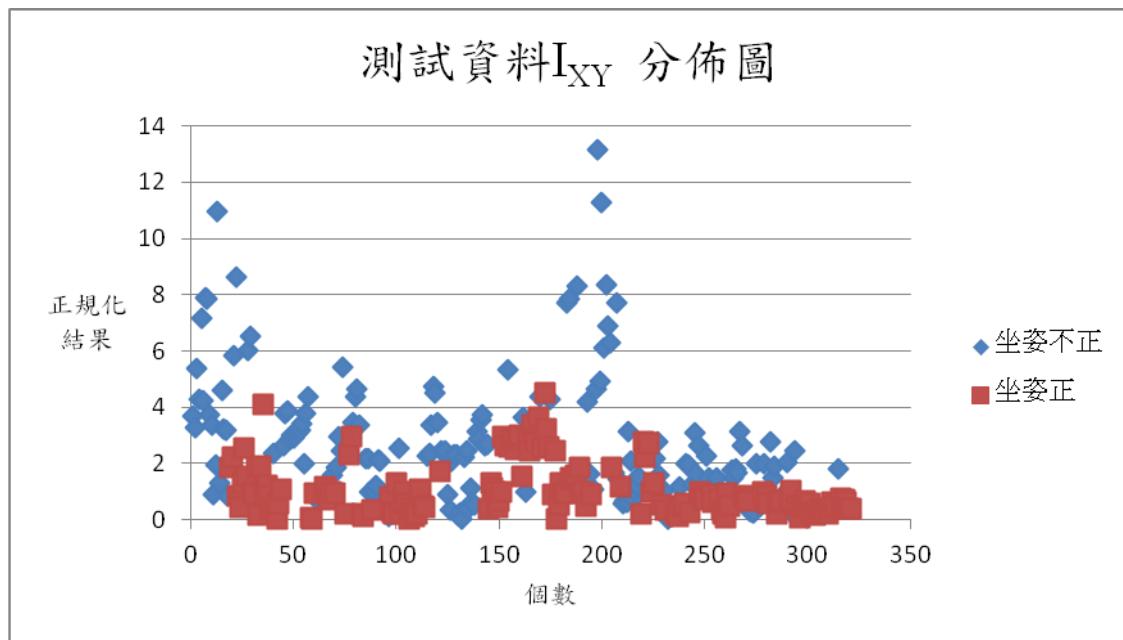
圖 5-2-4 測試資料在刪除小腿時參數正規化圖



(a) 測試資料 I_X 圖



(b) 測試資料 I_Y 圖



(c) 測試資料 I_{XY} 圖

圖 5-2-5 測試資料在刪除小腿、大腿與手時參數正規化圖

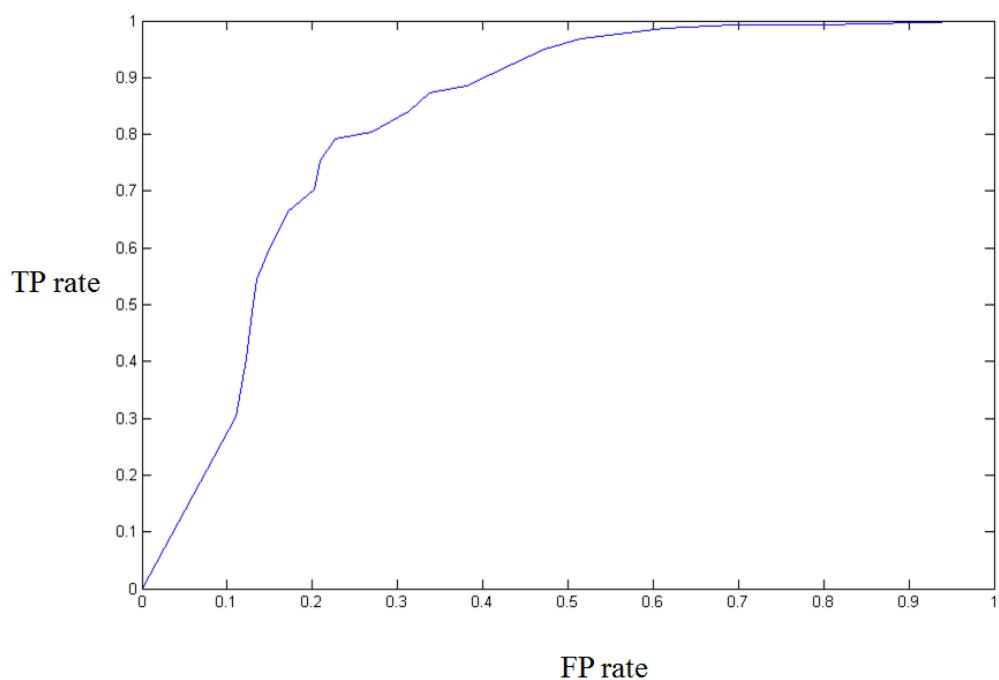


圖 5-2-6 測試資料在刪除小腿時 ROC 曲線圖

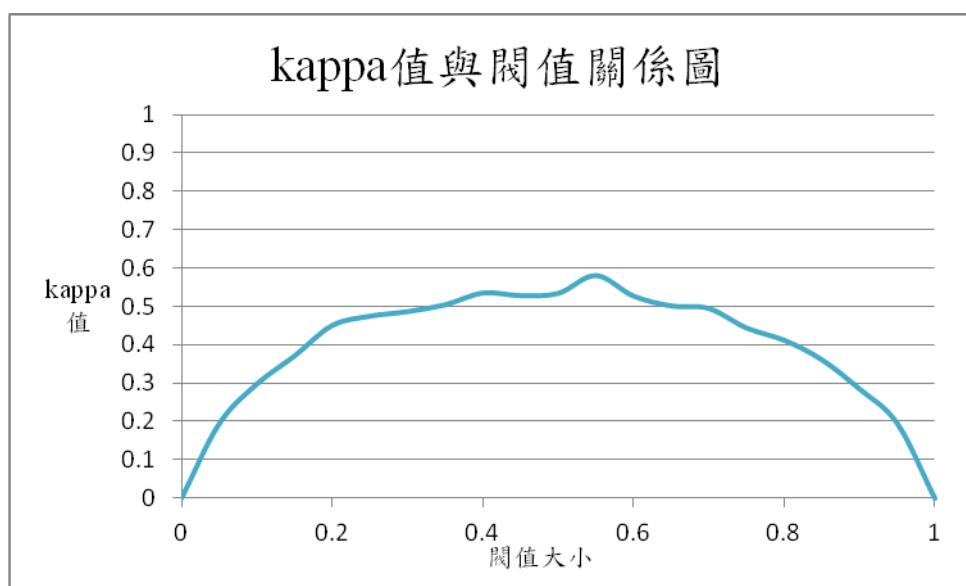


圖 5-2-7 測試資料在刪除小腿時不同閥值與 κ 值關係圖

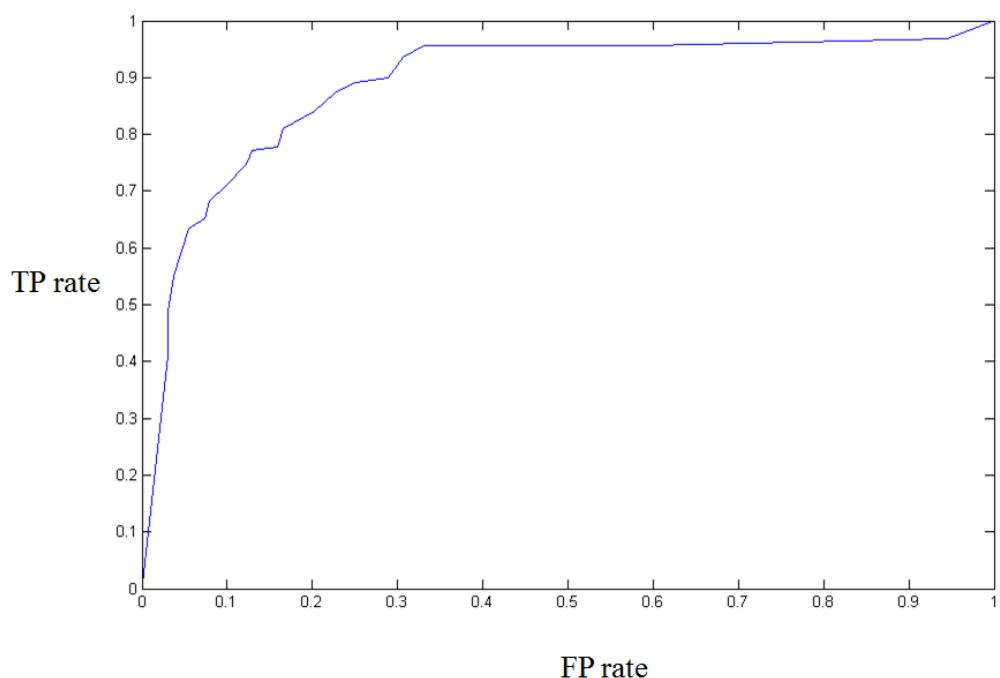


圖 5-2-8 測試資料在刪除小腿、大腿與手時 ROC 曲線圖

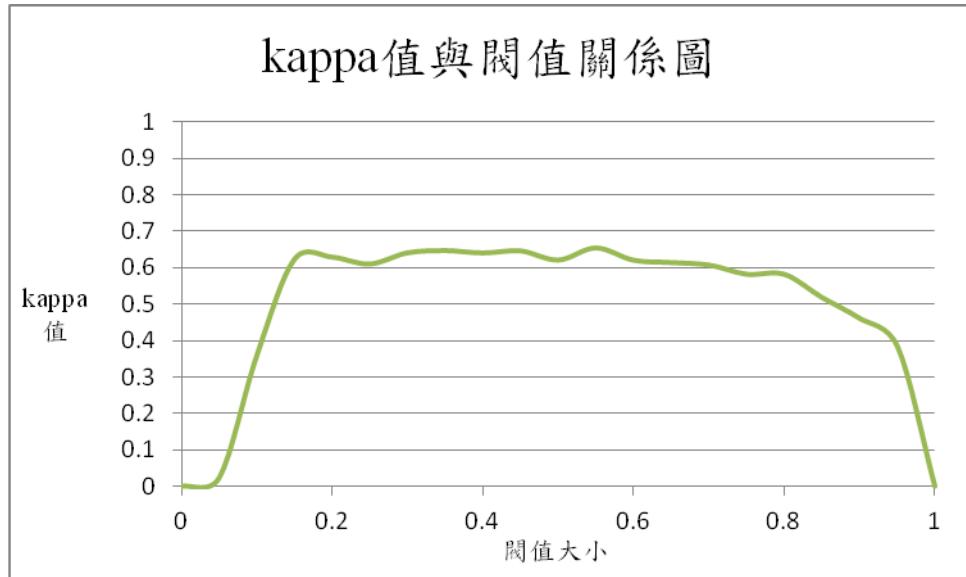


圖 5-2-9 測試資料在刪除小腿、大腿與手時不同閥值與 κ 值關係圖



圖 5-3-1 誤判姿勢類型 1



圖 5-3-2 誤判姿勢類型 2

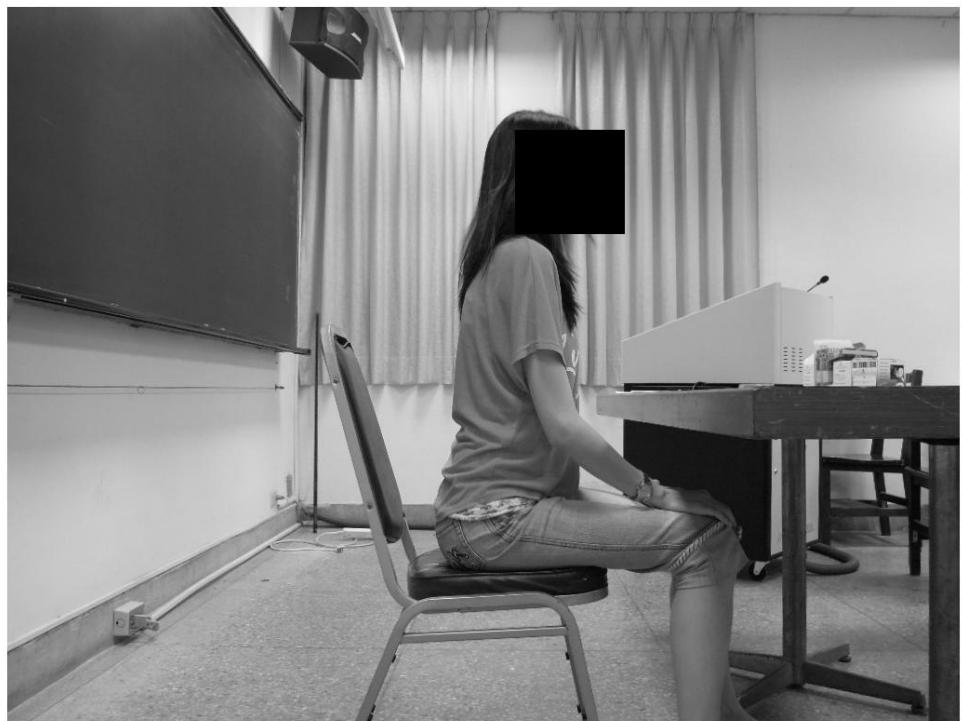


圖 5-3-3 誤判姿勢類型 3



圖 5-3-4 誤判姿勢類型 4



圖 5-3-5 誤判姿勢類型 5



圖 5-3-6 誤判姿勢類型 6



圖 5-3-7 誤判姿勢類型 7





圖 5-3-8 誤判姿勢類型 8

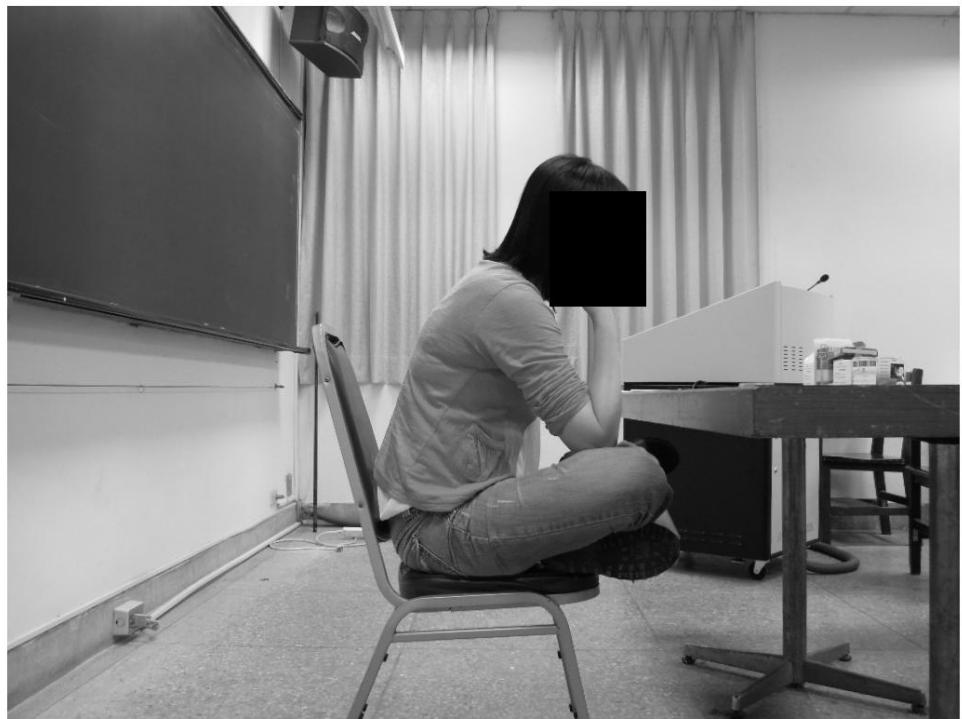




圖 5-3-9 誤判姿勢類型 9

表 5-2-1 測試資料在刪除小腿時各判斷值



		實際坐姿			
		正	不正		
判斷 坐姿	正	128	34	TP rate: 80%	FP rate: 21.25%
	不正	32	126	FN rare: 20%	TN rate: 78.75%
		160	160		

表 5-2-2 測試資料在刪除小腿、大腿與手時各判斷值

		實際坐姿			
		正	不正		
判斷 坐姿	正	125	20	TP rate: 78.13%	FP rate: 12.50%
	不正	35	140	FN rare: 21.87%	TN rate: 87.50%
		160	160		

第六章 結論與未來展望



本研究之目的在於使用人體影像進行坐姿的判讀，用以判斷當人坐在椅子上時的坐姿為正與不正，然而本研究並非醫學上嚴謹的從關節的角度對於姿勢做出判斷，因此精準度不若使用醫學儀器來得好。此判斷坐姿判斷系統內部的運算過程為影像處理、倒傳遞類神經網路與一判斷分類器，將藉由照相的方式將人體影像記錄下來，之後經過影像處理去除背景，再計算影像的慣性矩與慣性積等特徵參數，透過一標準坐姿將各參數值正規化變成類神經網路的輸入參數值，最後則是將類神經網路的輸出值做正與不正的判斷分類。然本研究目前受試者皆為年輕人族群，因此對於其他族群如老年人與兒童，其結果可能會有所差異。針對本研究所發展出的坐姿判斷系統可有以下結論：

1. 本研究的坐姿判斷系統依照對於影像擷取的部位不同，可分成兩大類：第一類為將原始坐姿圖檔只刪去小腿以下部分，第二類為刪除大腿、小腿與手的部份，第一類影像擷取的方式其判斷結果準確率為 79.38%，第二類為 82.82%。細部的辨識能力，第一類的 TP rate 為 80.00%，TN rate 為 78.82%；第二類的 TP rate 為 78.13%，TN rate 為 87.50%。



2. 過去研究中，採用壓力感測器方式進行坐姿判讀的準確率僅在 79%，相比之下，可以得知本研究的準確率較高且方便，更可即時對於坐姿做出判斷。
3. 本研究系統對於所有的姿勢判斷目前尚不能達到完全正確，有其侷限性在。是因為在本研究中所使用的慣性矩與慣性積這兩種參數不夠靈敏，在身體略微往前傾或後仰的坐姿下，其參數值與正姿參數值相比變化並不大，很容易會誤判產生。若要解決這樣的問題，需要再多增加一些輔助的參數幫助判斷。
4. 本研究的坐姿判斷並不需要穿帶任何感測器，因此對於工作效率不會有任何影響，對於身體也不會有任何負擔。

本研究之構想是希望發展出一套手機 APP 軟體，藉由自我監測的方式，適時的對於自身坐姿發出提醒，以避免將來發生其他病痛。然而本研究所發展的坐姿判斷系統目前僅為初步的階段，後續仍有许多發展與修正的空間：

1. 目前本研究只考慮在「椅子不動」且相機固定的情形下去做坐姿的分析與判定，將來如果能夠克服椅子移動或轉動等動態情形，則此坐姿判斷系統就能夠應用在所有類型與所有場合的椅子上。
2. 目前僅完成坐姿判讀內部的方法與流程，若未來能夠將此判斷方法在進一步寫成 APP 軟體，則可望發展成一套可攜式的坐姿判斷

監測系統，對於任何時候的坐姿皆能夠適時的與提醒，尤其是中學時期的學生身體正值發育期，如果能夠預防坐姿不良的情形，就能夠避免將來發生脊椎或是一些肩膀與腰部的病變。



3. 目前本研究仍在軟體的部分，將來若能夠有相關硬體部分持續發展，如發展出一張坐姿監測椅，或者是將坐姿判讀流程與微機電晶片做整合，當被判讀出坐姿不正確時，椅子能夠自動調整角度以其回復至正確姿勢。如此一來對於需要坐輪椅的復健者，便能夠省去醫療人力資源。
4. 未來如能夠將所有年齡層的坐姿收集完畢，並且以本研究的影像處理與類神經網路判讀，便可發展出適合各年齡層的坐姿監測系統。尤其老年人的身體比起其他族群來說衰退很多，進一步與看護系統做結合，更能有效達到醫療保健之功能。

參考文獻



1. A. Agresti , "An Introduction to Categorical Data Analysis" , Wiley-Interscience , America , 1996
2. Chien-Cheng Li , Yung-Yaw Chen , "Human Posture Recognition by Simple Rules" , IEEE , 2006
3. Gordon A. Vos , Jerome J. Congleton , J. Steven Moore , Alfred A. Amendola , Larry Ringer , "Postural versus chair design impacts upon interface pressure" , Elsevier , Applied Ergonomics 37 , 619-628 , 2006
4. Guodong Chen , Qun Wu , Zhongliang Yang , "Comfort Research of Pressure Distribution on Assistive Chair" , IEEE , 2009
5. Hisataka Nakane , Jun Toyama , Mineichi Kudo , "Fatigue Detection Using a Pressure Sensor Chair" , IEEE , 2011
6. Hong Z. Tan , Lynne A. Slivovsky , Alex Pentland , "A Sensing Chair Using Pressure Distribution Sensors" , IEEE , 2001
7. Hyun Jae Baek , Gih Sung Chung , Ko Keun Kim , Kwang Suk Park , "A Smart Health Monitoring Chair for Nonintrusive Measurement of Biological Signals" , IEEE , 2012
8. Jianjun Zhao , Yi Wei , Shihong Xia , Zhaoqi Wang , "Estimating

Human Body Segment Parameters Using Motion Capture Data” ,

IEEE , 2010



9. Juergen Gall , Carsten Stoll , Edilson de Aguiar , Christian Theobalt , Bodo Rosenhahn , Hans-Peter Seidel , BIWI , ”Motion Capture Using Joint Skeleton Tracking and Surface Estimation” , IEEE , 2009
10. Jun Ohya , Fumio Kishino , ”Human Posture Estimation from Multiple Images Using Genetic Algorithm” , IEEE , 1997
11. Kazutaka Kurihara , Shin'ichiro Hoshino , Katsu Yamane , Yoshihiko Nakamura , ”Optical Motion Capture System with Pan-Tilt Camera Tracking and Realtime Data Processing” , IEEE , 2002
12. Ko Keun Kim , Yong Kyu Lim , Kwang Suk Park , ”Common Mode Noise Cancellation for Electrically Non-Contact ECG Measurement System on a Chair” , IEEE , 2005
13. M. A. Adams , W. C. Hutton , ”The Effect of Posture on The Lumbar Spine” , The Journal of Bone And Joint Surgery , Vol 67-B. N0 4 , Augest , 1985
14. M. Yamada , K. Kamiya , M. Kudo , H. Nonaka , J. Toyama , ” Soft authentication and behavior analysis using a chair with sensors attached: hipprint authentication” , Pattern Anal Applic , 2009 , 12 : 251–260

15. Nan Liang , Lei Guo , Ying Wang , "An Approach on Characteristic Lines Extracting of Gray Images Using Moment of Inertia" , IEEE , 2010
16. Sam Murphy , Peter Buckle , Dave Stubbs , "The Use of The Portable Ergonomic Observation Method (PEO) to Monitor The Sitting Posture of Schoolchildren in The Classroom" , Elsevier , Applied Ergonomics 33 , 365-370 , 2002
17. Shinsuke Yoshioka , Akinori Nagano , Ryutaro Himeno , Enshi Fukashiro , "Computation of The Kinematics And The Minimum Peak Joint Moments of Sit-to-stand Movements" , BioMed Central , July , 2007
18. Yue Li , Rachid Aissaoui , "Smart Sensor, Smart Chair, Can it Predicts Your Sitting Posture" , IEEE , 2006
19. J. M. Gere 著 , 林震 , 劉興華 , 蘇金佳譯 , 材料力學 , 台北市 , 臺灣東華書局股份有限公司 , 2005
20. R. C. Gonzalez , R. E. Woods , Steven L. Eddins , 繆紹綱譯 , 數位影像處理: 運用 MATLAB , 臺北市 , 臺灣東華書局股份有限公司 , 2005
21. 王進德 , 蕭大全 , 類神經網路與模糊控制理論入門 , 台北市 , 全華圖書有限公司 , 2005

22. 李繼生, 李遠略, 慣性矩與慣性積的圖解法, *河南理工大學學報*,
第 29 卷第四期, 八月, 2010
23. 周鵬程, *類神經網路入門*, 臺北市, 全華圖書有限公司, 2006
24. 桑慧敏, *機率與推導統計原理*, 台北市, 高立圖書出版社, 2008
25. 張元翔, 運用視訊中人體特徵點追蹤與定位之代理人同部系統,
中原大學資訊工程學系碩士論文, 2007
26. 張東明, 影像處理於加工定位之應用, 國立台灣大學機械工程研
究所碩士論文, 1995
27. 張斐章, 張麗秋, *類神經網路導論: 原理與應用*, 台中, 滄海書
局, 2010
28. 張德豐, *MATLAB 神經網絡編程*, 北京, 化學工業出版社, 2011
29. 黃奕銘, 以類神經網路為最佳化途徑來完成影像強化, 國立台灣
大學電機電信電子產業研究所碩士論文, 2007
30. 黃鴻鈞, 利用類神經網路於輔助膝蓋動脈血管之 MRA 影像辨
識, 國立台灣大學電機工程研究所碩士論文, 2005
31. 羅建民, 剪力波速評估土壤液化潛能: 模糊類神經網路, 國立台
灣大學土木工程研究所碩士論文, 2003

