

國立台灣大學工學院工程科學及海洋工程學研究所

碩士論文

Department of Engineering

Institute of Engineering Science and Ocean Engineering

National Taiwan University

Master Thesis

以結合不同分類法與類神經網路為基礎在

吉他和弦即時辨識器之比較

Guitar chord real-time recognition system based on
different classifiers and Neural Network



王智弘

Chih-Hung Wang

指導教授：陳國在 博士

Advisor: Kuo-Tsai Chen, Ph.D

中華民國一百年六月

June.2011

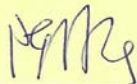
國立臺灣大學碩士學位論文
口試委員會審定書

以結合不同分類法與類神經網路為基礎在吉他和弦即時辨識器之比較

Guitar chord real-time recognition system based on
different classifiers and Neural Network

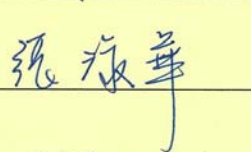
本論文係王智弘君 (R98525062) 在國立臺灣大學工程科學及海洋工程學系(所)完成之碩士學位論文，於民國 100 年 06 月 27 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

陳國在  (簽名)
(指導教授)

柯文俊 

周俊宏 

張淑華 

系主任、所長 宋家驥  (簽名)

致謝

時光飛逝，轉眼間兩年的研究生涯就要告一段落。首先能夠完成研究所的學業，最先感謝的是我的父母和姊姊，感謝你們無私的奉獻，讓我能在無後顧之憂下完成學業，在此獻上最誠摯的感謝。接著要感謝的是指導教授陳國在教授，感念老師引領我進入聲學的領域，並在這兩年來對我諄諄教誨，使我深受感激之心。

同時，要感激口試委員柯文俊教授對於本篇論文給予正面的肯定與力量，讓我心中踏實不少，以及感謝口試委員張淑華學姊和周俊宏學長對於本論文給予許多的建議和指導，讓我更明白論文敘述的完整性，在此致上萬分感激。

感謝學長阿周每個禮拜不吝嗇的指導，您的引領使我能步入研究的軌道；感謝學姊淑華的關心，與您聊天能夠沒有距離且充滿親切感；感謝學長信忻對我的研究指導，並給予我許多做人處事的態度，讓我能在碩二時更從容的面對未知的困難；感謝學長康寶，你的活潑帶給實驗室無限的回憶和歡樂，一起唱歌的日子在我心中永遠難忘；感謝一路走來的好夥伴小江，縱使你少了很多條筋，卻永遠可以看出你那執著於特定事情上的光芒；感謝學弟骨頭，因為有你的幫忙和快速的學習力，讓我安心將實驗室業務交給你；感謝學弟阿哲，因為有你的精實和對對對，讓我可以督促我自己也要更加認真向上。感謝朋友澤翰，沒有你不會有這篇論文，感謝你無私和耐心的指導，讓我從你身上看到許多必須學習的研究態度，在這邊獻上無限感謝。

最後，感謝一路走來陪伴我，關心我的每一個人，也希望，音響實驗室在未來能夠平順地走下去，甚至越來越好，在實驗室這邊的每一切，都會變成過程中的美好記憶，也希望大家都能身體健康，未來前程似錦。

摘要

音樂是人們生活中不可或缺的文化元素，隨著科技進步，樂音辨識伴隨著不少實用的價值，諸如：卡拉 ok 音準評分機、電子調音器、自動翻譜機…等都是樂音辨識的應用。

本研究著重在樂音辨識中的和弦辨識一環，本論文採用木吉他和弦，以類神經演算法為核心，並搭配其餘兩種演算法，與個人電腦結合，讓使用者可以透過使用者介面，搭配視訊麥克風，即可進行吉他和弦即時辨識。論文首先建立好吉他和弦的資料庫，接著分析三種不同辨識分類法的辨識程度，並做比較與討論。

實驗分成兩種架構進行，第一種架構利用建立好的 25000 筆吉他和弦資料庫，共 96 種不同和弦，比對三種辨識分類法的辨識程度。實驗結果發現，類神經演算法在吉他和弦資料庫的平均辨識度可達 99.26%，為三種分類器中辨識度最高者；貝氏分類法平均辨識度則可達 94.59%，Knn 分類法平均辨識度為 75.46%。

第二種架構為分別測試吉他和弦即時進行下所辨識的程度，採用四種不同的和弦進行，並使用四種不同音色的樂器(兩種真實木吉他，兩種虛擬音源)進行辨識度比較。實驗結果發現，類神經演算法在四種樂器的平均辨識度可達 80.58%，為三種分類器中辨識度最高者；貝氏分類法平均辨識度則可達 68.86%，Knn 分類法平均辨識度為 54.01%。

關鍵字：類神經網路、貝氏分類法、Knn 分類法、吉他和弦、和弦辨識

Abstract

In the culture for human life, music is necessary and elementary. Following the progress in science and technology, musical recognition, such as karaoke pitch-scoring machine, automatic tuning machine of music and electronic tuner, etc., did present a lot of practical values.

In this study, it is to concentrate on the chord recognition, which is a part of the music-recognition technology. Thus, this study tries to use acoustic guitar chords to make in-real-time recognition by users when using user interface to match with microphone through the combination of three different recognition-classification algorithms with personal computer. Accordingly, this study at first sets up the database for the guitar chord, and further analyses the recognition percentage for three different recognition-classification algorithms as used and then makes comparison and discussion.

Regarding to the experiment, two prototypes are used to perform. Between them, the first prototype uses the database of the guitar set up beforehand, which overall have ninety six items, to perform recognition percentage by using the above three different recognition -classification algorithms. The results as obtained for experiment find the averaged recognition percentages by 99.26, 94.59 and 75.46 of the guitar-chord database by using, respectively, neural network-, Naïve- Bayes classification- and Knn-algorithms are reached.

As regards to the second prototype, it, respectively, makes in real time recognition percentage of guitar chord in progress by using four different chords created, respectively, by four musical instruments of different acoustic characteristics, which include two actual acoustic guitars and two virtual sound sources, and then compare the results as obtained for recognition percentages. From the result as obtained from the experiment, it can find the associated averaged recognition percentages by 80.58, 68.86 and 54.01 to the four musical instruments by, respectively, using neural network-, Naïve-Bayes classification- and Knn-algorithms are reached.

Keyword : Neural Networks ,Naïve Bayes Classifier, K-th nearest neighbor

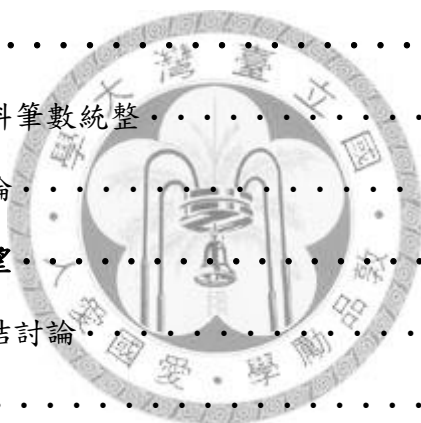
classifier,guitar chord,chord recognition

目 錄

致謝	I
摘要	II
Abstract	III
目錄	IV
圖目錄	VI
表目錄	XI

第一章 緒論	1
1.1 前言	1
1.2 文獻回顧	1
1.3 研究動機	4
1.4 論文章節概要	5
第二章 聲音訊號處理	7
2.1 聲音訊號的特性	7
2.2 基本聲音訊號特徵	9
2.3 高通濾波器	11
2.4 離散傅立葉轉換(DFT)與快速傅立葉轉換(FFT)	12
2.5 漢明窗(Hamming Window)	15
第三章 和弦特徵參數擷取	19
3.1 簡介	19
3.2 音樂與和弦	20
3.3 和弦特徵值擷取	26

第四章	分類與辨識	35
4.1	資料分類法簡介	35
4.2	K-th nearest neighbor classifier	36
4.3	Naïve Bayes Classifier	39
4.4	Neural Network	41
第五章	實驗設備與架構	57
5.1	實驗設備	57
5.2	實驗系統	59
5.3	實驗架構	59
5.4	實驗測試流程	61
第六章	實驗結果討論	65
6.1	訓練/預測 資料筆數統整	65
6.2	實驗結果與討論	67
第七章	結論與未來展望	134
7.1	辨識率數據總結討論	134
7.2	結論	136
7.3	未來展望	137
參考文獻		138



圖目錄

圖 1.1 傳統和弦偵測步驟	1
圖 1.2 Fukushima 提出的 PCP 演算法架構	3
圖 2.1 單一音框	8
圖 2.2 聲音音量	10
圖 2.3 高通濾波器	11
圖 2.4 高通濾波器之信號濾波前後比較	12
圖 2.5 漢明窗	16
圖 2.6 原始時域訊號與漢明窗訊號的時域差異	17
圖 2.7 原始頻域訊號與漢明窗訊號的頻域差異	17
圖 3.1 民謠吉他外部構造	19
圖 3.2 十二音律	20
圖 3.3 Am 和弦頻譜分析	24
圖 3.4 E 和弦頻譜分析	25
圖 3.5 和弦命名方式	25
圖 3.6 音高軸與頻率軸之刻度距離差	27
圖 3.7 weighting functions	28
圖 3.8 Am#和弦(組成音 A#+C#+F)之三種權重函數運算的 PCP	29
圖 3.9 A7#和弦(組成音 A#+D+F+G#)之三種權重函數運算的 PCP	30
圖 3.10 Harmonic Product Spectrum(HPS)	31
圖 3.11 C 和弦 PCP 擷取使用 HPS 之前後差異	32
圖 3.12 Cmaj7 和弦 PCP 擷取使用 HPS 之前後差異	33
圖 4.1 k=5 最近鄰居示意圖	36
圖 4.2 Knn 學習與辨識流程圖	38
圖 4.3 Naïve Bayes Classifier 學習與辨識流程圖	41

圖 4.4 生物神經元構造	43
圖 4.5 生物神經元網路傳遞流程	44
圖 4.6 神經元模型	45
圖 4.7 類神經網路常用的激發函數	46
圖 4.8 類神經前饋式網路架構	47
圖 4.9 類神經學習流程	48
圖 4.10 倒傳遞類神經網路架構圖	49
圖 4.11 類神經訓練網路的流程圖	56
圖 5.1 視訊麥克風	57
圖 5.2 實驗用吉他(GT1)	58
圖 5.3 實驗用吉他(GT2)	58
圖 5.4 GT1 和 GT2 的 Auto-Spectrum	58
圖 5.5 實驗架構一	60
圖 5.6 實驗架構二	61
圖 6.1 和弦類別個別資料筆數	66
圖 6.2 和弦類別個別資料筆數佔有率	66
圖 6.3 K 值對整體辨識率的影響	68
圖 6.4 樸真貝氏分類器回想/預測之辨識率比較	70
圖 6.5 回想過程中辨識率最高與最低的學習回合次數	75
圖 6.6 預測過程中辨識率最高與最低的學習回合次數	79
圖 6.7 不同神經元組合之回想\預測總辨識率之比較	80
圖 6.8 不同組合之神經元的平均學習回合次數	82
圖 6.9 最多學習回合次數與最少學習回合次數的收斂情形	83
圖 6.10 三種分類器對大三和弦和小三和弦之誤判情形比較	90
圖 6.11 三種分類器對增三和弦和減三和弦之誤判情形比較	91

圖 6.12	三種分類器對大七和弦和小七和弦之誤判情形比較	92
圖 6.13	三種分類器對屬七和弦和減七和弦之誤判情形比較	93
圖 6.14	和弦進行	94
圖 6.15	和弦進行即時辨識率討論示意圖	95
圖 6.16.1	和弦進行 I -GT1(NN)	96
圖 6.16.2	和弦進行 I -GT1(NB)	97
圖 6.16.3	和弦進行 I -GT1(KNN)	97
圖 6.17.1	和弦進行 I -GT2(NN)	98
圖 6.17.2	和弦進行 I -GT2(NB)	99
圖 6.17.3	和弦進行 I -GT2(KNN)	99
圖 6.18.1	和弦進行 I -Nylon Guitar(NN)	100
圖 6.18.2	和弦進行 I -Nylon Guitar(NB)	100
圖 6.18.3	和弦進行 I -Nylon Guitar(KNN)	101
圖 6.19.1	和弦進行 I -Classical Piano(NN)	102
圖 6.19.2	和弦進行 I -Classical Piano(NB)	102
圖 6.19.3	和弦進行 I -Classical Piano(KNN)	103
圖 6.20.1	和弦進行 II -GT1(NN)	104
圖 6.20.2	和弦進行 II -GT1(NB)	105
圖 6.20.3	和弦進行 II -GT1(KNN)	105
圖 6.21.1	和弦進行 II -GT2(NN)	106
圖 6.21.2	和弦進行 II -GT2(NB)	107
圖 6.21.3	和弦進行 II -GT2(KNN)	107
圖 6.22.1	和弦進行 II -Nylon Guitar(NN)	108
圖 6.22.2	和弦進行 II -Nylon Guitar(NB)	109
圖 6.22.3	和弦進行 II -Nylon Guitar(KNN)	109

圖 6. 23. 1 和弦進行 II -Classical Piano(NN)	110
圖 6. 23. 2 和弦進行 II -Classical Piano(NB)	111
圖 6. 23. 3 和弦進行 II -Classical Piano(KNN)	111
圖 6. 24. 1 和弦進行 III -GT1(NN)	113
圖 6. 24. 2 和弦進行 III -GT1(NB)	114
圖 6. 24. 3 和弦進行 III -GT1(KNN)	114
圖 6. 25. 1 和弦進行 III -GT2(NN)	115
圖 6. 25. 2 和弦進行 III -GT2(NB)	115
圖 6. 25. 3 和弦進行 III -GT2(KNN)	116
圖 6. 26. 1 和弦進行 III -Nylon Guitar(NN)	117
圖 6. 26. 2 和弦進行 III -Nylon Guitar(NB)	117
圖 6. 26. 3 和弦進行 III -Nylon Guitar(KNN)	118
圖 6. 27. 1 和弦進行 III -Classical Piano(NN)	119
圖 6. 27. 2 和弦進行 III -Classical Piano(NB)	119
圖 6. 27. 3 和弦進行 III -Classical Piano(KNN)	120
圖 6. 28. 1 和弦進行 IV -GT1(NN)	121
圖 6. 28. 2 和弦進行 IV -GT1(NB)	122
圖 6. 28. 3 和弦進行 IV -GT1(KNN)	122
圖 6. 29. 1 和弦進行 IV -GT2(NN)	123
圖 6. 29. 2 和弦進行 IV -GT2(NB)	123
圖 6. 29. 3 和弦進行 IV -GT2(KNN)	124
圖 6. 30. 1 和弦進行 IV -Nylon Guitar(NN)	125
圖 6. 30. 2 和弦進行 IV -Nylon Guitar(NB)	125
圖 6. 30. 3 和弦進行 IV -Nylon Guitar(KNN)	126
圖 6. 31. 1 和弦進行 IV -Classical Piano(NN)	127

圖 6.31.2 和弦進行IV-Classical Piano(NB) 127

圖 6.31.3 和弦進行IV-Classical Piano(KNN) 128

圖 6.32 使用者介面-選單 131

圖 6.33 使用者介面-選擇分類器 131

圖 6.34 使用者介面-啟動和弦辨識 132

圖 6.35 使用者介面-和弦辨識中 132



表 目 錄

表 3.1 樂音頻率與音高對照表	21
表 3.2 和弦組成音	22
表 4.1 三種分類器的演算類型和學習模式	36
表 4.2 類神經網路常用的激發函數	45
表 6.1 資料筆數與型態	65
表 6.2 K 值對整體辨識率的影響	67
表 6.3 Knn 分類器各類和弦辨識率	69
表 6.4 樸真貝氏分類器各類和弦辨識率	70
表 6.5 不同排列組合的神經元對回想過程中總辨識率的影響	73
表 6.6 回想過程中各類和弦的辨識率	73
表 6.7 不同排列組合的神經元對預測過程中總辨識率的影響	76
表 6.8 預測過程中各類和弦的辨識率	77
表 6.9 不同排列組合的神經元之神經網路對學習回合次數的影響	81
表 6.10 Knn 分類器在預測過程中的誤判情形	84
表 6.11 Naïve Bayes 分類器在回想過程中的誤判情形	85
表 6.12 Naïve Bayes 分類器在預測過程中的誤判情形	86
表 6.13 Neural Network 在回想過程中的誤判情形	87
表 6.14 Neural Network 在預測過程中的誤判情形	88
表 6.15 和弦進行 I -GT1	96
表 6.16 和弦進行 I -GT2	98
表 6.17 和弦進行 I -MIDI Nylon Guitar	100
表 6.18 和弦進行 I -MIDI Classical Piano	102
表 6.19 和弦進行 II -GT1	104
表 6.20 和弦進行 II -GT2	106

表 6.21 和弦進行 II-MIDI Nylon guitar	108
表 6.22 和弦進行 II-MIDI Classical Piano	110
表 6.23 和弦進行 III-GT1	113
表 6.24 和弦進行 III-GT2	115
表 6.25 和弦進行 III-MIDI Nylon guitar	117
表 6.26 和弦進行 III-MIDI Classical Piano	119
表 6.27 和弦進行 IV-GT1	121
表 6.28 和弦進行 IV-GT2	123
表 6.29 和弦進行 IV-MIDI Nylon Guitar	125
表 6.30 和弦進行 IV-MIDI Classical Piano	127
表 7.1 實驗數據總結	134
表 7.2 三和弦與七和弦辨識率比較	135
表 7.3 三種分類器之抗雜訊能力比較	136



第一章 緒論

1.1 前言

音樂，廣義而言可以指任何一種藝術的、讓人愉悅的或其他隨意方式排列起來的聲音，並包含了不同音階的節奏、旋律和伴奏之結合。

吉他是我們市面上最常見到的樂器之一，它屬於撥弦樂器，透過撥動琴弦，琴聲可藉由共鳴箱(木吉他)而放大。吉他的演奏方式非常廣泛，從古典吉他獨自一人詮釋經典演奏曲到電吉他扮演搖滾樂團的伴奏，都可以看見它的身影。然而，一般人對吉他的印象，最常扮演的角色就是做簡易的伴奏，只要透過簡單的幾個和弦搭配而成的和弦進行作為歌曲的伴奏，就可以邊彈邊唱。

和弦不同於單一樂音，它是由三個或三個以上的樂音所組合而成，也就是因為不同的樂音組合在一起，使得和弦在樂理上有和聲的效果，能被拿來做伴奏以配襯旋律。

最常見的吉他有六條弦，若根據吉他標準條弦法(standard tuning)，音程可由最低音 E2-82.4Hz 到最高音 F6-1396.9Hz(每把吉他能發出的最高頻率不盡相等，與吉他設計的琴格數有關，此數據僅供參考)，透過按壓指法，可以彈奏出和弦。吉他的和弦彈法很多元，在此僅討論初學者學習的刷弦，也就是右手利用吉他彈片由最低音弦(第六弦)刷到最高音弦(第一弦)，讓每條弦音一起混和而產生和弦的聲音。

1.2 文獻回顧

早期的和弦偵測(chord detection)研究，方法大致是將真實音樂訊號轉換到頻域上來識別和弦的組成音。根據和弦係由三個以上不同的樂音組成，可藉由其相關頻譜圖找出所有可能基頻出現的位置。

圖 1.1 所示為其傳統之和弦偵測步驟的演算法[1]：

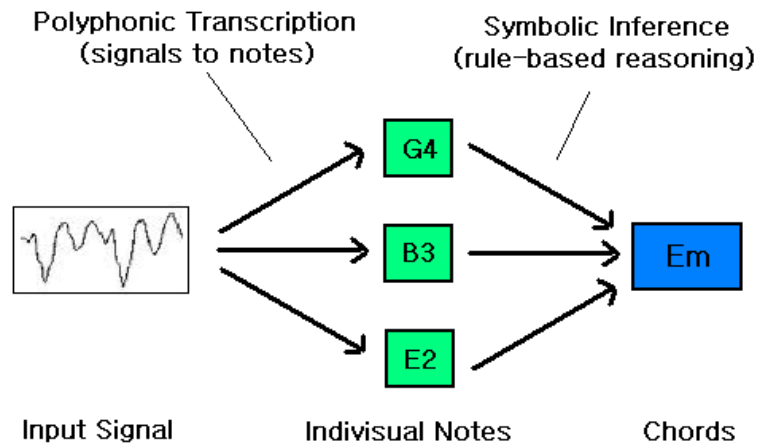


圖 1.1 傳統和弦偵測步驟

為了改善頻譜應用於樂音頻譜上之解析度問題，J.C.Brown 首先於 1990 年提出 Constant-Q(CQ) spectral transform[2]，其係利用改變擷取音框大小而不改變頻率與頻率之間的音程比例，致使頻譜的解析度能在低頻時較高，高頻時較低，成功解決了 DFT 頻率解析度僅能固定的問題，並使得 CQ spectrum 能更滿足樂音頻譜的需求。到了 2001 年，Nawab 利用 Constant-Q spectra 辨識和弦[3]，在實驗中藉著頻率產生兩個到四個音不等的訊號，並且配合和弦泛音的特性比對出和弦峰值出現的位置，成功抓出了樂音訊號的基頻。

Borching 在同年也提出新的和弦辨識技術[4]，冀望達到接近人腦與耳蝸的聽覺反應。他利用 wavelet transform(小波轉換)，且使用自製 SOM(self organized map)透過類神經學習來分類識別和弦，以 48 種不同和弦共計 480 個聲音樣本進行訓練，實驗結果以貝多芬第五號交響曲當作實驗樣本(貝多芬第五號交響曲乃由和弦 C，F，G 三個大三和弦所組成的和弦進行)，辨識率可達百分之百，並有良好的抗噪能力。

Y.Fukayama 也提出使用波包(Wavelet Packet)和延伸性 Kalman 濾波器(Extended Kalman Filter)來偵測單音和弦的演算法[5]，他使用和弦的完五度樂音

進行測試，並測試三種大三和弦(A, A#, B)和三種小三和弦(Am, Am#, Bm)實驗結果顯示，即使存在噪音的干擾下，和弦和單音的偵測也能有不錯的表現。

由於傳統和弦辨識乃偵測和弦各別組成音基頻，因為和弦由多音所組成，故存在多且複雜之樂音諧波，容易在擷取組成音的基頻時，因重疊的樂音諧波而造成誤判。所以後來的和弦辨識乃貼近音樂十二平均律的概念，將真實音樂訊號轉換到頻域後，對應到 12 個樂音做分類，並觀察其每個樂音的強度(Chroma Vector)。

PCP(Pitch Class Profile)理論在 1999 年由 Fujishima 提出[6]，其概念是將音樂訊號經由 DFT 的轉換後，擷取十二個樂音之能量大小，同時建構出二進位和弦樣板 (Chord Type Templates, CTT) 做為比對資料的標準，透過最近鄰居分類法(Nearest Neighbor Method)來配對 PCP 與 CTT，進行和弦分類。相關之演算法架構如下圖 1.2 所示：

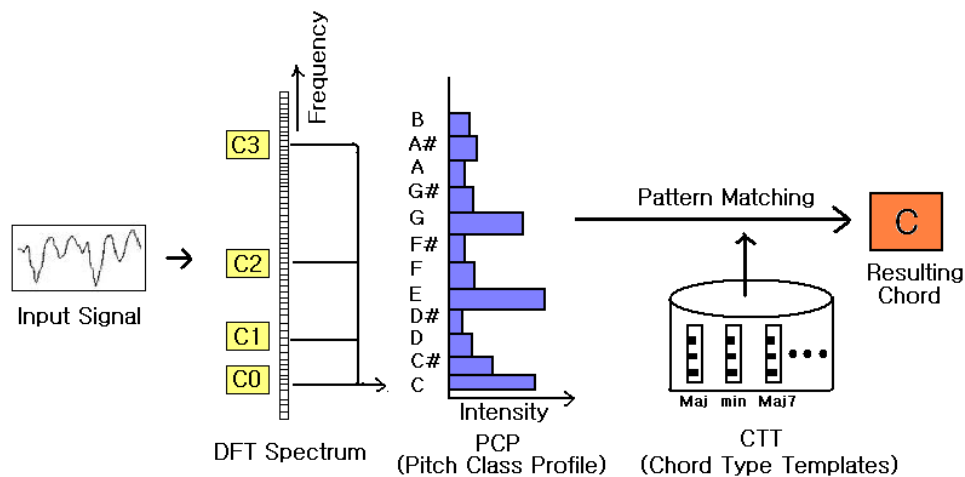


圖 1.2 Fukishima 提出的 PCP 演算法架構

PCP 的提出使得和弦辨識有了新的方向，在 2003 年，Shed 和 Ellis 以 PCP 的理論作為基礎[7]，提出一套和弦分類與辨識的統計學習法，並改良出 Weighted

Averaging of Rotated PCP Vectors 作為和弦特徵值，利用 Expectation Maximization 演算法(EM)計算出平均值，變異量的向量來訓練隱藏馬可夫模型(HMMs)，並以披頭四的二十首歌做為測試。另外，Gomez 認為泛音特性有助於分析和弦，在 2004 年也以 PCP 為基礎發展出 Harmonic Pitch Class Profile(HPCP)[8, 9]，並在試驗 525 首古典音樂作品中成功辨識出百分之六十的調性。

1.3 研究動機

研究動機，係源自於吉他初學者在學習彈奏吉他和弦時，往往只知道指法如何按，卻不曉得自己按的和弦其種類和名稱為何，如此，容易造成初學者在學習吉他和弦時，常常只是死背指法位置，而容易遇到樂理上的瓶頸。

為了方便各種樂器能調整音準，市售的電子調音器能幫助樂器彈奏者，準確地將每條弦調到標準的頻率。因此，實驗的動機是希望能建構出一台吉他和弦辨識器，能夠在彈奏者刷下吉他和弦的時候，辨識出該和弦的名稱與其組成音，幫助彈奏者學習樂理與記憶和弦。

樂器演奏家在長期彈奏下，耳朵對樂音的反應會變得比平常人敏銳，專業演奏家甚至不需依賴調音器就能靠著耳朵訓練出來的音感來判斷樂音的音準。耳朵就像是一台天然的頻譜分析儀，聲音訊號經耳朵內聽覺神經分析之後傳達給大腦做反覆的訓練和學習，久而久之就能培養出優越的絕對音感。據此角度出發，希望透過機器學習來取代大腦，並驗證其相似的效果。因此以類神經的生物學習精神為主體，並搭配分析其餘兩種分類器來做比較和測試。

目前市售的電子調音器價格昂貴，且僅能夠用來判定單一頻率的準確度。本研究僅需透過個人電腦以及普通的收音用視訊麥克風搭配程式，即可在自製的使用者介面上進行吉他和弦辨識。

1.4 論文章節概要

本論文共分成七個章節，即

第一章：緒論

第二章：聲音訊號處理

第三章：和弦特徵參數擷取

第四章：分類與辨識

第五章：實驗設備與架構

第六章：實驗結果與討論

第七章：結論與未來展望

參考文獻





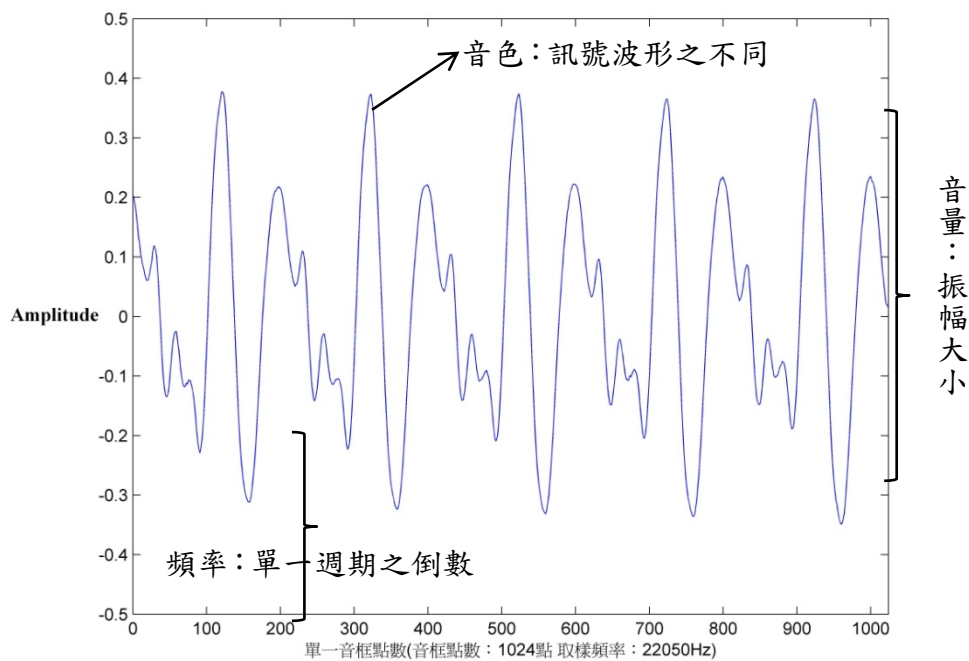
第二章 聲音訊號處理

2.1 聲音訊號的特性

在我們日常生活中，聲音無所不在，豐沛充盈了我們的聽覺感受。所謂的聲音訊號(Audio Signals)，簡稱「音訊」，泛指由人耳所聽到各種聲音之信息。一般來說，發音體(聲源)會產生振動，此震動會對空氣粒子產生壓縮與伸張而形成聲波，聲波會以一定的速度往外散播出去，就是我們所聽到的聲音。

聲音若以訊號的規律性來分類，可以分為準週期音與非準週期音。準週期音的波形具有規律性，可以看出週期的重複性，人耳亦可以判斷出其存在高穩定音，例如單音弦樂器。非準週期音的波形則不具規律性，所以無法看出明顯的週期變化，相對的，人耳也無法感覺出其具高穩定音。例如打雷聲、敲鑼打鼓聲…等。

當我們在分析聲音的時候，由於音訊在短時間內是相對的穩定，可視短時間內的波形為有規律性週期變化，所以我們會先將所需要分析的聲音訊號切成音框(Frame)來做短時距(Short-term)的分析，藉此觀察聲音訊號的變化與特徵。單一音框如圖 2.1 所示：



我們可以從圖 2.1 單一音框裡面，觀察到三個主要聲音訊號的特徵：

1. **音量(Volume)**：代表聲音的強度，也就是訊號的振幅。又稱為能量 (Energy)，或是強度(Intensity)。
2. **音高(Pitch)**：代表聲音的高低，在物理學上稱為頻率。可由基本頻率 (Fundamental Frequency)來類比。也是基本週期 (Fundamental Period)的倒數。
3. **音色(Timbre)**：代表聲音的內容，可由每一個波形在一個基本週期的變化來類比。

2.2 基本聲音訊號特徵

2.2.1 音量(Volume)

音量代表了聲音的強度，也稱為聲音的能量。在音訊處理上，可以計算單一音框內訊號振幅的大小來得知音量。

根據客觀音量和人耳主觀音量的感受不同，計算方式可以分為下列兩種[10]：

1. 求取音訊振幅能量的總和：

$$volume = \sum_{i=1}^n |S_i| \quad (2.1)$$

於式 2.1 中，符號表示如下：

S_i ：表示單一音框中的第 i 個取樣點

n ：表示單一音框的點數

此種計算方法透過整數加法的運算，程序上簡單且快速，適用於低階平台處理，如微處理機(MCU)。計算結果如圖 2.2(b)所示

2. 求取分貝值：

$$volume = 10 \times \log \left(\sum_{i=1}^n S_i^2 \right) \quad (2.2)$$

於式 2.2 中，符號表示如下：

S_i ：表示單一音框中的第 i 個取樣點

n ：表示單一音框的點數

此種方法所得到的數值以分貝(decibels)為單位，是一個相對強度的值，優點是比

較符合人耳對於大小聲音的感覺。計算結果如圖 2.2(c)所示：

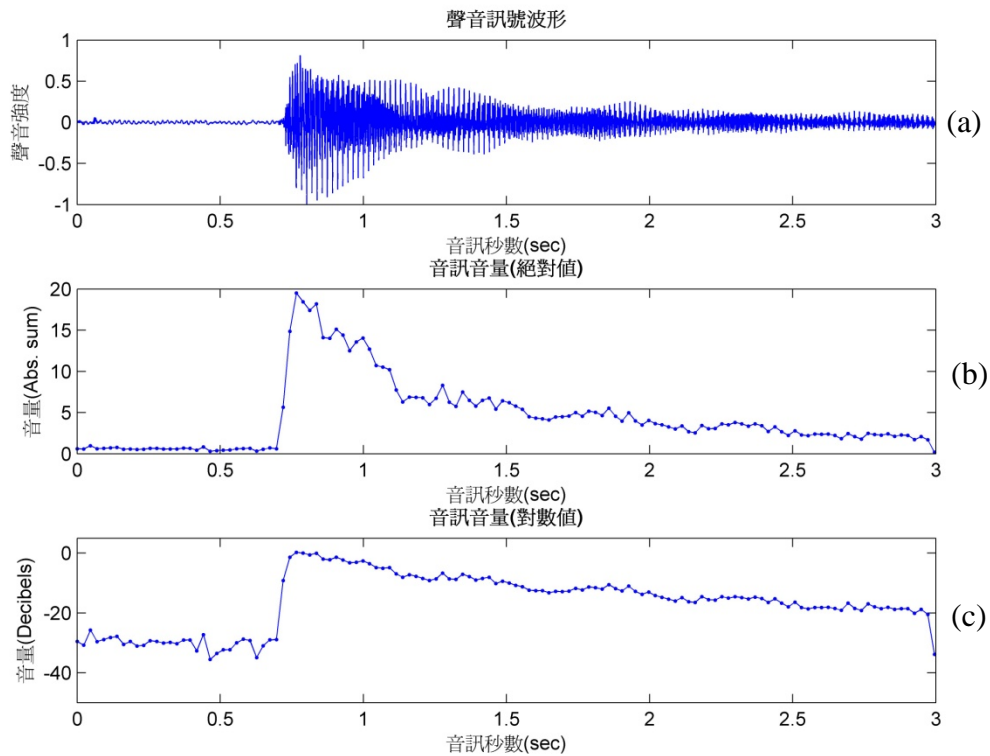


圖 2.2 聲音音量

2.2.2 音高(Pitch)

音高代表這頻率的高低，所謂的頻率，指的是基本頻率(Fundamental Frequency)，亦即基本週期(Fundamental Period)的倒數。透過觀察圖 2.1 單一音框可以發現，只要聲音穩定，就可以看出音訊波形之基本週期的存在。由單一週期的倒數則可以得到此音訊的基本頻率。

2.2.3 音色(Timbre)

音色泛指音訊的內容，直覺來看，音色的不同，代表基本週期的波形不同，因此我們可以使用基本週期的波形來代表音色，可以透過將每一個音框進行頻譜分析，並得到此時域音框的頻譜圖，來知道每個頻率的訊號強度。

2.3 高通濾波器

濾波器目的在於濾除不需要的頻率，依濾波的範圍，濾波器的種類可分為低通，高通和帶通濾波器。

本論文使用高通濾波器，是一種容許高頻信號通過，但大幅減弱頻率低於截止頻率信號通過的濾波器。利用高通濾波器將低於吉他最低音域的雜訊頻率濾除，以得到更佳的樂音訊號品質。

圖 2.3 和圖 2.4 分別為高通濾波器和訊號經高通濾波器之前後的比較：

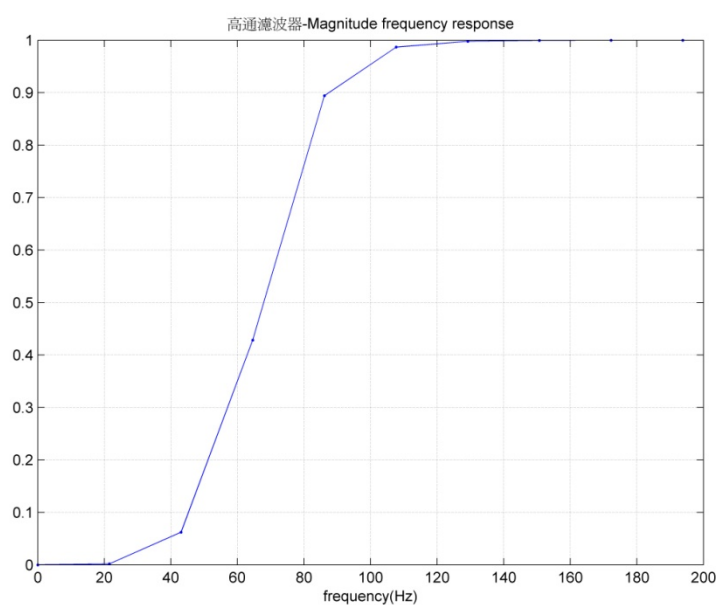


圖 2.3 高通濾波器

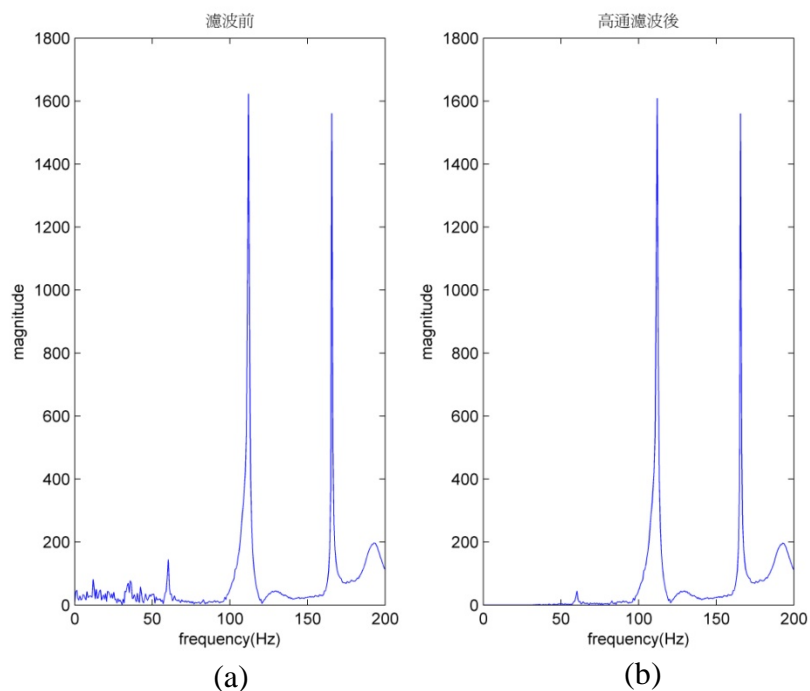


圖 2.4 高通濾波器之信號濾波前後比較

由圖 2.3 高通濾波器之圖示可發現，其截止頻率約是 90Hz；圖 2.4(a)為尚未通過濾波器前的吉他弦音之訊號，由圖 2.4(b)觀察訊號通過濾波後的差別，可發現高通濾波器濾掉了小於截止頻率的大部分訊號。

2.4 離散傅立葉轉換(DFT)與快速傅立葉轉換(FFT)

在訊號處理中，透過傅立葉轉換，可以將時域訊號轉變為具有頻率分量和振幅分量的頻譜圖。但最初的傅立葉轉換僅只能使用於連續時間的訊號上，無法用於離散時域者。為了在科學計算和數位訊號處理等領域中，使用計算機進行傅立葉轉換，必須先將函數 X_n 定義在離散點而非連續域內，始可於電腦中進行處理，且必須滿足有限性或週期性的條件。我們稱為離散傅立葉轉換(Discrete Fourier Transform)。

離散傅立葉轉換是傅立葉轉換在時域和頻域上都成離散的形式，對於聲音訊號 N 點序列 $\{x[n]\}_{0 \leq n \leq N}$ 其離散傅立葉轉換(DFT)為

$$\hat{X}[k] = \sum_{n=0}^{N-1} e^{-i\frac{2\pi nk}{N}} X[n] \quad k = 0, 1 \dots N-1 \quad (2.3)$$

於式 2.3 中，符號表示如下：

e ：自然對數的底數

i ：虛數單位

N 點 DFT 的頻譜解析度是 $2\pi/N$ ，透過補零可以觀察到更多的頻點。在滿足取樣定理的條件下，提高頻譜解析度可以透過增加時域採樣長度來實現。

快速傅立葉轉換(Fast Fourier Transform, FFT)，是離散傅立葉轉換的快速算法，由於速度很快，適合應用於計算大整數乘法，求解偏微分方程等。按照 DFT 的定義，計算一個長為 N 之序列的 DFT 所需要的計算複雜度，是階次 $O(n^2)$ ，至於快速傅立葉轉換的好處，在於可以計算出與直接計算相同的結果，但只需要階次 $O(n \log n)$ 的計算複雜度。由於 DFT 的逆轉換可以由 DFT 表示，所以 DFT 逆轉換的計算同樣可以由 FFT 完成。

Cooley-Tukey 演算法是最常見的 FFT 算法[11]，其使用的概念是計算機科學中的分治法，分治法簡單來說就是將一個複雜的問題分成兩個或更多的相同或類似的子問題，再把子問題分成更小的子問題直到最後子問題可以簡單的直接求解，而原問題的解即是所有子問題求得解的合成。透過分治法的概念，將訊號長度為 $N = N_1 N_2$ 的 DFT 分解成長度分別為 N_1 和 N_2 的兩個較短序列之 DFT，以及與

$O(N)$ 個旋轉因子的複數乘法。

Cooley-Tukey 演算法最著名的應用，是將長度 N 之訊號序列的 DFT 分割為兩個長為 $N/2$ 的子序列之 DFT，因此其應用僅適用於序列長度為 2^n 的 DFT 計算。實際上，Cooley-Tukey 演算法也可以用於序列長度 N 為任意因數分解形式的 DFT。儘管 Cooley-Tukey 演算法的基本想法是採用遞歸的方法進行計算，大多數傳統的算法實現都將顯示的遞歸算法改寫為非遞歸的形式。另外，因為 Cooley-Tukey 算法是將 DFT 分解為較小長度的多個 DFT，因此也可以同任一種其他的 DFT 演算法聯合使用。

我們用 N 次單位根 W_N 來表示 $e^{-j\frac{2\pi}{N}}$

其 W_N 具有三種性質，分別為：

1. 週期性：具有週期 N

2. 對稱性： $W_N^{k+\frac{N}{2}} = -W_N^k$

3. $W_N^{ikn} = W_N^{\frac{kn}{i}}$



為了簡單起見，我們下面設待轉換序列長度 $n = 2^r$ 。根據上面單位根的對稱性，求級數 $y_k = \sum_{n=0}^{N-1} W_N^{kn} X_n$ 時，可以將求和區間分為兩部分：

$$\begin{aligned} y_k &= \sum_{n=2i} W_N^{kn} X_n + \sum_{n=2i+1} W_N^{kn} X_n = \sum_i W_N^{\frac{ki}{2}} X_{2i} + W_N^k \sum_{n=i} W_N^{\frac{ki}{2}} X_{2i+1} \\ &= F_{even}(k) + W_N^k F_{odd}(k) \end{aligned} \tag{2.4}$$

於式 2.4 中， $F_{even}(k)$ 和 $F_{odd}(k)$ 是兩個分別關於序列 $\{X_n\}_0^{N-1}$ 奇數號和偶數號序列 $N/2$ 點轉換。由此式只能計算出 y_k 的前 $N/2$ 個點；至於後面 $N/2$ 個點，注意

$F_{even}(k)$ 和 $F_{odd}(k)$ 都是週期為 $N/2$ 的函數，由單位根的對稱性，於是以下轉換公式：

$$y_{k+\frac{N}{2}} = F_{even}(k) - W_N^k F_{odd}(k) \quad (2.5)$$

$$y_k = F_{even}(k) + W_N^k F_{odd}(k) \quad (2.6)$$

這樣，一個 N 點轉換就分解成了兩個 $N/2$ 點轉換。照這樣可繼續分解下去。這就是 Cooley-Tukey 算法的基本原理。根據主定理不難分析出此時算法的時間複雜度為 $O(n \log n)$

2.5 漢明窗(Hamming Window)

對於擷取一段音訊的離散時間訊號 $X(n)$ ，若單一音框的訊號兩端不連續，在聽覺上會產生額外的響聲，從頻域上看，聲音訊號的頻譜會被破壞。為了增加音框裡訊號左右兩端的連續性以降低頻譜的失真，我們常將分析的音框加上漢明窗來作分析。這樣的處理方式稱為加視窗(windowing)[12, 13]。漢明窗的特性為兩邊緩慢減小，在邊界上沒有明顯的不連續。

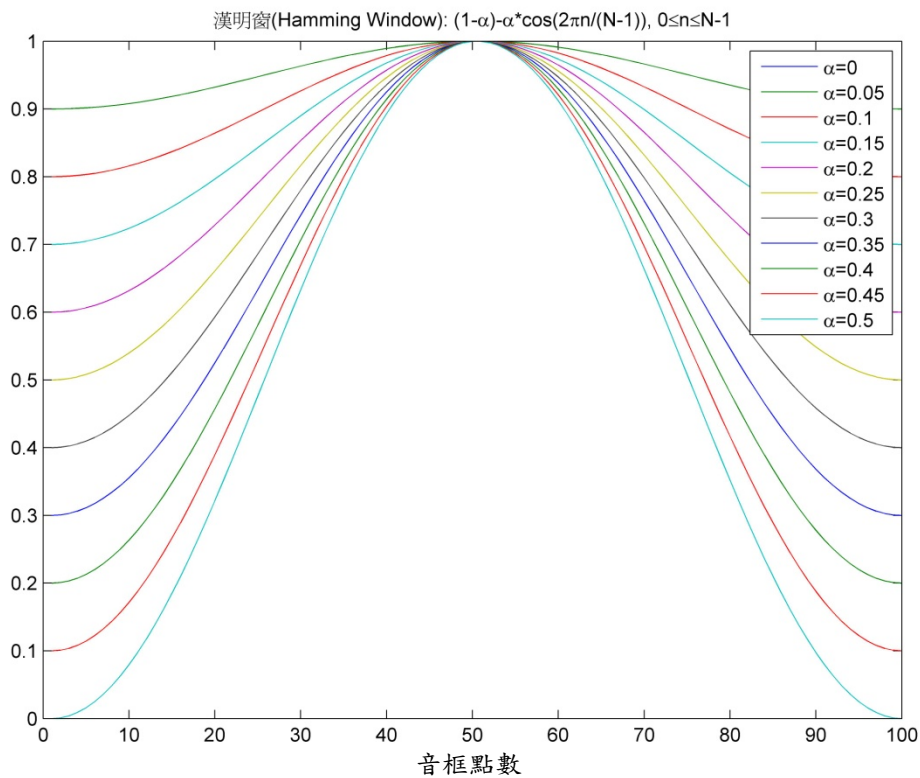
假設單一音框的訊號為 $X(n)$, $n = 0, \dots, N - 1$ ，則加上漢明窗之後的訊號為

$$X_{window} = X(n) \times W(n) \quad (2.7)$$

由式 2.7 中， $W(n)$ 的形式如下：

$$W(n, \alpha) = (1 - \alpha) - \alpha \cos(2\pi n / (N - 1)) \quad 0 \leq n \leq N - 1 \quad (2.8)$$

由式 2.8 中，不同的 α 值會產生不同的漢明窗，如圖 2.5 所示：



由圖 2.5 可以觀察不同 α 值對漢明窗兩旁大小的影響， α 值越高則窗形的兩旁坡度會越陡，一般使用的漢明窗為 $\alpha=0.46$ 。

漢明窗藉著降低擷取訊號兩邊的不連續，可有效降低頻譜的溢散。比較圖 2.6(a)之原始吉他訊號與圖 2.7 加漢明窗後的時域圖和頻域圖，可發現圖 2.6(b) 加漢明窗後的時域訊號兩端趨近平緩且連續；由圖 2.7(b)可發現，圖 2.6(b)所對應到的頻域能明顯突出基頻與諧波，並降低了頻譜的溢散。

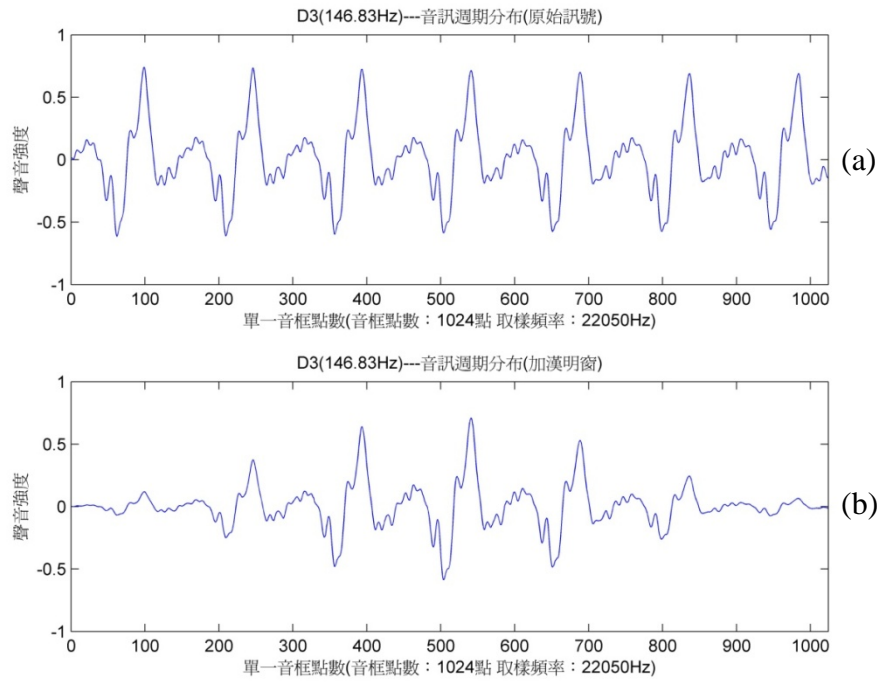


圖 2.6 原始時域訊號與漢明窗訊號的時域差異

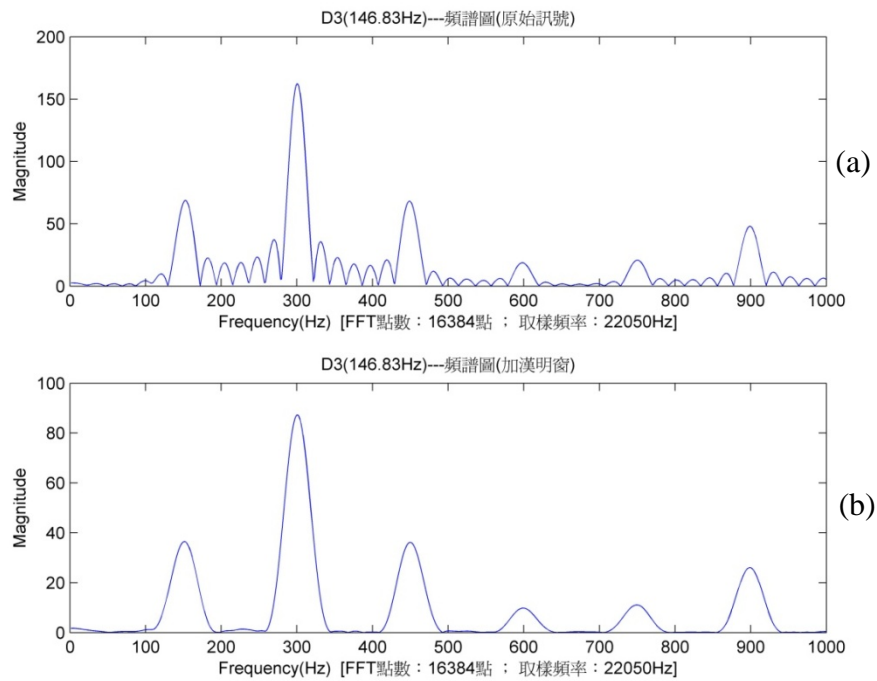


圖 2.7 原始頻域訊號與漢明窗訊號的頻域差異

由圖 2.6(b)可發現,加漢明窗後的時域訊號,其訊號的兩端趨近平緩且連續;
由圖 2.7(b)可發現,加漢明窗的訊號所對應到的頻域能明顯突出基頻與諧波,並
有效降低頻譜的溢散。



第三章 和弦特徵參數擷取

3.1 簡介

和弦(chord)在音樂樂理，係指三個音或三個音以上組合而成的聲音，就如同說話的抑揚頓挫令聽者有不同的感受，和弦在音樂裡也扮演著相類似的角色。不同的和弦的組成與和弦的進行，會影響音樂的曲風與曲型的結構。

3.2 音樂與和弦

3.2.1 木吉他構造簡介

吉他的類型，包含古典吉他，電吉他，民謠吉他等較常見的三種類別，但構造和基本彈奏方式皆大同小異，在此不多贅述，僅簡介民謠吉他的基本構造。



圖 3.1 民謠吉他外部構造

由圖 3.1 中，六條弦由左(第六弦)到右(第一弦)分別是最粗到最細弦，此六條弦被固定在上弦枕和下弦枕上，藉由撥動琴弦可以經由響孔(共鳴箱)的共鳴而產生樂音。

3.2.2 十二平均律

根據樂理與心理的分析，當樂音具 $2^{\frac{1}{12}} = 1.059$ 倍的頻率變化時，人耳始可分辨出其音高(pitch)或音程(interval)的差別，此差異在樂理上稱為半音(semitone)。依照十二平均律，可將八音度帶(octave band)分成 12 個樂音，如圖 3.2 所示：



圖 3.2 十二音律

由上圖 3.2 可知：

1. 英文字母代表樂音的音名。
2. 一個全音等於兩個半音，其中 E 到 F 的音程和 B 到 C 的音程為半音。
3. 「#」代表升記號，即該音名上升半音。

音高(pitch)，在物理上代表聲音訊號基本頻率的高低，亦是基本週期的倒數。由基本頻率轉換到半音的公式為：

$$\text{semitone} = 69 + 12 \times \log_2(\text{frequency}/440) \quad (3.1)$$

下表 3.1 為樂音頻率與音高的對照表，僅列出吉他音程範圍的頻率。

note	Octave2	Octave3	Octave4	Octave5	Octave6
A	55 (33)	110.0 (45)	220(57)	440(69)	880(81)
A#	58.3 (34)	116.5(46)	233.1(58)	466.2(70)	932.3(82)
B	61.7 (35)	123.5(47)	246.9(59)	493.9(71)	987.8(83)
C	65.4 (36)	130.8(48)	261.6(60)	523.3(72)	1046.5(84)
C#	69.3(37)	138.6(49)	277.2(61)	554.4(73)	1108.7(85)
D	73.4(38)	146.8(50)	293.7(62)	587.3(74)	1174.7(86)
D#	77.8(39)	155.6(51)	311.1(63)	622.3(75)	1244.6(87)
E	82.4(40)	164.8(52)	329.6(64)	659.3(76)	1318.5(88)
F	87.3(41)	174.6(53)	349.2(65)	698.5(77)	1396.9(89)
F#	92.5(42)	185.0(54)	370.0(66)	740.0(78)	1480.0(90)
G	98.0(43)	195.9(55)	392.0(67)	784.0(79)	1568.0(91)
G#	103.8(44)	207.7(56)	415.3(68)	830.6(80)	1661.2(92)

表 3.1 樂音頻率與音高對照表（單位：頻率(Hz)，括號內為 pitch)

由表 3.1 可知：

1. 依據吉他標準條弦法，粗體藍字頻率由小而大分別對應到吉他的第六弦(最粗)到第一弦(最細)。
2. 綠色粗體 **A440** 是樂理上的標準音高，也是吉他及其他樂器的頻率校準標準。

3.2.3 和弦的組成

和弦根據組成音的數目，又可分為三和弦和七和弦兩類。三和弦由三個樂音所組成，七和弦則由四個樂音所組成。在本論文裡總共探討八種類型的和弦，分

別為三和弦類型中的大三和弦、小三和弦、增三和弦、減三和弦以及七和弦類型中的大七和弦、小七和弦、屬七和弦和減七和弦。

表 3.2 為和弦組成音，定義了和弦的組成音與簡寫命名方式：

和弦名稱	和弦簡寫	組成音
大三和弦 (major)	省略	根音(1st)+大三度(2nd)+完五度(3rd)
小三和弦 (minor)	m	根音(1st)+小三度(2nd)+完五度(3rd)
增三和弦 (augmented)	aug	根音(1st)+大三度(2nd)+增五度(3rd)
減三和弦 (diminished)	dim	根音(1st)+小三度(2nd)+減五度(3rd)
大七和弦 (major7)	maj7	根音(1st)+大三度(2nd)+完五度(3rd)+大七度(4th)
小七和弦 (minor7)	m7	根音(1st)+小三度(2nd)+完五度(3rd)+小七度(4th)
屬七和弦 (dominant7)	7	根音(1st)+大三度(2nd)+完五度(3rd)+小七度(4th)
減七和弦 (diminish 7)	dim7	根音(1st)+小三度(2nd)+減五度(3rd)+小六度(4th)

表 3.2 和弦組成音

樂理定義：

由表 3.2 和弦組成音，定義：

根音：在和弦的組成音中，將最低的音定義為根音(root note)，根音決定了此和弦的旋律特性和低音部分。

小三度：音程差為 3 個半音。

大三度：音程差為 4 個半音。

減五度：音程差為 6 個半音。

完五度：音程差為 7 個半音。

增五度：音程差為 8 個半音。

小六度：音程差為 9 個半音。

小七度：音程差為 10 個半音。

大七度：音程差為 11 個半音。



3.2.4 頻譜上的和弦組成

和弦在樂理上是由三個或三個以上的樂音所組成，在物理上是由三個或三個以上的單一頻率所加成而得，其和弦的頻譜受單一頻率諧波的相互加成產生合成波。以下列舉兩種不同和弦，並觀察其和弦組成單音的頻譜，與加成後的和弦頻譜。

Am 和弦（組成音為 A+C+E），其和弦的頻譜如圖 3.3 所示：

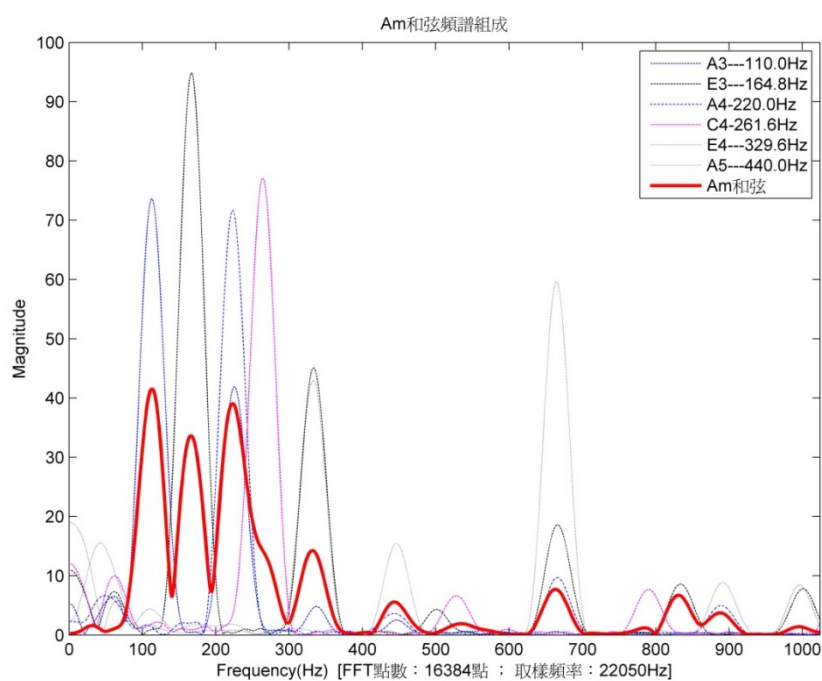
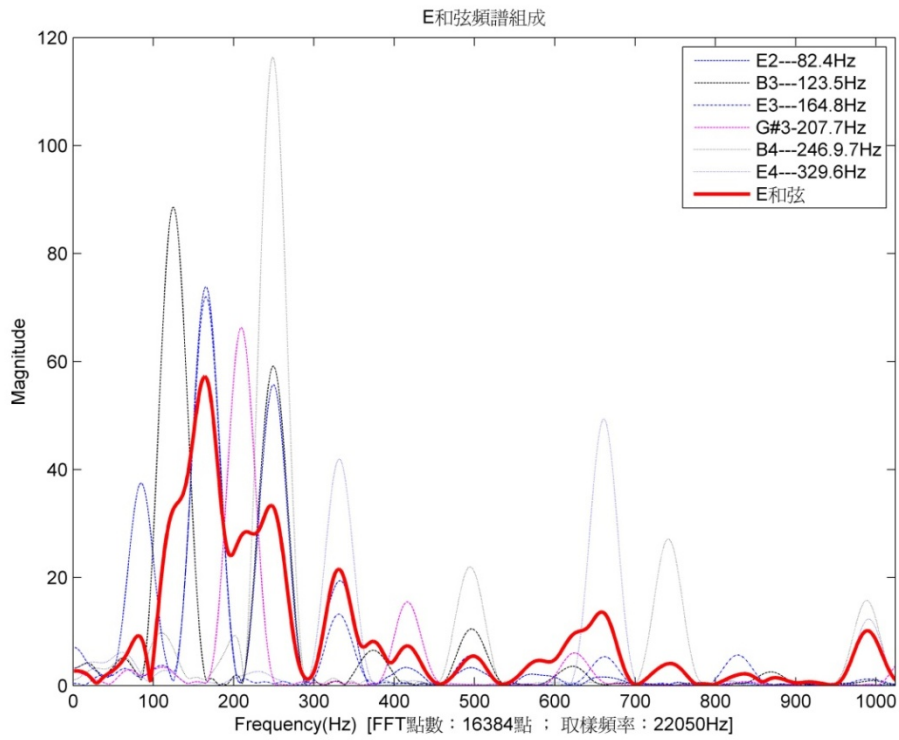


圖 3.3 Am 和弦頻譜分析

E 和弦（組成音為 E+G#+B），其和弦的頻譜如圖 3.4 所示：



3.2.5 和弦的屬性及命名

和弦屬性及命名常用簡寫表示，其寫法如下圖 3.5 所示：

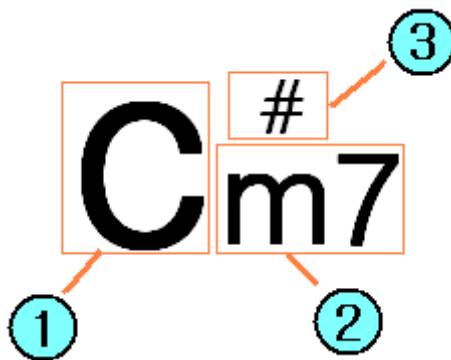


圖 3.5 和弦命名方式

由圖 3.5 所示，命名方式分為三個部分：

1. **第一部分**：表示和弦根音的音名，以英文字母標示。
2. **第二部分**：以和弦結構的縮寫作為命名，大三和弦命名時並不會加上標記，小三和弦則會加註「m」。7 代表七和弦，也代表了和弦第四個構成音的相關資訊，大七和弦可表示為「maj7」，小七和弦可表示為「m7」，屬七和弦可表示為「7」。
3. **第三部分**：代表升記號，可表示含有升記號的樂音。

3.3 和弦特徵值擷取

3.3.1 Pitch Class Profile

PCP 的取得，係首先將經過 FFT 轉換過後的頻域軸(Frequency bin)轉成音高軸(Pitch bin)，所以我們可以得到音高軸的刻度為：

$$p(k) = 69 + 12 \cdot \log_2(k/N \cdot f_{sr}/f_{ref}) \quad (3.2)$$

由式 3.2 中，我們訂定 $f_{ref} = 440\text{Hz}$ ，A440 代表 440Hz 的聲音音調，在西方樂理上，此音為標準音高，亦是樂器的頻率校準標準。

由於大部分的情況，頻率軸所對應到的音高軸無法與完全吻合所需量測的音程，Giordano Cabral 在 2005 年提出了音高軸距離計算對 PCP 擷取的影響，所以在 PCP 擷取上必須考慮所需計算的音符音高與音高軸刻度的距離差。如下圖 3.6 所示：

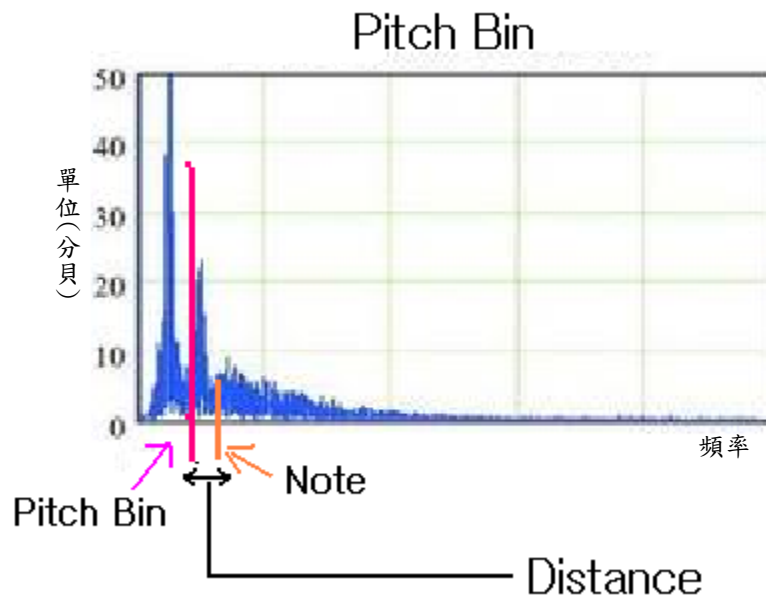


圖 3.6 音高軸與頻率軸之刻度距離差

由圖 3.6，若令距離差為 0~1 之間，則距離差可表示為

$$distance = 2 \times abs \left(\frac{12 \times \log \left(\frac{p_{2bin}}{440} \right)}{\log(2)} - \frac{12 \times \log \left(\frac{p_{2note}}{440} \right)}{\log(2)} \right) \quad (3.3)$$

於式 3.3 中，符號表示如下：

p_{2bin} ：Pitch Bin 接近或相同於所求音高的刻度

p_{2note} ：所求音高

由式 3.3 中可以知道，距離的範圍從 0~1，若欲量測的音高恰好座落於音高軸上的刻度，則距離差為 0；若欲量測的音高恰好座落於音高軸上兩刻度的中間，則距離差為 1。

Giordano Cabral 在實驗中採用了六種距離差的權重函數來研討其結果[14]，
下圖 3.7 列舉之三種權重函數，分別是圖 3.7(a)的 uniform 函數，圖 3.7(b)的 linear
函數和圖 3.7(c)的 quadratic 函數。

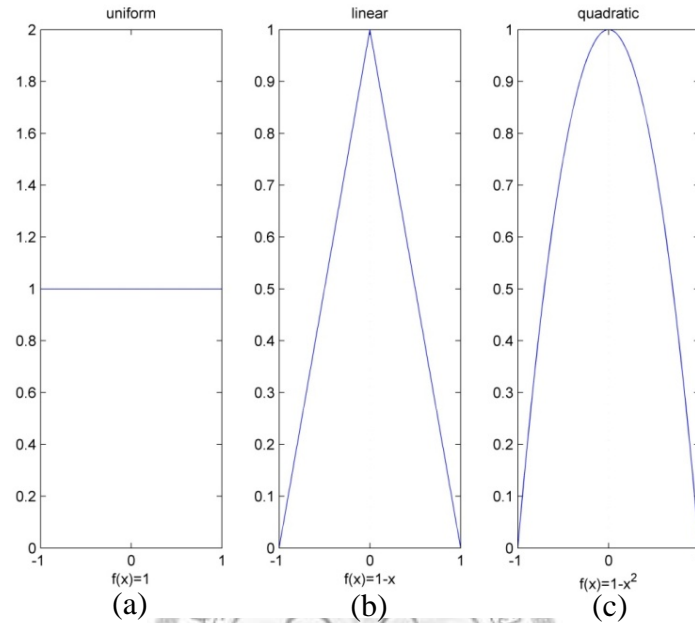


圖 3.7 weighting functions

接著實際以和弦擷取 PCP 值，並加入距離權重函數運算來觀看差異，圖 3.8
與圖 3.9 分別使用兩種不同和弦來擷取 PCP，並加入距離權重函數運算。可以發
現三種距離權重函數所算出的 PCP 值之強度略有不同，以 linear 的 PCP 強度表
現居於平均，而 uniform 函數和 quadratic 函數的 PCP 強度表現略有不同，但差
距不甚明顯。本實驗採用 uniform 函數擷取 PCP 值以減少運算的複雜度。

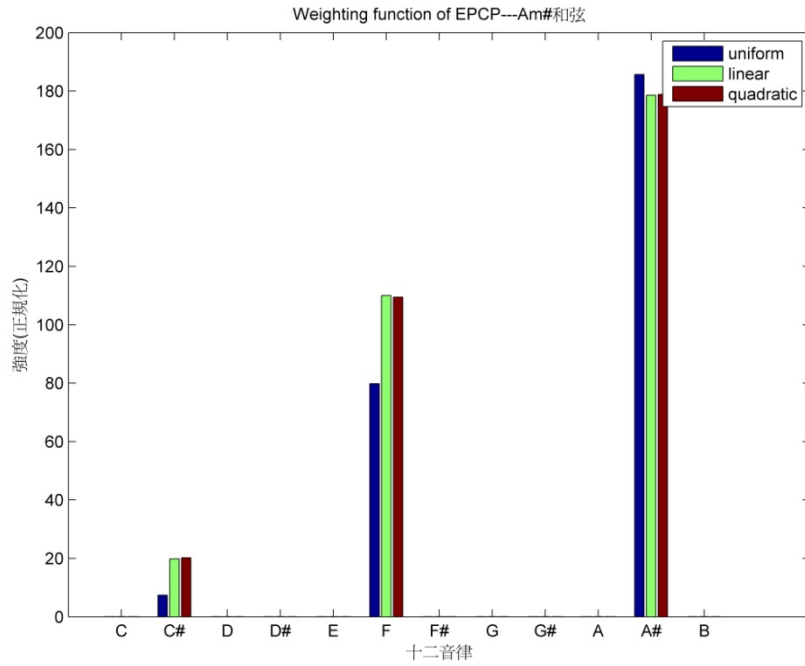


圖 3.8 Am#和弦(組成音 A#+C#+F)之三種權重函數運算的 PCP

由圖 3.9 觀察發現，使用 uniform 函數所算出的 PCP 值，擷取最多強度的音是 Am#和弦的組成音 A#；而其餘兩個 Am#和弦組成音 C#和 F，相對於使用 linear 函數和使用 quadratic 函數，uniform 函數擷取到比較少的強度。

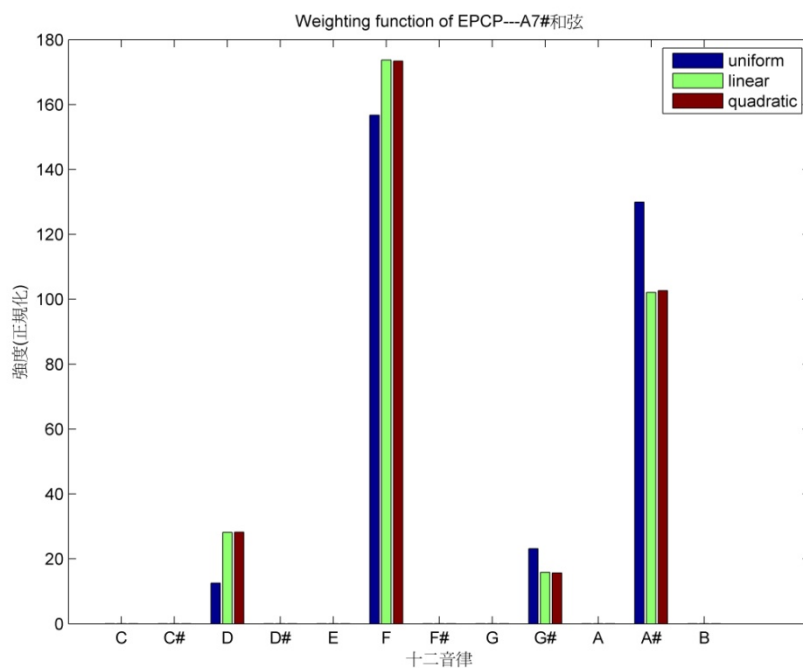


圖 3.9 A7#和弦(組成音 A#+D+F+G#)之三種權重函數運算的 PCP

由圖 3.9 觀察發現，使用 uniform 函數所算出的 PCP 值，擷取最多強度的音是 A7#和弦的組成音 A#和 G#；而其餘兩個 A7#和弦組成音 D 和 F，相對於使用 linear 函數和使用 quadratic 函數，uniform 函數擷取到比較少的強度

3.3.2 Harmonic Product Spectrum

Kyogu 在 2006 年，提出了 Enhanced Pitch Class Profile(EPCP)[15]，透過 Harmonic Product Spectrum(HPS)計算所對應的半音分類，可以避免泛音特性相近的和弦同時出現所造成的問題。

HPS 利用將頻譜訊號進行向下取樣而產生壓縮後的頻譜，將各個壓縮的頻譜相乘，便可以得到 Harmonic Product Spectrum。由於每一個壓縮後的訊號都會在

基頻附近產生一個高點，所以相乘之後便會凸顯出基頻的位置。HPS 可表示數學式如下：

$$HPS(\omega) = \prod_{m=1}^M |X(2^m \omega)| \quad (3.4)$$

於式 3.4 中，符號表示如下：

$HPS(\omega)$ ：表示 Harmonic Product Spectrum

$X(\omega)$ ：表示 FFT Spectrum

HPS 的運作機制如下圖 3.10 所示：

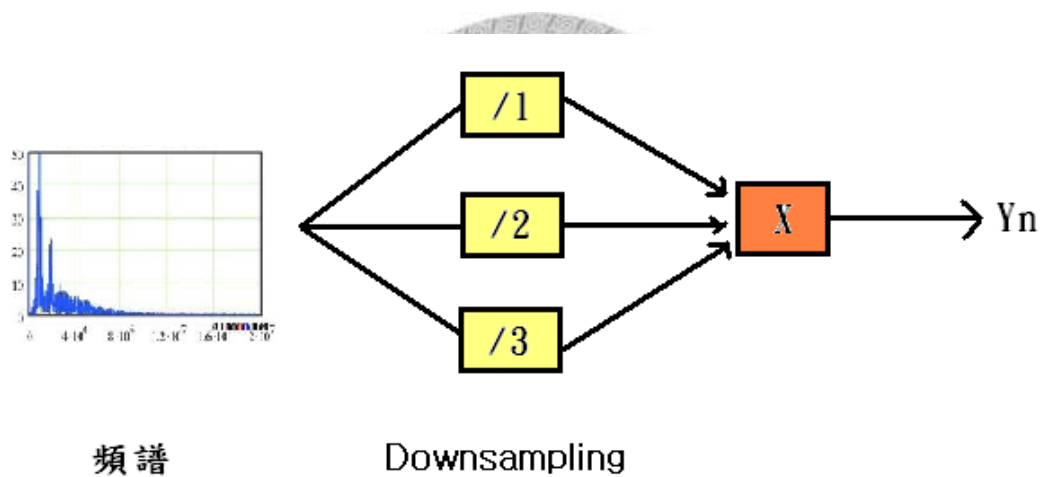


圖 3.10 Harmonic Product Spectrum(HPS)

圖 3.11 與圖 3.12，分別為 C 和弦(組成音 C+E+G) 及 Cmaj7 和弦(組成音 C+E+G+B)，在擷取 PCP 時使用 HPS 前後的差異：

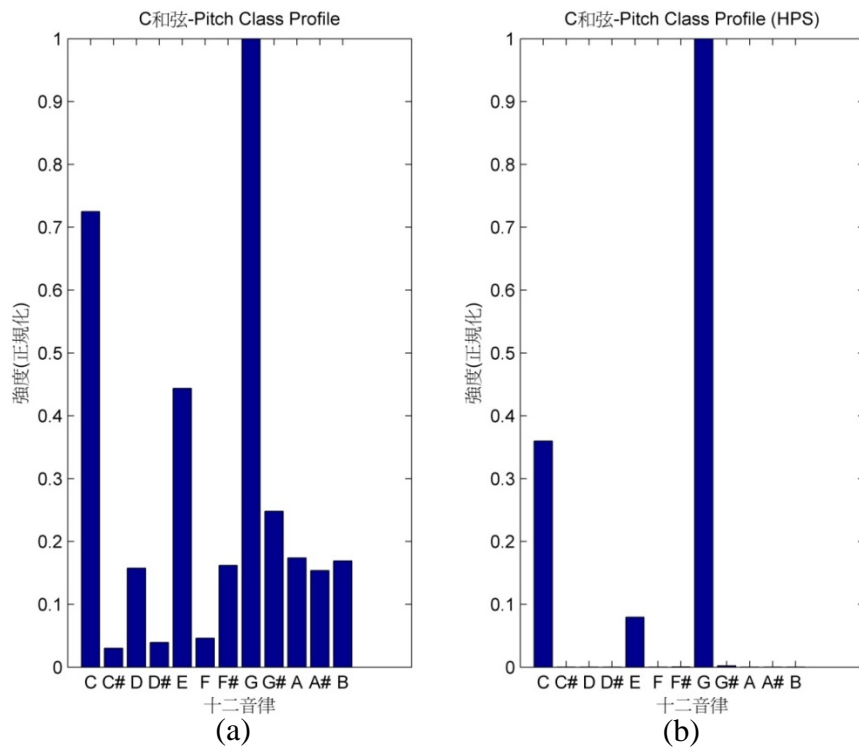


圖 3.11 C 和弦 PCP 擷取使用 HPS 之前後差異

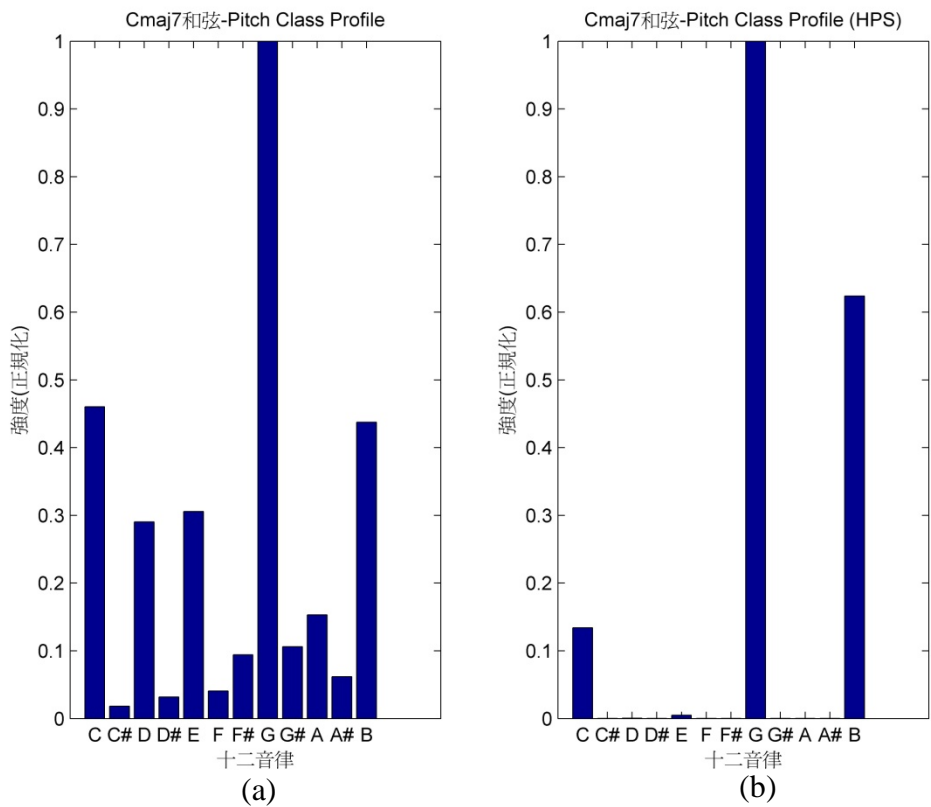


圖 3.12 Cmaj7 和弦 PCP 擷取使用 HPS 之前後差異

由圖 3.11(b)和圖 3.12 可發現，HPS 能將訊號的基頻明顯提高，並降低其他雜訊，這使我們能夠擷取到更乾淨的 PCP 值。



第四章 分類與辨識

4.1 資料分類法簡介

資料分類法[16, 17]是將已經給定的資料集合，依照資料的特性做適當的分析與歸類，並希望能透過機器學習所得的分類規則為依據，以提供未來做預測分類的依據。

分類的演算法眾多紛陳，根據機器學習模式，可以將學習過程分為監督式學習(Supervised Learning)和非監督式學習(Unsupervised Learning)。監督式學習是從問題領域中取得訓練範例(有輸入變數值，也有輸出變數)，並從中學習輸入變數與輸出變數的內在對應規則，一旦經過機器學習過後的數學模型，可以應用於未知的資料中，此時僅只有輸入變數，即可以推論出輸出變數值。

根據不同的問題類型與資料特徵型態，我們可以選擇適合的分類法，評估分類方法的優劣標準可大略分為以下四點：

1. 預測的正確度：辨識率的優劣
2. 速度和延展性：建造模型的時間和使用模型的時間
3. 健強性(robust)：處理雜值和缺值的能力
4. 延展性：有效率處理大型資料庫

本論文使用三種分類法來進行和弦分類辨識，分別是類神經網路(Neural Network)，樸真貝氏分類法(Naïve Bayes)與第 K 個最近鄰居分類法(Knn)，這三種分類法在訓練的過程皆屬於監督式學習，亦即在機器訓練時期會預先給定資料的分群，經過學習之後出來的模型可以用來預測未知的資料。

表 4.1 為三種分類器的演算機制與學習模式：

Classifier	Algorithm	Learning mode
Knn	Instance-based method	Supervised
Naïve Bayes	Probability	Supervised
Neural Network	Weighting changing	Supervised

表 4.1 三種分類器的演算類型和學習模式

4.2 K-th nearest neighbor classifier 第 K 個最近鄰居分類法

4.2.1 K-th nearest neighbor classifier 簡介

K-th nearest neighbor classifier (第 K 個最接近的鄰居分類法，簡稱 Knn)涵蓋了「物以類聚」的觀念，也就是將同一類的物件歸納聚集在一起，在數學上可以看成是距離最接近的點，其特徵最相似。因此，一個未知的資料可以透過數學空間，計算出與訓練資料庫中最相近的距離，來將此筆未知資料作歸類。K 值的意思表示前 K 個離訓練資料點最近的資料，運算完之後由多數決來判定資料的分群。透過 K 值的改變，可以應用於不同的特徵值問題。K 值的大小沒有一定的規則可循，可藉由試誤法來選擇 K 值的大小。圖 4.1 為採用 K=5 的最近鄰居演算法示意圖：

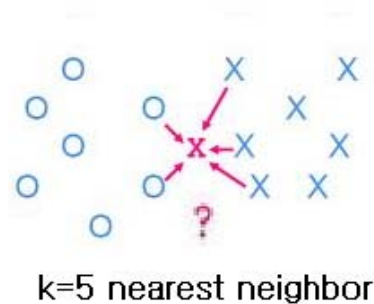


圖 4.1 K=5 最近鄰居示意圖

Knn 分類法，係利用空間距離來算出彼此之間的相似度差異，以決定分群的標準，是一種直覺式的分類法，需要事先準備學習的資料檔以供訓練作為分群的準則，在機器學習上屬於監督式學習。

4.2.2 K-th nearest neighbor classifier 理論

利用 Knn 分類法，首先必須收集一組包含類別資訊的訓練資料，在本論文之實驗中，準備了 12500 筆的資訊作為訓練，並事先將他們利用 96 種和弦做分類。資訊的部分維度為 12 維的 PCP 特徵向量。我們將訓練的資料為表示如下：

$$\{(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \dots, (X_n, Y_n)\} \quad n = 12500 \quad (4.1)$$

於式 4.1 中，符號表示如下：

(X_i, Y_i) ：代表第 i 筆資料 (X_i 是特徵向量， Y_i 是類別資訊)

則對於一筆新的測試資料 X ，若根據 1st nearest neighbor($k=1$)分類法，則最有可能的類別為

$$\text{nearest_neighbor}(x) = y_k \quad k = \arg \min_i \text{distance}(x - x_i) \quad (4.2)$$

由式 4.2 中，我們使用歐幾里德距離(Euclidean distance)作為我們的距離函數，可表示為：

$$\text{distance}(x - x_i) = |x - x_i|^2 \quad (4.3)$$

4.2.3 K-th nearest neighbor classifier 的分類運作

Knn 藉由直覺式的距離判斷，將未知資料分群，優點是計算簡單，缺點是抗雜訊能力差。當預測資料檔裡含有許多雜訊時，Knn 的效果很容易受到雜訊的影響而產生嚴重的誤判。此外，Knn 分類法不適用於資料維度很高的時候，因為會複雜化距離的運算，而導致花費較多的運算時間。

下圖 4.2，是 Knn 分類法的學習與辨識流程：

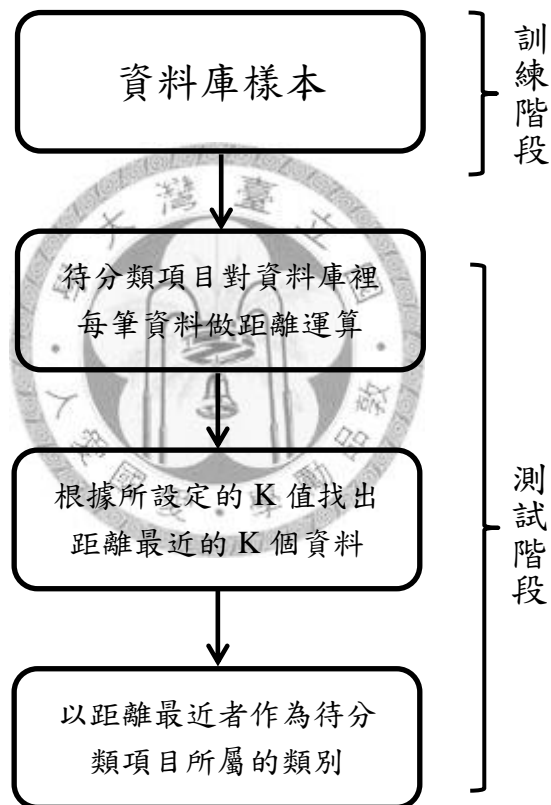


圖 4.2 Knn 學習與辨識流程圖

4.3 Naïve Bayes classifier 樸真貝氏分類器

4.3.1 樸真貝氏分類器簡介

貝氏分類法是一種利用機率分佈為基礎的分類法，因此也可以稱為是機率的學習法(probabilistic learning)，其演算法基本的核心基礎，是對於給出的待分類未知項，求解在此項出現的條件下各個類別出現的機率，判定哪個機率最大，就認定此待分類項屬於哪個類別。由於貝氏分類法是根據統計學中的貝氏定理 (Bayes' theorem) 下衍生而來，所以對每個事件(或特徵屬性)的發生機率都要是獨立時，此貝氏模型才最能代表這群資料筆的狀態，所以又稱為樸真貝氏分類法 (Naïve Bayes classifier)。

4.3.2 樸真貝氏分類法理論

樸真貝氏分類法屬於監督式機器學習，分為兩個階段。在最初之訓練階段時，將已知分群的資料庫由電腦學習而產生此問題的機率模型。第二階段利用建立好的貝氏機率模型，對未知的資料進行預測和分類。

樸真貝氏分類法的概念源自於貝氏定理，貝氏定理是一種機率論的結果，通常在事件 B 已經發生的前提下，事件 A 發生的機率與事件 A 已經發生的前提下，事件 B 發生的機率是不一樣的，然而，透過貝氏定理，可以將兩者的關係連結在一起。以上的因果關係，通常可以表示為：

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (4.4)$$

$P(A|B)$ ：表示事件 B 已經發生的前提下，事件 A 發生的機率

故由貝氏定理可知

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (4.5)$$

樸真貝氏分類法的學習階段分成訓練階段和預測階段

階段一：訓練階段

首先由資料庫建立訓練樣本集，其必須包含訓練的資料與已知的分類項集合，並統計得到各類別之各個特徵屬性的條件機率估計。即


$$P(a_1|y_1), P(a_2|y_1) \cdots P(a_m|y_1); P(a_1|y_2), P(a_2|y_2) \cdots P(a_m|y_2) \cdots \\ P(a_1|y_n), P(a_2|y_n) \cdots P(a_m|y_n);$$

其中

$\{a_1, a_2 \cdots a_m\}$ 表示此已知資料筆的特徵屬性集合， m 代表此特徵屬性的維度。

$\{y_1, y_2 \cdots y_n\}$ 表示分類項之類別集合， n 代表類別集合的個數。

若各個特徵屬性是條件獨立的，則根據貝氏定理可知


$$P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)}$$

(4.6)

其中 $P(x|y_i)P(y_i) = P(a_1|y_i)P(a_2|y_i) \cdots P(a_m|y_i) = P(y_i) \prod_{j=1}^m P(a_j|y_i)$

階段二：預測階段

假設 $X = \{a_1, a_2 \cdots a_m\}$ 為一個等待分類的項目，其中每個 a 為 X 的特徵屬性 (共 m 個特徵屬性)

此時預先設定好的類別集合 $C = \{y_1, y_2 \cdots y_n\}$ n 代表類別的個數

分別計算出 X 在所有類別出現的機率，即 $P(y_1|x), P(y_2|x) \cdots P(y_n|x)$

其中機率最大者則將 X 歸納到此類別中。

圖 4.3 所示，是貝氏分類法的學習與辨識流程：

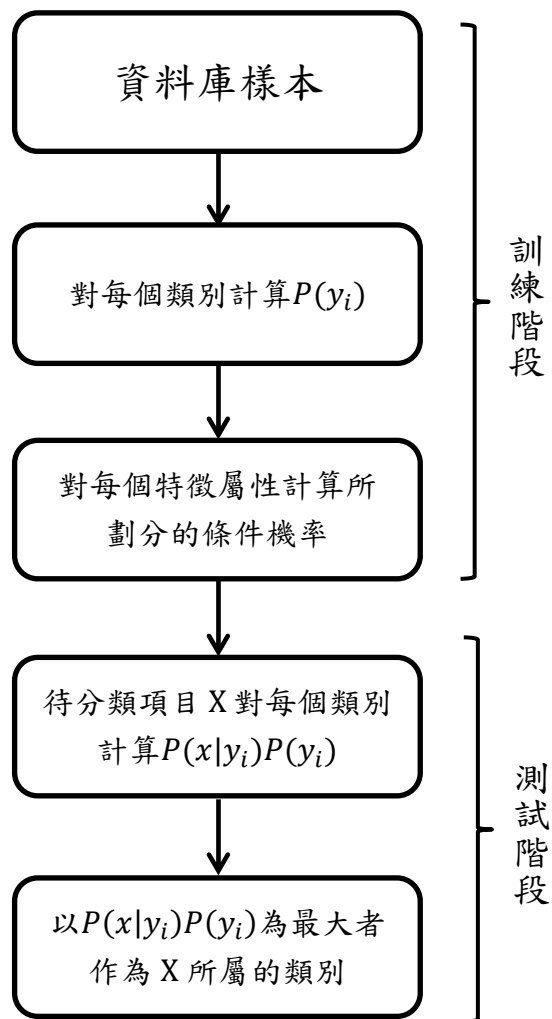


圖 4.3 Naive Bayes Classifier 學習與辨識流程圖

4.4 Neural Network 類神經網路

4.4.1 類神經網路簡介

類神經網路，或稱人造神經網路(Artificial Neural Network，簡稱 ANN)，是用來模仿生物神經網路(Biological Neural Network)所發展出來的數學模型。人類的大腦約由將近 10^{11} 個神經細胞所組成，每個神經細胞又含有 10^4 個突觸(Synapses)來與其他細胞相互連結成一個極複雜的神經網路。當人類感官受到外界刺激，神經細胞接受訊息之後會將訊息傳遞到大腦來讓大腦下達命令給相關受器(如四肢)做出反應。

為了在語音和影像辨識上獲致與人腦相似的功能，至 1940 年起，科學家便開始著手從事此方面的研究，並仿造出最簡單的人工神經元(Artificial Neurons)來模擬人的腦神經細胞。類神經網路模擬生物神經網路的連結，採用人工神經元和節點(Nodes)所組成，藉由人工神經元從外界環境獲取資訊，並以簡單的矩陣運算程序來輸出其結果或其他人工神經元。類神經網路的主要目的為模擬生物神經元間之運算訊息的能力，它最大的特點是能夠達到如生物神經系統所具備大量平行計算及分散儲存與處理的工作能力，這樣的設計很適合被用來處理各種需要大量資料運算的運用上，如語音辨識。

綜合言之，類神經網路可以歸納下列幾項特點：

1. **非線性(Nonlinearity)**：類神經網路內人工神經元之間的運算在處理非線性的系統上有很高的處理能力及優質的效能，能夠解決其他非線性演算法處理上的費時與計算上的繁雜，很適合用於語音辨識系統或非線性控制上。
2. **學習能力性(Learning Ability)**：透過使用者輸入不同的學習樣本來訓練類神經網路，並修改神經元的權重數值直到期望誤差函數(performance index)達到使用者所設定的收斂值。
3. **容錯性(Fault Tolerance)**：訓練過後類神經網路模型具有高容忍度，即使將來用來模擬的輸入訊號含有雜訊干擾時，仍不會影響其運作的精確度。除此之外，即使類神經網路內部部分神經網路失效也能照常運作。此特點很適合用來做音樂辨識以適應一般錄製音訊時所產生的雜訊干擾。
4. **關聯式記憶能力(Associative Memory)**：類神經網路在訓練的過程中，會記憶曾經訓練過的輸入函數和其所對應的目標函數。訓練過後的類神經網路模型

具有關聯式記憶，即便所輸入資料僅相似於訓練時的輸入函數，類神經網路仍能透過聯想和連結判斷相似的目標函數，這類似於人的耳朵聽到了小提琴彈奏一首歌曲，若改用鋼琴彈奏同樣一首歌，雖然彈奏的樂器不同，但是人卻能夠透過旋律的相似度，去聯想出這首歌的名稱。

5. **積體電路硬體實現(VLSI Implementation)**：類神經網路的結構高度的連接性(interconnection)，很容易將簡單且具規則性的類神經網路移植到積體電路上。

4.4.2 類神經網路理論

一. 生物神經元

如圖 4.4 所示，生物神經元的結構主要分成下面幾個部分[18]：

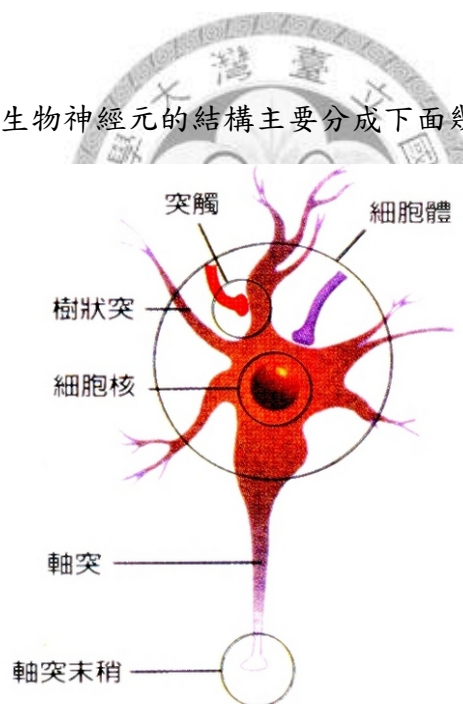


圖 4.4 生物神經元構造

細胞核(Nucleus)是神經細胞的中心體，主要功能是將樹狀突(Dendrite)所收集到的信號作加總並進行一次非線性轉換，然後透過軸突(Axin，或稱神經軸)將信號傳送到其他的神經細胞中。

神經樹狀突如同樹枝結構，可以接受突觸(Synapse)所傳入神經細胞體的信號，或傳送信號至另一神經元的突觸，突觸的主要功用則為記憶儲存先前所累積下來的記憶。

在人類大腦中，突觸代表了神經元之間的連結強度，當神經系統受到刺激，訊號透過軸突傳輸至突觸，突觸會不斷地改變本身的連結強度，來推測神經細胞的輸出以適應外在環境的變化，一直到達穩定狀態為止。簡單的神經網路傳遞流程如圖 4.5 所示：

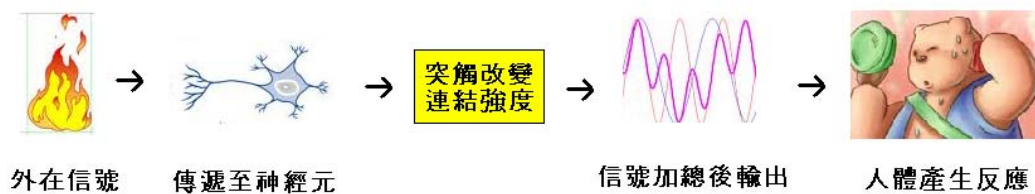


圖 4.5 生物神經元網路傳遞流程

二. 人工神經元

人工神經元是經由仿效生物神經元而建立，其模型如圖 4.6 所示，它被當作是類神經網路內的基本處理單元(Processing-element)，每個神經元皆包含了多數的輸入 x 及一個輸出 y ，而每個神經元所輸出的值會散發出去，成為其他神經元的輸入，其輸入與輸出之間的關係為[19]：

$$y = f\left(\sum_{i=1}^n W_i \cdot X_i - b\right) \quad (4.7)$$

於式 4.4 中，符號表示如下：

f : 表示神經元所使用的激發函數(Activation function)

W_i : 表示神經元之間的連接權重值(Weighting)

b : 表示神經元的偏權值(Bias)

n ：表示輸入神經元之數目

圖 4.6 為神經元的模型：

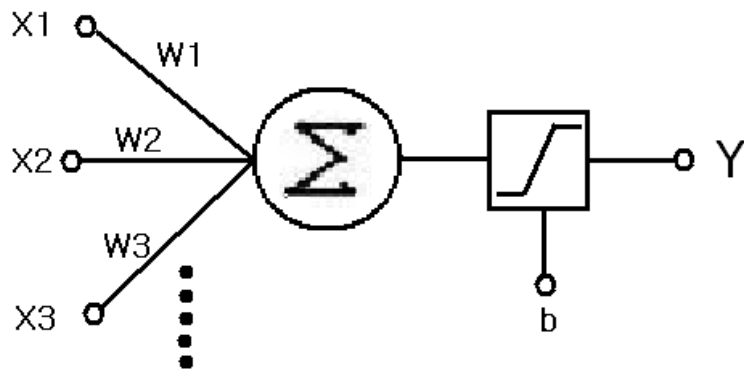


圖 4.6 神經元模型

於圖 4.6 中，輸入 X_i 等值於生物神經元中的神經樹，權重值 W_i 則相當於生物模型中的突觸。神經元的激發函數 f ，則涵蓋了線性和非線性兩種。常用的激發函數，如表 4.2 所列，其函數型態則如圖 4.7 所示：

表 4.2 為類神經網路常用的激發函數：

函數	$f(x)$	範圍
Purelin	$f(x) = k \cdot x$	
Logsig	$f(x) = \frac{1}{1 + e^{-x}}$	$0 < f(x) < 1$
Tansig	$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$	$-1 < f(x) < 1$

表 4.2 類神經網路常用的激發函數

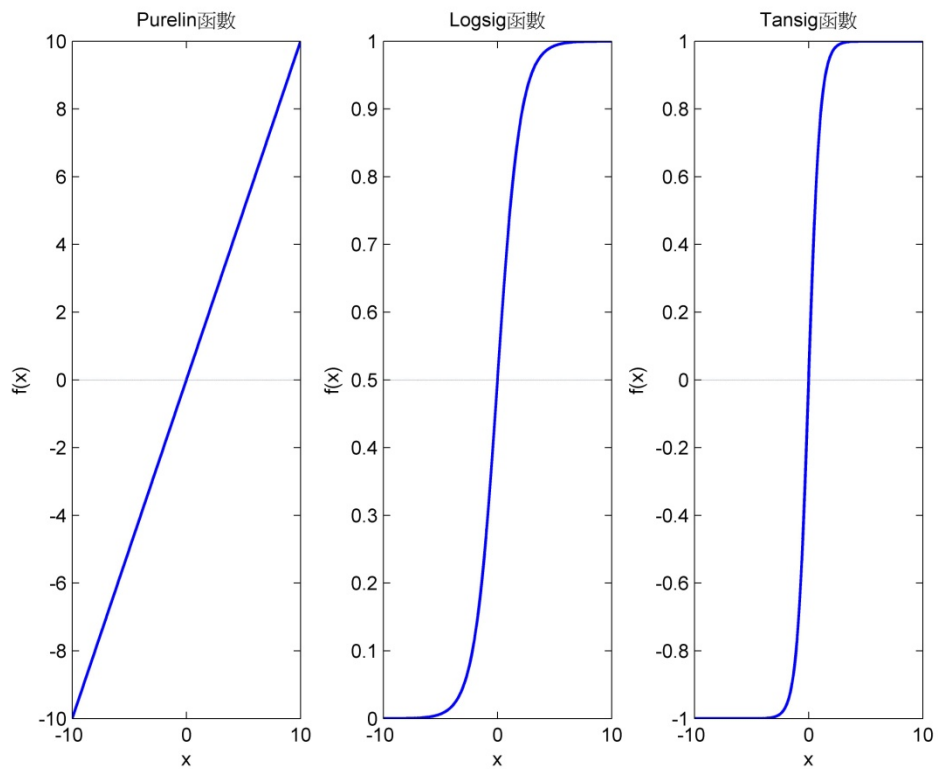


圖 4.7 類神經網路常用的激發函數



4.4.3 類神經網路的基本架構

如大腦內神經元的鍵結，類神經網路亦由基本處理元件的神經元相互連結形成龐大的網路架構。其架構可以分為前饋式網路(Feed-forward Net)及回饋式網路(Recurrent Net)，圖 4.8 為前饋式網路的架構。

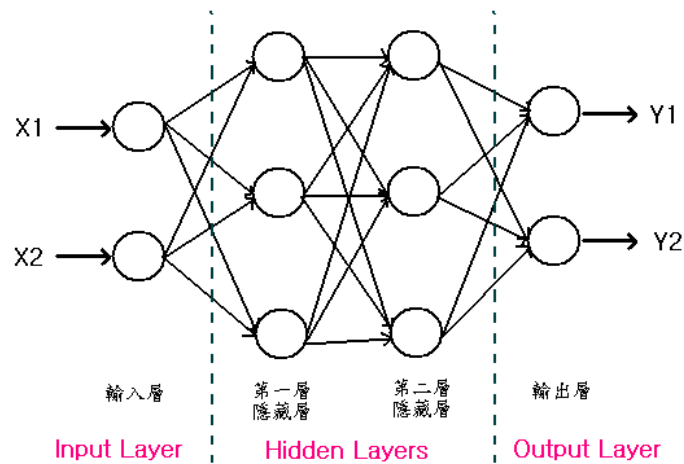


圖 4.8 類神經前饋式網路架構

由圖 4.8 所示，前饋式網路包含了三大部分，分別是輸入層，隱藏層以及輸出層。同一層內的神經元不做互相連結傳遞，且信號傳遞由輸入層到隱藏層均為單一方向進行。隱藏層的層數可隨問題的複雜度增加，但相同的，隱藏層的層數越多也會越增加運算的複雜度，隱藏層的層數給定，無一定數學法則，僅只能透過試誤法，找出最佳的層數。當信號進入類神經網路之後，輸入層可做線性轉換，隱藏層則主要使用非線性轉換函數來表現輸入間之交互的影響，輸出層則負責將信號輸出，同樣也可以做線性或非線性的轉換。

4.4.4 類神經網路的運作

類神經網路的運作過程可以分為學習過程(Learning)和回想過程(Recalling)兩部份。學習的過程是學習類組的輸入，以調整網路連結的權重值和神經元的偏權值。依照問題類型可以將學習過程分為監督式學習(Supervised Learning)和非監督式學習(Unsupervised Learning)。本論文，採用監督式學習。

監督式學習是從問題領域中取得訓練範例(有輸入變數值，也有輸出變數)，並從中學習輸入變數與輸出變數的內在對應規則，一旦經過學習過後的類神經網路，可以應用於新的案例，此時僅只有輸入變數即可以推論出輸出變數值。例如老師在黑板上寫上英文字母 ABCD，並且向學童說明此字母為英文字母

ABCD，學童透過認字來學習每個字母的特徵，當下次老師在黑板上寫上 ABCD 時，雖然新寫的字母筆跡與之前寫的原字母筆跡不完全相同，但學童仍能夠認出是哪個字母，此乃因為學童已將各別字母的特徵普遍化。

監督式學習最重要的兩大元素就是輸入所需要學習的問題與所對應學習問題的輸出答案(也稱為學習目標)，藉由誤差來調整神經元之間連結的權重值，使的類神經網路輸出可以逐漸接近目標值。有關類神經之學習流程，如圖 4.9 所示 [20]：

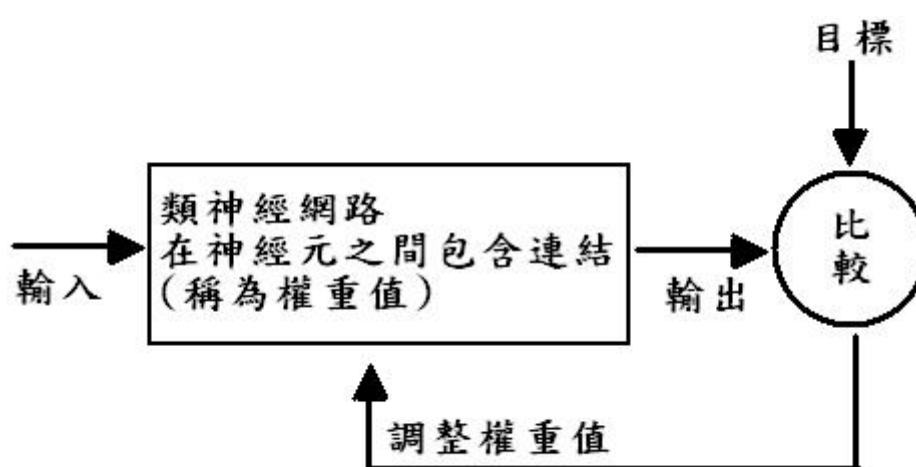


圖 4.9 類神經學習流程

本論文所使用的誤差倒傳遞演算法(Error Back-propagation Algorithm)，係採用監督式學習演算法則，並透過網路目標輸出與推論輸出之間的誤差，由類神經網路的輸出層往輸入層倒傳遞回推，以誤差值作為修正網路權重值和偏權值的參考依據。學習過程結束之後，則進入回想過程接受測試資料，以產生最佳的推論輸出值。

4.4.5 倒傳遞類神經網路

一. 倒傳遞網路架構

下圖 4.10 為倒傳遞類神經網路架構圖，於此網路中，隱藏層的層數與使用的神經元數目無明確的參考準則，取決於問題的複雜度，但通常隱藏層層數以不超過兩層為基準。透過增加隱藏層的層數，可以強化處理問題複雜度的能力，通常依據實驗的結果，來修正神經元個數和層數，以期待能得到最佳輸出結果。

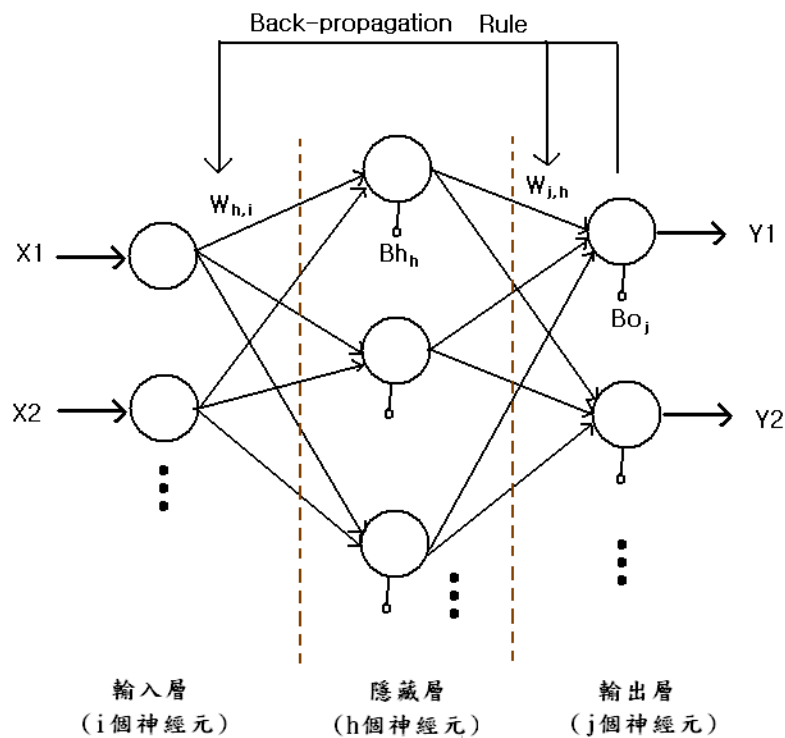


圖 4.10 倒傳遞類神經網路架構圖

二. 前饋網路

於圖 4.10 之單層倒傳遞類神經網路架構中，相關符號表示如下：

i ：輸入層神經元個數

h ：隱藏層神經元個數

j ：輸出層神經元個數

$W_{h,i}$ ：輸入層到隱藏層的連結權重

$W_{j,h}$ ：隱藏層到輸出層的連結權重

Bh_h ：隱藏層內神經元的偏權值

Bo_j ：輸出層內神經元的偏權值

V_h ：隱藏層的輸出值

Y_j ：輸出層的輸出值

X_i ：網路輸入變數值

由圖 4.10，則隱藏層的第 h 個神經元，其輸出值 V_h 可表示為

$$V_h = f \left\{ \left(\sum_{i=1}^i W_{h,i} \cdot X_i \right) + Bh_h \right\}$$

(4.8)

於式 4.8 中， f 代表激發函數

由圖 4.10，其輸出層的第 j 個神經元，其輸出值 Y_j 可表示為

$$Y_j = f \left\{ \left(\sum_{h=1}^h W_{j,h} \cdot V_h \right) + Bo_j \right\}$$

(4.9)

接著，定義誤差函數為

$$E = \frac{1}{2} \sum_j (t_j - y_j)^2$$

(4.10)

於式 4.10 中，符號表示如下：

t_j ：代表輸出層目標輸出值

y_j ：代表輸出層前饋輸出值

三. Levenberg-Marquardt Backpropagation 演算法

以往的倒傳遞類神經網路乃採用最陡坡降法(gradient descend)的觀念，將目標值與實際輸出值之間的誤差函數最小化，藉著訓練時調整網路的連結權重，以達到類神經學習的目的。但是採用最陡坡降法會有以下幾個主要缺點：

1. 容易收斂至局部最小值，而無法獲取全域最小值。
2. 當搜尋結果越靠近最小值時，由於梯度變小，權值更新的速率變慢，因此疊代次數變多，學習時間會拉長。

為了解決最陡坡降法的缺點，可以使用牛頓法來尋找最小值。牛頓法在靠近最佳點時擁有良好的收斂性。標準的牛頓法疊代公式為：

$$X_{k+1} = X_k - A_k^{-1} g_k \quad (4.11)$$

於式 4.11 中，

A_k 是矩陣誤差向量對當前權重值和偏權值的二次微分所構成的 Hessian 矩陣

X_k 是當前的權重值和偏權值的向量

X_{k+1} 是疊代後之權重值和偏權值的向量

故 A_k 和 g_k 可表示如下：

$$A_k \equiv \nabla^2 F(x)|_{x=x_k} \quad (4.12)$$

$$g_k \equiv \nabla F(x)|_{x=x_k} \quad (4.13)$$

Levenberg-Marquardt 演算法主要的精隨就是結合牛頓法跟最陡坡降法的優點，在疊代的初期，使用最陡坡降法來向梯度方向接近；疊代的後期則使用牛頓法來趨近最小值。

在 Levenberg-Marquardt 的算法中，類神經網路的連結權重之更新公式可表示為

$$\Delta w = (J^T J + \mu I)^{-1} J^T e \quad (4.14)$$

於式 4.14 中，符號表示如下：


Δw ：表示類神經網路所有連結權重與偏權值的更新向量

I ：表示單位矩陣

μ ：表示結合係數

J ：Jacobian Matrix

其中 Jacobian Matrix J 和誤差向量 e 可以表示為



$$J = \begin{bmatrix} \frac{\partial e_{11}}{\partial w_1} & \frac{\partial e_{11}}{\partial w_2} & \dots & \frac{\partial e_{11}}{\partial w_N} \\ \frac{\partial e_{12}}{\partial w_1} & \frac{\partial e_{12}}{\partial w_2} & \dots & \frac{\partial e_{12}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{1j}}{\partial w_1} & \frac{\partial e_{1j}}{\partial w_2} & \dots & \frac{\partial e_{1j}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{p1}}{\partial w_1} & \frac{\partial e_{p1}}{\partial w_2} & \dots & \frac{\partial e_{p1}}{\partial w_N} \\ \frac{\partial e_{p2}}{\partial w_1} & \frac{\partial e_{p2}}{\partial w_2} & \dots & \frac{\partial e_{p2}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{pj}}{\partial w_1} & \frac{\partial e_{pj}}{\partial w_2} & \dots & \frac{\partial e_{pj}}{\partial w_N} \end{bmatrix} \quad e = \begin{bmatrix} e_{11} \\ e_{12} \\ \dots \\ e_{1j} \\ \dots \\ e_{p1} \\ e_{p2} \\ \dots \\ e_{pj} \end{bmatrix}$$

(4.15)

於式 4.15 中，符號表示如下：

p ：表示訓練組數(patterns)的數目

N ：類神經網路內連結權重的數目(包含權重值和偏權值的數目)

j ：output 的數目

w ：連結權重

誤差向量 e 表示為

$$e_{pm} = d_{pm} - o_{pm} \quad (4.16)$$

於式 4.16 中，符號表示如下：

d_{pm} ：輸入函數所對應的目標輸出值

o_{pm} ：類神經網路的輸出值

由式 4.10 的定義，可得誤差能量函數為

$$E(w) = \frac{1}{2} \sum_{p=1}^P \sum_{j=1}^J e_{pj}^2 \quad (4.17)$$

由式 4.12，此時可以得到 $N \times N$ 的 Hessian Matrix H

$$H = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 E}{\partial w_1 \partial w_N} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2^2} & \cdots & \frac{\partial^2 E}{\partial w_2 \partial w_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 E}{\partial w_N \partial w_1} & \frac{\partial^2 E}{\partial w_N \partial w_2} & \cdots & \frac{\partial^2 E}{\partial w_N^2} \end{bmatrix} \quad (4.18)$$

於式 4.18，符號表示如下：

N ：表示為連結權重的數目(包含權重值和偏權值的數目)

於式 4.18 中，矩陣內的元素可表示為：

$$\frac{\partial^2 E}{\partial w_a \partial w_b} = \sum_{p=1}^P \sum_{j=1}^J \left(\frac{\partial e_{pj}}{\partial w_i} \frac{\partial e_{pj}}{\partial w_j} + \frac{\partial^2 e_{pj}}{\partial w_i \partial w_j} e_{pj} \right) \quad (4.19)$$

於式 4.19 中， a, b 為權重順序

此時，由高斯牛頓法，可假設式 4.19 中之 $\frac{\partial^2 e_{pj}}{\partial w_i \partial w_j} e_{pj} \approx 0$ [21, 22]


因此，式 4.19 可近似於

$$\frac{\partial^2 E}{\partial w_a \partial w_b} \approx \sum_{p=1}^p \sum_{j=1}^j \left(\frac{\partial e_{pj}}{\partial w_i} \frac{\partial e_{pj}}{\partial w_j} \right) = q_{ij} \quad (4.20)$$

於式 4.20 中

q_{ij} ：代表 quasi-Hessian matrix

若結合式 4.15 與式 4.20，式 4.18 的 Hessian 矩陣可以近似於：



$$H \approx J^T J \quad (4.21)$$

於式 4.16 中，假設一個 Q 為 quasi-Hessian matrix，則 Q 為

$$Q \approx H \quad (4.22)$$

假設一個 $N \times 1$ 的梯度向量 g 為

$$g = \left[\frac{\partial E}{\partial w_1} \quad \frac{\partial E}{\partial w_2} \quad \dots \quad \frac{\partial E}{\partial w_N} \right]^T \quad (4.23)$$

則將式 4.17 代入式 4.23 中，則梯度向量 g 內的元素可表示為：

$$g_i = \frac{\partial E}{\partial w_i} = \sum_{p=1}^P \sum_{m=1}^M \left(\frac{\partial e_{pm}}{\partial w_i} e_{pm} \right) \quad (4.24)$$

由式 4.15 和式 4.24，可得此向量梯度為

$$g = J^T e \tag{4.25}$$

由式 4.21、式 4.22 和式 4.25，可將式 4.14 更新權重之公式改寫為：

$$\Delta w = (Q + \mu I)^{-1} g \tag{4.26}$$

其中，如果標量因子 $\mu = 0$ 時，就剛好是使用近似化 Hessian 矩陣的牛頓法了。當 μ 值很大時，上式變成具有小的步階大小之梯度下降 (gradient descend)。牛頓法在誤差極小值附近較快且比較正確，所以上式的目的就是要盡可能快速地移向牛頓法。因此在每個成功的步階 (亦即步階內性能函數降低) 就降低 μ 值

圖 4.11 是類神經訓練網路的流程圖：

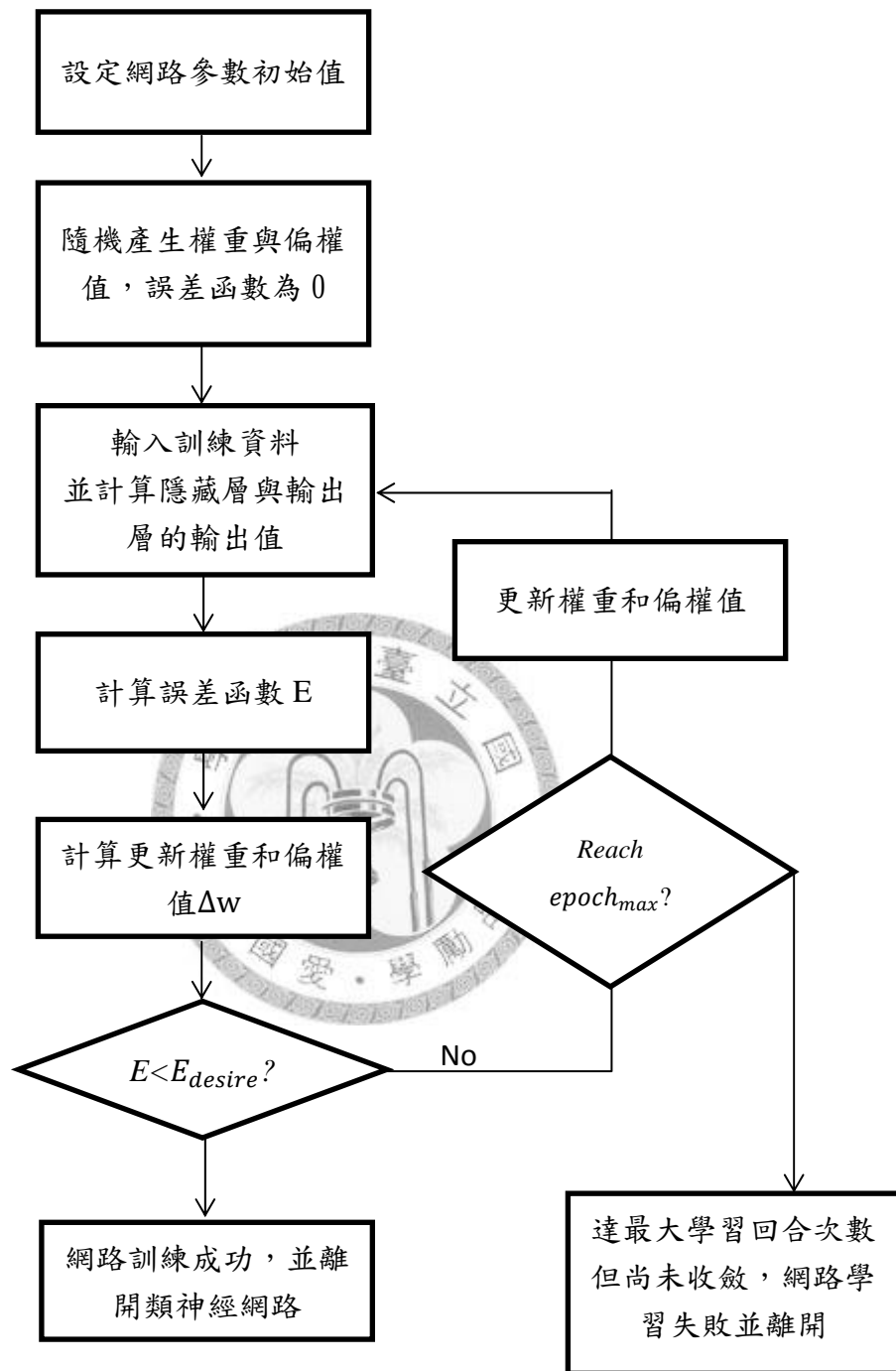


圖 4.11 類神經訓練網路的流程圖

第五章 實驗設備與架構

5.1 實驗設備

5.1.1 視訊麥克風

採用如圖 5.1 之直立式桌上型電容視訊麥克風，作用為接收吉他的聲音訊號，其規格如下：

頻率響應：30Hz~15KHz

阻抗： $<10k \text{ Ohm}$

靈敏度： $-58\text{dB}\pm 3\text{dB}$

輸入線路：3.5mm 立體聲插頭



圖 5.1 視訊麥克風

5.1.2 民謠吉他

本實驗使用如圖 5.2 與圖 5.3 之兩把吉他進行實驗，Guitar 1 (GT1)用於錄製音訊檔以建立和弦資料庫，而 Guitar 2 (GT2)則用來測試辨識率。



圖 5.2 實驗用吉他(GT1)



圖 5.3 實驗用吉他(GT2)

圖 5.4 為 GT1 和 GT2 此兩把吉他的 Auto-Spectrum，透過敲擊兩把木吉他的琴聲身，來產生共鳴箱的頻率，藉此觀看兩把木吉他的頻率響應分佈的曲線：

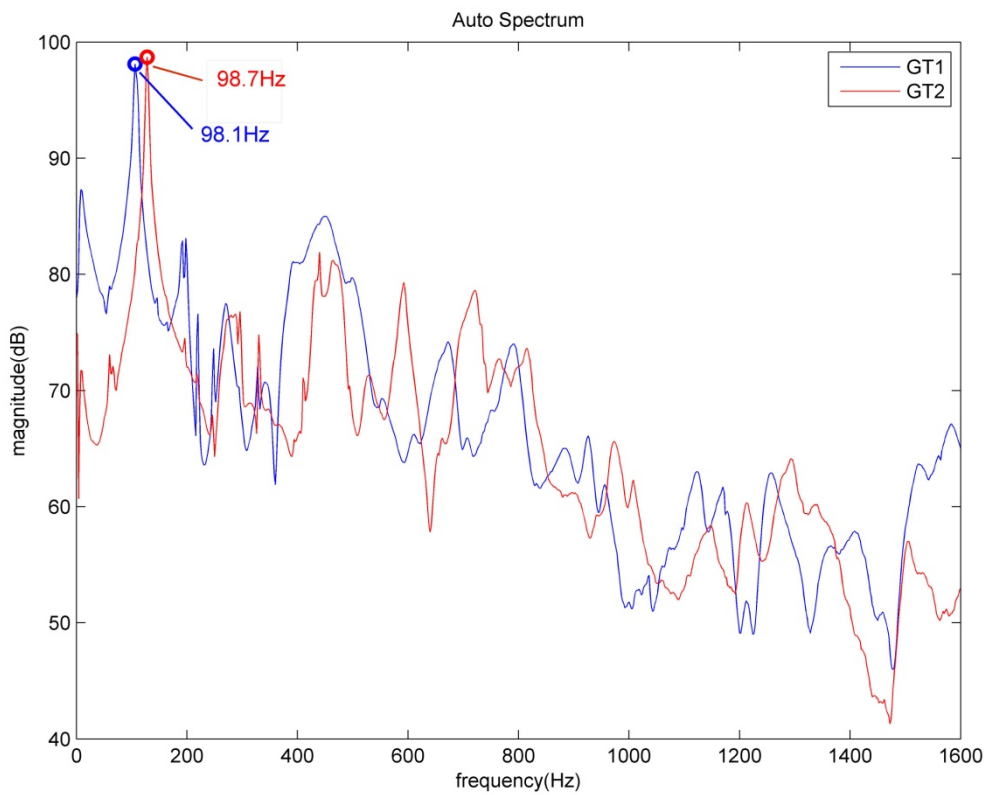


圖 5.4 GT1 和 GT2 的 Auto-Spectrum

5.2 實驗系統

本論文所涉及之實驗系統，包括個人電腦、訊號擷取卡與虛擬音源等三部分，即

1. 個人電腦：

處理器：AMD Phenom(tm) II X4 945 Process 3.0GHz

記憶體：3GB

程式：MATLAB R2007a

2. 訊號擷取卡

Realtek HD Audio 音效卡(內建在個人電腦中)，取樣頻率最高為 44.1kHz，本實驗使用的取樣頻率為 22050Hz

3. 虛擬音源

- 軟體：Guitar Pro



5.3 實驗架構

本論文之實驗架構分成兩種：

如圖 5.5 之實驗架構一所示，可利用三種分類法建立辨識模型，並測試未知資料樣本來觀察其辨識率。

實驗架構一流程：

將訓練樣本 12500 筆資料筆數給三種分類法做為學習的範例，利用產生出來的辨

識模型，去測試未知的 12500 筆資料筆數測試樣本，並觀察其辨識率的好壞。

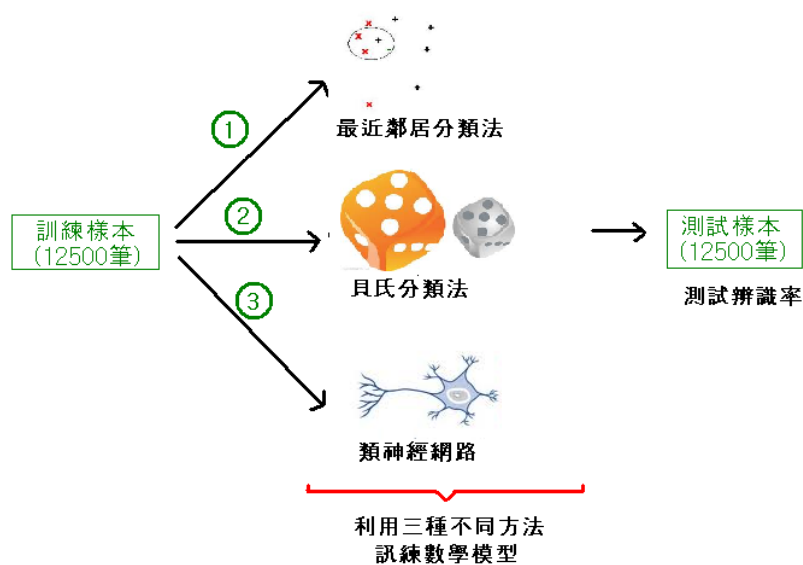


圖 5.5 實驗架構一

其次，利用圖 5.6 實驗架構二之吉他和弦即時辨識所示，可將訓練好的辨識數學模型拿來使用，在彈奏吉他時將吉他共鳴箱對準麥克風收音，透過程式運算，即可在自製的使用者介面(GUI)即時顯示出和弦名稱。

實驗架構二流程：

彈奏時將吉他靠近麥克風，讓麥克風接收音訊，接收到的音訊先進行音量判定，若音量過低，則判定沒有彈奏；若音量達到標準之上，則將訊號進行數位訊號處理，並求出 PCP 特徵值，再給以建立好的三種辨識模型進行和弦的辨識，得到的辨識結果，將透過 GUI 介面顯示出來給使用者觀看。流程示意圖如圖 5.6 所示：

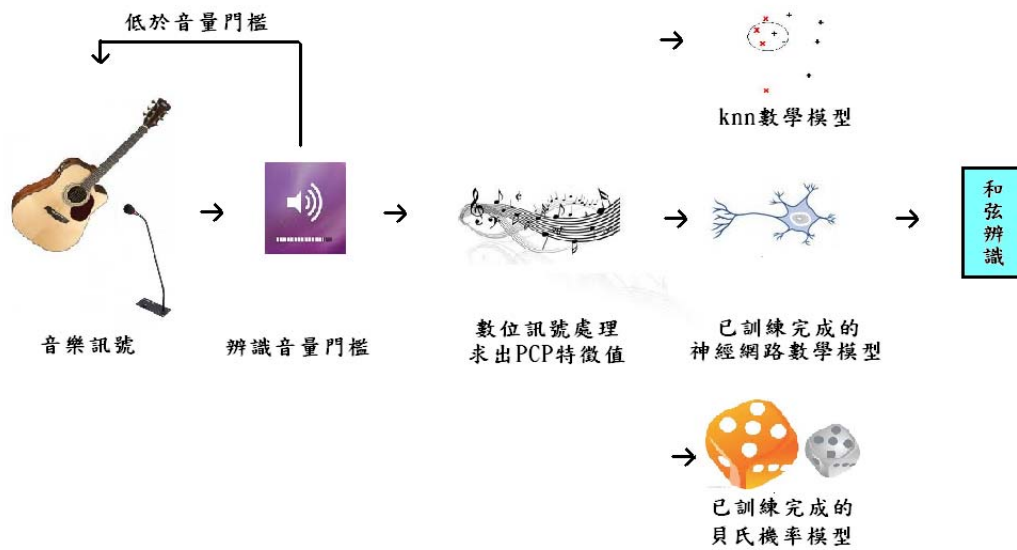


圖 5.6 實驗架構二

5.4 實驗測試流程

實驗架構一和實驗架構二將實驗結果，分成以下三個部份來進行討論：

第一部分-辨識率討論：探討三種分類器在本實驗中的辨識率

第二部分-誤判率討論：探討三種分類器在本實驗中的誤判率

第三部分-即時和弦彈奏辨識討論：討論三種分類器在即時和弦辨識時的辨識率

5.4.1 第一部分 辨識率討論

訓練資料經過電腦訓練完所產生的模型，將其進行回想和預測兩個步驟。

回想(Recalling)：將建立好的辨識模型，利用訓練資料集進行辨識率測試，若辨

識率不佳則代表訓練的模型需要重新架構；若辨識率很好，

也不全然代表於預測時其辨識率具最佳性。

預測(Prediction)：將建立好的辨識模型，利用測試資料集裡測試辨識率，若辨識不佳，則代表此數學模型本身不適合拿來做辨識系統，必須考慮重新學習；若辨識率很好，則代表此模型在面對未知到問題時能有較佳的預測能力，適合做為辨識的模型。

論及辨識之討論，則可分成整體辨識率的討論和各類和弦辨識率的討論：

1. **整體辨識率的討論**：將所有訓練/測試的資料集一併討論，可觀察整體辨識率

整體辨識率，表示方法如式 6.1 所示：


$$\text{整體辨識率} = \frac{\text{辨識成功的資料筆數}}{\text{整體資料筆數 (12500筆)}} \quad (6.1)$$

2. **各類和弦辨識率討論**：整體辨識率好，並不代表每類和弦辨識率就較佳，藉

由分析觀察來發現哪類和弦的辨識率較佳？哪種分類器辨識哪類和弦的辨識率較佳？

各類和弦之辨識率，表示方法如式 6.2 所示：

$$\text{各類和弦辨識率} = \frac{\text{各類和弦辨識成功的資料筆數}}{\text{各類和弦資料筆數}} \quad (6.2)$$

5.4.2 第二部分 誤判率討論

透過誤判討論，可以了解分類器將原本的資料誤判為何種資料，藉此可以修正分類器的數學模型，或修訂所給定的資料型態。

5.4.3 第三部分 即時和弦彈奏辨識討論

採用四組不同的和弦進行，每組和弦進行由七種不同的和弦組成，以拍速 60bpm，每小節四拍，一個和弦停留一小節，並以 0.5 秒辨識一次，每組和弦進行共費時 28 秒，並產生 56 個辨識結果。除此之外，每組和弦進行將分別使用四種樂器做測試，分別是兩種真實的木吉他，兩種虛擬音色。

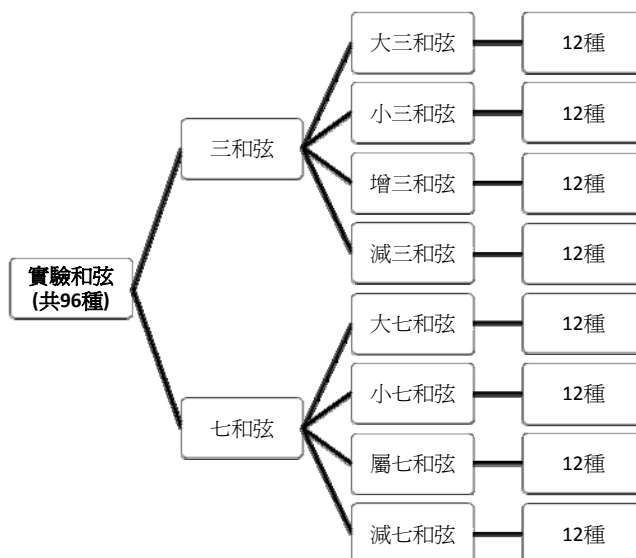




第六章 實驗結果與討論

6.1 訓練/預測 資料筆數統整

1. 本實驗使用如表 6.1 所示之 96 種不同和弦，進行辨識率測試。



2. 訓練和測試資料筆數各為 12500 筆，個別數目如下：

和弦類別	大三和弦	小三和弦	增三和弦	減三和弦	小七和弦	屬七和弦	大七和弦	減七和弦
資料筆數	2100	1900	1200	1200	1800	1900	1200	1200

3. 表 6.1 為各類和弦資料筆數與資料型態

檔案類型	聲道數	取樣頻率	取樣時間	資料型態
wav.	mono	22050kHz	3 sec	int 16

表 6.1 資料筆數與型態

訓練和測試資料筆數的資料分布長條圖與圓餅圖，如圖 6.1 與 6.2 所示：

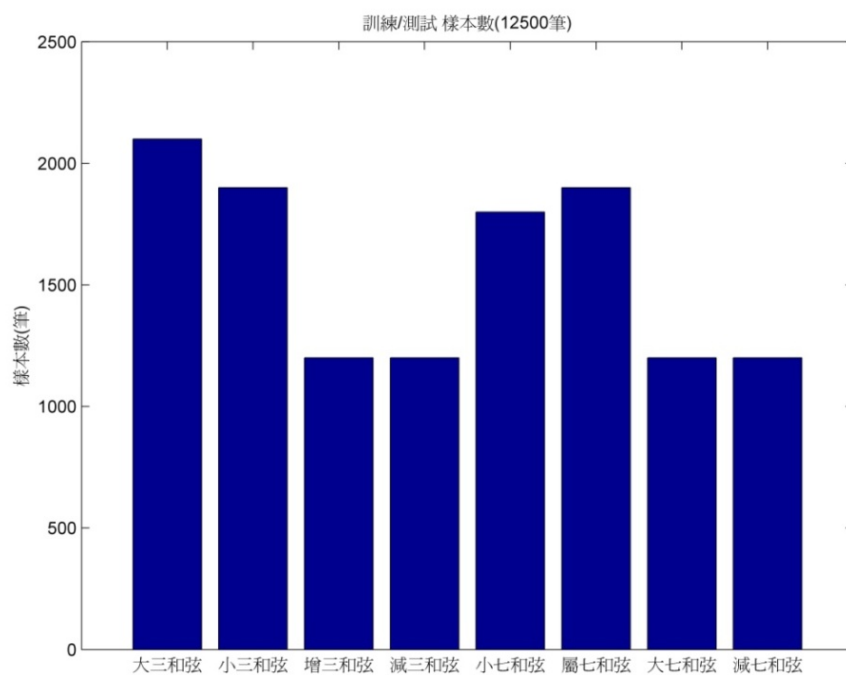


圖 6.1 和弦類別個別資料筆數

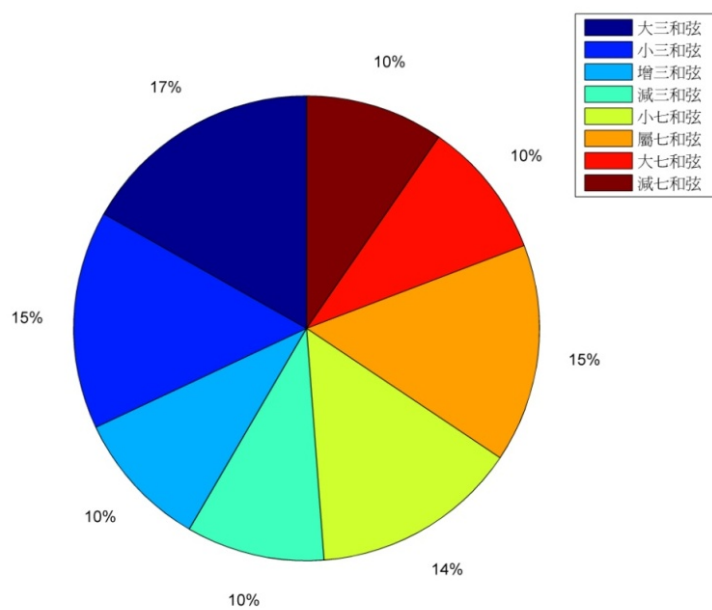


圖 6.2 和弦類別個別資料筆數佔有率

6.2 實驗結果與討論

一. Knn method

1. Knn - K 值與整體辨識率的影響

首先必須先找出 **K** 值對本實驗資料辨識率的影響，實驗從 **K=1~15**，利用試誤法觀察並找出最適合的 **K** 值，其實驗結果分別列於表 6.2 與圖 6.3:

K 值	1	2	3	4	5	6	7
整體辨識率(%)	75.46	72.64	72.33	70.82	70.24	69.2	68.88

K 值	8	9	10	11	12	13	14	15
整體辨識率(%)	67.9	67.26	66.62	66.04	65.9	95.35	65.04	64.59

表 6.2 K 值對整體辨識率的影響



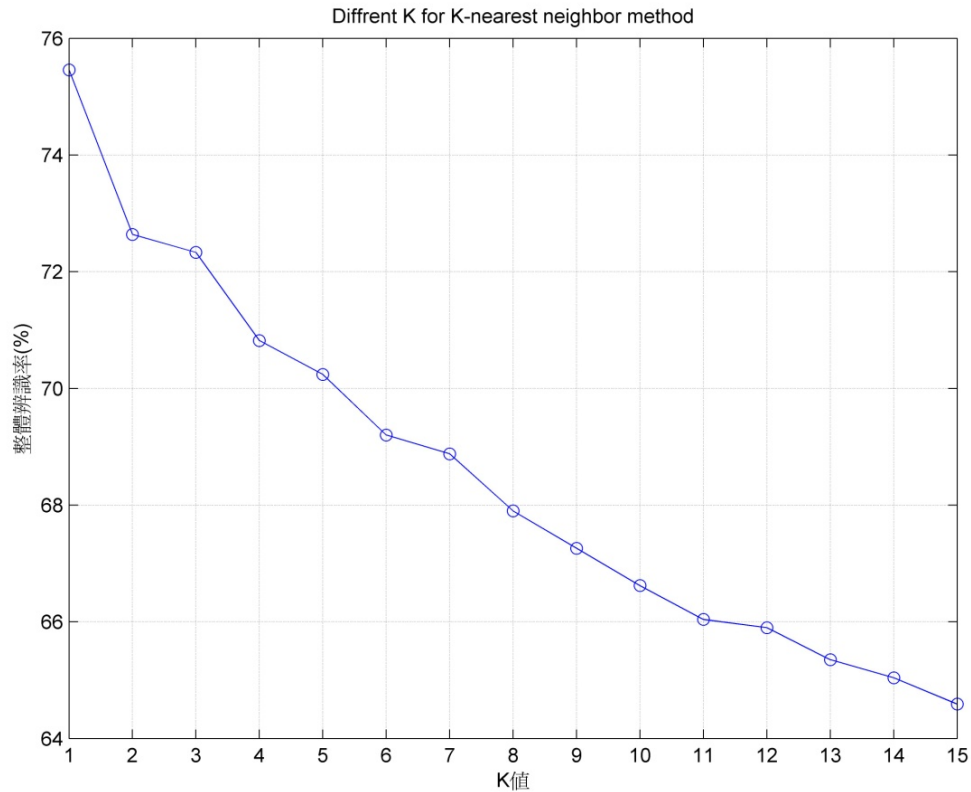


圖 6.3 K 值對整體辨識率的影響

●分析討論：

由圖 6.3 實驗數據可發現，使用 $K=1$ 時可以得到最高的整體辨識率，故本實驗的 Knn 分類器採用的 K 值為 1。

2. Knn 辨識率結果：

由於 Knn 是利用資料筆數之間的距離來判別歸類的群組，所以利用訓練的模型去測試訓練集(回想過程)則一定可以達到 100%的辨識率，所以我們要討論其預測過程(prediction)的辨識率才會有其意義。

將 12500 筆預測資料給 Knn 辨識分類，所得到的和弦辨識率如表 6.3 所示：

1-st Nearest Neighbor	大三和弦	小三和弦	增三和弦	減三和弦	屬七和弦	小七和弦	大七和弦	減七和弦
辨識率(%)	68.29	77.04	96.67	77.25	54.58	67.25	93.5	81.5

表 6.3 Knn 分類器各類和弦辨識率

●分析討論：

由表 6.3 觀察發現，使用 Knn 分類法辨識的結果，可得到總辨識率為 75.46%，從各類和弦辨識率的討論中，可發現辨識率較高的和弦類別為大七和弦和增三和弦；辨識率較低的和弦類別為大三和弦和屬七和弦。



二. 樸真貝氏分類器 Naïve Bayes Classifier

1. Naïve Bayes Classifier 辨識率結果

分別觀察樸實貝氏分類器的回想過程和預測過程的各類和弦辨識率，實驗結果如表 6.4 所示：

貝氏辨識率	大三和弦	小三和弦	增三和弦	減三和弦	屬七和弦	小七和弦	大七和弦	減七和弦
回想 (%)	96.83	94.88	99.92	98.58	92.75	94.63	98	97.25
預測 (%)	93.04	96.25	98.42	97.08	85.58	95.04	93	98.33

表 6.4 樸真貝氏分類器各類和弦辨識率

圖 6.4 係將表 6.4 畫成長條圖，所做之比較結果：

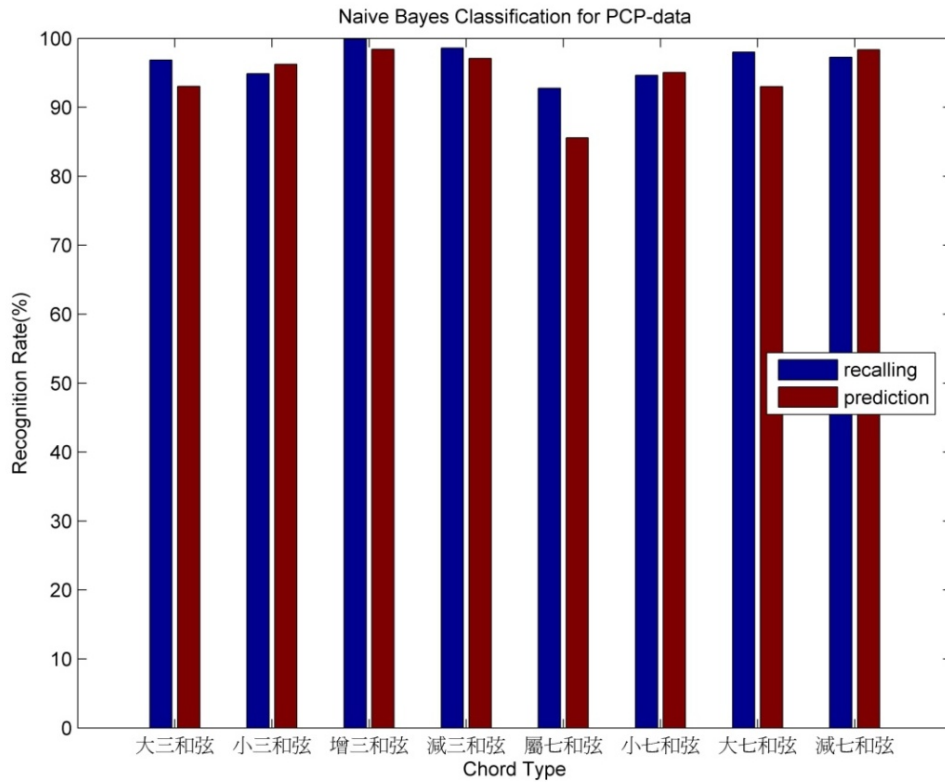


圖 6.4 樸真貝氏分類器回想/預測之辨識率比較

●分析討論：

由表 6.4 與圖 6.4 可發現：

1. 使用貝氏分類法，發現每個類別的和弦其預測過程中的辨識率會略高於回想過程中的辨識率。
2. 使用貝氏分類法回想辨識和預測辨識的結果，發現辨識率較高的和弦類別為增三和弦和減三和弦；辨識率較低的和弦類別為屬七和弦和小七和弦。



三. Neural Network 類神經演算法

使用類神經網路，需先給定基本的初始設定，其分別包含以下五項：

1. 輸入與輸出均為 12 個神經元
2. 兩層隱藏層，分別給定神經元數為 (n, n) ， $n = 11 \sim 21$ ，共 11 種組合，每種組合各測試三次。
3. 學習速率：0.01
4. 限制學習回合次數：7000 次內
5. 收斂目標：0.001

茲將類神經演算法之辨識結果，分述於下：

1. Recalling 回想

1.1 回想過程之總辨識率討論

表 6.5 為使用類神經演算法進行本實驗回想過程總辨識率的結果，共有 11 種不同排列組合的神經元，每種組合皆測試三次，並運算其辨識率的總平均。

隱藏層神經元個數		回想過程之總識別率(%)			
第一層	第二層	測試 1	測試 2	測試 3	總平均
11	11	98.94	98.86	98.83	98.88
12	12	99.00	98.98	98.95	98.98
13	13	98.90	98.96	98.83	98.90
14	14	99.11	99.02	99.08	99.07
15	15	99.10	99.05	99.10	99.08
16	16	98.97	99.07	99.06	99.03
17	17	99.11	98.90	99.26	99.09
18	18	98.93	99.13	99.10	99.05

19	19	99.23	99.15	99.07	99.15
20	20	99.22	99.02	99.03	99.09
21	21	98.99	99.17	99.22	99.13

表 6.5 不同排列組合的神經元對回想過程中總辨識率的影響

●分析討論：

由上表 6.5 可發現

1. 總辨識率就單次測試而言，以神經元組合(17, 17)的第三次測試辨識率最高(99.26%)；以神經元組合(11, 11)的第三次辨識率最低(98.83%)。

2. 總辨識率就三次測試平均而言，以神經元組合(19, 19)的平均測試辨識率最高(99.15%)；以神經元組合(11, 11)的平均測試辨識率最低(98.88%)。

1.2 回想過程中之各類和弦個別辨識率討論

上述討論回想過程中的總辨識率，接下來討論回想過程中，各類和弦之個別辨識率，其結果列於表 6.6。

表 6.6 為回想過程中各類和弦的辨識率：

隱藏層 神經元個數	11, 11 (平均值)	12, 12 (平均值)	13, 13 (平均值)	14, 14 (平均值)	15, 15 (平均值)	16, 16 (平均值)
大三和弦	99.68	99.55	99.70	99.62	99.51	99.44
小三和弦	99.74	99.47	99.67	99.35	99.51	99.42
增三和弦	99.64	99.52	99.58	99.56	99.83	99.58
減三和弦	98.17	99	99.03	98.42	98.50	98.47
大七和弦	96.07	98.74	96.44	98.93	99.20	98.94
小七和弦	99.67	99.12	99.52	99.37	99.31	99.28

屬七和弦	99.09	97.87	98.47	98.35	97.95	98.05
減七和弦	99.31	98.97	99.44	99.19	99.25	99.56

隱藏層 神經元個數	17, 17 (平均值)	18, 18 (平均值)	19, 19 (平均值)	20, 20 (平均值)	21, 21 (平均值)
大三和弦	99.60	99.63	99.71	99.56	99.62
小三和弦	99.39	99.56	99.47	99.32	99.47
增三和弦	99.89	99.75	99.81	99.92	99.83
減三和弦	99.22	99.08	98.50	98.44	99.17
大七和弦	97.98	98.07	99.13	98.78	98.89
小七和弦	99.31	99.26	99.33	99.33	99.19
屬七和弦	98.60	98.23	98.19	98.28	97.86
減七和弦	99.33	99.44	99.33	99.58	99.69

表 6.6 回想過程中各類和弦的辨識率

●分析討論：

由表 6.6 之實驗結果可發現：

1. 根據表 6.6 之各類和弦在回想過程中的三次平均辨識率，可發現不同的神經元組合會使各類和弦的辨識率略有不同。
2. 另由觀察發現，在回想過程中，大體而言，大三和弦和增三和弦的辨識率較高；
大七和弦和屬七和弦的辨識率則相對較低。

1.3 回想過程中辨識率最高與最低的神經網路之收斂情形

類神經網路學習的好壞會決定辨識率的優劣，可由收斂情形觀察其收斂的狀

況，由表 6.5 得知使用神經元(17, 17)的組合，在第三次測試時可達到回想之辨識率 99.26%，為所有神經元組合測試中，回想辨識率最高者；使用神經元(11, 11)的組合，在第三次測試時回想之辨識率為 98.83%，為所有神經元組合測試中辨識率最低者。圖 6.5 顯示出在回想過程中，辨識率最高(99.26%)與最低(98.83%)的類神經網路之學習收斂情形。

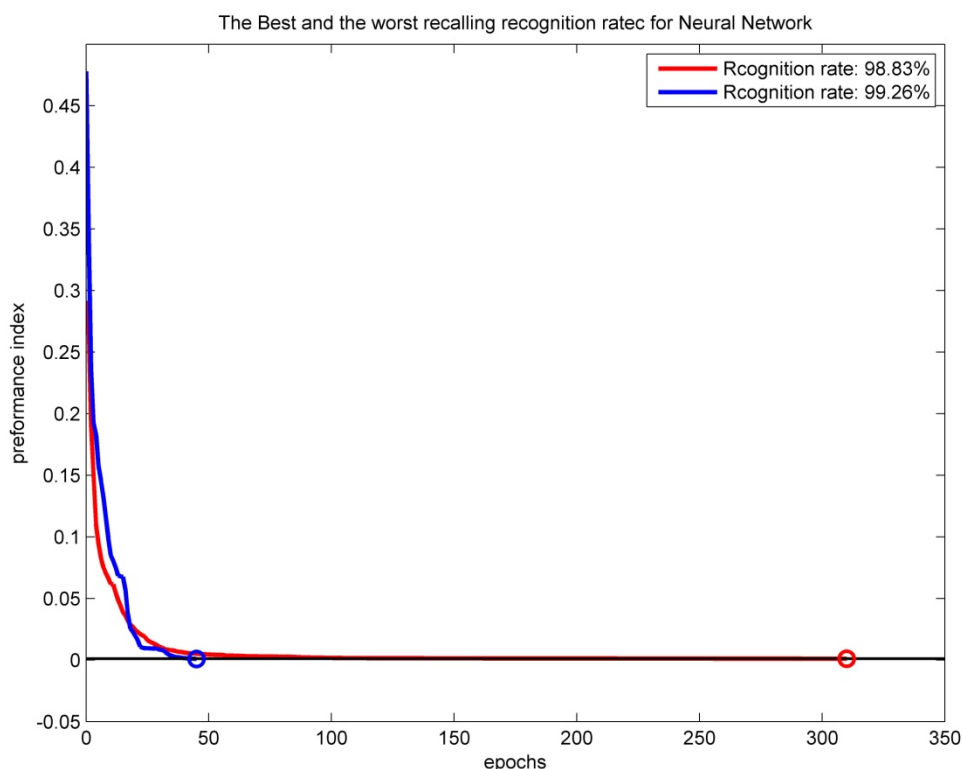


圖 6.5 回想過程中辨識率最高與最低的學習回合次數

●分析討論：

由圖 6.5 發現，辨識率最高的類神經網路模型，其收斂速度快，僅約 50 次就達收斂；辨識率最低的類神經網路模型，其收斂速度在尾端時極慢，超過了 300 次後才收斂至 0.001。

2. Prediction 預測

2.1 預測過程之總辨識率討論

與回想過程相同，將訓練好的模型用到測試資料集裡進行預測，並觀察其不同排列組合的神經元之預測過程中的總辨識率，表 6. 7 為不同排列組合的神經元對預測過程中總辨識率的影響：

隱藏層神經元個數		預測過程之總識別率(%)			
第一層	第二層	測試 1	測試 2	測試 3	總平均
11	11	94.05	95.11	95.16	94.77
12	12	95.69	95.08	94.79	95.19
13	13	95.46	93.57	94.93	94.65
14	14	94.27	94.35	95.26	94.63
15	15	90.32	94.14	95.89	93.45
16	16	94.75	94.34	94.89	94.66
17	17	94.70	95.08	95.68	95.15
18	18	93.88	94.98	93.28	94.05
19	19	95.14	94.65	94.37	94.72
20	20	94.56	92.18	94.07	93.60
21	21	91.48	94.42	95.01	93.64

表 6.7 不同排列組合的神經元對預測過程中總辨識率的影響

●分析討論：

由表 6.7 實驗結果可發現：

1. 總辨識率就單次測試而言，以神經元組合(15, 15)的第三次測試辨識率最高(95.89%)；以神經元組合(21, 21)的第一次辨識率最低(91.48%)。

2. 總辨識率就三次測試平均而言，以神經元組合(12, 12)的平均測試辨識率最高(95.19%)；以神經元組合(15, 15)的平均測試辨識率最低(93.45%)。

2.2 預測過程中之各類和弦個別辨識率討論

完成預測過程中的總辨識率討論，繼而討論預測過程中，各類和弦之個別辨識率，茲將其結果，分列於表 6.8。

表 6.8 為預測過程中各類和弦的辨識率：

隱藏層 神經元個數	11, 11 (平均值)	12, 12 (平均值)	13, 13 (平均值)	14, 14 (平均值)	15, 15 (平均值)	16, 16 (平均值)
大三和弦	98.10	97.40	96.95	96.73	96.76	96.06
小三和弦	97.65	97.75	97.51	97.46	97.77	98.32
增三和弦	95.81	97	96.61	96.08	97.75	97.03
減三和弦	94.03	94.17	94.53	94.78	94.92	94.44
大七和弦	90.78	95.24	92.65	93.26	96.06	94.65
小七和弦	94.20	95.31	94.72	95.02	93.65	95.17
屬七和弦	93.26	90.32	90.86	91.16	90.79	88.21
減七和弦	95.97	96.31	95.86	94.50	95.81	96.42

隱藏層 神經元個數	17, 17 (平均值)	18, 18 (平均值)	19, 19 (平均值)	20, 20 (平均值)	21, 21 (平均值)
大三和弦	95.78	96.16	96.27	95.94	95.14
小三和弦	98.32	98.16	98.07	96.89	96.61
增三和弦	96.97	96.33	97	96.42	97.17
減三和弦	94.44	94.25	93.61	93.19	92.75

大七和弦	94.83	94.28	95.85	95.5	94.20
小七和弦	95.54	92.63	92.93	91.65	93.04
屬七和弦	91.26	88.19	90.12	87.65	87.77
減七和弦	99.33	99.44	99.33	99.58	99.69

表 6.8 預測過程中各類和弦的辨識率

●分析討論：

表 6.8 為各類和弦在預測過程中的三次平均辨識率，由觀察發現，在預測過程中，小三和弦和減七和弦的辨識率較高；屬七和弦的辨識率相對較低。



2.3 預測過程中辨識率最高與最低的神經網路之收斂情形

由表 6.7 得知使用神經元(15, 15)的組合，在第三次測試時可達到預測之辨識率 95.89%，為所有神經元組合測試中，預測辨識率最高者；使用神經元(21, 21)的組合，在第一次測試時預測之辨識率為 91.48%，為所有神經元組合測試中，預測辨識率最低者。

圖 6.6 為預測能力最好(95.89%)與最壞(91.48%)的類神經網路之學習收斂情形。

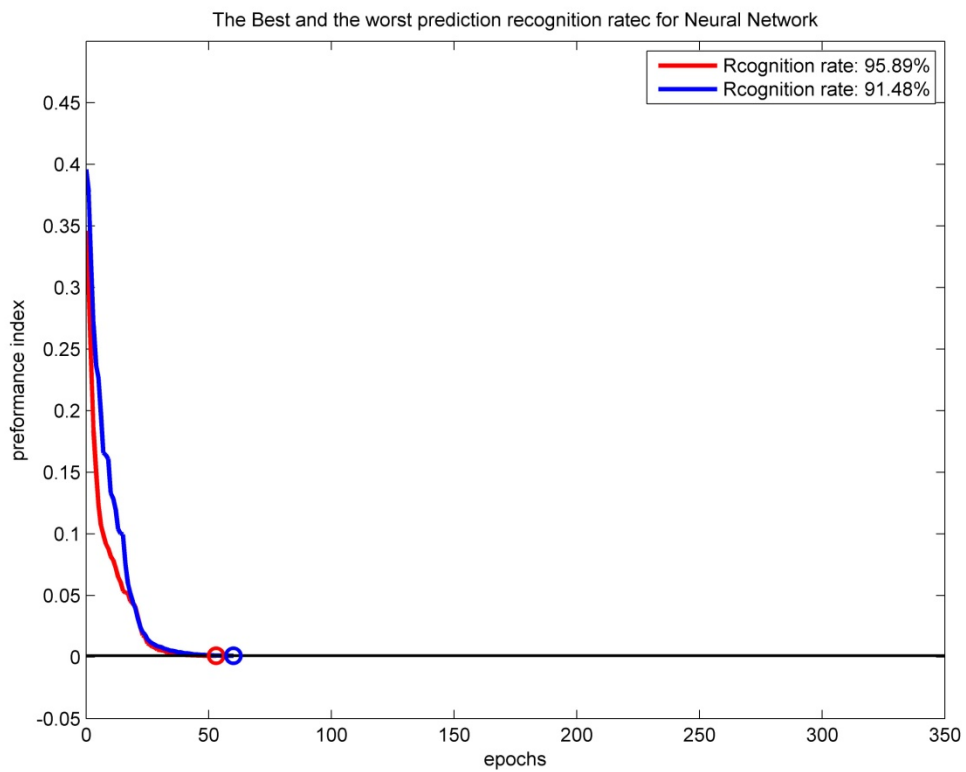


圖 6.6 預測過程中辨識率最高與最低的學習回合次數

●分析討論：

由圖 6.6 可以發現，預測能力較佳的類神經模型，其收斂曲線較為平緩；相對於預測能力差的類神經模型，其收斂曲線則略為不平順。

圖 6.7 係將不同神經元組合之回想和預測過程的總辨識率，畫成長條圖以方便比較。

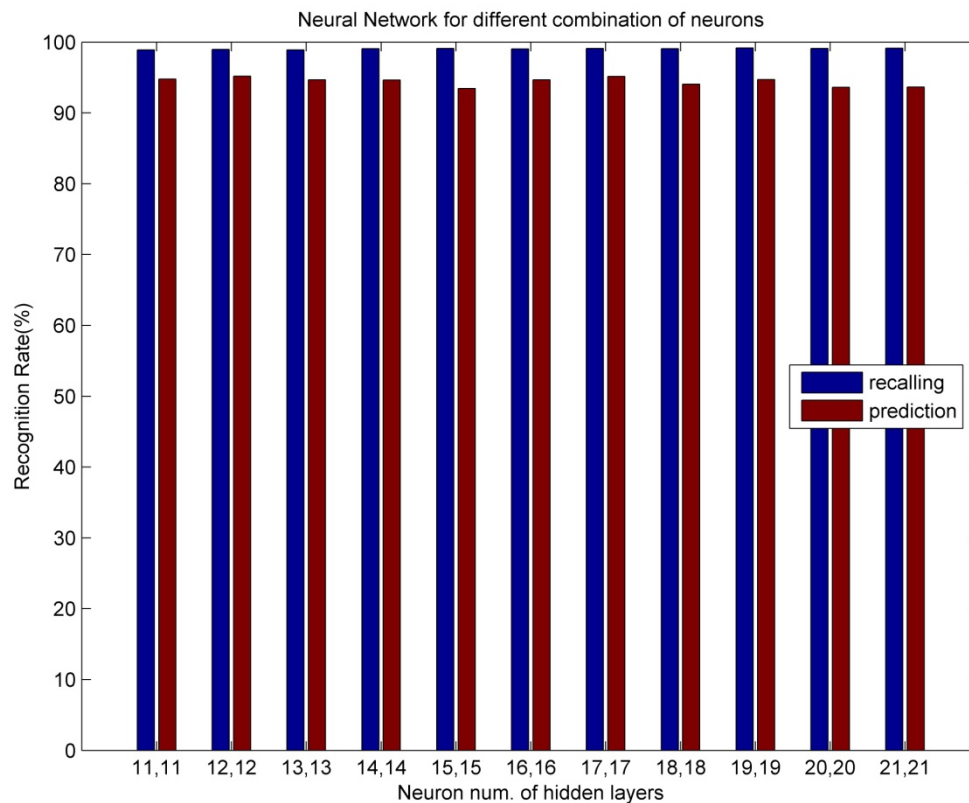


圖 6.7 不同神經元組合之回想\預測總辨識率之比較

●分析討論：

由圖 6.7 可以發現，利用類神經演算法來進行和弦辨識時，回想過程的總辨識率會高過於預測過程的總辨識率。然而神經元不同排列組合的搭配對總辨識率的影響則沒有明顯的規則，此部分需靠試誤法，方可知道最佳組合解。

3. 類神經學習的收斂情況與次數討論

使用類神經演算法學習，往往不同之隨機初始權重設定，會產生不同的收斂情況。若到達收斂以前所學習次數越多，則表示所花的時間越長；收斂的曲線愈平滑代表學習狀況穩定。然而收斂次數的多寡與訓練模型的好壞，沒有絕對的關係。透過尋找學習回合之次數少，且訓練模型完整的神經元組合，可讓我們能更快速的完成初始權重和所需的神經元個數之設定。表 6.9 為 11 種不同排列組合的神經元在三次測試時所需的學習回合次數：

隱藏層神經元個數		到達收斂次數時學習回和次數(epoch)			
第一層	第二層	測試 1	測試 2	測試 3	總平均
11	11	251	219	310	260
12	12	52	65	76	64.3
13	13	72	182	269	174.3
14	14	61	59	150	90
15	15	52	83	53	62.7
16	16	89	100	69	86
17	17	64	99	45	69.3
18	18	104	59	86	83
19	19	60	219	69	116
20	20	52	104	52	69.3
21	21	60	69	60	63

表 6.9 不同排列組合的神經元之神經網路對學習回合次數的影響

●分析討論：

由表 6.9 可發現，不同排列組合的神經元之神經網路與學習回合次數，並無太絕對的關聯性，然而可以發現神經元組合為(11, 11)的類神經網路，需要花費較多的學習回合次數(平均 260 次)才能到達收斂；而神經元組合為(21, 21)則花費較少的學習回合次數(平均 63 次)即可收斂。顯然神經元越多，越能處理複雜的情況且費時較短。將表 6.9 的數據畫成長條圖所作之比較，如下圖 6.8：

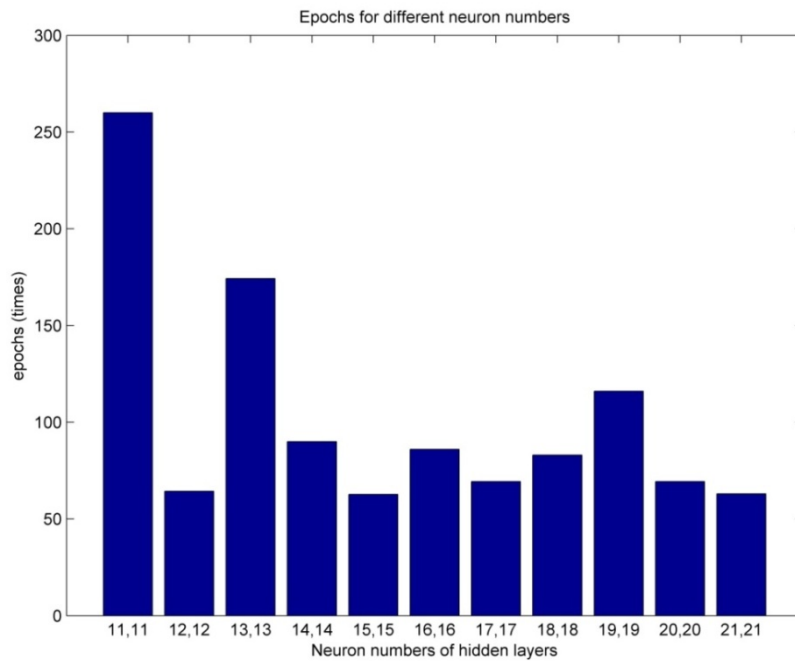


圖 6.8 不同組合之神經元的平均學習回合次數

圖 6.9 為所有測試中，學習回合次數最長(310 次)的類神經網路(神經元組合-11, 11 的第三次測試)和學習回合次數最短(45 次)的類神經網路(神經元組合-17, 17 的第三次測試)之收斂情形。

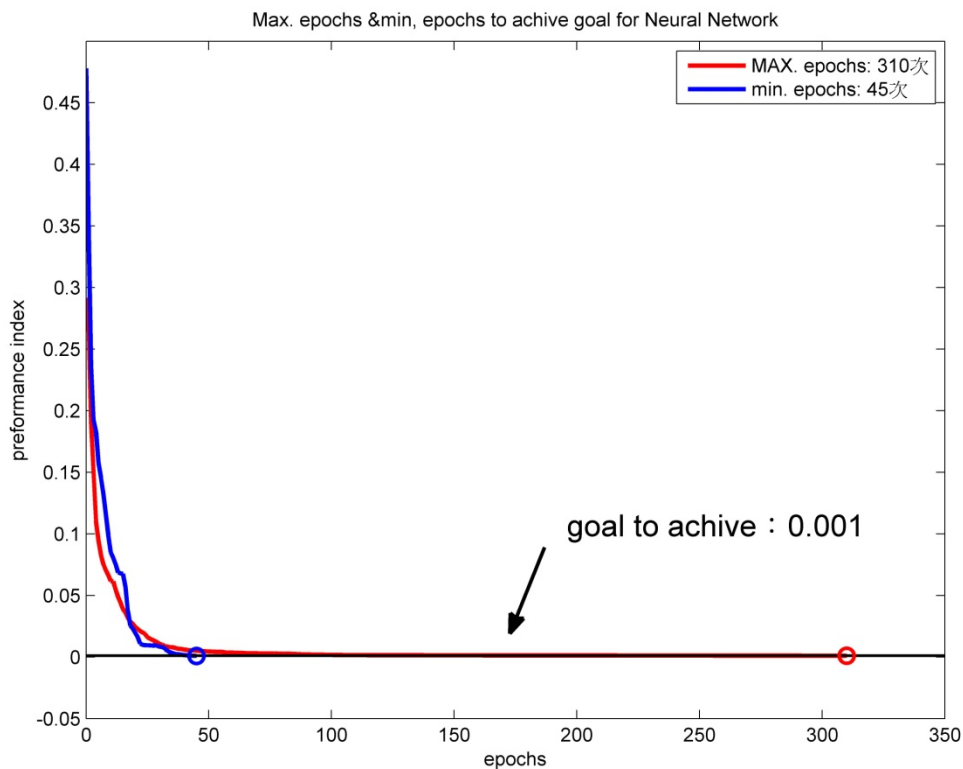


圖 6.9 最多學習回合次數與最少學習回合次數的收斂情形



6.2.2 第二部分 誤判率討論

當分類器對和弦歸類無法做正確類別歸屬時，就會造成誤判之發生。誤判的嚴重與否，除了和本身使用分類器是否具有高容錯性之外，也與本身給定的訓練資料之好壞有關係。以下係根據誤判表，來觀察三種分類法的誤判情形。

一. Knn 誤判討論

表 6.10 為使用 Knn 演算法進行預測過程時，所造成的誤判情形。

預測過程誤判 (單位：%)

誤判 資料	大三 和弦	小三 和弦	增三 和弦	減三 和弦	大七 和弦	小七 和弦	屬七 和弦	減七 和弦
大三		13.79	0.14	0.28	1.58	1.32	13.98	0.40
小三	12.79		0.125	0.83	0.125	5.29	3.625	0.125
增三	0.25	0.1108			0.8325	0.555		0.083
減三	0.25	0.166	8.416		0.083	0.83	4.416	8.583
大七	1.91	0.66	0.833	0.416	0.166	0.25	0.916	1.33
小七	5.625	13.375	0.33	0.583	0.041		12.5	0.291
屬七	2.408	9.041	3		0.458	7.79	0.042	1
減七	0.33	0.042	0.042	3.083			1.083	0.042

表 6.10 Knn 分類器在預測過程中的誤判情形

●分析討論：

由表 6.10 中，從 Knn 分類器之預測過程中誤判結果來分析可發現：

原先資料	最容易誤判成	原因
大三和弦	小三和弦	與大三和弦的第一組成音和第三組成音相同
	屬七和弦	與大三和弦前三個組成音相同，僅多了第四個組成音
小三和弦	大三和弦	與小三和弦的第一組成音和第三組成音相同
增三和弦	大七和弦	與增三和弦第一組成音和第二組成音相同

減三和弦	減七和弦	與減三和弦的前三個組成音皆相同，只多了第四個組成音
大七和弦	大三和弦	與大七和弦的前三個組成音皆相同，只少了第四個組成音
小七和弦	小三和弦	與小七和弦的前三個組成音皆相同，只少了第四個組成音
	屬七和弦	與小七和弦的第一、三和四個組成音相同
屬七和弦	小三和弦	純粹誤判
減七和弦	減三和弦	與減七和弦的前三個組成音皆相同，只少了第四個組成音

二. Naive Bayes 誤判討論

表 6.11 為使用 Naïve Bayes 演算法進行回想過程時，所造成的誤判情形：

回想過程誤判（單位：%）

誤判 資料	大三	小三	增三	減三	大七	小七	屬七	減七
大三		2.49		0.35	0.5		0.04	0.04
小三	3.96					0.29	0.83	0.04
增三								0.08
減三				0.92				0.5

大七		0.92				0.5	0.583	
小七	0.29	0.54	0.25		0.13		3.79	
屬七	4.29	0.66		0.79		1.33		
減七				2.66			0.16	

表 6.11 Naïve Bayes 分類器在回想過程中的誤判情形

表 6.12 為使用 Naïve Bayes 演算法進行預測過程時，所造成的誤判情形：

預測過程誤判（單位：%）

誤判 資料	大三	小三	增三	減三	大七	小七	屬七	減七
大三	0.125	3.305	0.208	0.125	0.166	0.291	2.26	0.388
小三	1.33				1.25	0.5	0.416	0.25
增三	0.75							0.83
減三				1.16		0.083	0.25	1.416
大七	1.5	1.83	0.25		0.166	0.583	0.5	2.16
小七	1.208	0.083	0.33		0.125		2.33	0.125
屬七	7.08	2.83	0.166		0.0416	3.75	0.0416	0.5
減七				1.25			0.104	

表 6.12 Naïve Bayes 分類器在預測過程中的誤判情形

●分析討論：

從表 6.11 與上表 6.12 之 Naïve Bayes 分類器之回想過程和預測過程中的誤判結果來分析可發現以下之綜合結果：

原先資料	最容易誤判成	原因
大三和弦	小三和弦	與大三和弦的第一組成音和第三組成音相同
小三和弦	大三和弦	與小三和弦的第一組成音和第三組成音相同
屬七和弦	大三和弦	與屬七和弦前三個組成音相同，僅少了第四個組成音
小七和弦	屬七和弦	與小七和弦的第一、三和四個組成音相同
減七和弦	減三和弦	與減七和弦的前三個組成音皆相同，只少了第四個組成音

三. Neural Network 誤判討論

此節著重於討論預測過程時總辨識率最高的數學模型-亦即討論神經元組合為(15, 15)的第三次測試，此神經網路預測之整體辨識率為 95.89%。類神經網路在學習時所給定的目標函數是和弦所對應的二進位 PCP，若預測所得結果之 PCP 未在 96 種和弦之內，則將其和弦定義為無對應資料。

回想過程誤判 (單位：%)

表 6.13 為使用 Neural Network 演算法進行回想過程時，所造成的誤判情形

誤判 資料	大三	小三	增三	減三	大七	小七	屬七	減七	無對應 資料
大三		0.14				0.05	0.10		0.14

小三	0.16						0.05		0.26
增三									0.08
減三				0.083			0.17	0.42	0.75
大七	0.42	0.83				0.83			
小七	0.05	0.67					0.05		
屬七	1.31			0.37					0.16
減七				0.67			0.08		0.16

表 6.13 Neural Network 在回想過程中的誤判情形

預測過程誤判 (單位：%)

表 6.14 為使用 Neural Network 演算法進行預測過程時，所造成的誤判情形

誤判 資料	大三	小三	增三	減三	大七	小七	屬七	減七	無對應 資料
大三		0.43			1.19	0.10	0.81		1.33
小三	0.32				0.11	0.42			1.10
增三						0.08			0.5
減三	0.17			0.08			1.33	1.17	3.25
大七	1.75	1.25			0.17	0.08			0.58
小七	0.11	2.89		0.05	0.05		0.11		1.05
屬七	4.74	0.95		0.05		1.05	0.05		1.21
減七				1.5			1		0.92

表 6.14 Neural Network 在預測過程中的誤判情形

●分析討論：

從表 6.13 與表 6.14 之 Neural Network 分類器之回想和預測過程中的誤判結果來分析可發現以下之綜合結果：

原先資料	最容易誤判成	原因
大三和弦	小三和弦	與大三和弦的第一組成音和第三組成音相同
	大七和弦	與大七和弦前三個組成音相同，僅少了第四個組成音
小三和弦	大三和弦	與小三和弦的第一組成音和第三組成音相同
大七和弦	大三和弦	與大三和弦前三個組成音相同，僅多了第四個組成音
屬七和弦	大三和弦	與屬七和弦前三個組成音相同，僅少了第四個組成音
小七和弦	小三和弦	與小三和弦前三個組成音相同，僅多了第四個組成音
減七和弦	減三和弦	與減七和弦的前三個組成音皆相同，只少了第四個組成音

四. 三個分類器的誤判比較

圖 6.10 至圖 6.13 分別為三種分類器的誤判比較，由此可明顯發現 Knn 分類器的誤判率最高，其次是貝氏分類器，誤判率最低者則是類神經網路。故此可明確論斷，類神經網路在本實驗中具有高準確度之預測能力且容錯性高，即使進行測試的資料集裡，可能存在無法預期的雜訊，類神經網路仍具有很高的辨識率。

由圖 6.10(上)發現，Knn 演算法最容易將大三和弦誤判成小三和弦和屬七和弦；Naïve Bayes 演算法最容易將大三和弦誤判成小三和弦和屬七和弦；Neural Network 演算法最容易將大三和弦誤判成大七和弦和屬七和弦。

由圖 6.10(下)發現，Knn 演算法最容易將小三和弦誤判成大三和弦，小七和弦和屬七和弦；Naïve Bayes 演算法最容易將小三和弦誤判成大三和弦和大七和弦；Neural Network 演算法最容易將小三和弦誤判成大三和弦。

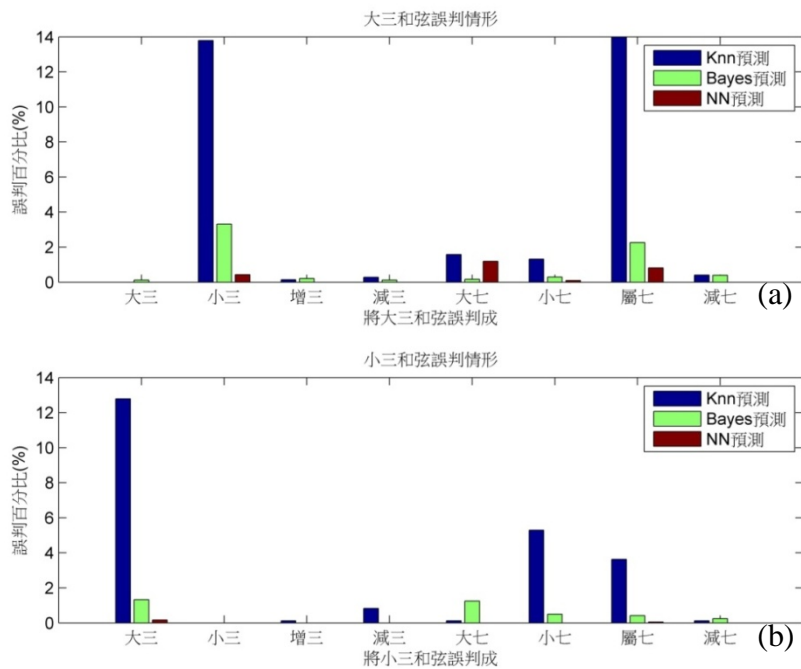


圖 6.10 三種分類器對大三和弦和小三和弦之誤判情形比較

由圖 6.11(上)發現, Knn 演算法最容易將增三和弦誤判成大七和弦和小七和弦;
Naïve Bayes 演算法最容易將增三和弦誤判成大三和弦和減七和弦。

由圖 6.11(下)發現, Knn 演算法最容易將減三和弦誤判成增三和弦和減七和弦;
Naïve Bayes 演算法最容易將減三和弦誤判成減三和弦和減七和弦; Neural
Network 演算法最容易將減三和弦誤判成屬七和弦和減七和弦。

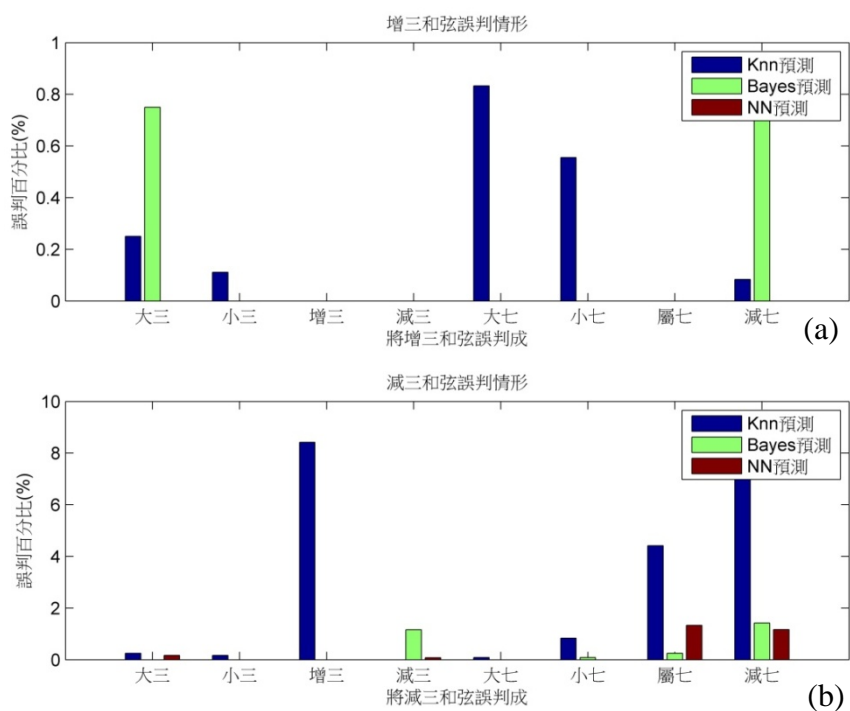


圖 6.11 三種分類器對增三和弦和減三和弦之誤判情形比較

由圖 6.12(上)發現, Knn 演算法最容易將大七和弦誤判成大三和弦和減七和弦;
Naïve Bayes 演算法最容易將大七和弦誤判成小三和弦和減七和弦; Neural
Network 演算法最容易將大七和弦誤判成小三和弦和小七和弦。

由圖 6.12(下)發現, Knn 演算法最容易將小七和弦誤判成小三和弦和屬七和弦;
Naïve Bayes 演算法最容易將小七和弦誤判成大三和弦和屬七和弦; Neural
Network 演算法最容易將小七和弦誤判成小三和弦。

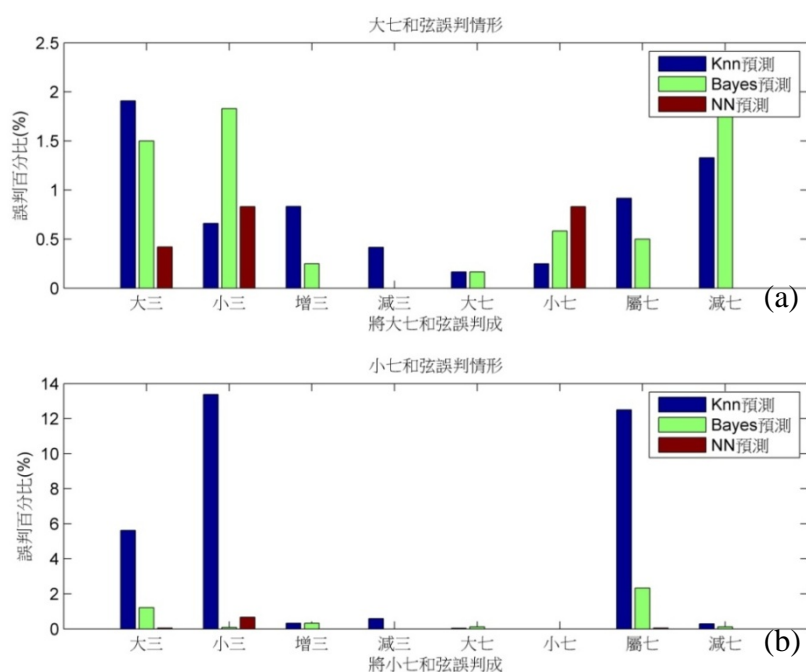


圖 6.12 三種分類器對大七和弦和小七和弦之誤判情形比較

由圖 6.13(上)發現, Knn 演算法最容易將屬七和弦誤判成小三和弦和小七和弦;
Naïve Bayes 演算法最容易將屬七和弦誤判成大三和弦和小七和弦; Neural
Network 演算法最容易將屬七和弦誤判成大三和弦和減三和弦。

由圖 6.13(下)發現, Knn 演算法最容易將減七和弦誤判成減三和弦和屬七和弦;
 Naïve Bayes 演算法最容易將減七和弦誤判成減三和弦和屬七和弦; Neural
 Network 演算法最容易將減七和弦誤判成減三和弦。

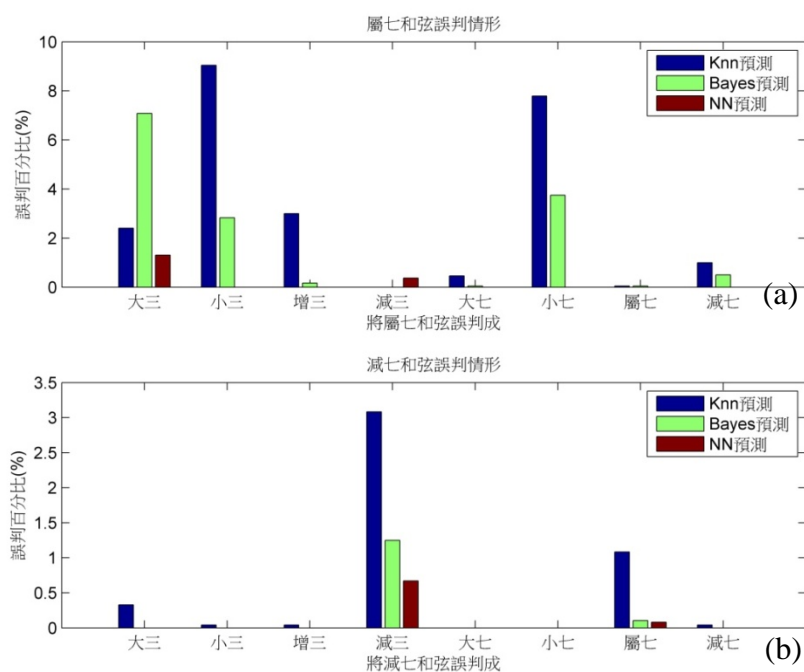


圖 6.13 三種分類器對屬七和弦和減七和弦之誤判情形比較

第三部分 即時和弦彈奏辨識討論

圖 6.14 顯示和弦之實驗進行現狀，其係以每分鐘 60 拍(60bpm)的拍速，每個和弦進行一小節，一小節停留四拍(全音符)。一組和弦進行，係由 7 個不同的和弦所構成。

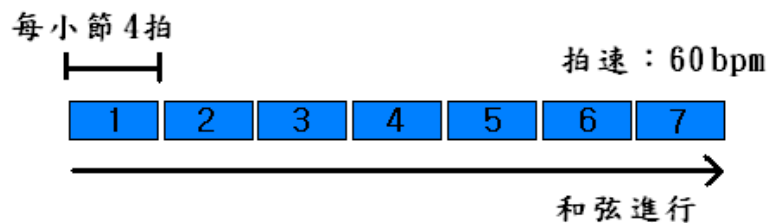


圖 6.14 和弦進行

測試方法：

使用兩把真實的民謠吉他和由 midi 訊號產生的兩種不同樂器之虛擬音色，分別使用三種辨識方法進行測試，以測試其三種分類器的抗噪性和廣用度。

實驗時每 0.5 秒辨識一次，由於每個和弦停留四拍，亦即四秒，所以一次完整的和弦進行會維持 28 秒，並會辨識產生 56 個結果。

測試樂器，共四種樂器如下：

GT1：guitar 1 (原先拿來建立 PCP 資料庫的吉他)

GT2：guitar 2 (民謠吉他)

MIDI：古典尼龍弦吉他(Nylon Guitar) 虛擬音源 由喇叭放出

MIDI：古典鋼琴(Classical Piano) 虛擬音源 由喇叭放出

和弦進行，共四組搭配如下：

進行 I：C→Dm→Em→F→G→Am→Bdim (大調順階三和弦)

進行 II：Cm→Ddim→D#→Fm→Gm→G#→A# (小調順階三和弦)

進行 III：Cmaj7→Dm7→Em7→Fmaj7→G7→Am7→Bdim7 (大調順階七和弦)

進行 IV：Cm7→Ddim7→Dmaj7#→Fm7→Gm7→Gmaj7#→A7# (小調順階七和弦)

圖 6.15 為和弦進行即時辨識所產生出來的示意圖：

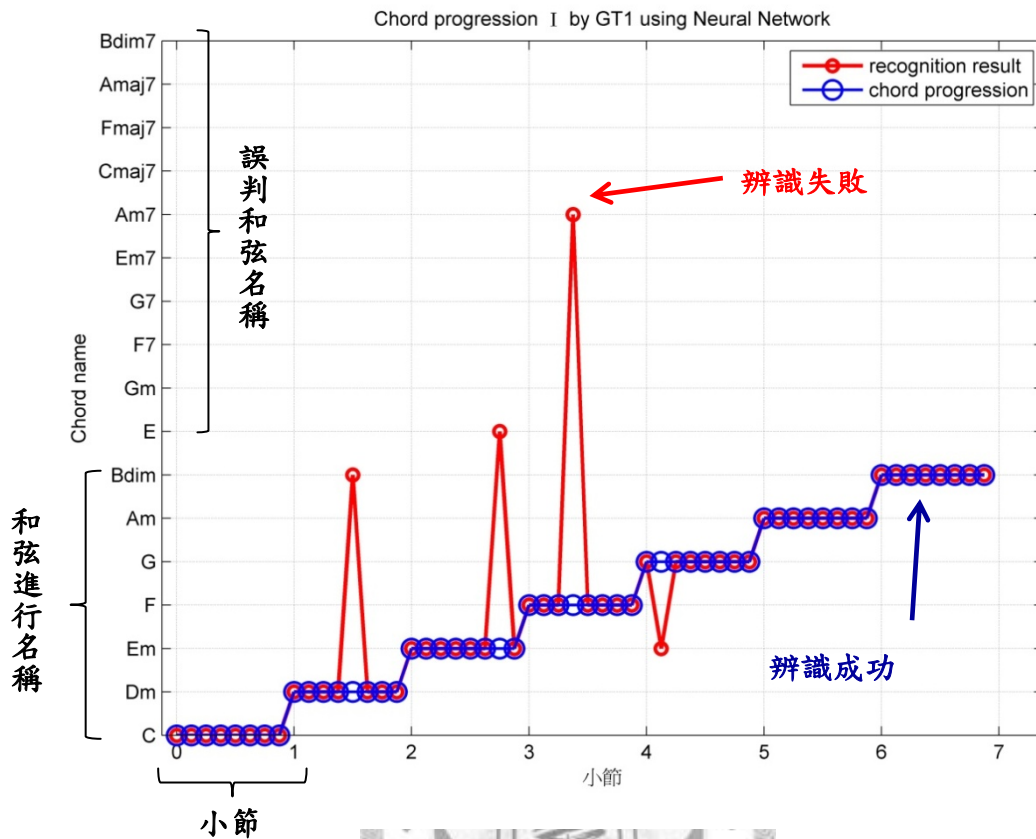


圖 6.15 和弦進行即時辨識率討論示意圖

由圖 6.15 所示：

藍圈：代表和弦進行

紅圈：代表辨識和弦結果

1. 和弦進行 I : C→Dm→Em→F→G→Am→Bdim

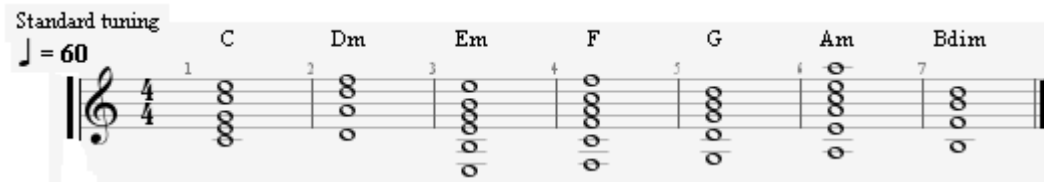


表 6.16 所測試的樂器為 GT1，圖 6.16.1 至圖 6.16.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：GT1（和弦進行 I）	
<p>Neural Network (GT1)</p> <p style="text-align: center;">圖 6.16.1 和弦進行 I -GT1(NN)</p>	<p>1. 和弦進行之整體 辨識率：92.85%</p> <p>2. 和弦最高辨識率： C/Am/Bdim (100%)</p> <p>3. 和弦最低辨識率： Dm/Em/F/G (87.5%)</p>
<p>Naïve Bayes Classifier (GT1)</p>	<p>1. 和弦進行之整體</p>

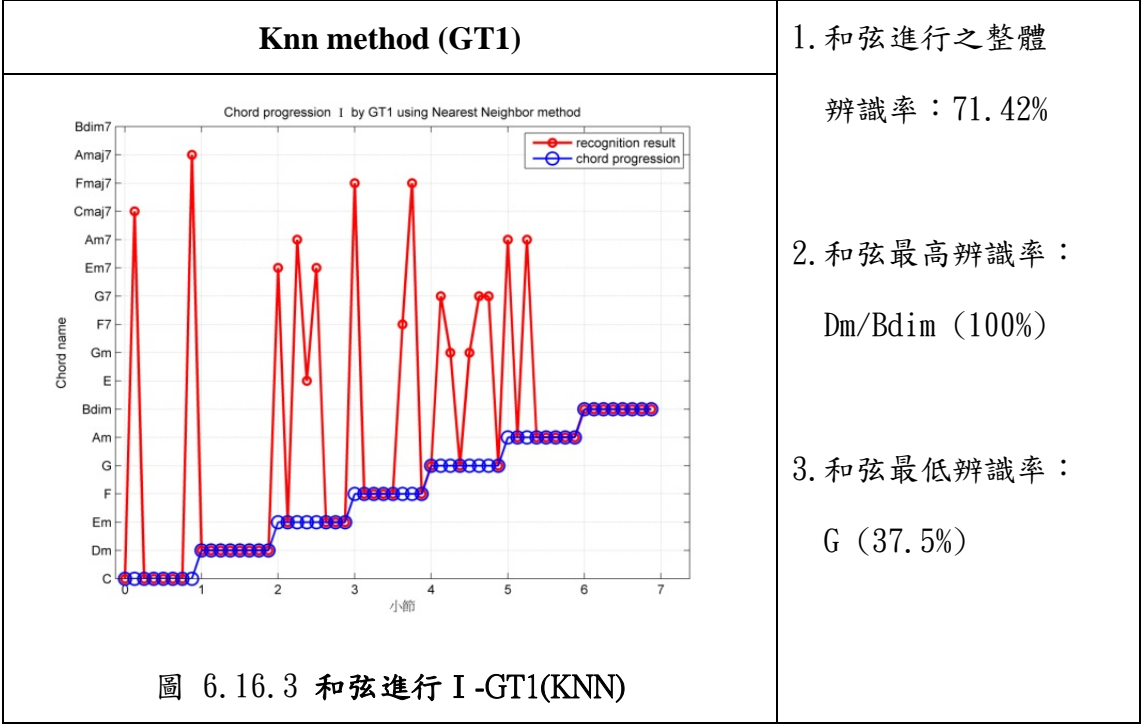
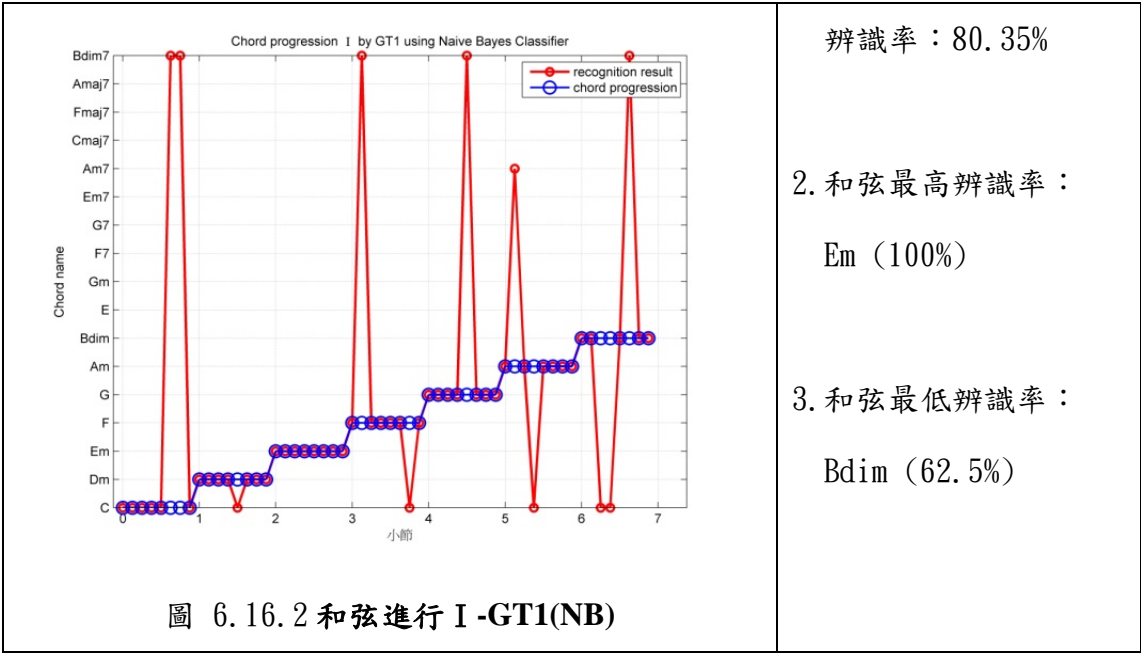
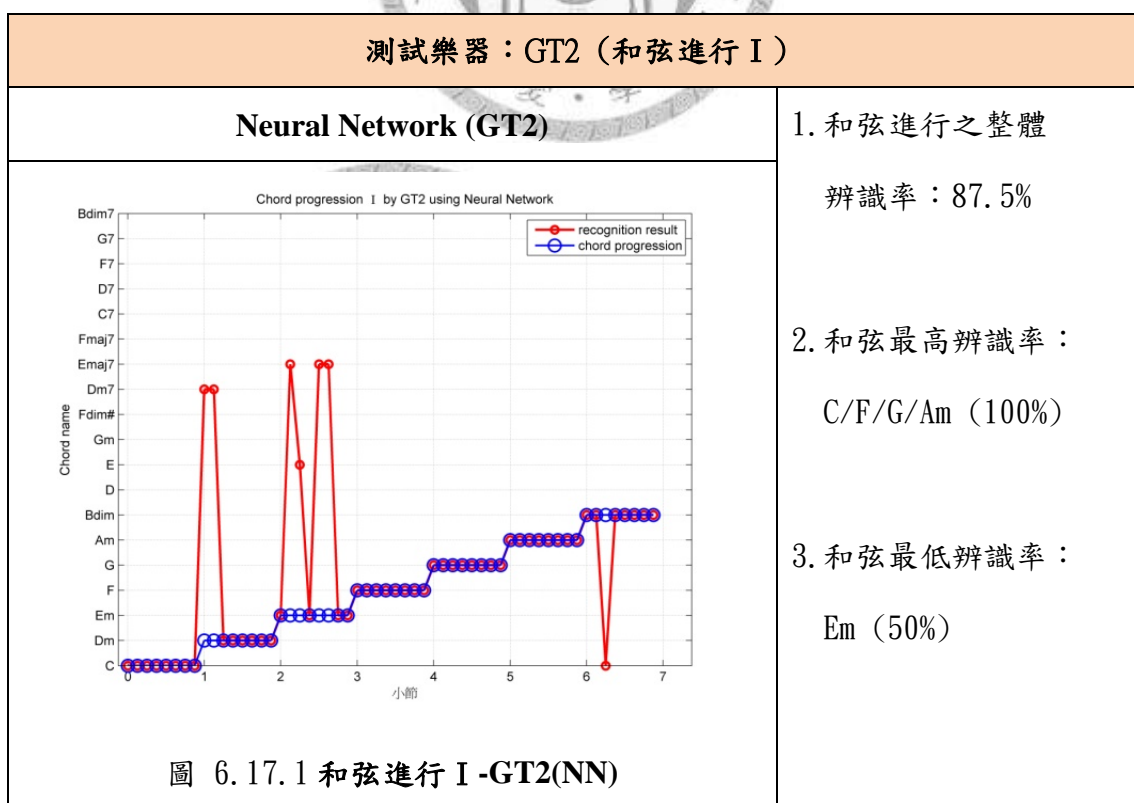


表 6.15 和弦進行 I -GT1

由圖 6.16.1 至圖 6.16.3，可發現使用類神經演算法來辨識和弦，可達到較高的辨識率；相對的，採用 Knn 演算法則誤判率較高。

表 6.16 所測試的樂器為 GT2，圖 6.17.1 至圖 6.17.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。



<p style="text-align: center;">Naïve Bayes Classifier (GT2)</p> <p style="text-align: center;">圖 6.17.2 和弦進行 I -GT2(NB)</p>	<ol style="list-style-type: none"> 和弦進行之整體 辨識率：75% 和弦最高辨識率： Dm/Am (100%) 和弦最低辨識率： Em (37.5%)
<p style="text-align: center;">Knn method (GT2)</p> <p style="text-align: center;">圖 6.17.3 和弦進行 I -GT2(KNN)</p>	<ol style="list-style-type: none"> 和弦進行之整體 辨識率：64.28% 和弦最高辨識率： F (87.5%) 和弦最低辨識率： Em (37.5%)

表 6.16 和弦進行 I -GT2

由圖 6.17.1 至圖 6.17.3 可發現，使用 GT2 來進行和弦辨識時，相對於 GT1，辨識率略為下降。而三種辨識法中，又以類神經演算法的辨識率最高，驗證了類

神經抗雜訊與容錯性優於其他兩種分類器。

表 6.17 所測試的樂器為 MIDI-Nylon Guitar，圖 6.18.1 至圖 6.18.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：MIDI-Nylon Guitar (和弦進行 I)	
<p style="text-align: center;">Neural Network (MIDI-Nylon Guitar)</p> <p style="text-align: center;">圖 6.18.1 和弦進行 I -Nylon Guitar(NN)</p>	<p>1. 和弦進行之整體 辨識率：82.14%</p> <p>2. 和弦最高辨識率： C/Am/Bdim (100%)</p> <p>3. 和弦最低辨識率： F(37.5%)</p>
<p style="text-align: center;">Naïve Bayes Classifier (MIDI-Nylon Guitar)</p> <p style="text-align: center;">圖 6.18.2 和弦進行 I -Nylon Guitar(NB)</p>	<p>1. 和弦進行之整體 辨識率：85.71%</p> <p>2. 和弦最高辨識率： C/Em/F/G(100%)</p> <p>3. 和弦最低辨識率： Dm/Bdim (62.5%)</p>

測試樂器：MIDI-Nylon Guitar (和弦進行 I)

Knn method (MIDI-Nylon Guitar)

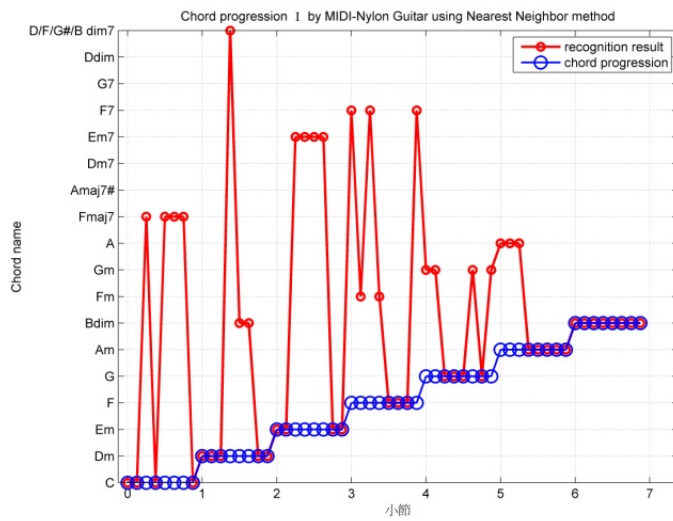


圖 6.18.3 和弦進行 I -Nylon Guitar(KNN)

1. 和弦進行之整體

辨識率：58.92%

2. 和弦最高辨識率：

Bdim (100%)

3. 和弦最低辨識率：

F (37.5%)

表 6.17 和弦進行 I -MIDI Nylon Guitar

由圖 6.18.1 至圖 6.18.3 可發現，使用軟體音源所產生的古典吉他和弦音，使 Knn 演算法的辨識率明顯下降；類神經演算法的辨識率則仍然維持在 80% 以上；辨識率表現最優者為貝氏分類法。

表 6.18 所測試的樂器為 MIDI-Classical Piano，圖 6.19.1 至圖 6.19.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：MIDI-Classical Piano (和弦進行 I)	
Neural Network (MIDI-Classical Piano)	1. 和弦進行之整體 辨識率：92.85% 2. 和弦最高辨識率： Dm/Em/F/G/Bdim (100%) 3. 和弦最低辨識率： C/Am (75%)
<p style="text-align: center;">圖 6.19.1 和弦進行 I -Classical Piano(NN)</p>	1. 和弦進行之整體 辨識率：89.28% 2. 和弦最高辨識率： Em/F/G(100%) 3. 和弦最低辨識率： C/Dm (75%)
<p style="text-align: center;">圖 6.19.2 和弦進行 I -Classical Piano(NB)</p>	

測試樂器：MIDI-Classical Piano (和弦進行 I)

Knn method (MIDI-Classical Piano)

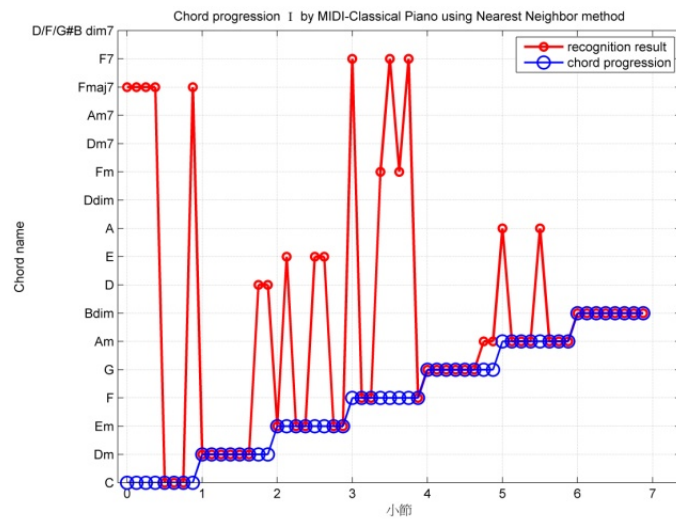


圖 6.19.3 和弦進行 I -Classical Piano(KNN)

1. 和弦進行之整體

辨識率：58.92%

2. 和弦最高辨識率：

Bdim (100%)

3. 和弦最低辨識率：

F (37.5%)

表 6.18 和弦進行 I -MIDI Classical Piano

由圖 6.19.1 至圖 6.19.3 可發現，使用軟體音源產生古典鋼琴的和弦，在使用 Knn 演算法的辨識之下，無法獲得良好的辨識率；然而，使用類神經演算法來辨識，可達到辨識率 90%以上。

2. 和弦進行 II : Cmaj7→Dm7→Em7→Fmaj7→G7→Am7→Bdim7

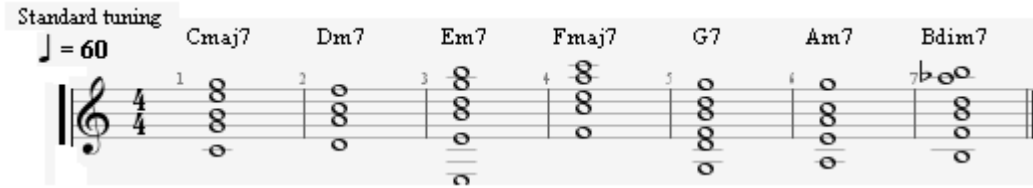


表 6.19 所測試的樂器為 GT1，圖 6.20.1 至圖 6.20.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

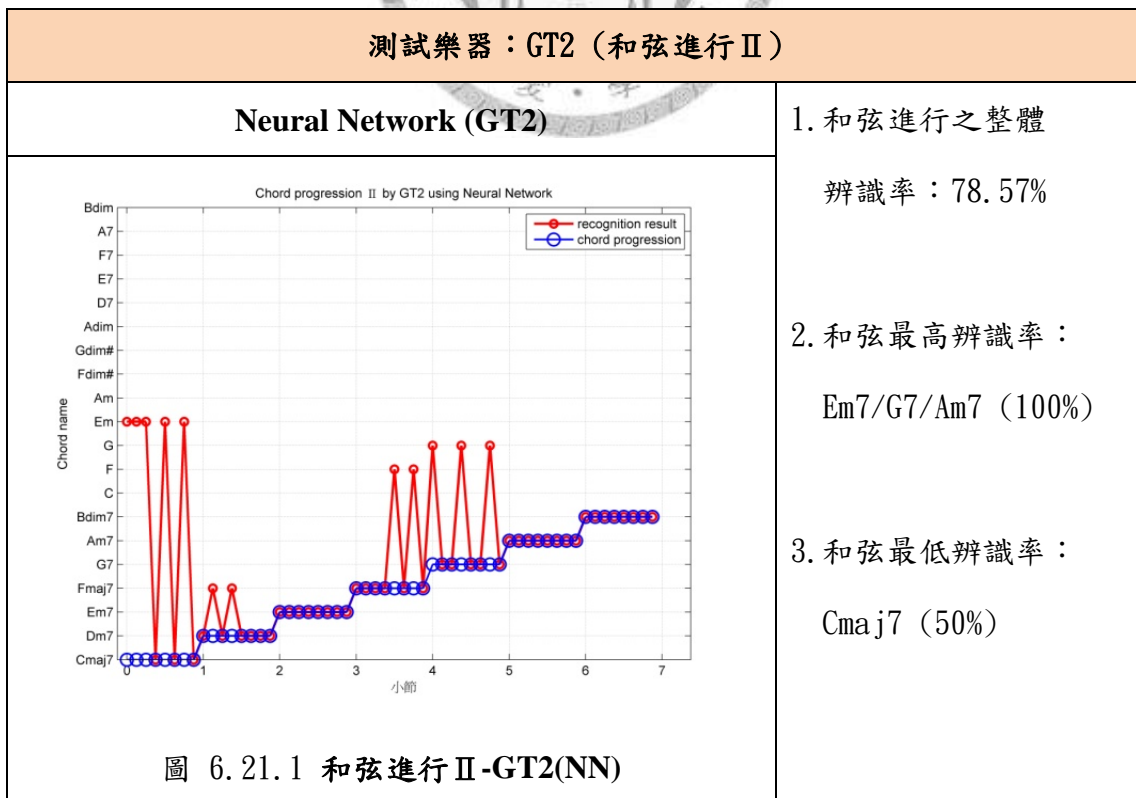
測試樂器：GT1（和弦進行 II）	
<p>Neural Network (GT1)</p> <p style="text-align: center;">圖 6.20.1 和弦進行 II-GT1(NN)</p>	<p>1. 和弦進行之整體 辨識率：75%</p> <p>2. 和弦最高辨識率： Em7 (100%)</p> <p>3. 和弦最低辨識率： Cmaj7/Fmaj7 (50%)</p>
<p>Naïve Bayes Classifier (GT1)</p>	<p>1. 和弦進行之整體</p>

<p>圖 6.20.2 和弦進行 II-GT1(NB)</p>	<p>辨識率：73.21%</p> <p>2. 和弦最高辨識率： Dm7 (100%)</p> <p>3. 和弦最低辨識率： Fmaj7/G7 (50%)</p>
<p>Knn method (GT1)</p>	<p>1. 和弦進行之整體</p>
<p>圖 6.20.3 和弦進行 II-GT1(KNN)</p>	<p>辨識率：71.42%</p> <p>2. 和弦最高辨識率： Em/Am7/Bdim7 (100%)</p> <p>3. 和弦最低辨識率： Fmaj7 (0%)</p>

表 6.19 和弦進行 II-GT1

由圖 6.20.1 至圖 6.20.3 可發現，相對於圖 6.16.1 至圖 6.16.3，雖然使用相同的樂器 GT1，但和弦進行的不同，可明顯發現，三種辨識器在七和弦的和弦進行下，辨識率低於三和弦的和弦進行，可歸納結論為七和弦的組成音含四個單音，其谐波成分，比三和弦複雜，辨識率亦容易下降。

表 6.20 所測試的樂器為 GT2，圖 6.21.1 至圖 6.21.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。



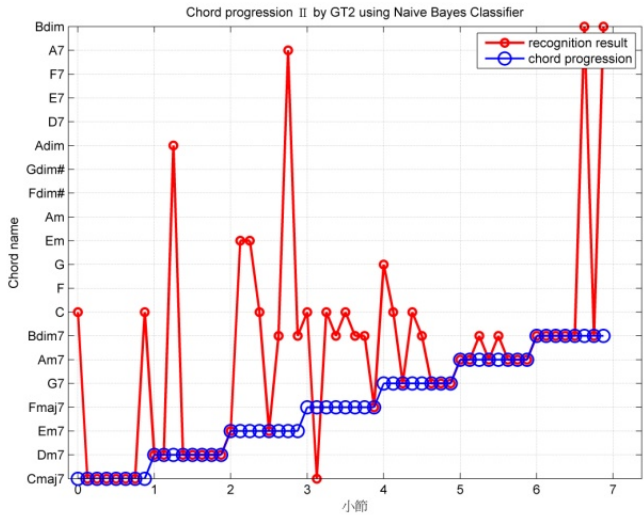
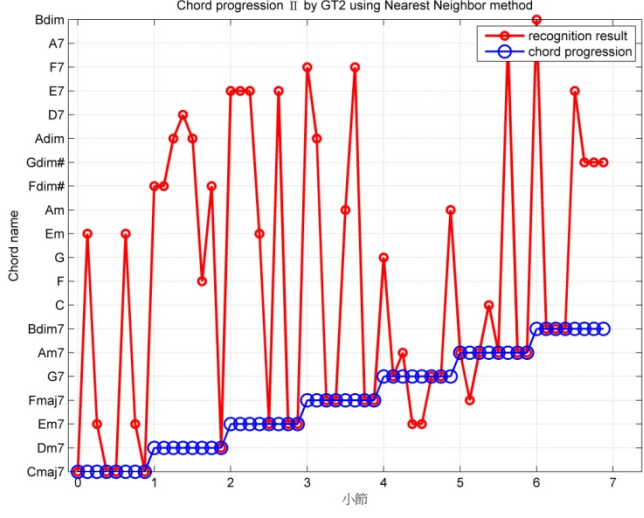
<p style="text-align: center;">Naïve Bayes Classifier (GT2)</p>	<p>1. 和弦進行之整體 辨識率：57.14%</p>
 <p style="text-align: center;">圖 6.21.2 和弦進行 II -GT2(NB)</p>	<p>2. 和弦最高辨識率： Dm7 (87.5%)</p> <p>3. 和弦最低辨識率： Fmaj7 (12.5%)</p>
<p style="text-align: center;">Knn method (GT2)</p>	<p>1. 和弦進行之整體 辨識率：41.07%</p>
 <p style="text-align: center;">圖 6.21.3 和弦進行 II -GT2(KNN)</p>	<p>2. 和弦最高辨識率： Am7 (62.5%)</p> <p>3. 和弦最低辨識率： Dm7 (12.5%)</p>

表 6.20 和弦進行 II -GT2

由圖 6.21.1 至圖 6.21.3 可發現，相較於圖 6.20.1 至圖 6.20.3，實驗發現，使用類神經演算法進行辨識時，並沒有因為受到樂器的不同而影響辨識率，反而提高了辨識率；相對的，貝氏分類法與 Knn 演算法的辨識率下降，可由此知道，類神經演算法具有良好的抗雜訊和學習能力。



表 6.22 所測試的樂器為 MIDI-Nylon Guitar，圖 6.22.1 至圖 6.22.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：MIDI-Nylon Guitar (和弦進行 II)	
Neural Network (MIDI-Nylon Guitar)	1. 和弦進行之整體 辨識率：64.28%
	2. 和弦最高辨識率： Em7 (100%)
圖 6.22.1 和弦進行 II -Nylon Guitar(NN)	3. 和弦最低辨識率： Cmaj7(37.5%)

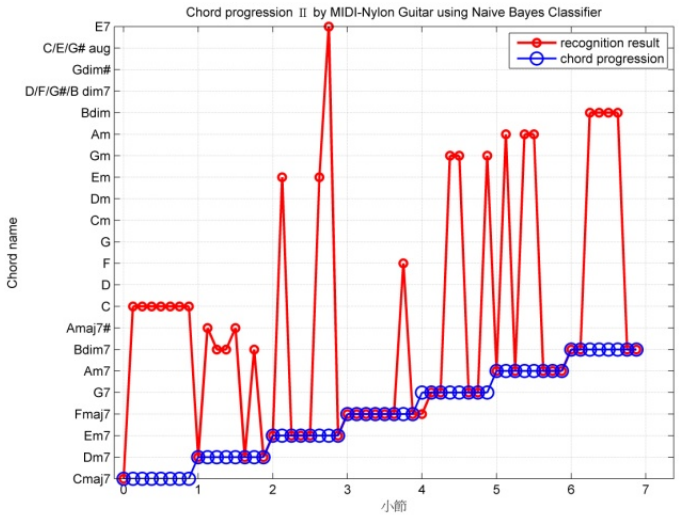
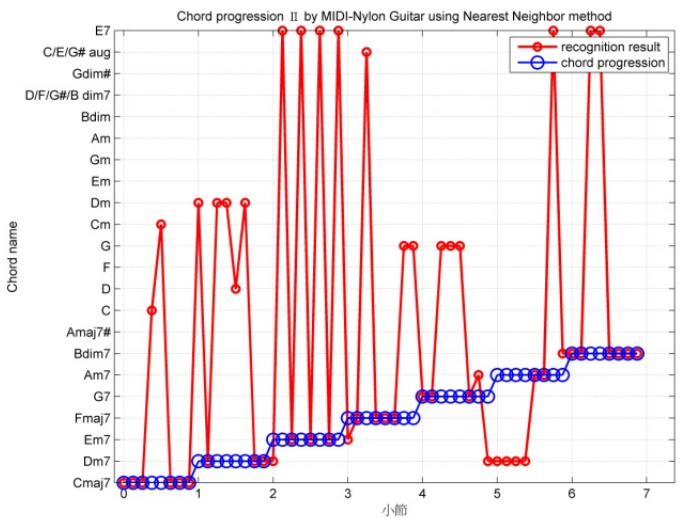
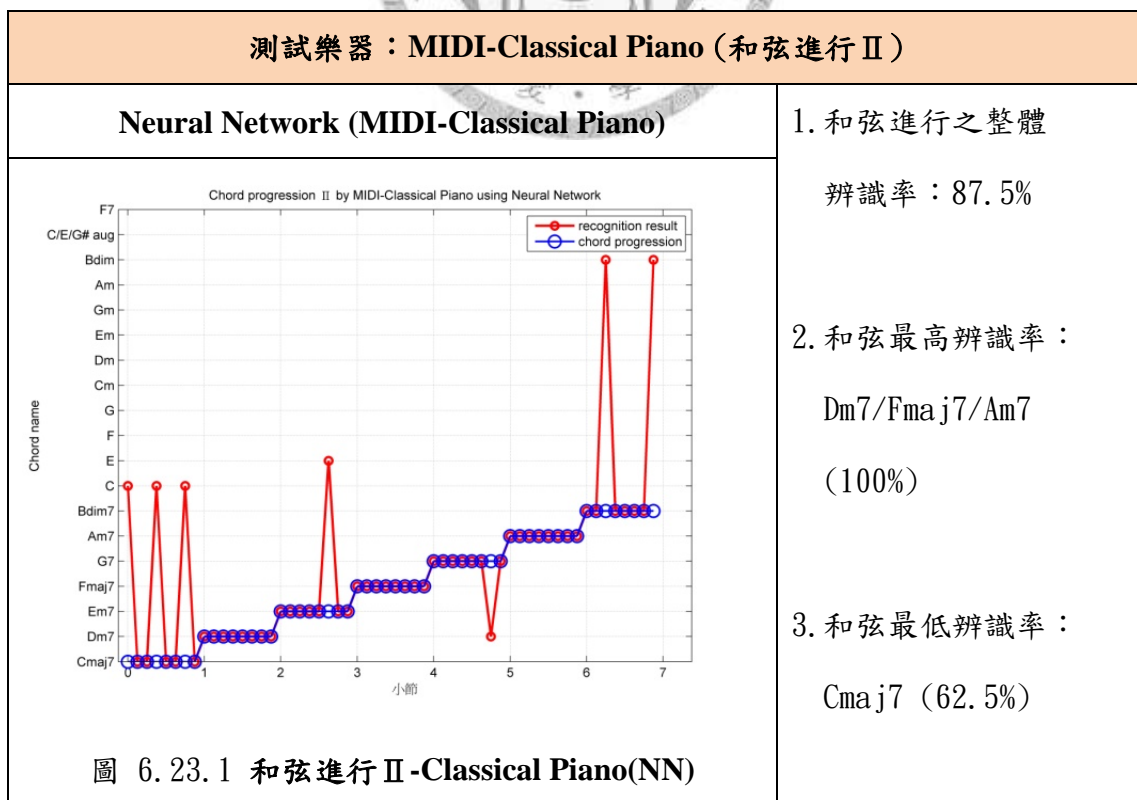
<h3 style="text-align: center;">Naïve Bayes Classifier (MIDI-Nylon Guitar)</h3>	
 <p style="text-align: center;">圖 6.22.2 和弦進行 II -Nylon Guitar(NB)</p>	<ol style="list-style-type: none"> 1. 和弦進行之整體 辨識率：51.78% 2. 和弦最高辨識率： Fmaj7(87.5%) 3. 和弦最低辨識率： Cmaj7 (12.5%)
<h3 style="text-align: center;">Knn method (MIDI-Nylon Guitar)</h3>	
 <p style="text-align: center;">圖 6.22.3 和弦進行 II -Nylon Guitar(KNN)</p>	<ol style="list-style-type: none"> 1. 和弦進行之整體 辨識率：48.21% 2. 和弦最高辨識率： Cmaj7(75%) 3. 和弦最低辨識率： Am7 (25%)

表 6.21 和弦進行 II -MIDI Nylon guitar

由圖 6.22.1 至圖 6.22.3 可發現，使用軟體音源產生 MIDI 古典吉他的和弦，使得三種辨識器在辨識率上略微下降，但仍然以類神經演算法的整體辨識率最高。

表 6.22 所測試的樂器為 MIDI-Classical Piano，圖 6.23.1 至圖 6.23.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。



<h3>Naïve Bayes Classifier (MIDI-Classical Piano)</h3>	
<p>Chord progression II by MIDI-Classical Piano using Naive Bayes Classifier</p> <p>Chord name</p> <p>小節</p> <p>Legend: recognition result (red circle), chord progression (blue circle)</p>	<ol style="list-style-type: none"> 和弦進行之整體 辨識率：85.71% 和弦最高辨識率： Em7/Fmaj7/Am7(100%) 和弦最低辨識率： Bdim7 (62.5%)
<p>圖 6.23.2 和弦進行 II-Classical Piano(NB)</p>	
<h3>Knn method (MIDI-Classical Piano)</h3>	
<p>Chord progression II by MIDI-Classical Piano using Nearest Neighbor method</p> <p>Chord name</p> <p>小節</p> <p>Legend: recognition result (red circle), chord progression (blue circle)</p>	<ol style="list-style-type: none"> 和弦進行之整體 辨識率：33.92% 和弦最高辨識率： Am7 (75%) 和弦最低辨識率： Dm7/Em7 (0%)
<p>圖 6.23.3 和弦進行 II-Classical Piano(KNN)</p>	

表 6.22 和弦進行 II-MIDI Classical Piano

由圖 6.23.1 至圖 6.23.3 可發現，使用軟體音源產生 MIDI 古典鋼琴的和弦，使

用類神經演算法和貝氏分類法進行辨識，可獲得高於85%的整體辨識率；相對的，Knn 的辨識率則非常不理想。



3. 和弦進行Ⅲ：Cm7→Ddim7→Dmaj7#→Fm7→Gm7→Gmaj7#→A7#



表 6.23 所測試的樂器為 GT1，圖 6.24.1 至圖 6.24.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：GT1（和弦進行Ⅲ）	
<p>Neural Network (GT1)</p> <p style="text-align: center;">圖 6.24.1 和弦進行Ⅲ-GT1(NN)</p>	<p>1. 和弦進行之整體 辨識率：73.21%</p> <p>2. 和弦最高辨識率： Cm7/Gmaj7# (100%)</p> <p>3. 和弦最低辨識率： Gm7 (37.5%)</p>
<p>Naïve Bayes Classifier (GT1)</p>	<p>1. 和弦進行之整體</p>

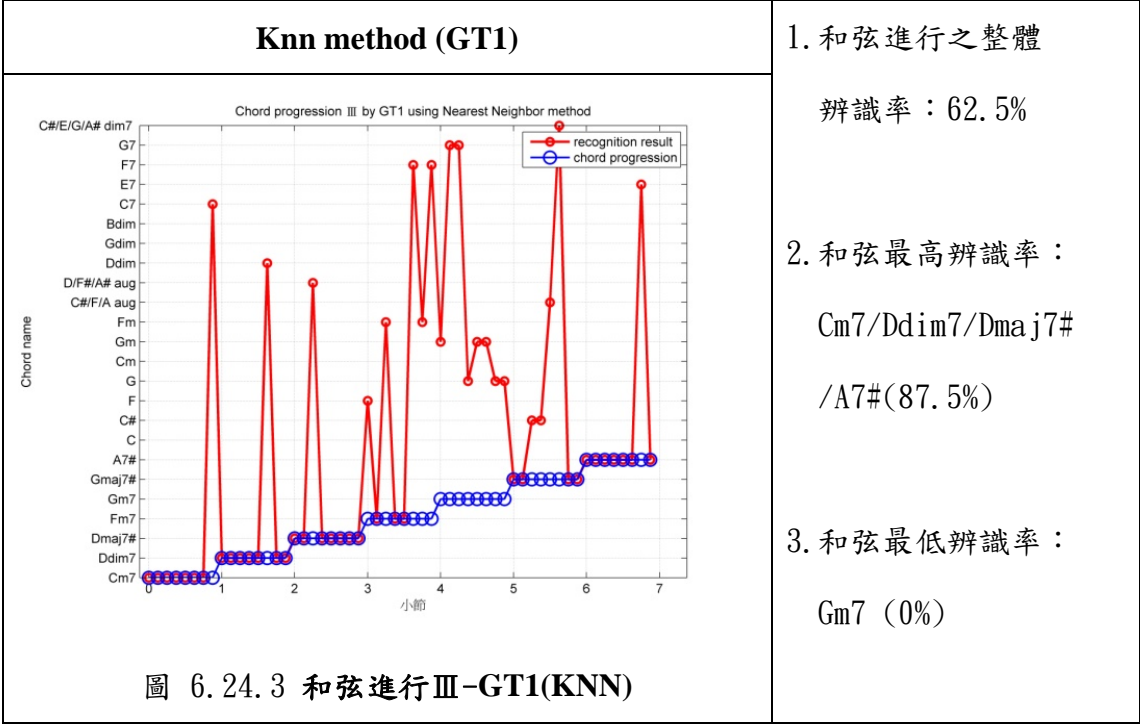
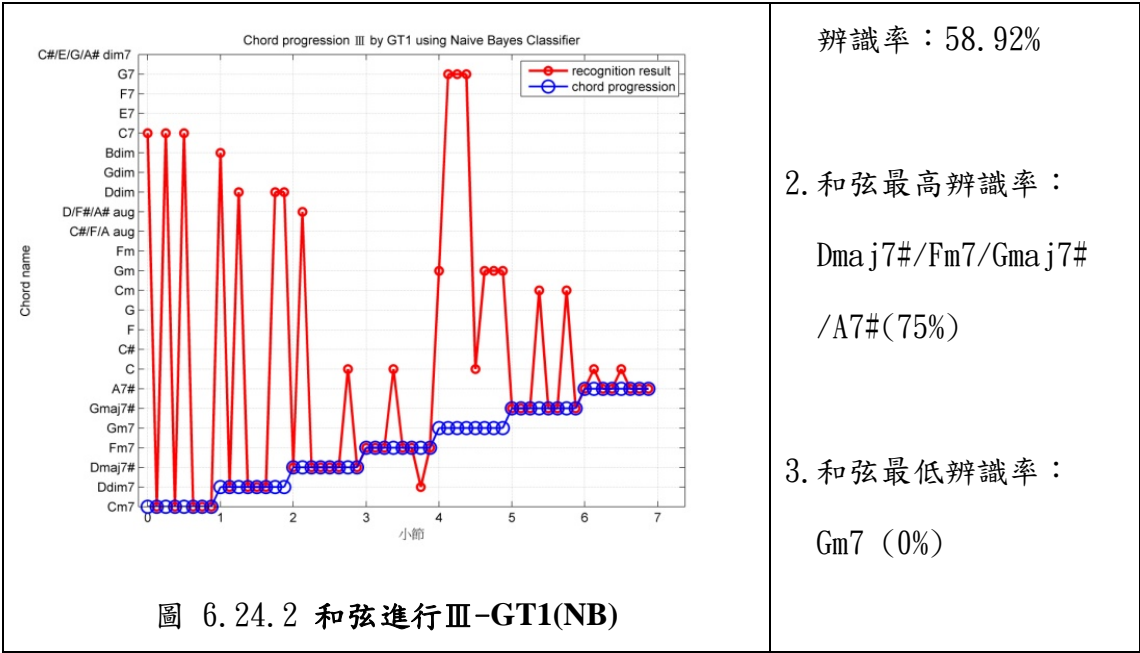


表 6.23 和弦進行III-GT1

由圖 6.24.1 至圖 6.24.3，與圖 6.20.1 至圖 6.20.3 相較，七和弦的和弦進行使得辨識率略低於三和弦的和弦進行。此和弦進行，整體辨識率最高者為類神經演算法；整體辨識最低者為貝氏分類法。

表 6.24 所測試的樂器為 GT2，圖 6.25.1 至圖 6.25.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：GT2（和弦進行Ⅲ）	
Neural Network (GT2)	1. 和弦進行之整體 辨識率：58.92% 2. 和弦最高辨識率： Cm7 (100%) 3. 和弦最低辨識率： Gm7 (25%)
	Naïve Bayes Classifier (GT2)
	1. 和弦進行之整體 辨識率：41.07% 2. 和弦最高辨識率： A7# (62.5%) 3. 和弦最低辨識率： Fm7 (12.5%)
圖 6.25.1 和弦進行Ⅲ-GT2(NN)	圖 6.25.2 和弦進行Ⅲ-GT2(NB)

測試樂器：GT2（和弦進行Ⅲ）

Knn method (GT2)

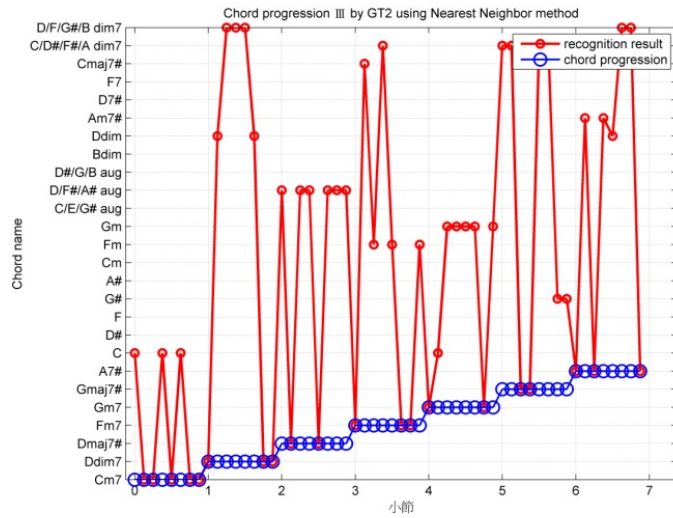


圖 6.25.3 和弦進行Ⅲ-GT2(KNN)

1. 和弦進行之整體

辨識率：35.71%

2. 和弦最高辨識率：

Cm7 (62.5%)

3. 和弦最低辨識率：

Dmaj7#/Gmaj7# (25%)

表 6.24 和弦進行Ⅲ-GT2

由圖 6.25.1 至圖 6.25.3，與圖 6.24.1 至圖 6.24.3 相比較，採用不同的樂器 GT2，使得三種分類器在整體辨識率上明顯下降。其中，類神經演算法對於此和弦進行的抗雜訊能力較低，使得辨識率無法達到 60% 以上。其餘兩種分類器也明顯有相同的問題，致使辨識率偏低。

表 6.25 所測試的樂器為 MIDI-Nylon Guitar，圖 6.26.1 至圖 6.26.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：MIDI-Nylon Guitar (和弦進行 III)	
<p style="text-align: center;">Neural Network (MIDI-Nylon Guitar)</p> <p style="text-align: center;">圖 6.26.1 和弦進行 III-Nylon Guitar(NN)</p>	<p>1. 和弦進行之整體 辨識率：67.85%</p> <p>2. 和弦最高辨識率： Dmaj7#/Gm7 (100%)</p> <p>3. 和弦最低辨識率： Gmaj7# (0%)</p>
<p style="text-align: center;">Naïve Bayes Classifier (MIDI-Nylon Guitar)</p> <p style="text-align: center;">圖 6.26.2 和弦進行 III-Nylon Guitar(NB)</p>	<p>1. 和弦進行之整體 辨識率：55.35%</p> <p>2. 和弦最高辨識率： Ddim7(100%)</p> <p>3. 和弦最低辨識率： Gmaj7# (0%)</p>

測試樂器：MIDI-Nylon Guitar (和弦進行Ⅲ)

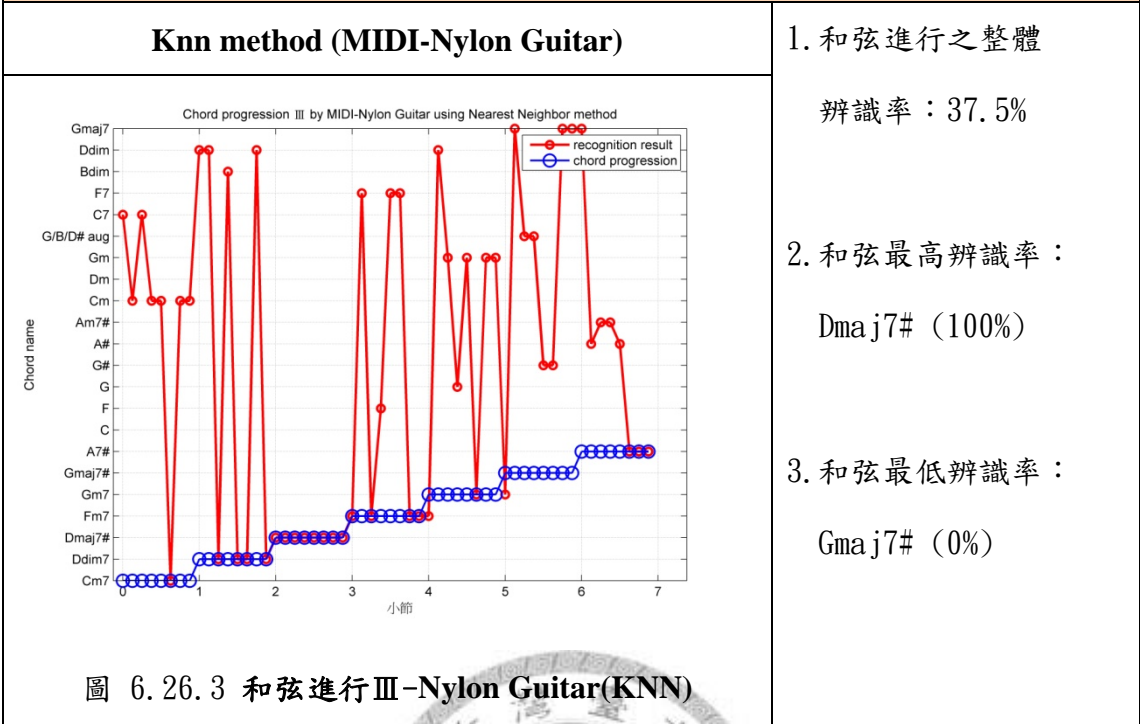


表 6.25 和弦進行Ⅲ-MIDI Nylon guitar

由圖 6.26.1 至圖 6.26.3，使用軟體音源MIDI 古典吉他產生和弦，三種辨識器中，僅只有類神經演算法的整體辨識率能超過 60%；整體辨識率最低者為 Knn 演算法，僅只有 37.5%。

表 6.26 所測試的樂器為 MIDI-Classical Piano，圖 6.27.1 至圖 6.27.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：MIDI-Classical Piano (和弦進行Ⅲ)	
<h3>Neural Network (MIDI-Classical Piano)</h3> <p style="text-align: center;">圖 6.27.1 和弦進行Ⅲ-Classical Piano(NN)</p>	<ol style="list-style-type: none"> 和弦進行之整體 辨識率：76.78% 和弦最高辨識率： Gmaj7#/A7# (100%) 和弦最低辨識率： Cm7 (50%)
<h3>Naïve Bayes Classifier (MIDI-Classical Piano)</h3> <p style="text-align: center;">圖 6.27.2 和弦進行Ⅲ-Classical Piano(NB)</p>	<ol style="list-style-type: none"> 和弦進行之整體 辨識率：75% 和弦最高辨識率： A7#(100%) 和弦最低辨識率： Ddim7/Gm7 (50%)

測試樂器：MIDI-Classical Piano (和弦進行Ⅲ)

Knn method (MIDI-Classical Piano)

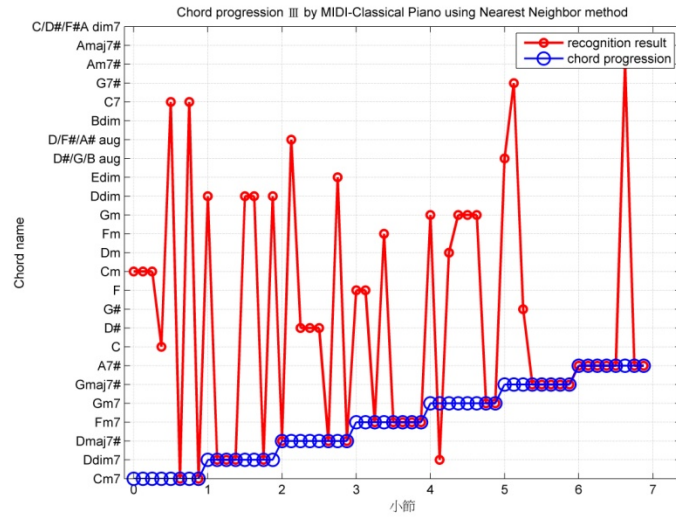


圖 6.27.3 和弦進行Ⅲ-Classical Piano(KNN)

1. 和弦進行之整體

辨識率：50%

2. 和弦最高辨識率：

A7# (87.5%)

3. 和弦最低辨識率：

Cm7/Gm7 (25%)

表 6.26 和弦進行Ⅲ-MIDI Classical Piano

由圖 6.27.1 至圖 6.27.3，使用軟體音源 MIDI 古典鋼琴產生和弦，其使用類神經演算法和貝氏分類法辨識和弦，都有達 75% 以上的辨識率；採用 Knn 演算法所得到的整體辨識相對偏低，說明了 Knn 演算法在抗雜訊以及學習能力上，明顯劣於類神經演算法與貝氏分類法。

4. 和弦進行IV：Cm→Ddim→D#→Fm→Gm→G#→A#



表 6.27 所測試的樂器為 GT1，圖 6.28.1 至圖 6.28.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：GT1（和弦進行IV）	
<p>Neural Network (GT1)</p>	<p>1. 和弦進行之整體 辨識率：89.28%</p> <p>2. 和弦最高辨識率： Cm/Fm/Gm/A#（100%）</p> <p>3. 和弦最低辨識率： Ddim（62.5%）</p>
<p>Naïve Bayes Classifier (GT1)</p>	<p>1. 和弦進行之整體</p>

圖 6.28.1 和弦進行IV-GT1(NN)

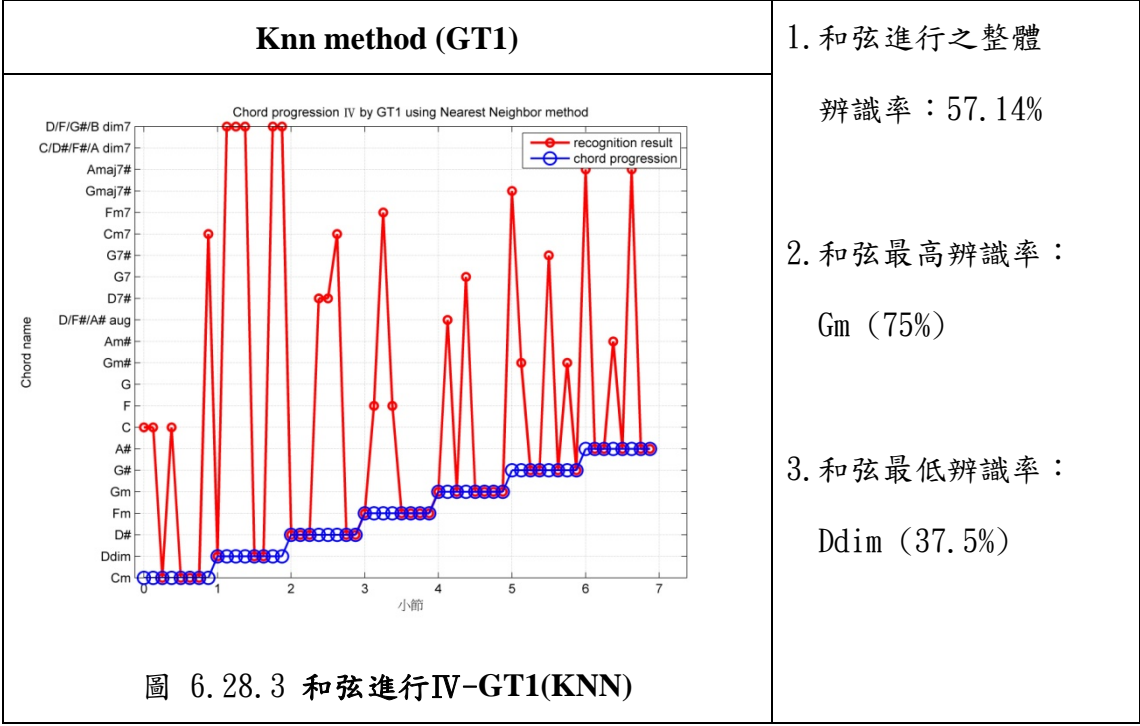
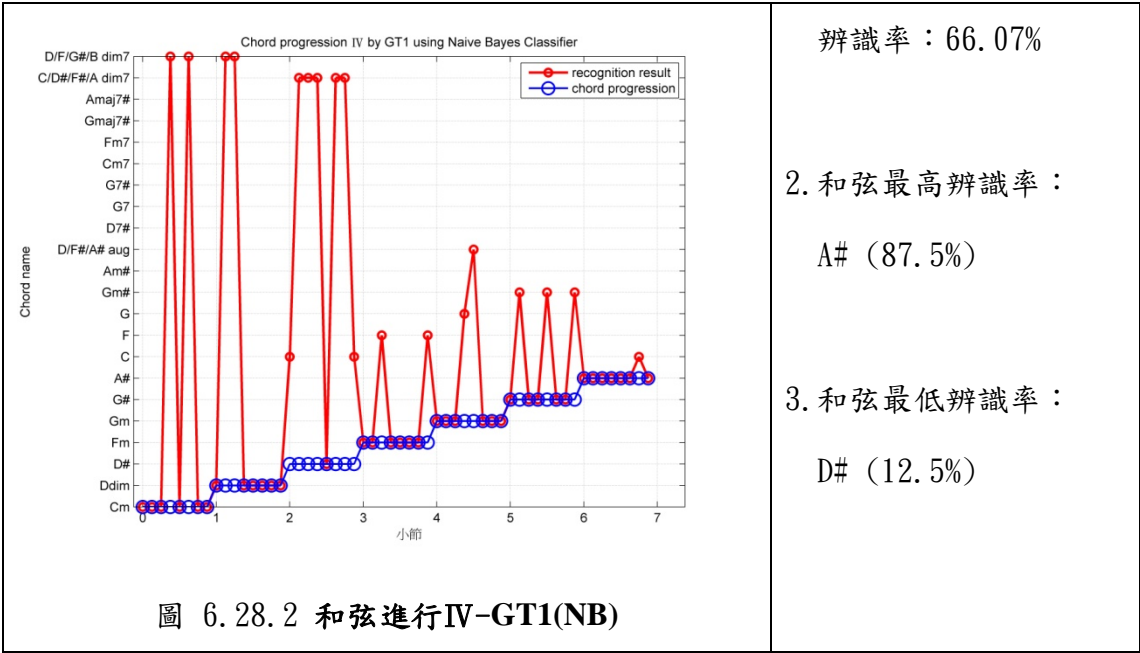


表 6.27 和弦進行IV-GT1

由圖 6.28.1 至圖 6.28.3，在三和弦的和弦進行之下，三種辨識器皆可獲得較佳的辨識率。其中，以類神經演算法的整體辨識率接近 90% 為最高；相對的，Knn 演算法在整體辨識率上仍然不盡理想，僅只有不到 60% 的辨識率。

表 6.29 所測試的樂器為 GT2，圖 6.29.1 至圖 6.29.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：GT2（和弦進行IV）	
<h3>Neural Network (GT2)</h3> <p style="text-align: center;">圖 6.29.1 和弦進行IV-GT2(NN)</p>	<ol style="list-style-type: none"> 和弦進行之整體 辨識率：85.71% 和弦最高辨識率： Cm/Fm/Gm (100%) 和弦最低辨識率： Ddim (62.5%)
<h3>Naïve Bayes Classifier (GT2)</h3> <p style="text-align: center;">圖 6.29.2 和弦進行IV-GT2(NB)</p>	<ol style="list-style-type: none"> 和弦進行之整體 辨識率：62.5% 和弦最高辨識率： Fm (87.5%) 和弦最低辨識率： D# (25%)

測試樂器：GT2 (和弦進行IV)

Knn method (GT2)

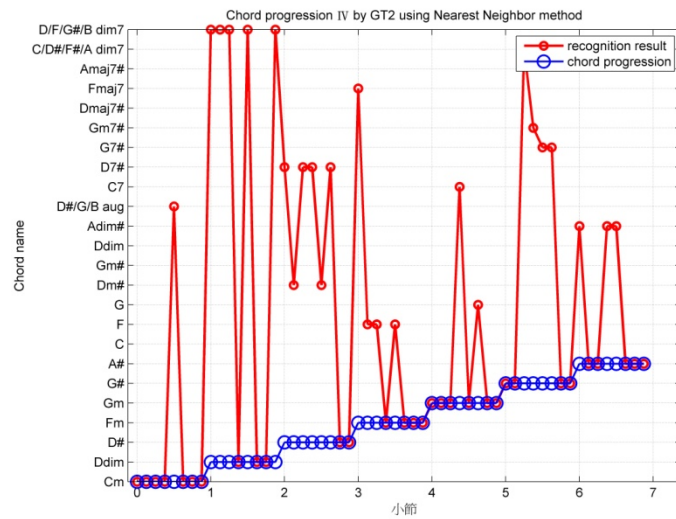


圖 6.29.3 和弦進行IV-GT2(KNN)

1. 和弦進行之整體

辨識率：55.35%

2. 和弦最高辨識率：

Cm (87.5%)

3. 和弦最低辨識率：

D# (25%)

表 6.28 和弦進行IV-GT2

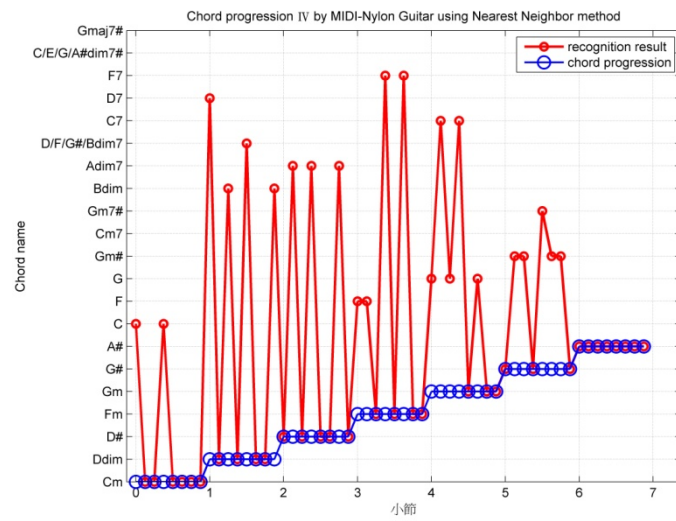
由圖 6.29.1 至圖 6.29.3，比較圖 6.28.1 至圖 6.28.3，可發現即使辨識樂器改為 GT2，使用類神經演算法進行辨識時，仍能維持在 85% 以上的辨識率；使用貝氏分類法進行辨識時，也仍能維持在 60% 以上的辨識率。

表 6.29 所測試的樂器為 MIDI-Nylon Guitar，圖 6.30.1 至圖 6.30.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：MIDI-Nylon Guitar (和弦進行IV)	
Neural Network (MIDI-Nylon Guitar)	1. 和弦進行之整體 辨識率：87.5% 2. 和弦最高辨識率： Ddim/G#/A# (100%) 3. 和弦最低辨識率： D#(62.5%)
<p style="text-align: center;">圖 6.30.1 和弦進行IV-Nylon Guitar(NN)</p>	1. 和弦進行之整體 辨識率：64.28% 2. 和弦最高辨識率： Gm(87.5%) 3. 和弦最低辨識率： Fm (37.5%)
<p style="text-align: center;">圖 6.30.2 和弦進行IV-Nylon Guitar(NB)</p>	

測試樂器：MIDI-Nylon Guitar (和弦進行IV)

Knn method (MIDI-Nylon Guitar)



1. 和弦進行之整體

辨識率：58.92%

2. 和弦最高辨識率：

A# (100%)

3. 和弦最低辨識率：

Gm/G# (37.5%)

圖 6.30.3 和弦進行IV-Nylon Guitar(KNN)

表 6.29 和弦進行IV-MIDI Nylon Guitar

由圖 6.30.1 至圖 6.30.3，使用軟體音源產生古典吉他的和弦，實驗發現，使用類神經演算法進行辨識，可獲得高達 85% 以上的整體辨識率，明顯優於其於兩種分類器的整體辨識率。

表 6.30 所測試的樂器為 MIDI-Classical Piano，圖 6.31.1 至圖 6.31.3 所使用的演算法分別為 Neural Network、Naïve Bayes Classifier 和 Knn method。

測試樂器：MIDI-Classical Piano (和弦進行IV)	
Neural Network (MIDI-Classical Piano)	1. 和弦進行之整體 辨識率：89.28% 2. 和弦最高辨識率： Cm/Fm/Gm/G#/A# (100%) 3. 和弦最低辨識率： Ddim/D# (62.5%)
	圖 6.31.1 和弦進行IV-Classical Piano(NN)
Naïve Bayes Classifier (MIDI-Classical Piano)	1. 和弦進行之整體 辨識率：80.35% 2. 和弦最高辨識率： G#/A#(100%) 3. 和弦最低辨識率： Ddim (50%)
	圖 6.31.2 和弦進行IV-Classical Piano(NB)

測試樂器：MIDI-Classical Piano (和弦進行IV)

Knn method (MIDI-Classical Piano)

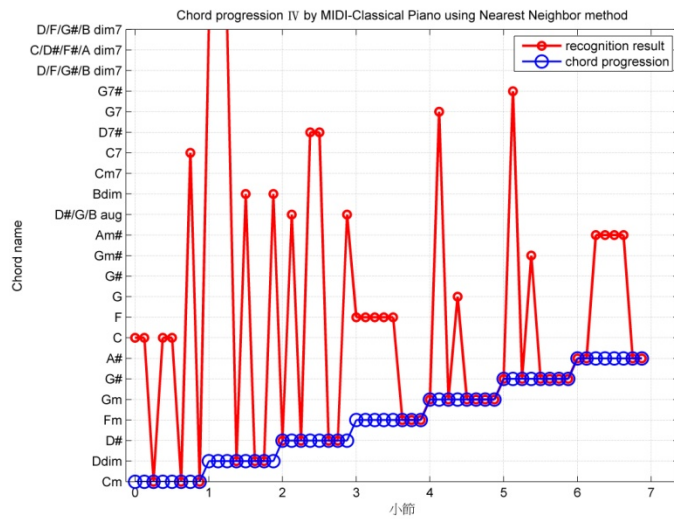


圖 6.31.3 和弦進行IV-Classical Piano(KNN)

1. 和弦進行之整體
辨識率：51.78%
2. 和弦最高辨識率：
G# (75%)
3. 和弦最低辨識率：
Cm/Fm (37.5%)

表 6.30 和弦進行IV-MIDI Classical Piano

由圖 6.31.1 至圖 6.31.3，使用軟體音源產生古典鋼琴的和弦，實驗發現，使用類神經演算法進行辨識，可獲得高達接近 90%的整體辨識率；而使用貝氏分類法進行辨識，也可得到 80%以上的整體辨識率。

接著，將以上所討論的吉他和弦進行即時辨識以表格整理如下：

1. 和弦進行 I 辨識率比較 (單位：百分率 %)

樂器 分類器	GT1	GT2	MIDI-Nylon Guitar	MIDI-Piano
類神經	92.85	87.5	82.14	92.85
貝氏分類器	80.35	75	85.71	89.28
Knn	71.42	64.28	58.92	66.07

2. 和弦進行 II 辨識率比較 (單位：百分率 %)

樂器 分類器	GT1	GT2	MIDI-Nylon Guitar	MIDI-Piano
類神經	75	78.57	64.28	87.5
貝氏分類器	73.21	57.14	51.78	85.71
Knn	71.42	41.07	48.21	33.92

3. 和弦進行 III 辨識率比較 (單位：百分率 %)

樂器 分類器	GT1	GT2	MIDI-Nylon Guitar	MIDI-Piano
類神經	73.21	58.92	67.85	76.78
貝氏分類器	58.92	41.07	55.35	75
Knn	62.5	35.71	37.5	50

4. 和弦進行IV 辨識率比較 (單位：百分率 %)

樂器 分類器	GT1	GT2	MIDI-Nylon Guitar	MIDI-Piano
類神經	89.28	85.71	87.5	89.28
貝氏分類器	66.07	62.5	64.28	80.35
Knn	57.14	55.35	58.92	51.78

●分析討論：

1. 由實驗結果可以發現，類神經演算法的辨識率為三者分類器中最高，其次是貝氏分類法，而 Knn 則為辨識率最低者。

2. 由不同之和弦進行可以發現，和弦進行 I 和 IV(三和弦進行)有較優的辨識率，和弦進行 II 和 III(七和弦進行)的辨識率則較差。也說明了三種辨識器在辨識三和弦的能力上優於七和弦。

3. 由改變樂器觀察分類器的辨識率中可以發現，相較於貝氏分類器與 Knn，類神經演算法即使在樂器不同的情況下仍有較佳的辨識度，也證明了類神經演算法其抗雜訊的能力與外在相容性能力較佳。

使用者介面 (Graphical User Interface)

使用自製的使用者介面(GUI)，可使操作者方便使用，由圖 6.32 至圖 6.35 分別為使用者介面的操作畫面：

由圖 6.32 所示，使用者可以在左邊自由選擇使用分類器：



圖 6.32 使用者介面-選單

由圖 6.33 所示，選擇分類器之後，按開始鈕即可開啟辨識系統：



圖 6.33 使用者介面-選擇分類器

由圖 6.34 所示，將麥克風對準吉他共鳴箱進行彈奏：



圖 6.34 使用者介面-啟動和弦辨識

由圖 6.35 所示，辨識結果會由螢幕即時顯示：



圖 6.35 使用者介面-和弦辨識中

實驗總結討論

實驗的過程中，在一般的環境裡，使用簡易之視訊麥克風接收吉他音訊時，在音訊品質上自然無法要求完美，必定摻雜許多不必要的雜訊。使用類神經演算法的最大優點是抗雜訊能力強，透過更新神經元內的權值來逼近目標函數，使得網路模型在雜訊的容忍度上大，即使在背景噪音有干擾的情況下，仍能使辨識率不受太大影響。

貝氏分類法主要並非透過逼近目標函數的方法來進行演算，而是靠資料之特徵屬性機率的大小，來判定資料分群的標準。所以若所擷取的 PCP 值無法很乾淨，也就沒辦法能很準確地將資料內每個屬性的機率分布拉大，自然就無法很明顯地判定資料所屬的分群，也因此容易造成誤判。

Knn 演算法是貼近資料分群直觀想法之分類法，利用資料與資料點間，歐幾里得之距離，視為資料與資料間相似程度的多寡，以便決定分群的標準。使用距離運算來決定分群的最大缺點就是，倘若資料的屬性維度一高，雜訊一多，就很容易誤判，所以相對於類神經網路與貝氏分類法，使用 Knn 演算法來進行和弦的分類較不理想且運算速度慢。

由上表 7.3 發現，類神經演算法不僅在辨識率程度上獲致良好的辨識率，將其使用於不同的樂器上進行辨識，相較於其餘兩種辨識器，類神經演算法其辨識度仍能維持一定的水平，並沒有因為接受不同的音樂訊號而產生嚴重誤判，這也印證了類神經演算法，其學習相容性高與抗雜訊能力強的特點。

第七章 結論與未來展望

本論文以類神經演算法為核心進行和弦辨識，並利用貝氏分類器與 Knn 分類法做比較，分別實驗了 96 種不同和弦的辨識率，根據實驗結果，可以得到表 7.1

7.1 辨識率數據總結討論

1. 由表 7.1 實驗結果可以得知，類神經演算法的辨識率效果，為三種辨識器當中最者；其次是貝氏分類法，最差者為 Knn 分類法。

表 7.1 為辨識率數據總結：

分類法 測試種類	類神經網路	貝氏分類法	Knn method
總辨識率 (回想)-資料庫	99.04%(總平均) 最佳者為 98.88%	96.61%	
總辨識率 (預測)-資料庫	94.41%(總平均) 最佳者為 95.19%	94.59%	75.46%
吉他和弦進行 即時辨識	81.69%(GT1)	69.64%(GT1)	65.62%(GT1)
	77.68%(GT2)	58.92%(GT2)	49.10%(GT2)
	75.44% (MIDI-Nylon Guitar)	64.28% (MIDI-Nylon Guitar)	50.89% (MIDI-Nylon Guitar)
	86.60% (MIDI-Piano)	82.59% (MIDI-Piano)	50.44% (MIDI-Piano)

表 7.1 辨識率數據總結

2. 在實驗的 96 種不同種類的和弦中，有 48 種三和弦與 48 種七和弦。表 7.2 實驗結果顯示，三種分類器在辨識三和弦的能力上皆優於七和弦；此外，三種分類器當中，又以類神經演算法的辨識率效果最佳，Knn 分類法則辨識率為最差。

表 7.2 為三和弦與七和弦辨識率之比較：

辨識方法 和弦進行	類神經網路	貝氏分類法	KNN method
三和弦	88.39%	75.44%	60.49%
七和弦	72.76%	62.27%	47.54%

表 7.2 三和弦與七和弦辨識率比較



3. 利用四種不同樂器來做和弦進行辨識，以測試其三種辨識器，在接收不同樂器音色的聲音訊號時，能否具有優良的學習相容性與抗雜訊能力。使用平均差來觀察總體各單位標誌值的變動程度，其平均差公式如下：

$$\text{平均差} = \frac{\sum |X - \bar{X}|}{N}$$

其中 N=4 (4 種樂器)，X 為使用四種樂器所得到的各別辨識率， \bar{X} 為四個各別辨識率的平均值。

討論結果如表 7.3 所示，可發現在比較使用四種不同樂器的情況下，所得到的辨識率，其平均差以類神經最低，其次是 Knn 分類法，最高者則為貝氏分類法。由於平均差越小，則代表總體中，數值的變異指標越小，因此可得到結論為，即使使用不同的樂器進行和弦辨識，採用類神經模型可使整體辨識率的差異不會太明顯，也顯示出類神經網路的抗雜訊和相容學習能力明顯優於其餘兩者辨識器。

表 7.3 為三種分類器針對四種不同樂器的辨識率之平均值比較：

辨識方法 不同樂器 之辨識率差異	類神經	貝氏分類法	KNN 分類法
平均差	3.79	7.25	5.80

表 7.3 三種分類器針對四種不同樂器的辨識率之平均值比較

7.2 結論

本論文突破類神經網路應用在系統控制的既定印象，將其使用在分類與辨識的基礎上。類神經網路符合生物神經之學習的運作，亦符合演奏者靠聽神經學習樂音的概念。運用類神經網路抗雜訊強與學習相容性佳的優點，將實驗設備簡單化來符合一般使用者的需求；並且使用類神經辨識的最大優點，亦即訓練好之辨識模型，僅只是結構簡單的矩陣，在數學運算上能大幅提升速度，若將此之硬體實現，則可以達到即時的運算能力。

貝氏分類器運用在即時和弦進行辨識上，效果不如類神經演算法來的突出。受機率分布的影響，各種樂器所發出不同的音色，擁有不同之諧波所組成，易在 PCP 值的強度上造成差異而影響機率分布的判斷，所以導致貝氏分類器在辨識各種不同樂器，彼此的辨識率差異大。

在本論文中，使用 Knn 演算法進行和弦辨識的效果不盡理想，此乃因為資料的特徵屬性維度高，容易因為資料含有雜訊而造成誤判，因此辨識效果遠低於類神經演算法和貝氏分類法。

7.3 未來展望

1. 加入更多類型的和弦，增加其辨識範圍的廣度。
2. 嘗試不同的程式語言撰寫，將程式做成一個可廣泛應用的軟體
3. 採用精度更好的收音麥克風，並利用濾波器改進頻譜分析時的雜訊。
4. 嘗試將本論文的架構硬體化，將寫好的程式移植到單晶片上做成實體。



參考文獻

- [1] Yushi Aono, Haruhiro Katayose and Seiji Inokuchi ,” *A Real-time Session Composer with Acoustic Polyphonic Instruments*” , In *Proceedings of the International Computer Music Conference(ICMC)*, 1998, pp. 236-239.
- [2] Judith C. Brown, ” *Calculation of a constant Q spectral transform*”, J. Acoustical Society of America , 89(1) , January 1991
- [3] S.Hamid Nawab, Salma Abu Ayyash,and Robert Wotiz ,” *Identification of Musical Chords Using Constant-Q Spectra*”, In *Proceedings 2001 IEEE International Conference on Acoustics,Speech,andSignal Processing(ICASSP'01)*, 2001
- [4] Borching Su andShyh-Kang Jeng ,” *Multi-Timbre Chord Classification Using Wavelet Transform and Self-Organized Map Neural Networks*” , In *Proceedings 2001 IEEE International Conference on Acoustics,Speech,andSignal Processing(ICASSP'01)*, 2001
- [5] Yukio Fukayama , ” *A detection algorithm for single tones and chords applying wavelet packets and the extended Kalman filter*” ,*The 47th IEEE International Midwest Symposium on Circuits and Systems*, 2004.
- [6] T. Fujishima, “*Realtime Chord Recognition of Musical Sound:a System Using Common Lisp Music*” , In *Proceedings of the International Computer Music Conference(ICMC)* , 1999 , pp. 464-467
- [7] Alexander Sheh and Daniel P.W. Ellis , “*Chord Segmentation and Recognition using EM-Trained Hidden Markov Models*” , In *Proceedings of the International Symposium on Music Information Retrieval(ISMIR)* , 2003, pp. 185-191
- [8] Emilia Gomez and Perfecto Herrera , ” *Automatic extraction of tonal metadata from polyphonic*”,In *Proceedings of the Audio Engineering Society,London,2004*

- [9] Bee Suan Ong, Emilia Gomez, Sebastian Streich, "Automatic Extraction of Musical Structure Using Pitch Class Distribution Features", LSAS, 2006, pp. 53-65
- [10] Jyh-Shing Roger Jang, "Audio Signal Processing and Recognition," (in Chinese) available at the links for on-line courses at the author's homepage at <http://www.cs.nthu.edu.tw/~jang>
- [11] James W. Cooley, and John W. Tukey, "An algorithm for the machine calculation of complex Fourier series", American Mathematical Society, 1965.
- [12] 王小川, 語音訊號處理, 全華科技圖書股份有限公司, 2009.
- [13] Ian McLoughlin, *Applied Speech and Audio Processing*, Cambridge University Press, NY, 2009
- [14] Giordano Cabral, Jean-Pierre Briot, Francois Pachet, "Impact of Distance in Pitch Class Profile Computation".
- [15] Kyogu Lee, "Automatic Chord Recognition from Audio Using Enhanced Pitch Class Profile", In *Proceedings of the International Computer Music Conference (ICMC)*, 2006
- [16] Jyh-Shing Roger Jang, "Data Clustering and Pattern Recognition," available at the links for on-line courses at the author's homepage at <http://mirilab.org.jang>.
- [17] Dorian Pyle, *Data Preparation for Data Mining*, Morgan Kaufman, California, 1999
- [18] Chin-Teng Lin and C.S. George Lee, *Neural Fuzzy System: A Neural-Fuzzy Synergism to Intelligent Systems*, Prentice Hall, NJ, 1996
- [19] Toshio Fukuda and Takanori Shibata, "Theory and applications of Neural Network for Industrial Control System", *IEEE Transaction on Industrial Electronics*, 39(6), 1992, pp. 472-489
- [20] 羅華強, 類神經網路-Matlab 的應用, 高立圖書有限公司, 2005

- [21] Martin T. Hagan and Mohammad B. Menhaj, “*Training feedforward networks with the Marquardt algorithm*” IEEE Transactions on Neural Network, 1994. pp.989-993
- [22] Bogdan M. Wilamowski, *Fellow, IEEE* and Hao Yu, “*Improved Computation for Levenberg-Marquardt Training*” , IEEE Transactions on Neural Network, 2010, pp.930-937

