國立中興大學統計學研究所

碩士學位論文

利用 HE 預訓練之 CNN 方法於中文單音之辨識

Applying the method of He pre-trained CNN on the Mandarin Monosyllable Recognition

指導教授：李宗寶 Dr. Chung-Bow Lee

研 究 生：黃莘揚 Sin-Yang Huang

中華民國 一○八 年 八 月

國立中興大學 統計 學研究所

碩 士學位論文

題目：利用HE預訓練之CNN方法於中文單音之辨識

姓名：黃華揚 學號：71060l8025

經 口 試 通 過 特 此 證 明

論文指導教授 李家寶

論文考試委員 郭仁泰

邱國致

李家寶

中華民國 108 年 7 月 2 日

摘要

本文利用卷積神經網路(Convolutional neural network, CNN)來對中文單音進行學習及辨識。主要實驗方向為把單音拆成子音與母音，並在同一模型下預測出子、母音類別，最後組合出單音。其中子音總有 36 個類別，母音則有 160 的類別，單音組合則有 1391 個類別。資料特徵求取方法選用梅爾倒頻譜系數(MFCC)，並以此作為模型輸入數值。本論文將實驗不同卷積層層數、特徵圖(feature map)數和全連接層(full connection layer, FC)的層數、神經元個數對辨識結果的影響。同時地，會探討不同的活化函數(activation function)、初始化方法和 BN(batch normalization)、dropout 技術的有無是否會影響分類結果。實驗結果發現在使用 4 層卷積層、3 層全連接層並且使用何初始化(He initialization)和 BN 下獲得最高的單音辨識率。子音、母音和單音辨識正確率分別達到: 96.49%、97.40% 和 94.49%。

關鍵字: 類神經網路、卷積神經網路、MFCC、活化函數、初始化、dropout

Abstract

This thesis is mainly to apply convolutional neural network (CNN) in Chinese monophonic. The main experimental direction is to split the single monophonic into the consonant and the vowel, and predict the consonant and vowel categories under the same model, finally combine them to the monophonic. There are 36 categories of consonants, 160 categories of vowels, and 1391 categories of monophonic. The data feature extraction method uses the Mel-Frequency cepstral coefficients (MFCC) and uses this as the model input value. This thesis will experiment with the effects on the identification results of different convolution layer numbers, feature map numbers, full connection layer (FC) numbers, and number of neurons. At the same time, we will discuss whether different activation functions, initialization methods, batch normalization (BN), and dropout techniques will affect the classification results. The experimental results show that the highest monophonic recognition rate is obtained using four CNN layers, three full connection layers, He initialization and BN. The recognition rates of consonant, vowel and monophonic are: 96.49%, 97.40% and 94.49%, respectively.

Keywords: Neural network, Convolutional neural network, MFCC, activation function, Initialization, dropout

目錄

附圖目錄

表目錄

第一章　緒論

1.1 研究動機

類神經網路在許多領域已經獲得了驚人的成就，在人臉辨識、影像辨識和語音辨識上的成果更是遠遠超過以往研究，其中卷積神經(Convolutional neural network, CNN)網路占了十分重要的地位。在影像辨識比賽各屆冠軍的網路架構中，皆可以看到卷積網路的各種應用，其實用性與成果可見一斑。卷積神經網路主要目的為捕抓特定的特徵，並且在複雜、高維度的空間裡依然可以表現良好，這也是為什麼本文採用此方法為研究主要方向的原因。

過往要讓電子產品例如:電腦、手機，正確辨識出一段語音檔是十分困難的，然而現今有了類神經網路作為辨識方法，出現了許多具有高辨識率的相關產品，諸如手機上的語音助理等，可見類神經網路於現代社會的重要性。未來或許會出現更多使人類生活更便利的應用產品問世。

相比與傳統統計方法例如: 多變量分析、回歸分析等，類神經網路缺乏堅實理論基礎。但是，其結構化網路能為許多不同問題相對容易的建立模型的特性是其一大優勢。

1.2 研究目的

基於上述研究動機，本研究有下列幾項研究目的:

(1) 實驗卷積層層數、特徵圖數、全連接層數、神經元個數對網路的辨識結果是否有影響。

(2) 探討不同的活化函數、初始化方法和 BN、dropout 技術的有無對辨識率的影響。

1.3 相關研究

近幾年隨著硬體設備不斷的進步，以及 Cuda 和平行運算的 Spark 持續的發展，以往需要龐大計算量的演算法得以實現。類神經網路也受益於此開始蓬勃發展，從最早期的 MLP 到 DNN、CNN 以及 RNN 等，新的模型不斷推出，更快、更有效益的演算法也持續提出，並且應用在多個領域，例如: 語音、影像與自然語言處理等，都獲得了巨大的進步。

在 2002 年時，Ahad et al. [1]就使用了多層感知器對數字 0 到 9 的聲音資料進行辨識，其中已經使用了倒傳遞進行權重的修正，但當時硬體設備發展尚未成熟，故只進行了較簡單的辨識任務。Graves et al. [2]在 2013 年利用長短期記憶 RNNs 在 TIMIT 資料集上獲得了 17.7%的錯誤率，他們結合了深度網路以及語音資料具有上下文相關性的特性，於當時獲得了最低的錯誤率。Qian et al. [3]的論文於 2016 年提出，其中實驗了各種超參數的組合和使用不同的模型對辨識率的影響。超參數的實驗包括了模型的深度、厚度等等，模型方面實驗了深層的 CNN 與 LSTM 的模型進行比較，也探討了處理噪音方法的特徵值對辨識率的影響。本論文主要參考鄒振宏 [4]的論文，以不同的方式對中文單音進行辨識，最後再進行兩篇論文辨識結果的比較。

1.3 語音辨識介紹

1.3.1 何謂語音辨識

語音辨識主要目的為讓機器從語音內容辨識出正確類別。為監督學習的一種，即在給定已知分類結果下，建立一個模型函數可視為如下:

$$f(input) \rightarrow output$$

其中 input 在本研究中使用梅爾倒頻譜系數，output 則為相對應的正確類別。

1.3.2 語音特性

本小節將介紹何謂語音特性，和語音特性對於本研究結果會產生何種影響:

(1) 相同語者:

相同語者代表發出語音者為同一人，即便是相同人也不能保證每次發音皆固定，也可能因為身體因素、聲量和語調等原因影響。

(2) 不同語者:

不同語者代表發出語音者為不同人，除了有上述干擾的可能性外，不同語者間口腔、口音方式大多不同，辨識上更加困難。

(3) 環境噪音:

除了上述語者的因素外，還有可能因為外在因素例如: 喧嘩聲、收音設備的品質等等因素影響到辨識結果，這些原因統稱環境噪音。

1.3.3 語音辨識的應用

語音辨識運用在現今社會可說是越來越多，由於物聯網的快速發展，凡舉手機上的打電話、發訊息或者是導航系統、智慧管家等等。凡是需要讓機器「辨識」語音訊息的皆和語音辨識相關。

未來如果能加入自然語言處理(Natural Language Processing)，機器將不只能「辨識」人類講的話，甚至能開始「理解」其中內容，將能提供更廣泛的服務例如: 法律諮詢、醫療諮詢等，可以舒緩醫療資源不足的問題。

第二章　語音訊號前處理與特徵參數求取

在進行語音辨識前，如果把原始資料直接丟入模型內，不只維度過大外，還包括雜訊過多、在 1.3.2 節所提及的語音特性干擾等，所以我們會先把原始資料進行一系列的預處理，其中不外乎降維、降噪等等手段。本章會介紹實驗所用之資料進行過的處理流程，以及資料如何轉換成我的需要的特徵質作為輸入。圖 1 為語音訊息進行前處理的流程。

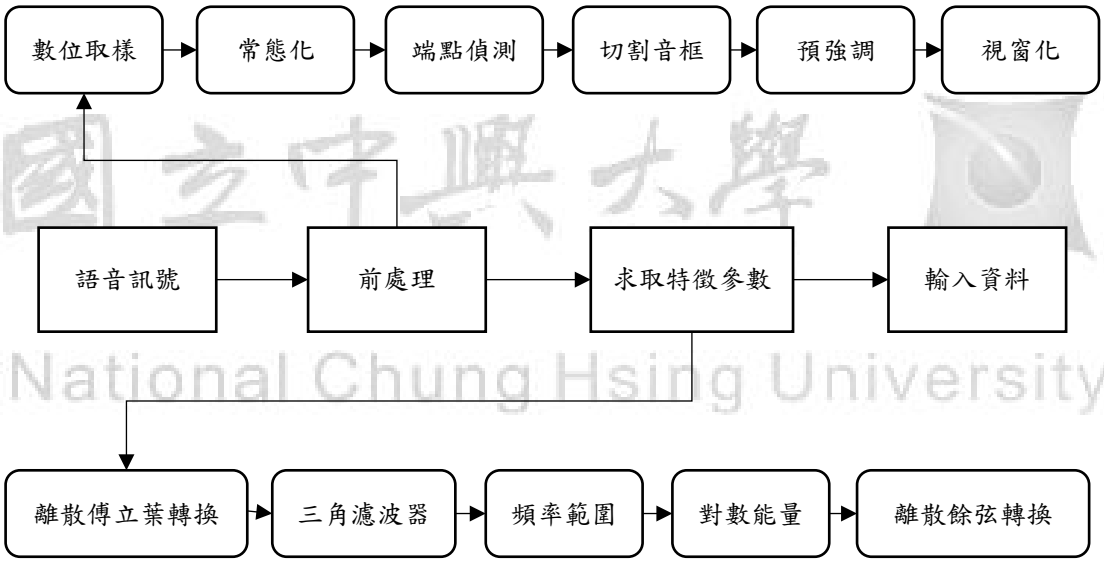

圖 1 語音訊號處理流程

2.1 語音訊號

本篇實驗類別採用 1391 個中文單音來進行子音、母音和單音的辨識。實驗樣本的蒐集採用方法為: 二十位不同語者對指定單音錄製十次。蒐集到的資料即為我們的原始資料，但原始資料是模型難以識別的類比訊號，所以我們需要對資料進行預處理獲得所需特徵值。

2.2 資料前處理

2.2.1 數位取樣

獲得了語音原始資料後，由於其資料型態為類比訊號，我們無法直接使用，所以我們需要進行數位取樣。數位取樣的概念為將資料轉換為「0」、「1」的數位訊號，並且在固定時間間隔下擷取樣本點，樣本點的值作為震幅大小。本論文所使用之資料庫採樣頻率固定為每秒 11025 個樣本點。

2.2.2 常態化

進行錄音時，會因為許多外在與人為因素導致獲得資料變異過大，為了避免資料太小被當成起點或終止點，以及確保網路權重不會因為變異過大導致訓練困難，我們會對資料進行常態化，目的為降低資料間的變異程度。本論文將原始資料範圍從[-32768, 32768]調整至[-10, 10]。

2.2.3 端點偵測

一段錄音黨中，不會只包含到所想要的中文單音資訊，往往會包含到其他我們不需要的雜訊，例如: 氣音、電子設備雜音等等。所以我們需要切割出語音真正開始和結束的位置，使辨識結果不會被干擾。本論文使用的端點偵測方法為「能量量測法」及「越零率法」。

2.2.4 切割音框與視窗化

經過上述處理後，我們獲得了隨時間變化的向量。為了使其便於分析，我們將語音資料進行切割的動作，切割的方法為每 20~30 毫秒切成一個音框，在將其堆疊成一矩陣。最後產生一個單音有 53 個音框，其中子音 24 個音框，母

音則有 29 個音框，每個音框內有則 256 個樣本點。

根據以往經驗，我們會捨棄子音、母音前後各兩個音框避免其中雜訊干擾。在切割音框時會破壞其連續性，為了使端點附近更為平滑，我們會將音框內的樣本點乘上一個視窗函數。

2.2.5 預強調

聲音在傳遞的過程中會產生高頻能量的損失，而人類耳朵能有高頻強調的特性機器沒有，為了彌補其中的損失，我們會讓語音通過一個高通濾波器，此過程就稱為「預強調」。

2.3 特徵參數的求取

完成資料之前處理後，我們需要對其做特徵參數的求取。本論文使用之方法為梅爾導頻譜係數特徵(Mel-Frequency cepstral coefficient, MFCC)。本節將簡述其流程以及所使用之方法。

首先，因為我們的資料是屬於時域上的資料型態，所以須將其轉變為頻域上的資料型態。這裡我們使用「離散傅立葉轉換」來達成我們的目的。做好轉換之後，再將資料通過一組梅爾濾波器，該濾波器是由 M 個非線性的三角形濾波器所組成。資料通過三角濾波器組後，我們會將其乘與一個函數並加總後再取對數值，此時我們獲得的數值稱為「對數能量」。最後，對資料進行「離散餘弦轉換」得到的結果即為「梅爾倒頻譜係數(MFCC)」。

第三章　研究方法

3.1 介紹

本章節將介紹在模型中所使用到網路架構的概念、數學原理等，除了模型架構外，也會探討架構中使用的活化函數、優化器等在不同設置下的影響。

3.2 感知器

首先，我們先介紹簡單架構的單層感知器(perception)，圖 2 即為該模型架構圖。左邊為一個單層感知器架構，其基本構造包含了一個輸入層(input layer)、一個隱藏層(hidden layer)和最後的輸出層(output layer)。右邊則是一個神經元內部的構造，以圖形符號為例，神經元的輸入為(1)式:

$$\sum \omega * x \tag{1}$$

其中 w 為權重，主要透過倒傳遞(back propagation)方式訓練。在輸出資料前，我們會用一個函數對資料進行轉換或稱活化，這個函數一般稱為活化函數(activation function)。經過活化後的資料才會進入下一層繼續訓練。如果我們增加了隱藏層層數即形成了一個多層感知器(multilayers perception)如圖 2 所示，或者可稱其為深度神經網路(deep neural network, DNN)。

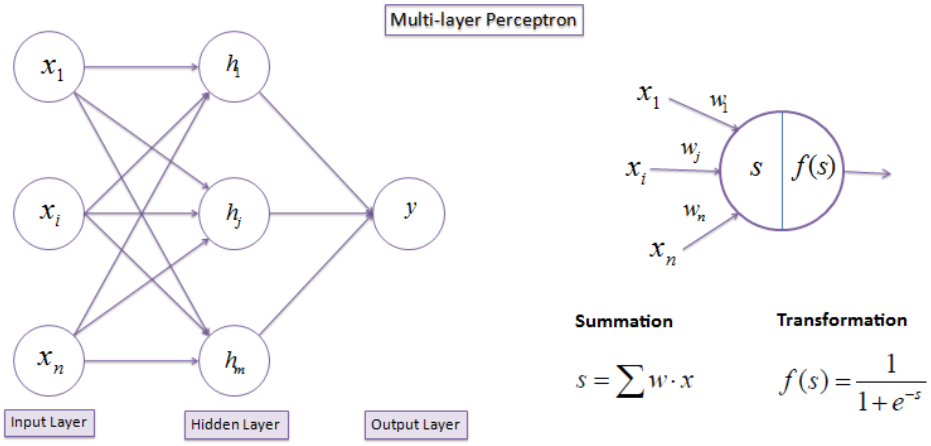

圖 2 簡單多層感知器

(http://www.saedsayad.com/artificial_neural_network_bkp.htm)

3.2.1 活化函數

圖 2 中使用 sigmoid function 作為活化函數，其方程式如圖 2 所示，涵式圖形則為圖 3。

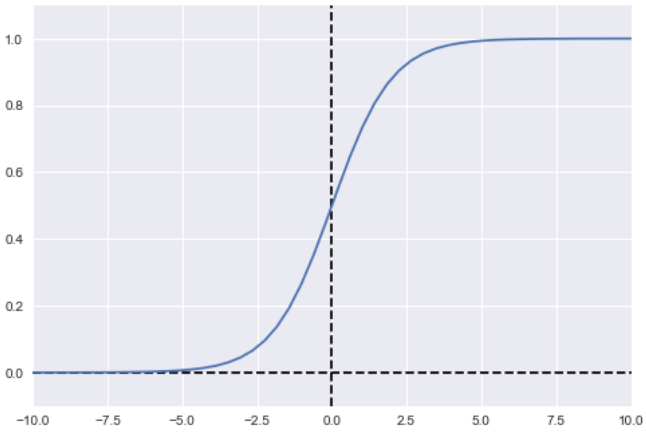

圖 3 Sigmoid Function

Sigmoid 為類神經發展早期所使用的活化函數，但由於其諸多問題例如: 在數值過大或過小時，會產生函數值不敏感、產生梯度消失(gradient vanishing)和在靠近函數值 1 和 0 時會難以訓練參數等問題目前已多被 ReLU 函數所取代，但本論文依舊會在實驗中比較其成果與其他活化函數的辨識率差距。

- Rectified Linear Unit (ReLU)

$$y = max(0, x) \qquad ,\forall\ x \in R \tag{2}$$

ReLU 為目前較主流的活化函數，因為其小於 0 輸出為 0，大於 0 為線性輸出的特性，可以大大降低計算量。並且高於 0 才輸出的特性與生物上的神經元特性相同，作為神經網路活化函數也有一定說服力。

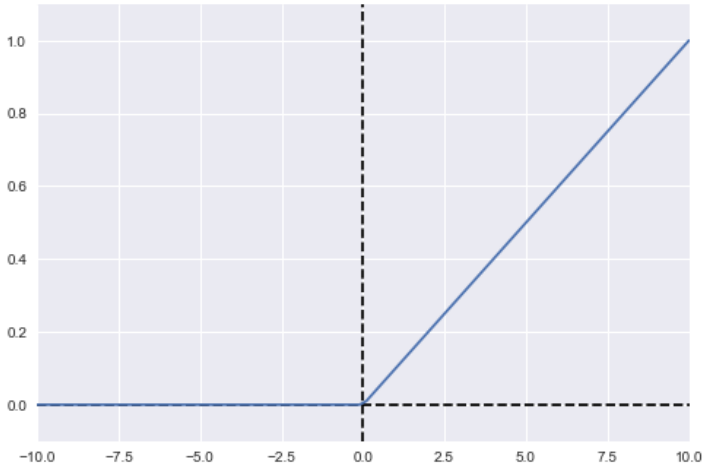

圖 4 ReLU Function

- Exponential Linear Unit (ELU)

$$f(x) = \begin{cases} x & ,x > 0 \\ \alpha * (exp(x) - 1) & ,else \end{cases} \tag{3}$$

Elu 為 ReLU 活化函數的一種變體，其圖形如圖 5 所示，提出者認為在 ReLU 函數中，所有小於 0 的數皆輸出為 0 會丟失許多訊息，所以提出了在小於 0 時輸出為$\alpha * (exp(x) - 1)$。其中 α 為一個可訓練的係數，訓練方式一樣透過倒傳遞進行，並且採用與「權重共享」(shared weights)一樣的方式，在同一層特徵圖內 α 皆相同。此方法的優點為只增加少量參數，增加的計算量微乎其微。在提出者的實驗中，此方法獲得了不錯的成效，故本論文把此方法納入實驗中進行比較。

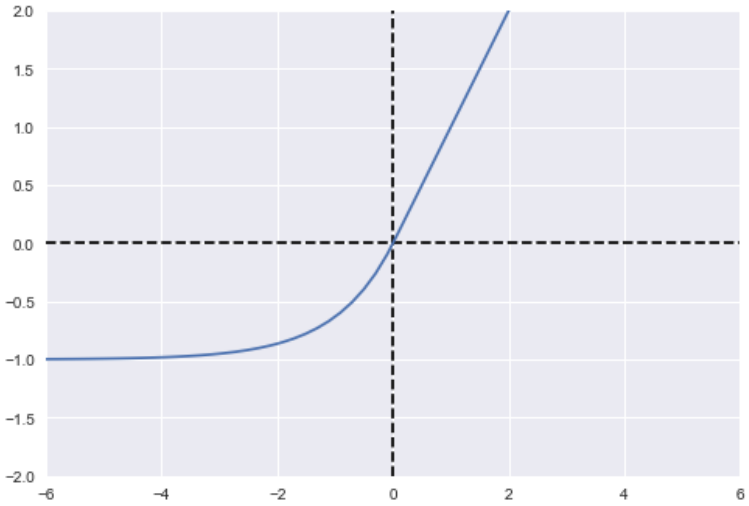

圖 5 ELU Function

- Softmax

$$f_i(\mathrm{x}) = \frac{e^{x_i}}{\sum_i^K e^{x_i}} \; for \;\; i = 1, \ldots \ldots K \tag{4}$$

Softmax 作用是把原始輸出壓縮至總和為 1，目的是使輸出擁有機率的特性，形成一機率空間。此函數一般不放在隱藏層內做為活化函數，通常做為輸出層的活化函數，使輸出結果擁有機率的意義。

3.2.2 多層感知器

了解單層感知器後，我們將介紹多層感知器。圖 6 為該模型架構，其中隱藏層是我們可以任意添加或刪減的。由於單層感知器只能去擬合線性可分的問題，在遇到線性不可分問題時，預測效果往往很差。此時，通用的方法為增加層數。理論上來說，如果問題的解屬於 convex 函數，透過增加層數可以非常逼近的擬合此函數，最終也會獲得較好的預測結果。

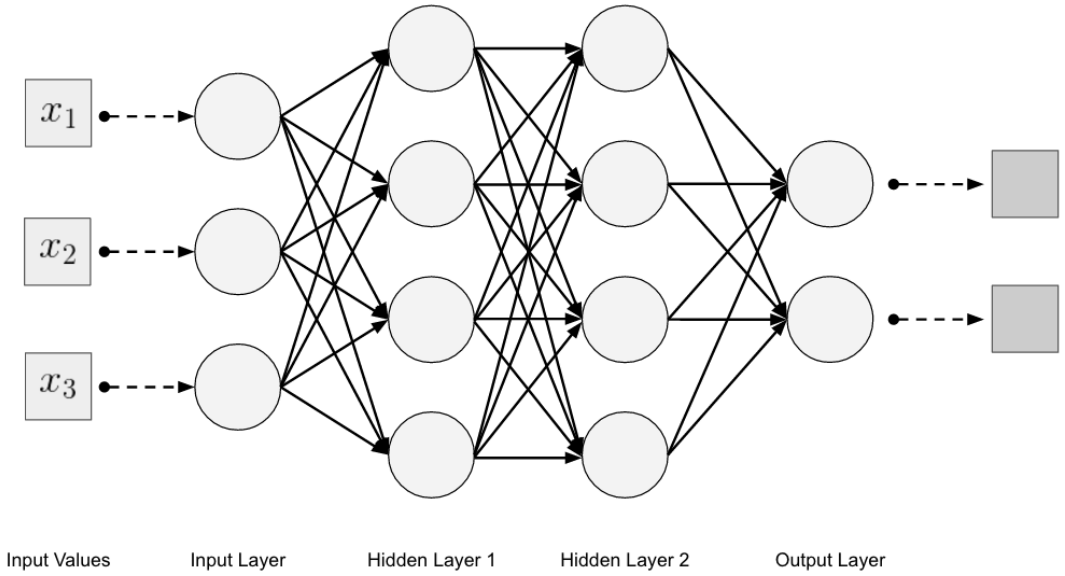

圖 6 多層感知器
(https://www.oreilly.com/library/view/deep-learning/9781491924570/ch04.html)

3.3 初始化方法

在 3.2 節裡有提到，我們的輸入如公式(1)所示，其中權重的選擇方式即稱作初始化方法。一般來說，權重通常使用均勻分布去隨機選取，這裡我們參考 tensorflow 默認的初始化方法，其公式與定義如下:

$$f(x) = \begin{cases} \frac{1}{b-a} & for\ \ a \le x \le b \\ 0 & elsewhere \end{cases} \tag{5}$$

其中 a 和 b 的選取採用以下方法:

$$a = -b = \sqrt{\frac{6}{(fan_{in} + fan_{out})}}$$

fan_{in}是輸入權重張量的數量，fan_{out}是輸出權重張量的數量。

由於使用默認的方法會有收斂速度慢或者更嚴重的難以收斂等問題。所以參照 [5]裡所提之方法，我們使用了何初始化(He initialization)來與傳統方法進行比較。何初始化是一個平均值為 0 變異數為$\sqrt{2/n}$的高斯分布，其數學式如下:

$$w \sim G\left(0, \sqrt{\frac{2}{n}}\right) \tag{6}$$

至於為什麼不用 Xavier 初始化方法，是因為其推導假設活化函數是線性函數。很顯然的，我們所使用的 ReLU 和 ELU 函數皆不屬於線性函數，故在本論文中並無將此方法納入討論與實驗中。

3.4 Batch normalization

在訓練一個深層網路時，除了在 3.3 節討論的收斂問題外，還需要面對梯度消失、爆炸等問題。特別是網路越深問題越容易出現。為了解決，本論文採用了[6]文章內所使用的方法: Batch normalization(BN)。

BN 的計算細節於 [6]中已經講解詳細，在此不再贅述。此節主要介紹使用此方法的優點。神經網路最主要的作用為學習數據分佈，所以一旦測試數據與訓練數據分佈不同，其網路泛化能力便會降低。BN 即是處理此問題的方法，於每層輸入前加入 BN 層可以讓每層屬於同一分佈。此舉還可以大大提升網路訓練速度。此外，深層網路還很容易有過度擬合問題產生，BN 之前大多採取 dropout、L2 正則化方法來解決。在使用 BN 之後有無 dropout 層已經無影響，因為 BN 提高了模型的泛化能力解決了過度擬合的問題。

3.5 卷積神經網路

卷積神經網路(Convolution neural network, CNN)主要作用於特徵擷取，其功能強大之處於高維、複雜的空間中依然作用良好。運作概念如圖 7 所示，其中紅色方形稱為一個 kernel 或叫卷積核，卷積核內會有許多不同權重來決定此核的卷積方式，其中權重皆可進行訓練，卷積核的卷積方式為滑動此核來做數據做特徵擷取。本論文主要在實驗卷積網路的輸出數或稱作特徵圖數、卷積網路的層數對辨識結果的影響。

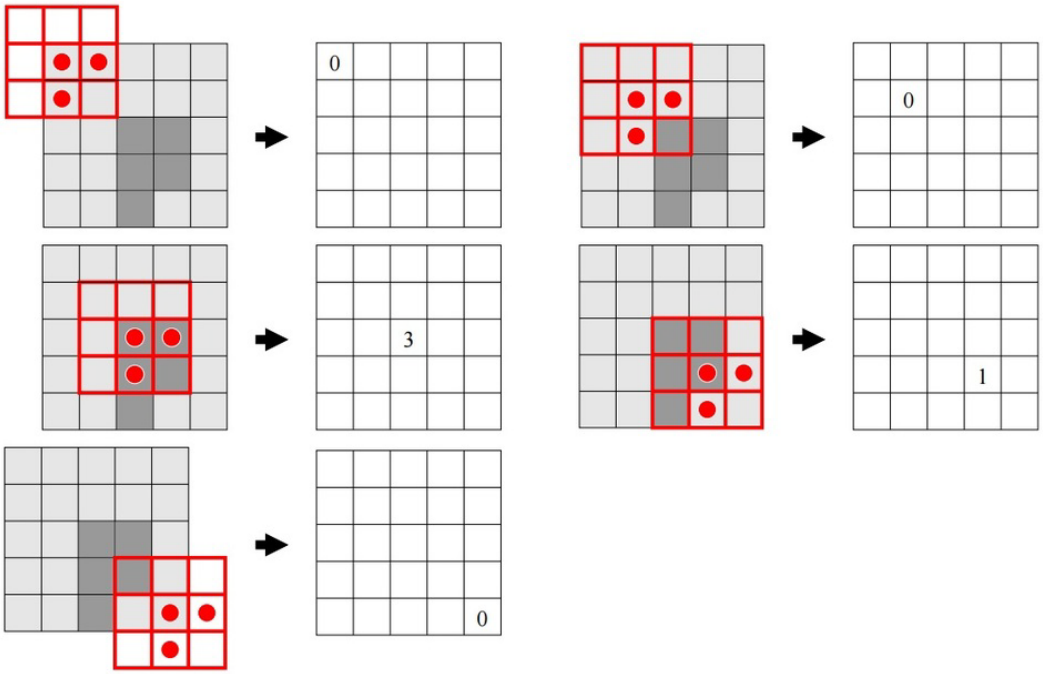

圖 7 神經網路卷積過程

(https://sflscientific.com/data-science-blog/2015/12/4/image-recognition-getting-value-from-visual-data)

3.6 池化層

此層存在目的為資料降維，語音資料與影像資料類似，通常位置相近的資料其相關性較高，故使用最大池化層在降維過後依然可以保留良好的特徵資訊，最大池化層也是本論文所使用的池化方法。池化層作用過程與卷積層類似，圖 8 為其示意圖。

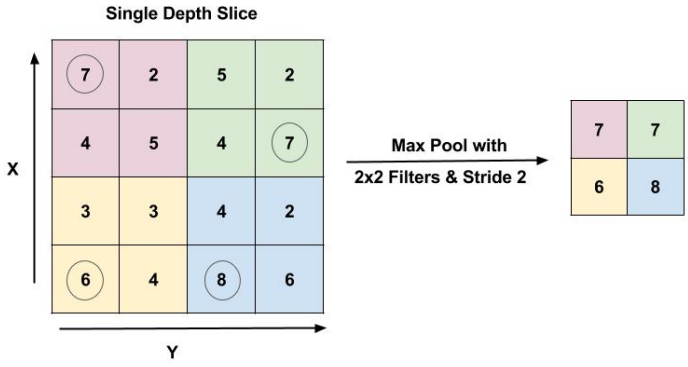

圖 8 池化

(https://www.learnopencv.com/image-classification-using-convolutional-neural-networks-in-keras/)

3.7 卷積層結構

本論文模型結構主要是使用卷積層加上全連接層來辨識中文單音，其中我們定義一層卷積層包含以下動作，如圖 9 所示。此定義為參照一般卷積網路架構之設定，在每一次卷積後都需加入 BN 層，因為我們使用之資料非常龐大，不對每層做標準化會導致 3.4 節所提及的問題造成網路無法訓練，最後才使用池化的原因為避免資料被過度降維遺失了包含在其中的資訊。由於層數、特徵圖數、神經元個數皆是我們實驗的目標，故更詳細的模型架構以及其對辨識成果的影響將會在第四章裡詳述。

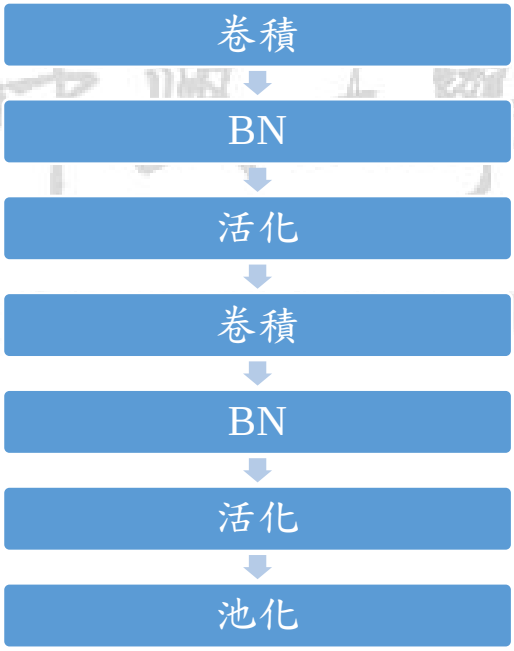

圖 9 本論文使用之卷積層

第四章　實驗與結果

此章節將說明各實驗所使用之模型架構與想法，並討論與比較各結果以獲得最終最高辨識率之模型。

4.1 實作軟體

本論文使用 Python 程式語言中 GPU 版本的 tensorflow 套件，Python 相較於傳統統計上常用的 R 語言，有運算速度更快更適合處理巨量資料等特性。並且 tensorflow 有許多機器學習相關的資源可以使用，對於建造機器學習模型十分便利，在搭配 GPU 版本可以讓運算速度更加快速，故本論文採用此軟體實作。

4.2 資料來源

本論文所使用之資料庫是由二十位語者所錄製之語音資料，總共有 1391 個中文單音類別。每一位語者會對同一單音重複錄製十次，其中我們固定抓第四次錄製資料作為我們的驗證組，即此組資料不參與訓練只當作辨識用途。本論文主要辨識目標為中文單音辨識，但除了單音外我們也會對子音、母音進行辨識。詳細過程將會於 4.3 節詳述。

4.3 主要模型設計

本論文辨識目標為 1391 個中文單音，但直接做 1391 個分類問題可能導致辨識結果不夠好。故本論文的模型設計為:

(1) 先使用同一模型同時分辨子音(36 類)、分聲調母音(160 類)的類別。

(2) 依照辨識結果可能性分別由高至低排列，開始組合成單音。

(3) 刪除不存在於 1391 個類別中的結果。

(4) 取前十高的辨識結果，此即我們的 TOP10 辨識率。

此模型設計可以避免直接做 1391 個類別過多所產生的問題，並且使用同一模型同時對子音以及母音做分類，可以節省運算資源以及時間，不用再分別訓練兩個模型再組合結果。此模型的架構圖如圖 10 所示。

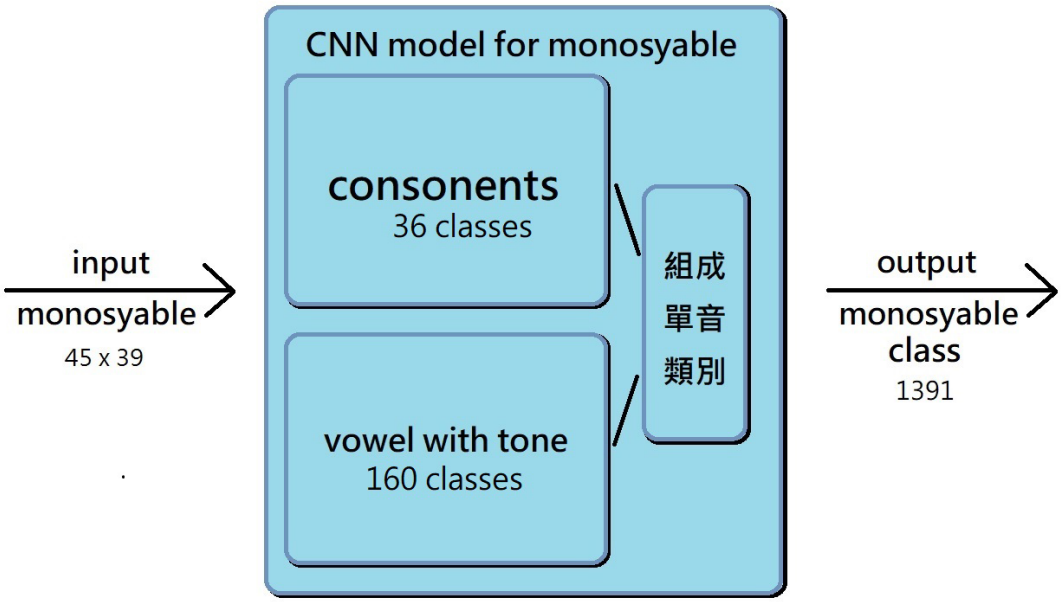

圖 10 本論文之模型設計

4.4 超參數最佳化

超參數(hyperparameter)是指需要被人工調整的參數，相對於神經網路裡的參數，例如: 權重、偏誤等等，這些參數是可以透過倒傳遞讓網路自己訓練修正的。相反的，超參數一般是人工調整、測試結果才能找出最好的配置。本論文主要討論之超參數最佳化有以下幾個面向:

(1) 卷積層深度

(2) 卷積層寬度

(3) 活化函數

(4) 初始化方法

一般而言，造一個越深的網路辨識效果應該是更好的。但是，實例上來說，一個網路架構到一定深度再繼續增加對於辨識結果的提升幾乎微乎其微，多餘的層數

反而佔據了計算資源。在一些研究中也有顯示，層數深度夠時影響的關鍵因子反而是「寬度」。故在深度測試後接著進行寬度，也就是特徵圖數的測試。這裡補充一下，本論文在進行深度測試時，下一層的特徵圖數皆是上一層的兩倍。接下來，對活化函數與初始化方法進行檢測。由於時間成本的問題，本論文會挑選一個較小的模型進行此兩種方向的測試。

4.5 實驗結果

4.5.1 深度測試

本小節將對模型的深度進行實驗，由於本論文專注的是 CNN 層對辨識結果的影響，故全連接層在每次實驗中皆是固定的層數與神經元數。表 1 為實驗結果，卷積層的數字代表特徵圖的數目，TOP5 辨識率表示最高的前 5 個辨識結果包含真實分類的辨識率，其他依此類推。子音以及母音的辨識結果我們將放在附錄供大家參考。由表 1 可以看到在只有兩層時辨識結果只有八成而已，原因是卷積層數不夠造成辨識結果不夠好，在第三、四、五層時辨識結果已經可以將近九成。但是，可以看到從第三層開始，加層數對辨識率並沒有明顯上升，故可以知道在層數四層左右深度已經足夠。所以在接下來的寬度測試，我們使用三層、四層和五層的網路進行寬度的測試。

表 1 單音深度測試正確率之結果

卷積層	TOP1	TOP3	TOP5
(8, 16)	0.8079	0.8154	0.8227
(8, 16, 32)	0.8972	0.9670	0.9680
(8, 16, 32, 64)	**0.8994**	0.9708	0.9740
(8, 16, 32, 64, 128)	0.8979	**0.9717**	**0.9754**

4.5.2 寬度測試

在一開始，由於時間成本的因素，我們先使用三層的卷積網路來進行寬度測試。在測得最佳的特徵圖配置後，開始比較三層、四層和五層的辨識結果。由表 2 可以發現，在三層配置時，一開始增加特徵圖數對辨識結果獲得了很好的提升，從 89.59%增加至了 91.13%上升了約 1.5%左右的辨識率，但在不增加層數的限制下，辨識率上升率從 1.5%開始慢慢下降。到最後兩項實驗配置(64, 128, 256)和(128, 256, 512)時，其辨識率只有比前一層多出 0.65%和 0.3%。故本論文認為三層的寬度已經接近其極限，接著進行四層特徵圖的測試。我們接續著三層的測試結果，在第四層時直接使用三層辨識率最高的配置。我們可以發現在第三層到第四層時辨識率上升了 0.76%，比第三層最後的上升率高了兩倍左右。由於時間成本和設備上的考量，本論文特徵圖最高的設置為 2048，故在四層配置中第二次實驗已經到達最高配置。在四層的最佳配置中，可以看到我們最終的辨識率為 94.15%的成果。最後我們也有實驗五層的配置，但結果如 4.5.1 深度測試的結果一致，辨識率反而比四層的還要低。故經過 4.5.1 和 4.5.2 兩節的實驗可以驗證我們一開始的假設：深度足夠時，寬度反而是影響的關鍵。

表 2 單音寬度測試正確率之結果

卷積層	TOP1	TOP3	TOP5
(8, 16, 32)	0.8959	0.9667	0.9701
(16, 32, 64)	0.9113	0.9694	0.9721
(32, 64, 128)	0.9223	0.9713	0.9733
(64, 128, 256)	0.9288	0.9728	0.9748
(128, 256, 512)	0.9318	0.9727	0.9749
(128, 256, 512, 1024)	0.9394	0.9765	**0.9788**
(256, 512, 1024, 2048)	**0.9415**	**0.9766**	0.9784
(128, 256, 512, 1024, 2048)	0.9397	0.9761	0.9782

4.5.3 活化函數測試

本節主要探討活化函數對辨識率的影響，主要參考的文章為 [9]，此篇文章提出了新的活化函數: ELU。並且在 [9]的實驗結果內也獲得了不錯的成效，故本論文也將此活化函數放入實驗結果討論，其數學式細節已於 3.2.1 節討論過在此不再贅述。

本節實驗用模型街採用卷積層三層、特徵圖數依序為(32, 64, 128)的卷積網路，加上三層的全連接層進行實驗。原因為節省時間成本，並且在辨識率依舊維持一定水準的情況下進行實驗。

由表 3 可以發現，雖然 ELU 在其發表文章中表現都高於 ReLU，但在本論文的資料集中，ReLU 表現都比 ELU 還要好的。不過值得注意的是 ELU 在收斂速度上是比 ReLU 還要快的，但本論文採用的依舊為 ReLU 函數，因為本論文認為辨識率為最重要的，在沒有巨大的時間差距下，還是以辨識率為第一考量。至於 Sigmoid 函數可以看到，雖然時間花費較少但辨識結果差距較大。ELU 差距 0.26% 但 Sigmoid 差距則有 1.04%，其原因已在 3.2.1 討論過，在此只作為如今少使用此活化函數的驗證。

表 3 單音活化函數測試正確率之結果

活化函數	TOP1	TOP3	TOP5	Time(sec)
Sigmoid	0.9122	0.9687	0.9718	7065.83
ReLU	**0.9226**	**0.9729**	**0.9754**	**7337.43**
ELU	0.9207	0.9683	0.9709	7085.57

4.5.4 初始化方法

本節主要在探討權重的初始化方法對於辨識結果的影響，在 3.3 節我們討

論過 tensorflow 默認的初始化方法和我們想要比較的 He 初始化方法的數學式。此節則是就實驗結果進行討論。在表 4 中我們使用了較小的網路，只包含三層的卷積層和三層全連接層。即使如此，我們還是可以看到 He 初始化也是比默認的方法便是率還要高，雖然並無顯著上升但本論文還是採用能增進辨識結果的方法。並且在 [5]裡面也有提到，當網路越深 He 初始化的方法對於收斂的幫助越好，故本論文最終採用 He 初始化方法。

表 4 初始化方法測試正確率之結果

初始化方法	TOP1	TOP3	TOP5	Time(sec)
Default	0.9216	0.9716	0.9741	7065.83
He	0.9225	0.9704	0.9726	7085.57

4.5.5 最終模型

經過 4.5.1 節~4.5.4 節對模型結構以及超參數的測試之後，我們最後採用的模型如表 5 所示。與之前實驗方式不同，在測試最終模型時，我們把實驗的訓練週期(Epoch)提高到 500 輪。如前所說，越長的訓練時間理論上可以獲得越好的訓練成果。本論文並未設置提前終止條件，但為了避免最後一次訓練結果並非最佳值，本論文在每次訓練完後會比較辨識結果並且如果辨識結果比原本的更好會輸出模型參數資訊。在訓練完 500 輪後會使用最佳模型參數資訊來進行辨識，如此即可得到這 500 輪訓練中最好的參數和最佳的辨識結果。本實驗結果在第 478 輪得到了最好的辨識率，可見在 478 輪後面的訓練模型已經學習不到更有用的資訊，故本論文沒有繼續提高訓練輪數進行實驗。在圖 11 中也可以觀察到，模型學習力十分良好，在大約 100 輪訓練內就已經開始收斂，其中母音的辨識率又大約高於子音 1%左右。雖然模型在 100 輪左右就開始收斂，本論文為了獲得更好的訓練結果依舊是提高訓練輪數至 500 輪，也確實於 478 次訓練獲得了最好的結果。表 6 為使用最終模型得到的辨識率結果，如前所述母音

辨識率幾乎高於子音 1%，如何改善子音辨識率可以與母音相當或許是未來可以研究的一個方向。

表 5 模型資訊

卷積層	(256, 512, 1024, 2048)
全連接層	(512, 1024, 2048)
活化函數	ReLU
初始化方法	He
Epoch	500

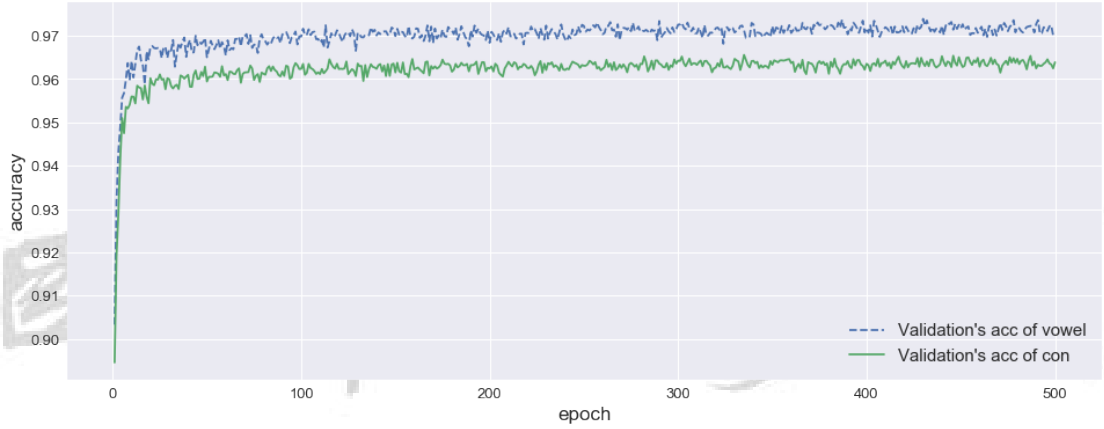

圖 11 子母音辨識率變化

表 6 最終辨識正確率之結果

	TOP1	**TOP3**	**TOP5**	**TOP10**
子音	0.9649	0.9966	0.9986	0.9997
母音	0.9740	0.9975	0.9988	0.9996
單音	0.9449	0.9666	0.9680	0.9707

最後，我們將與前人的辨識率進行比較。在表 7 中，我們比較了 [4]的模型辨識結果，模型 I 為直接對 1391 個單音類別進行分類，模型 II 為搭建兩個模型分別對子音、母音進行辨識再組合成單音，模型 III 則為對子音進行辨識時先考慮母音，最後模型 I+II+III 則為模型投票。在 [4]中辨識率是使用 TOP1 故我們只對 TOP1 辨識率進行比較。可以看到本論文使用之模型在辨識率方面獲得了很

大的提升， [4]中使用了模型投票作為最終輸出模型，本論文只使用了單個模型不但獲得了更好的辨識率，也省下了搭建其他模型的時間成本。不過在將來為了提升整體辨識率，搭建多個模型進行投票或許是一個可行的方法。

表 7 辨識結果的比較

	TOP1	TOP3	TOP5	TOP10
本論文使用模型	**0.9449**	0.9666	0.9680	0.9707
比較模型 I	0.8289			
比較模型 II	0.8276			
比較模型 III	0.8046			
比較模型 I+II+III	0.8405			

第五章結論

5.1 總結

本論文主要探討 CNN 網路各種超參數的組合對於 1391 個中文單音辨識率的影響，本小節主要總結在 4.5 節中各項實驗的結果，和分析與前人辨識結果不同的原因。

1. 首先，對於一個網路最重要的還是其深度。在許多文章內都有表示，深度不夠時模型擬合能力不足，造成辨識成果低落，也與我們的表 1 結果一致。在深度足夠時，寬度變成了影響模型的主要原因。增加足夠多的特徵圖，可以讓模型更好的擷取特徵，進而改進辨識結果。
2. 除了深度與寬度之外，活化函數的選擇也是相當重要，如表 3 所表示，好的活化函數可以在相同的寬度與深度之下改進模型的辨識率，雖然說本文使用的 ReLU 函數增加了些訓練時間，但是本論文認為其增加的訓練時間並不顯著是可以接受的時間成本。
3. 除了模型自身結構的變動之外，訓練時間也是影響辨識率的一大要素，我們最後於 4.5.5 節把最終模型訓練次數拉長至 500 輪，可以看到我們獲得了最好的辨識結果，當然也有因素是我們使用了最佳的超參數配置，不過這並不影響訓練時間對於辨識結果的重要性。
4. 在 4.5.5 節我們和前人的辨識結果進行了比較，本論文認為獲得辨識率的進步與我們於 4.3 內討論的原因有關，直接進行 1391 個單音辨識容易造成辨識錯誤產生，先進行子音母音的辨認最後再組合這種方法較符合現實，並且也獲得了較好的辨識成果。

5.2 改善與建議

1. 選取好的特徵值一直都是建模過程中很重要的一環，本論文是使用 MFCC 特徵來做為輸入。但 CNN 也是擅長汲取特徵的模型，或許未來可以嘗試不透過 MFCC，直接使用 CNN 網路對資料進行特徵的汲取。
2. 可以搭建多個網路，以不同角度去分析辨識單音，再搭配模型投票進行辨識。可能可以把目前的辨識率更加往上提升。
3. 本論文主要使用 CNN 網路加上全連接層進行辨識，但 CNN 主要功能還是特徵汲取，如果搭配上 RNN 再進行單音的辨識，或許可以再增加辨識率。
4. 本論文做的是單音辨識，其性質較相近於分類問題。但語音往往是連續一段，並且具有意義的句子。故未來如果能加入 LSTM 讓模型對資料具有記憶性，進而讓機器「理解」一段話的語意，這是相當值得研究的部分。

參考文獻

[1] A. Ahad, A. Fayyaz and T. Mehmood, "Speech recognition using multilayer perceptron," *IEEE,* pp. 103-109, 2002.

[2] A. Graves, A.-r. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural network," *IEEE International Conference on Acoustics, Speech and Signal Processing,* 2013.

[3] Y. Qian, M. Bi, T. Tan and K. Yu, "Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* 12 2016.

[4] 鄒振宏 且 李宗寶, "利用 CNN 類神經法於中文單音之辨識," 國立中興大學統計學研究所, 2018.

[5] Kaiming He, Xiangyu Zhang, Xiangyu Zhang, Jian Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," 2015.

[6] Sergey Ioffe, Christian Szegedy, "Batch Normalization: Accelerating Deep

Network Training by Reducing Internal Covariate Shift,” 2015.

[7] 蘇木春 且 張孝德, 機器學習：類神經網路、模糊系統以及基因演算法則。修訂二版, 全華, 2003.

[8] 蘇奕銘 且 李宗寶, 應用 MLP、RBF 及 DNN 類神經網路方法於中文母音辨識。碩士論文, 台中: 國立中興大學統計學研究所, 2016.

[9] D.-A. Clevert, T. Unterthiner and S. Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” 2015.

附錄

附錄 表 1 子音深度測試結果

卷積層	TOP1	TOP3	TOP5
(8, 16)	0.9102	0.9688	0.9746
(8, 16, 32)	0.9309	0.9939	0.9978
(8, 16, 32, 64)	**0.9342**	0.9939	0.9979
(8, 16, 32, 64, 128)	0.9324	**0.9940**	**0.9983**

附錄 表 2 母音深度測試結果

卷積層	TOP1	TOP3	TOP5
(8, 16)	0.8079	0.8154	0.8227
(8, 16, 32)	0.9516	0.9955	0.9986
(8, 16, 32, 64)	**0.9539**	**0.9967**	**0.9989**
(8, 16, 32, 64, 128)	0.9522	0.9959	0.9987

附錄 表 3 子音寬度測試結果

卷積層	TOP1	TOP3	TOP5
(8, 16, 32)	0.9299	0.9935	0.9980
(16, 32, 64)	0.9427	0.9950	0.9981
(32, 64, 128)	0.9512	0.9950	0.9985
(64, 128, 256)	0.9539	0.9962	0.9988
(128, 256, 512)	0.9569	0.9961	0.9985
(128, 256, 512, 1024)	0.9616	0.9959	0.9988
(256, 512, 1024, 2048)	**0.9640**	0.9970	**0.9990**
(128, 256, 512, 1024, 2048)	0.9633	**0.9971**	0.9987

附錄 表 4 母音寬度測試結果

卷積層	TOP1	TOP3	TOP5
(8, 16, 32)	0.9507	0.9956	0.9988
(16, 32, 64)	0.9587	0.9967	0.9987
(32, 64, 128)	0.9625	0.9971	0.9988
(64, 128, 256)	0.9666	0.9975	0.9990
(128, 256, 512)	0.9663	0.9967	0.9985
(128, 256, 512, 1024)	0.9710	**0.9978**	0.9989
(256, 512, 1024, 2048)	**0.9715**	0.9973	0.9989
(128, 256, 512, 1024, 2048)	0.9696	0.9972	0.9990

附錄 表 5 子音活化函數測試結果

活化函數	TOP1	TOP3	TOP5	Time(sec)
Sigmoid	0.9440	0.9938	0.9975	7065.83
ReLU	**0.9527**	**0.9956**	**0.9982**	**7337.43**
ELU	0.9467	0.9942	0.9978	7085.57

附錄 表 6 母音活化函數測試結果

活化函數	TOP1	TOP3	TOP5	Time(sec)
Sigmoid	0.9560	0.9955	0.9984	7065.83
ReLU	**0.9609**	**0.9963**	**0.9988**	**7337.43**
ELU	0.9628	0.9956	0.9981	7085.57

附錄 表 7 辨識錯誤之預測結果

1. NAME & WORD	**2. 注音**	**3. TOP1-(0.9443), TOP2-(0.9646), TOP3-(0.9666), TOP4-(0.9672), TOP5-(0.9680), TOP6-(0.9684), TOP7-(0.9690), TOP8-(0.9697), TOP9-(0.9703), TOP10-(0.9707)**
M13-4-八	ㄅㄚ	阿-ㄚ(0.9999), 八-ㄅㄚ(0.5001), 發-ㄈㄚ(0.5000), 紮-ㄗㄚ(0.5000), 擦-ㄘㄚ(0.5000), 撒-ㄙㄚ(0.5000), 媽-ㄇㄚ(0.5000), 殺-ㄕㄚ(0.5000), 差-ㄔㄚ(0.5000), 拉-ㄌㄚ(0.5000)

F20-4-發	ㄈㄚ	八-ㄅㄚ(1.0000), 吧-ㄅㄚ˙(0.5000), 蔔-ㄅㄛˊ(0.5000), 唄-ㄅㄟˊ(0.5000), 寶-ㄅㄠˇ(0.5000), 爸-ㄅㄚˋ(0.5000), 播-ㄅㄛˋ(0.5000), 敗-ㄅㄞˋ(0.5000), 貝-ㄅㄟˋ(0.5000), 抱-ㄅㄠˋ(0.5000)
⋮	⋮	
M25-4-子 B	ㄗˇ	子 a-ㄗˇ(0.7853), 子 b-ㄗ˙(0.7147), 組-ㄗㄨˇ(0.5000), 駔 a-ㄗㄤˇ(0.5000), 怎 b-ㄗㄣˇ(0.5000), 攢 b-ㄗㄢˇ(0.5000), 早-ㄗㄠˇ(0.5000), 仔-ㄗㄞˇ(0.5000), ㄗㄚˇ-ㄗㄚˇ(0.5000), 走-ㄗㄡˇ(0.5000)
M08-4-ㄙ˙	ㄙ˙	匙-ㄕ˙(1.0000), 脽-ㄕㄨㄟˊ(0.5000), 蛇-ㄕㄜˊ(0.5000), 神-ㄕㄣˊ(0.5000), 熟-ㄕㄡˊ(0.5000), 誰-ㄕㄟˊ(0.5000), 叔 a-ㄕㄨˊ(0.5000), 繩-ㄕㄥˊ(0.5000), 勻-ㄕㄠˊ(0.5000), 啥-ㄕㄚˊ(0.5000)

註:

1. 第一列分別代表: 聲音來源(人)-第 i 次錄音-正確單音。
2. 第三列列名代表 TOP1~10 的辨識率，列裡則是模型預測出的單音括號內為模型預測之機率。
3. 總共有 697 列這裡只挑選一小部份作為展示。