國立中興大學統計學研究所

碩士學位論文

利用 CNN 類神經法於中文單音之辨識

The Mandarin Monosyllable Recognition by Using the Method of Convolutional Neural Network

指導教授：李宗寶　Chung-Bow Lee

研 究 生：鄒振宏 Chen-Hung Tsou

中華民國一○七年八月

國立中興大學統計學研究所

碩士學位論文

題目：利用 CNN 類神經法於中文單音之辨識

姓名：鄒振宏

學號：7105018033

經 口 試 通 過 特 此 證 明

論文指導教授 李宗寶

論文考試委員

郭仁豪

邱國欽

李宗寶

中華民國 107 年 7 月 5 日

摘要

本論文主要探討卷積類神經網路（Convolutional neural network, CNN）在中文單音上的辨識。將 20 個不同語者所錄製的 1391 個單音，進行數位採樣、音框切割、視窗化等一系列的前處理後，取得梅爾倒頻譜係數（Mel-Frequency cepstral coefficients, MFCC）作為模型的輸入特徵。本方法將利用卷積、池化、批標準化等過程，對原始特徵做進一步的擷取，最後再輸入多層感知機（Multilayer perceptron, MLP）進行分類。除了將全部 1391 個單音直接分類外，也嘗試了其他的分類方法，如先分母音、再分子音的模型設計，或者進一步將母音聲調作分類，共 3 個主要模型，辨識率分別為 82.89%、82.76%、80.46%。最後再透過模型不加權投票，得到最佳辨識率 84.05%。

關鍵字: 多層感知機, 梅爾倒頻譜係數, 卷積類神經網路, 語音辨識, 機器學習

Abstract

This thesis mainly discusses the speech recognition using CNN(convolutional neural network) in Chinese monophonic. MFCC(Mel-Frequency cepstral coefficients) were obtained as models after a series of pre-processing such as digital sampling, frame cutting and windowing were performed on a total of 1391 single tones recorded by 20 different speakers as the input of model. This method will use the convolution, pooling, batch normalization and other layers to further extract the original features, and finally input the MLP(multilayer perceptron) for classification. In addition to directly classifying all 1391 monophonic, other classification methods have been tried, such as model design of first denominator and re-molecular sound, or further classification of vowel tones. There are 3 main models with recognition rates of 82.89. %, 82.76%, 80.46%. Finally, through the model unweighted voting, the best recognition rate is 84.05%.

Keywords: MLP, MFCC, CNN, Speech recognition, Machine learning

目錄

附圖目錄

附表目錄

第一章　緒論

1.1 研究動機

許多數十年前即被提出，惟礙於計算量太大在當代硬體無法實現的演算法，如類神經網路、機器學習、深度學習等，漸漸地被實作出來，也在語音辨識、影像辨識、自然語言處理取得許多意想不到的成果，大數據、資料科學、人工智慧等名詞應運而生。

其中在影像辨識方面最主要的成果，幾乎都來自於卷積類神經網路（Convolutional neural network, CNN）以及其衍生模型。CNN 比起多層感知機，在前端多了數個「卷積層」，每一層中又利用數個「卷積核」(kernel)的轉換，建立更具結構化的特徵，最後再透過多層感知機分到所屬類別。

而在語音辨識中，原始的語音資料是一段連續的聲波，經過離散化、切割音框等前處理後，得到類似矩陣的 2 維特徵。由於其本身來自於連續的波，我們合理假設，轉換後的特徵，與影像資料有相似的結構。即矩陣中的元素間具相關性，以此為嘗試 CNN 模型的契機。

雖然與傳統統計方法，如多變量分析比起來，類神經模型有缺乏解釋能力、不容易一般化等缺點待克服；但在語音辨識這種高頻、高維度，並且重視預測能力優先的問題上，有相當強的實用性。

1.2 相關研究

隨著硬體技術的發展，以及 Cuda、Spark 等運算框架的蓬勃發展，許多龐大計算量的演算法得以實現，類神經相關演算法尤是。從簡單的 MLP 到 DNN(deep neural network)、CNN、RNN(recurent neural network)等模型，接連被應用在語音、影像、自然語言處理等領域上，取得了不俗的成果；更快、更有效率的演算法也不斷地被提出。除了類神經網路的方法之外，還有其它不同的方法如最近鄰居法(K-nearest neighbor, KNN)、支持向量機(Support vector machine, SVM)等也能夠用

來語音辨識，也都有顯著的成效。

早在 2002 年，Abdul Ahad et al.[4]即使用最簡單的 MLP 搭配倒傳遞(Back propagation, BP)修正梯度，在語音資料上發展。此時的電腦還無法負荷較複雜的模型，實驗也採用較簡單的 0~9 的數字辨識任務為主。

Alex Graves et al.[3] 在 2013 年因應語音資料連續、隨時間前後相關等特性，使用 RNN 類神經網路進行語音辨識，比起不考慮時序的單純 MLP 提昇了不少性能，達到 17.7%的錯誤率。可惜 RNN 運算量較大且不適合平行運算的特性，暫時還難以實用。

Neelima Rajput et al.[9] 在 2014 年使用倒傳遞類神經網路應用於英文字母的語音辨識。該模型在輸入階段為向前傳遞，計算梯度時卻是向後傳遞，大幅提高了梯度修正時的速度，進而使模型能夠容納更多的神經元個數，辨識率也得到提昇。

Ossama et al.[10] 在 2014 年基於隱藏馬可夫模型(Hidden Makov model, HMM)在英文語音辨識中不錯的成果，歸功語音資料具有一定的結構性，適合使用 CNN 類神經模型。實驗結果也顯示，比起 DNN，卷積層的加入能夠讓模型錯誤率下降 6% - 10%不等。若使用限制波茲曼機(Restricted Boltzmann machine, RBM)對參數進行預訓練，在 DNN 上能讓錯誤率下降約 1.7%；在 CNN 則下降 0.8%。預訓練的效果在 CNN 上較不明顯。

Yanmin Qian et al. 在 2016 年的論文[11]，使用了較深層的 CNN，並與 RNN 中的 LSTM(Long short-term memory)模型進行比較。此時的硬體技術已較為發達，因此能夠試著對各種超參數做更多的探索，包括神經元的深度、厚度、順序等，做了相當多的嘗試，以及各種處理噪聲的方法取出的特徵，對辨識率的幫助。

上述多為單音的研究成果，然而真實的資料多為一段連續的語音。Xu Tian et al.在 2017 年的論文[12]，就使用 LSTM 對連續的語音作辨識，透過對上下文給定不同的權重，找出相關性最高的關鍵字，同時避免了 RNN 在運算上的某些問題。

綜合上述，近年來類神經模型在語音方面的研究、實作，多是基於 CNN 搭配 LSTM，且以兩者搭配截長補短較多。

1.3 語音辨識介紹

1.3.1 何謂語音辨識

語音辨識是人工智慧的一個領域，目標是透過演算法，讓機器能夠將人類的語音內容，轉換成相應的文字。可以視為一種監督式學習(supervised learning)或是分類(classification)問題。亦即在已知答案的情況下，建立一個模型函數

$$f：\text{語音特徵值} \rightarrow \text{相應文字} \tag{1.1}$$

來描述語音資料與文字之間的關係。而所謂的建模則是指，找一個理想的$\hat{f}$作為f的估計量，使得辨識正確率能達到最佳。

1.3.2 語音特性

本節將介紹一些常會對辨識結果造成干擾的語音特性，也就是第二章中資料前處理的濾波目標。

(1) 相同語者:
即使是同一個人發出的相同語音，仍可能受到音量大小、語調、感冒等因素影響，導致發音方式有所改變，影響辨識結果。

(2) 不同語者:
語音發自不同人時，除了前述的干擾依然存在外，更多了腔調、口音等的影響，辨識上比起相同語者更加困難。

(3) 外在環境噪音:
其他例如環境吵雜、錄音設備等過於明顯的雜訊混入語音訊號中，也會增加辨識的困難度。

因此，為了降低這些干擾因子對語音辨識的影響，必須對原始資料做適當的

特徵提取，使得轉換後的特徵值，投影在一個良好的特徵空間上。細節將在第二章介紹。

1.3.3 專有名詞介紹

簡介本論文會用到的語音概念。

(1) 單音(monosyllable)

一個完整的單音節語音，在中文裡一個字即為一個單音。每個單音又可以分解成母音、子音以及音調。

(2) 聲調(tone)

中文是一個分聲調的語言，普通話分為四個聲調，即一般所稱的陰平（第一聲）、陽平（第二聲）、上聲（第三聲）、去聲（第四聲）。

(3) 子音(consonant)

又稱為輔音。在發聲過程中，口腔內氣流明顯的阻塞，此時聽到的聲音稱為子音。其中根據發音方法的不同，又區分成鼻音(通過鼻腔)、塞音(氣流被完全阻塞)、或是近音(近似母音)。如:「八，ㄅㄚ」當中的「ㄅ」即為子音。

(4) 母音(vowel)

又稱元音。在發聲過程中，口腔內的氣流並未阻塞且聲帶產生震動，則稱為母音。如:「八，ㄅㄚ」當中的「ㄚ」即為母音。有些字音又被稱為雙元音，它是由一個元音開始，轉動到另一個元音上面。常見的雙元音有「ㄠ」、「ㄟ」、「ㄞ」、「ㄡ」。

根據以上的發音原理，一個被收錄的單音訊號，結構上前半段是子音、後半段是母音的，在特徵選取時須特別注意。

1.3.4 語音辨識的應用

在物聯網發展飛快的今日，語音辨識的應用場景只能越來越多。舉凡打電

話、發訊息、導航系統，未來說不定都能在一個穿戴式裝置、智慧管家完成。可以說，只要是與「溝通」有關的服務，均能透過語音辨識的技術實現自動化。

若是能搭配自然語言處理(Natural language processing, NLP)相關的技術，能夠實現的服務將變得更加廣泛，一些簡易的醫療問診、法律諮詢等特定封閉領域的問題，均可使用人工智慧來取代，讓機器變的更善解人意。

第二章　語音訊號前處理與語音特徵求取

物理上，語音的本質是一個連續的波，經電腦讀取並數位化後，以一連串離散的數值儲存。我們可以把同一類別的數值資料，看做一個多變量機率分布。然而由於 1.3.2 節中提到的語音特性，若直接使用原始數值作為分類特徵，除了結果不盡理想，消耗的時間成本也極大。

本章節的資料前處理，主要目的不外乎降低維度、去除雜訊，將原始數值投影到適當的特徵空間，讓同一分類的變異降低、不同分類的變異提高。

當語音訊號輸入進行辨識之前，需要經過複雜的前處理過程，因為一段語音訊號的訊息量非常龐大，如果直接使用原始訊號進行辨識，消耗的時間、計算成本將非常驚人；因此，需要將語音訊號經過一系列的前處理，進而轉換成所需的資料。而本章將介紹語音資料的性質，以及如何將其轉換成適當特徵值，當作模型輸入資料。圖 1 為語音訊號前處理的流程。

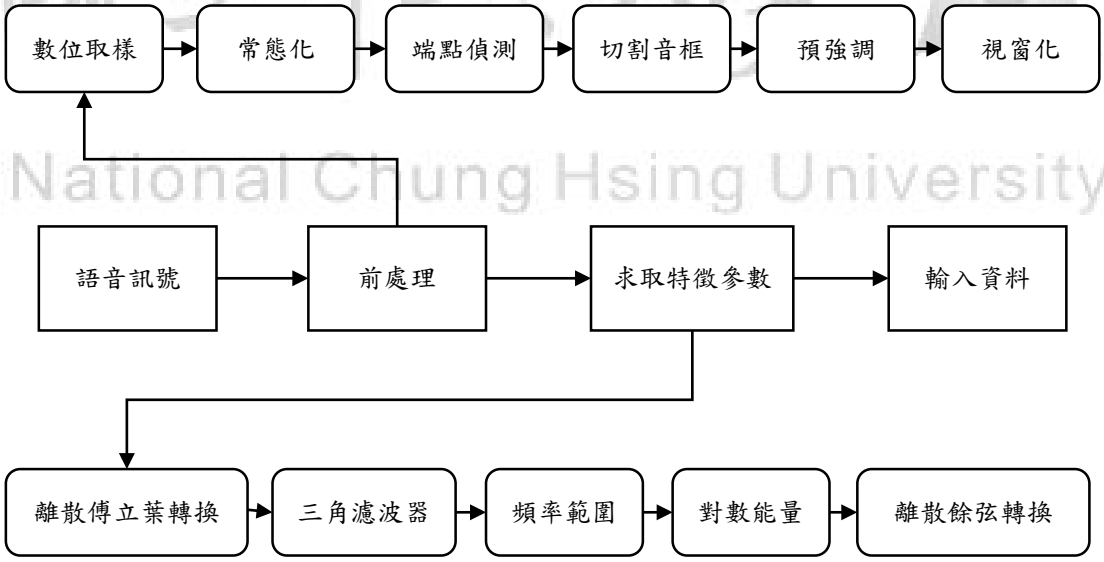

圖 1　語音訊號前處理的流程

2.1 語音訊號

本篇共採用 1391 個中文單音的母音來進行辨識。每個中文單音的組成包含母音以及聲調再加上子音。母音的波型通常較為規律，因此母音的辨識是語音辨識能夠成功很重要的一環。本論文資料庫共有二十位不同語者，在一般環境的情況下，每個中文單音會請每位語者錄製十次，得到的資料即為我們的語音訊號。然而錄製好的語音是屬於電腦無法直接讀取的類比訊號，因此需要對這些資料做適當前處理。

2.2 資料前處理

2.2.1 數位取樣

語音錄製完畢之後，是以類比訊號的方式被記錄下來。類比訊號(analog signal)是指在時域上其形式為連續函數的訊號。因電腦的底層是以 0、1 訊號所構成的二進制運算所構成，若要處理連續型資料，則必須以數值方法做逼近。

數位取樣(digital sampling)就是對一個連續的聲波，沿著時域以特定間隔大小、等距的擷取波的振幅值，最後得到一組向量，作為聲波的特徵。而該間隔大小稱為「取樣頻率」。取樣頻率的選擇，視使用者的需求而定，取樣頻率越大，越接近類比訊號，失真的越少；反之取樣頻率越小則損失越多訊息。本論文所使用之資料庫，取樣頻率皆為每秒 11025 個取樣點。

2.2.2 常態化

收錄語音時，會因為設備差異、音量大小、環境雜音等因素，影響其振幅值的大小，使得取樣值的範圍過於分散。而在語音辨識中，在乎的是波的相對形狀，而非其個別取樣點的數值大小。因此為了消弭這些因素所造成的變異，

我們利用常態化方法，保持波形的同時，將取樣值的範圍從[-32768, 32768]調整至[-10, 10]。

2.2.3 端點偵測

錄製語音的過程中，幾乎不可能收到一段長度剛好的語音，該語音訊號的前後通常會有長短不一的靜音、氣音等不需要的訊號。所以在收錄之後，需要一個判斷起點、終點的方法，確實捕捉到語音訊號，並刪除前後雜訊。本篇論文所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

經過前述處理後，得到的語音資料是一個隨時間變化的向量。為了更利於分析，我們會將語音訊號切割成多個 20~30 毫秒的音框，再將其堆疊起來，成為一個矩陣的結構。

其中每個音框裡面有 256 個樣本點，子音總共會有 24 個音框，母音則有 29 個。根據經驗，我們會捨棄掉子音與母音的前後各兩個，容易有雜訊的音框。

同時，切割音框的動作將造成訊號的不連續性。為了使每個音框的端點附近更平滑，我們會將音框內的每個取樣值乘上一個視窗函數。

2.2.5 預強調

人類耳朵的傳音系統具有高頻強調的特性。為了讓機器模擬這一特性，我們會讓語音通過一個高通濾波器，以彌補傳遞時的損失，這個動作就稱為「預強調」。

2.3 特徵參數的求取

經 2.2 節一系列的前處理之後，我們必須找出代表每個單音的特徵值，本節簡單說明如何求取本論文所使用之梅爾導頻譜係數特徵(Mel-Frequency cepstral coefficient，MFCC)。

2.3.1 離散傅立葉轉換

離散傅立葉轉換主要是將時域上的資料型態轉換為頻域上的資料型態。其公式如(2.1)式：

$$X(k) = \sum_{n=0}^{N-1} S(n) e^{\frac{-2\pi jkn}{N}} \quad , k = 0, 1, \dots, N-1 \tag{2.1}$$

其中$X(k)$為頻域上的樣本，$S(n)$為時域視窗化後的語音訊號，N為音框數。藉由尤拉公式：

$$\begin{cases} e^{jw} = \cos(w) + jsin(w) \\ e^{-jw} = \cos(w) - jsin(w) \end{cases} \tag{2.2}$$

代入傅立葉轉換公式得到下列式子:

$$X(k) = \sum_{n=0}^{N-1} S(n) \cos\left(\frac{2\pi kn}{N}\right) - j \sum_{n=0}^{N-1} S(n) \sin\left(\frac{2\pi kn}{N}\right) = A(k) - jB(k) \tag{2.3}$$

藉由上面換算後的公式我們得到頻譜強度為$|X(k)| = \sqrt{A(k)^2 + B(k)^2}$，相位角為$\theta_k = \tan^{-1}\left(\frac{B(k)}{A(k)}\right)$。

2.3.2 三角濾波器

接下來將語音訊號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，其中第m個三角濾波器通常有兩種選擇分別如式(2.4)、(2.5)所示:

$$H_m[k] = \begin{cases} 0, & k < f[m-1] \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1])}, & f[m-1] \le k \le f[m] \\ \dfrac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f[m+1] - f[m])}, & f[m] \le k \le f[m+1] \\ 0, & k > f[m+1] \end{cases} \tag{2.4}$$

$$H'_m[k] = \begin{cases} 0 & , k < f[m-1] \\ \dfrac{(k - f[m-1])}{(f[m] - f[m-1])} & , f[m-1] \le k \le f[m] \\ \dfrac{(f[m+1] - k)}{(f[m+1] - f[m])} & , f[m] \le k \le f[m+1] \\ 0 & , k > f[m+1] \end{cases} \tag{2.5}$$

$f[m]$代表第m個頻帶中心頻率，M為頻帶的數目。

2.3.3 頻率範圍

首先定義:

- f_1: 三角濾波器組中最低頻率。
- f_h: 三角濾波器組中最高頻率。
- M：濾波器數目。
- N： 音框取樣點數目。
- F_s： 取樣頻率。

頻率範圍公式如(2.6)式:

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_1) + m\frac{B(f_h) - B(f_1)}{M+1}\right) \quad (2.6)$$

其中

$B^{-1}(\mathrm{b}) = 700(\exp\left(\frac{b}{11025}\right) - 1\ )$，所以$B(f) = 11025 \times \ln\left(1 + \frac{f}{700}\right)$。

2.3.4 對數能量

通過三角濾波器組之後我們會將各個頻率的能量乘上(2.3)式並且加總起來，再對其加總的結果取對數值，公式如(2.7)式:

$$S[m] = \ln[\textstyle\sum_{k=0}^{N-1}|X_a[k]|^2 H_m[k]]\ \ , m = 1, 2, \ldots, M \quad (2.7)$$

其中$|X_a[k]|$是頻譜強度。

2.3.5 離散餘弦轉換

我們將從(2.7)式得到的對數能量，對其做離散餘弦轉換，得到的結果就我們所要的梅爾頻率倒頻譜係數(MFCC)，公式如(2.8)式:

$$c[n] = \sum_{m=1}^{M} S[m]\cos\left(\frac{\pi n\left(m - \frac{1}{2}\right)}{M}\right), \quad 0 \le n \le M \quad (2.8)$$

$c[n]$就是我們求得的梅爾倒頻譜係數，M為每個音框中 MFCC 的維度。

第三章　研究方法

3.1 介紹

本章節從簡單的 MLP 出發，到進一步的 DNN、CNN 神經網路模型，介紹各個模型的想法、運作原理以及背後的假設，最後說明本論文的模型設計。另外也針對包括活化函數、優化器、卷積層等元件的使用與調校、超參數的選擇等問題做進一步的探討。

是故本模型設計的重點，主要考慮如何利用卷積層提取具代表性的特徵(3.3 節)、以及如何將不同類別的特徵值(3.2 節)分開兩方面。

3.2 人工類神經網路

人工類神經網路(Artificial neural network, ANN)是機器學習的一種，相較於傳統事先給定規則、再讓電腦進行判斷的人工智慧，該演算法透過模仿生物神經元(neuron)(圖 2)的突觸、刺激、活化、傳遞等運作，希望讓電腦具有生物般的自我學習、修正能力。如此一來，人工智慧可不必再受限於人類知識自行成長，甚至人類可以參考電腦找出的結果，作進一步解釋。

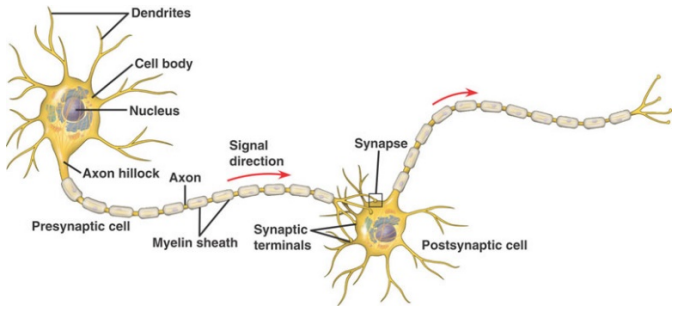

圖 2　生物神經元

(http://biomedicalengineering.yolasite.com/neurons.php)

3.2.1 計算圖

本論文大部分的模型，都使用計算圖(computational graph)表達，所以由此談起。

計算圖由一個個節點(node)所構成，每個節點透過箭頭輸入資料，做完該節點內部定義的運算之後再輸出至下一個節點，舉例如圖 3：

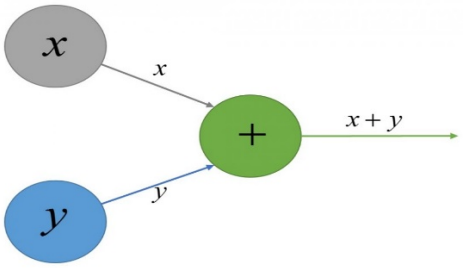

圖 3　計算圖

計算圖在類神經、機器學習等很多演算法中都是很重要的概念，並且不侷限在上述一種描述方法，依模型狀況不同，節點或可展開成另一張計算圖，本論文所使用的實作軟體 Tensorflow 套件，亦是基於計算圖所實作而成。

而人工類神經網路所謂的神經元，即是一種計算圖的縮寫，如圖 3，每一個節點即代表一個神經元。節點裡放的是一個純量，鍵結值則是一般的純量乘法。若資料是向量甚至張量，則用數個節點結構化的表達，即為一般類神經網路的一個「層」(layer)。

一般而言，任何資料均可以表達成張量，語音資料也可以看成一個矩陣或二維張量。依情況不同，可能直接輸入多維度張量結構的資料，或是將張量攤平成一維度的向量再行輸入。

3.2.2 感知機

感知機(perceptron)是一種可堆疊的分類器模型，也是一種簡單的類神經網路。只有單層的話是一個線性分類器，僅能處理線性可分割的資料。一個單層感知機

的數學表達如式(3.1):

$$\mathbf{y} = \mathrm{f}(\mathbf{x}\,;\,\mathbf{w},\mathbf{b}) = \mathrm{act}(\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b}) \tag{3.1}$$

其中，

(1) **x** 代表輸入資料，是一個長度 m 的向量，若資料是多維的張量，則一律攤平為向量。
(2) **y** 代表正確分類，為了後續的演算法，在此我們使用 one-hot 編碼:若資料一共有 k 類，則為長度 k 的向量，除第 k 個元素為 1，其餘為 0。
(3) **w** 為可訓練的權重參數，是一個k × m的矩陣，將長度 m 的 X 映射至長度 k 的 Y。
(4) b 為可訓練的偏移參數，是一個長度 k 的向量，稱為模型的偏移量，相當於一般線性迴歸中的截距項。
(5) act 為活化函數(activate function)，將$\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b}$的運算結果，映射到希望的範圍，常用的有 sigmoid、tanh、relu、softmax 等。

由上可知，單層感知機模型中可訓練的參數僅有 **w**、**b**，故只能描述線性可分的資料，但真實世界的情況絕大多數為非線性資料；這時候只要將數個單層感知機，以合成函數(composite function)的方式疊加起來，就可模擬非線性的情況，稱為「多層感知機」。考慮一個 L 層的多層感知機，如式(3.2):

$$\mathbf{y} = f_L\left(f_{L-1}\left(\cdots f_1(X)\right)\right) \tag{3.2}$$

其中f_1為第 1 層的函數，稱為輸入層(input layer)、f_L為第 L 層的函數，稱為輸出層(output layer)，其他則稱為隱藏層(hidden layer)。輸入層、輸出層須配合資料的維度及類別數設定參數個數；其他隱藏層則有較大的彈性及不確定性，需透過實驗最佳化之。為區別兩者，將上式改寫如式(3.3):

$$\mathbf{y} = f_o(f_h\left(\cdots f_1\left(f_i(X)\right)\right) \tag{3.3}$$

f_i、f_o分別是輸入層、隱藏層，整體稱為一個含有 h 個隱藏層的 MLP。一般所稱的 MLP，多指較為淺層的神經網路(如圖 4)，因早期訓練高層數網路的硬體、方法尚未成熟，沒有能力處理過於複雜的模型；而所謂深度類神經網路(Deep neural network, DNN)則是一個更廣泛的領域，不僅深度增加可達數十、數百層，也慢慢的加入了更多複雜的手法，如卷積層、殘差層、批標準化層…等手法(如圖 5)。可以說，MLP 即是 DNN 的一個簡單特例。在 DNN 中，或者說在「層」這

樣的概念中，我們亦稱 MLP 這種不同層之間，每一個神經元均互相連接的層為 fully connected layer、dense layer。

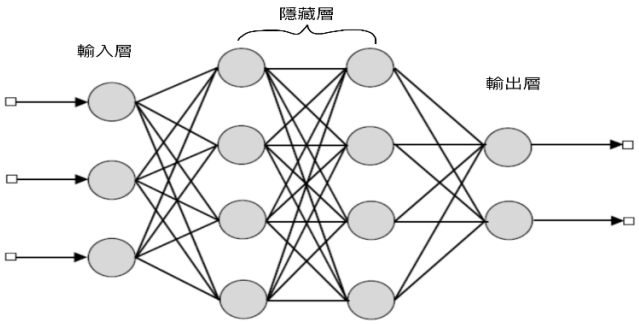

圖 4　MLP

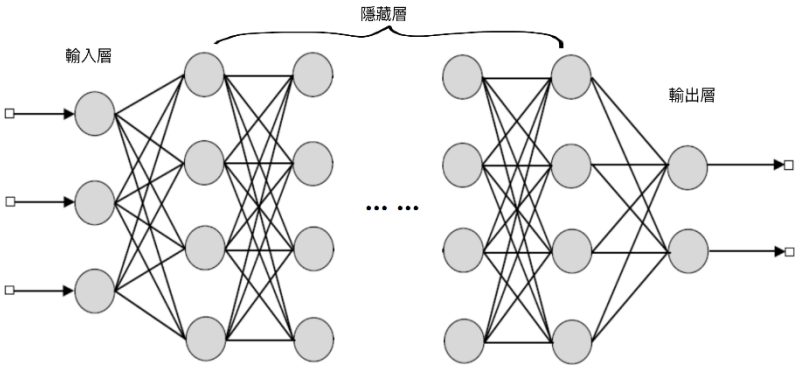

圖 5　DNN

透過層數的不斷加深，模型的參數量、非線性程度也大幅提昇，能夠擬和出更加複雜的情況，對於事物的建模、抽象表達能力更強。亦透過以「深度換取寬度」的想法，增加深度的同時，降低每一層的神經元個數。這樣的設計，能夠在同樣的表達力下，節省參數量、增加建模的效率。然而隨著深度的增加，訓練的難度、效率變成一個挑戰，將在後續章節探討。

3.2.3 活化函數

在不特別限制權重 w 與誤差項 b 的情況下，wx + b的值可能落在實數的任何範圍而難以掌控。尤其深度愈是增加，愈可能落在極端值。活化函數是一個

位在神經元後端，相當於閘門的函數，能夠重新定義該節點的輸出範圍。大多數的活化函數通常是定義在實數上的一對一、非線性、可微分函數，使得控制範圍的同時，保有良好的微分性質，方便最佳化(詳見 3.2.5 節)；非線性活化函數的加入，也提昇了模型對非線性情況的擬和能力。簡單介紹幾個常用的活化函數如(3.4)式、(3.5)式、(3.6)式，圖形可見圖 6。

- Sigmoid

$$y = \sigma(x) = \frac{1}{1 - e^{-x}} \ , \qquad \forall \ x \in R \tag{3.4}$$

早期使用較多，能夠將實數映射至(0, 1)區間，常用來模擬機率值。但由於 sigmoid 在極大、極小值時容易過早飽和，造成梯度過小而無法對參數修正、模型收斂緩慢，現已較少使用。

- Tanh

$$y = 2\sigma(2x) - 1 = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \ , \qquad \forall \ x \in R \tag{3.5}$$

tanh 可視為 sigmoid 的一種變形。值域落在(-1, 1)且均值為 0，使其可往正、負方向修正。雖較 sigmoid 佳，但過早飽和的問題依然存在。

- ReLU(Rectified linear unit, ReLU)

$$y = \max(0, x), \qquad \forall \ x \in R \tag{3.6}$$

一般生物神經元在刺激未達一定強度時不會被激發，必須在達到某一強度時，才會引起神經衝動，稱為「全有全無律」。而 relu 在大於 0 時是一個線性函數，小於 0 時則一律輸出 0，相當於神經元不激發，恰好的捕捉了這個生物神經元此一特性。計算量方面，不僅相較 sigmoid、tanh 的指數運算來的有效率，在梯度下降相關演算法下，模型收斂速度也快上許多。因此近年來，多以 relu 或其衍生(leaky relu、elu、selu)作為多數情況的默認選項。本論文亦使用 relu 作為活化函數。

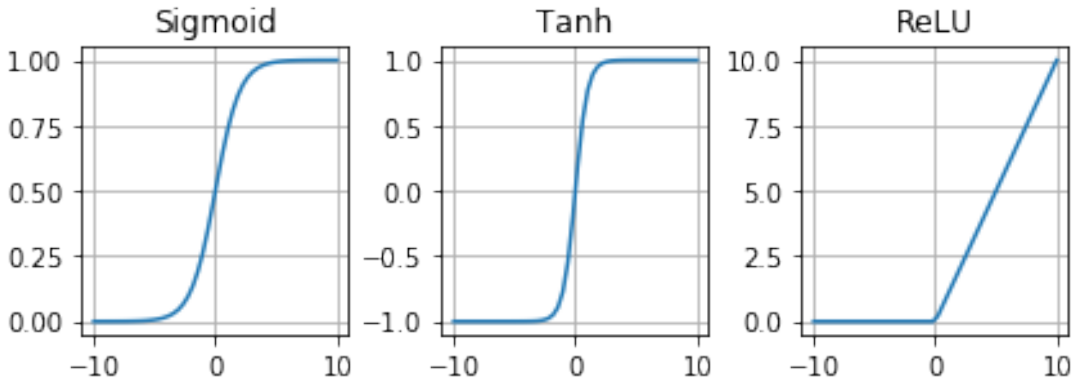

圖 6　常用活化函數

- Softmax

$$\mathbf{Y} = f_i(\mathbf{X}) = \frac{e^{x_i}}{\sum_i^K e^{x_i}}, \qquad for\ \ i = 1, \dots \dots K\ \ ;\ \ X, Y \in R^K \tag{3.7}$$

Softmax 函數的輸入、輸出均是一組長度相同的向量，輸出向量的和為 1 且所有元素均介於 0、1 之間，符合機率空間定義。與其他活化函數用途不同，一般不放在隱藏層中，而是放在分類器模型中的輸出層來模擬機率，並以其值的大小判斷該資料偏向各類別的程度。

3.2.4 損失函數與準確率

為了讓模型能夠有效率、且正確的學習，必須有適當的損失函數作為學習目標。有了式(3.1)的模型後，利用隨機變數的方式，生成隨機一組 **w**、**b** 初始值，得到模型估計，即(3.8)式:

$$\hat{y} = \hat{f} = f\left(X; \hat{\boldsymbol{w}}, \hat{\boldsymbol{b}}\right) \tag{3.8}$$

在(3.8)式之下，為了評估模型的績效，定義損失函數如式(3.9):

$$\text{loss} = \text{L}\left(\text{y}; \hat{f}\right) \tag{3.9}$$

所以為了估計$\hat{f}$，學習目標就變成式(3.10):

$$\hat{f} = \underset{f}{\operatorname{argmin}}\, E(y, loss) \tag{3.10}$$

那如何定義損失函數呢? 在一般迴歸分析中，我們會直接將正確答案 **y** 與估計值 $\hat{\boldsymbol{y}}$對答案，並計算準確率(accuracy)，所以損失函數即為式(3.11):

$$\text{loss} = 1 - \text{accuracy} \tag{3.11}$$

但這樣的損失函數在分類問題中，僅能衡量預測值的正確與否，無法表現出預測值對正確分類接近程度。所以在分類器模型中，一般使用交叉熵(cross entropy)來評估模型的績效，定義如下:

$$\text{cross entropy} = -\sum_{i=1}^{n}\sum_{j=1}^{c} y_i^{(k)} \log\left[softmax\left(f(x_i)\right)_k\right] \tag{3.12}$$

其中 n 為樣本數、c 為分類數，$y_i^{(k)}$則表示當正確分類為第 k 類時，在 one-hot 編碼下第 i 個元素的值。如此一來就能將只有正確與否的二元問題，轉成一個連續的數值做為評估績效，能夠更有效的修正模型參數。

3.2.5 梯度下降、參數最佳化

神經網路中所謂「訓練」、「學習」的目的，就是試著找出能夠盡量降低損失函數的一組模型參數，亦即最佳參數；而解決這種問題的方法，稱為「最佳化(optimization)」。訓練機器學習相關模型時，通常使用梯度下降法(Gradient descent, GD)或其衍生方法。透過往損失函數的負梯度方向前進，以迭代的方式逐漸逼近式(3.10)的目標。考慮:

(1) 損失函數$l(\boldsymbol{\theta})$在$\widehat{\boldsymbol{\theta}}$上可微分。
(2) 所有可訓練參數: $\boldsymbol{\theta} = (\boldsymbol{w}, \boldsymbol{b})$。
(3) 學習率α，決定每次迭代修正的幅度，通常選擇 0~1 之間的正數，影響甚大，將在 3.4.6 節中討論。

迭代步驟如下:
步驟一:隨機生成參數初始值$\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(0)}$，由式(3.11)得到估計值$\hat{f}(\boldsymbol{x}; \boldsymbol{\theta}^{(0)})$，再代入

式(3.9)得到式(3.13):

$$l(\boldsymbol{\theta}) = l\left(\boldsymbol{\theta}^{(0)}\right) = \mathrm{L}\left(\mathrm{y}; \hat{f}\left(\boldsymbol{x}; \boldsymbol{\theta}^{(0)}\right)\right) \tag{3.13}$$

步驟二:計算 loss 對$\boldsymbol{\theta}^{(0)}$的梯度，式(3.14):

$$\nabla l\left(\boldsymbol{\theta}^{(0)}\right) = \frac{\partial l}{\partial \boldsymbol{\theta}} \mid_{\boldsymbol{\theta}^{(0)}} \tag{3.14}$$

步驟三:更新模型參數

$$\boldsymbol{\theta}^{(1)} = \boldsymbol{\theta}^{(0)} - \alpha \cdot \nabla l\left(\boldsymbol{\theta}^{(0)}\right) \tag{3.15}$$

步驟四:重複步驟二、三得到$\boldsymbol{\theta}^{(2)}, \boldsymbol{\theta}^{(3)}, \ldots\ldots$，直到$l(\boldsymbol{\theta})$足夠小為止，我們就認為

$$\widehat{\boldsymbol{\theta}} \cong \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, l(\boldsymbol{\theta}) \tag{3.16}$$

需要注意的是，梯度下降法受到損失函數的凹凸性、初始值的選擇以及學習率(learning rate)大小等等因素影響，並不能保證得到損失函數的全域最小值，亦無法保證收斂。因此在使用梯度下降作參數最佳化的模型中，損失函數以及其初始值的選擇至關重要。而學習率則關係到每次迭代中，參數修正的幅度，太大會影響模型準確率，太小則會降低收斂速度。關於學習率的調整策略的改進，將在 3.3.4 節中詳述。

另外，在深度學習中，關於計算梯度的另一個重要議題-梯度消失(gradient vanishing)與梯度爆炸(gradient exploding)。隨著神經網路的深度增加，意謂著資料輸入之後，經過了更多層的合成函數，因此在計算梯度時，容易上升、或下降過快而落入極端值，造成參數無法更新。本論文模型使用批標準化(batch normalization)層解決這一問題，將在 3.4.4 節詳述。

3.3 卷積類神經網路

卷積類神經網路（Convolution neural network, CNN）是一種基於 ANN、DNN 模型的一種延伸，本章節僅介紹不同的部分。CNN 適合處理如影像圖片這樣的高維度、結構相關資料；而本論文中所使用的 MFCC 語音資料特徵，亦具有相似的資料特性。

3.3.1 卷積運算子

給定:

(1) f(u,v):被提取函數，通常是資料。

(2) k(x,y):卷積核(kernel)。

一個 2 維卷積運算的定義如式(3.17):

$$f(u,v) * k(x,y) = \iint f(u,v)k(x-u, y-v)dudv \tag{3.17}$$

卷積運算可視為一個線性算子，藉由一個卷積核 k(x,y)作為「濾鏡」，以類似內積的方式對資料特徵做「濾波」，進一步的提取、開採。在計算機的實作上，多使用黎曼和(Riemann sum)作逼近，且本論文使用的語音特徵為離散形式，故有以下離散形式如式(3.18):

$$\sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f(u,v)k(x-u, y-v) \tag{3.18}$$

把 f、k 定義域以外的地方當成 0，則變為有限和，如式(3.19):

$$\sum_{u=-m}^{m} \sum_{v=-n}^{n} f(u,v)k(x-u, y-v) \tag{3.19}$$

其中 m、n 分別是 2 個維度的長度，可以將 k 視為一個m × n的矩陣。
使用一個3 × 3的卷積核 k，對5 × 5的資料 f 作卷積，實際運作如圖 7，

令

$$k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix},\ f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.20}$$

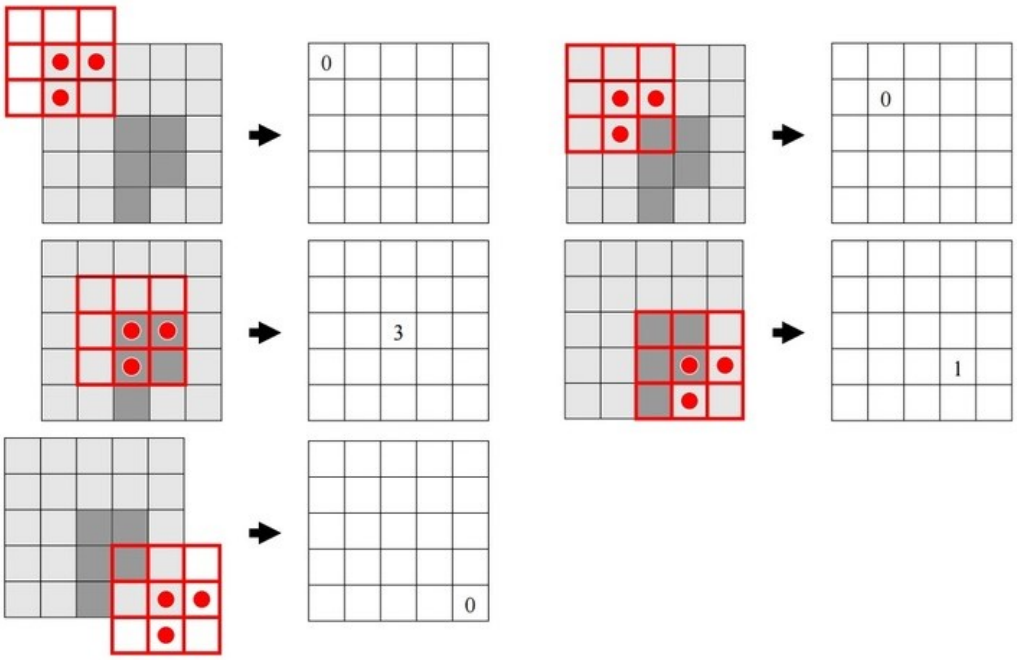

圖 7　卷積運算

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

由圖 7 可以看到，假設對於 f 中的每一個元素 f_{ij}，卷積後的結果 f'_{ij}，即為將卷積核 k 的中心點放置在該點上時，逐點相乘後的總和，相當於兩個矩陣攤平之後再內積的結果。並且以固定的移動間隔，重新建立一個長寬相同的新特徵 f'(移動間隔預設為 1)。

直觀的來看，卷積運算即是對資料中的每個元素，藉由卷積核與該元素附近的相鄰元素建立線性關係。而在 CNN 模型中，卷積核中的元素均視為模型的學習參數，可使用梯度下降法進行最佳化，以得到更好的特徵值，提高模型辨識率。

3.3.2 池化

高維度的資料，相關性往往是非線性且高複雜度的。為了更好的擬和，一個卷積層(convolutional layer)內，往往會使用數十、數百個卷積核，多數模型也會有不只一個卷積層。在這樣的情況下，特徵值通過卷積層後，維度將大幅的上升。如此除了計算量暴增外，特徵的雜訊也會變多。

池化(pooling)就是一個濾波以縮小維度的手段。與卷積不同，池化並沒有學

習參數，僅是將資料切割成數個區域，再從每個區域中提取特徵值。最常用的池化有 2 種:

(1) 平均池化(average pooling)
將矩陣或張量切割成數個區塊，再從每個區域取平均值，重新組合出新的矩陣或張量。

(2) 最大池化(max pooling)
同上，不同的是每個區塊中取出的是最大值。本論文中的模型，均使用最大池化。

舉例最大池化如下圖 8:

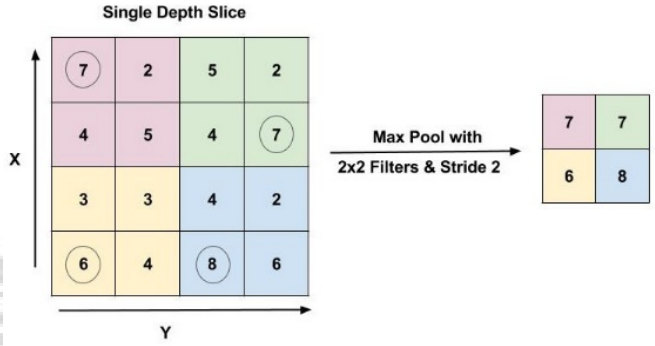

圖 8　池化

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

池化也跟卷積，可以選擇適當的區塊大小與移動間隔。圖 8 舉例為2×2的區塊、移動間隔為 2。

3.3.3 卷積層

承 3.3.2 節，一般基本的卷積層，均包含卷積、池化等動作。另外同一般全連接層在 3.2.3 節的理由，卷積層中為了控制數值範圍與增加非線性程度，也必須加入活化函數。根據資料特性、目的不同，卷積層中各種運算的先後順序、次數等超參數，均有不同的組合，也可能加入更多的處理手法。經實驗以及參考 VGG16[2]中的部分設計，本論文使用的卷積層如圖 9:

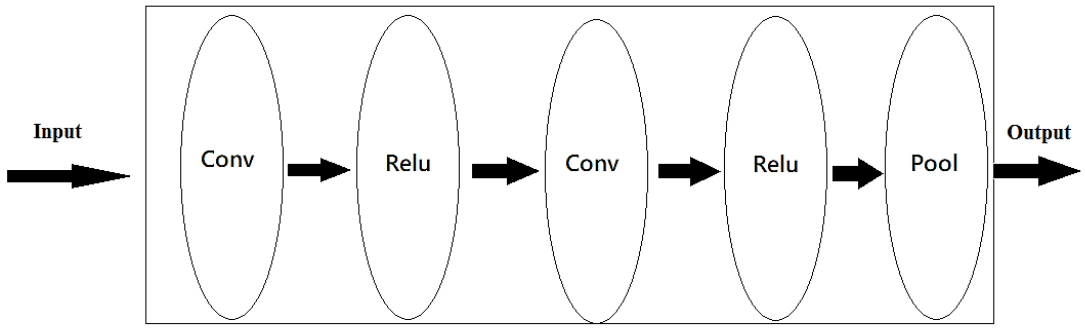

圖 9　本論文使用之卷積層

維度變化示意如圖 10:

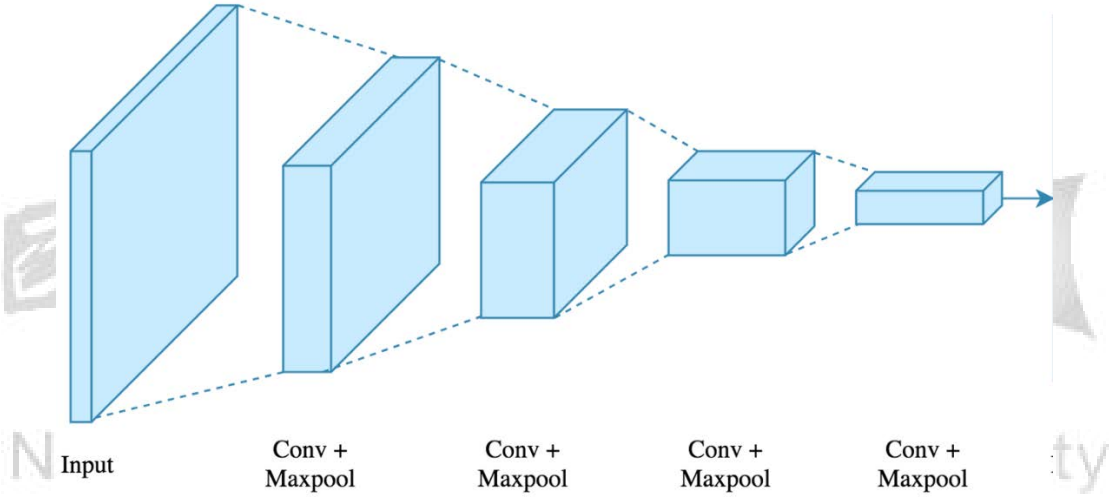

圖 10 資料被卷積過程

(https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2)

可以看到原始資料的維度，每經過一層卷積層之後，經由數個卷積核使厚度增加再經過池化層使長寬縮小，最後得到一疊低維度的特徵。

3.3.4 CNN 分類器模型

原始特徵值經過一個或數個卷積層後，會輸出一組提取過後的高階特徵，再將這些特徵輸入 MLP 中，藉由 MLP 將這些把這些高階特徵分開來，如此就

是一個完整的 CNN 模型。舉例圖 11 為本論文所使用之 CNN 模型，通過 3 個卷積層、3 層全連接層。其中:

(1) Input:輸入層。
(2) Conv:卷積層。
(3) BN:批標準化層(Batch normalization)，將在 3.3.4 節說明。
(4) Relu:使用 relu 函數之活化層。
(5) Max pooling:最大池化層。
(6) Flat:攤平層，將任何張量攤平成 1 維向量。
(7) FC:全連接層(fully connected laye)，即一般的多層感知機。
(8) Softmax:如式(3.7)所示。
(9) Output:輸出層。

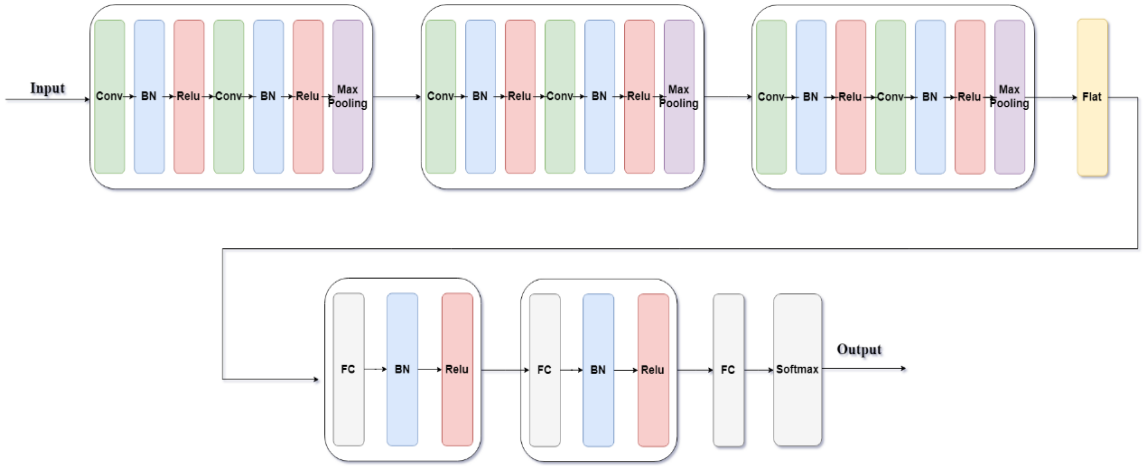

圖 11 本論文所使用三層 CNN 模型

3.4 附加方法

除了前面所談的基本核心外，這邊介紹本論文模型用到的其他神經層，以及一些訓練技巧。

3.4.1 切割訓練與測試集

本論文之資料集包含 20 位語者，每位語者每個單音錄製 10 次，取第 4 次錄製之單音作為測試集，不加入訓練；其餘為訓練集。待模型訓練完成之後，再將測試集代入模型計算準確率，以此評斷模型的好壞，確保模型的泛化能力。

3.4.2 GPU 運算

由於 CNN 模型的運算量極大，最佳化時又有與內積相似的運算特性，相當適合使用 GPU 運算，可以大幅增加建模效率。依條件不同，約可提昇 10~30 倍不等。

3.4.3 隨機批次輸入

每次迭代均從訓練集中隨機抽取適當批次大小(batch size)輸入。隨機化能夠避免模型遵循特定的資料排列順序，不斷的往相同方向修正，因而對特定資料產生過擬和(overfitting)。最佳批次大小依模型的參數量、硬體條件而有所不同。配合使用 GPU 運算，批次大小建議為該 GPU 核心數的整數倍，能夠最有效率的使用計算資源。

3.4.4 批標準化

延續 3.2.5 節的問題，批標準化(batch normalization)是一種強制性重整輸出值的手段。令

(1) 輸入值$\boldsymbol{x}$、輸出$\boldsymbol{y}$ ，維度皆為 n × d
(2) 學習參數:γ, β

(3) 中繼參數$\begin{cases} \mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{i,j} \\ \sigma_j^2 = \frac{1}{n}\sum_{i=1}^{n}(x_{i,j}-\mu_j)^2 \end{cases}$，$1 \le i \le n$，$1 \le j \le d$ (3.21)

即該批資料的平均數、變異數。與一般學習參數不同，在訓練階段，需在該神經層輸入後才有值，輸出之後也不保留；但會保留一組中繼參數供驗證階段使用。

接著可以得到經調整過後的式(3.22)

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}}, \qquad x_{i,j} \in \boldsymbol{x} \tag{3.22}$$

其中ε是一個為小正數，用來避免分母為 0 的情況。
最後神經層輸出式(3.23)

$$y_{i,j} = \gamma_j \hat{x}_{i,j} + \beta_j, \qquad \forall y_{i,j} \in \boldsymbol{y} \tag{3.23}$$

經過學習參數最佳化之後，將標準化後的這一批數據，加以一個適當的伸縮與平移參數後再輸出，如此就可將每一神經層的輸出控制在一定範圍內，除了大大的加快了訓練速度，也有降低對初始值的依賴、減少過擬和的優點。

3.4.5 訓練中斷條件

在訓練過程中，一般難以確定何時模型才停止收斂，不僅耗時，也容易有訓練不足或訓練過度的問題。所以本文採用一個最大長度為 3 的佇列(queue)記錄訓練集的 loss，每經過一定的迭代次數時(視模型收斂快慢而定)就進行記錄並檢查，若發現連續 2 次 loss 下降的幅度不夠，就停止訓練。

3.4.6 學習率調整

學習率代表了每一次迭代，參數修正的步伐大小，在 3.2.5 節中，提到了選擇學習率的重要。由於梯度僅能指向當下這一批數據下的最佳參數，無法保證在所有資料都適用;而我們的目標是訓練出一個具泛化能力的模型，修正的步伐若太大，模型將因過度擬和該批數據，而在其他數據上表現變差；但若步伐太小，

除了訓練效率差外，也容易卡在局部最小值，無法走向最佳參數。

因此在本篇論文中，採用 Adam(Adaptive moment estimation)[6]，做學習率的動態估計。Adam 是梯度下降的延伸，主要優點在於經過偏移校正之後，每一次迭代的學習率都能有一個確定的範圍，對每一個參數也有不同的學習率，使得參數的修正較平穩，是近年較流行的最佳化方法之一。

3.4.7 模型投票

本論文一共使用了 3 種不同想法的 CNN 模型對中文單音做分類，分別經過超參數的最佳化後，再從不同模型的最佳結果中進行投票，以期提高準確率。

具體投票的方法:
每一個分類模型，最後一層均使用 softmax 活化函數，再取機率最大的維度作為分類。本論文將 3 個 CNN 模型經過 softmax 的輸出結果，經過加權後做向量加總，再經過一次 softmax 活化，最後得到綜合三個模型的一組機率值。如圖 12 所示

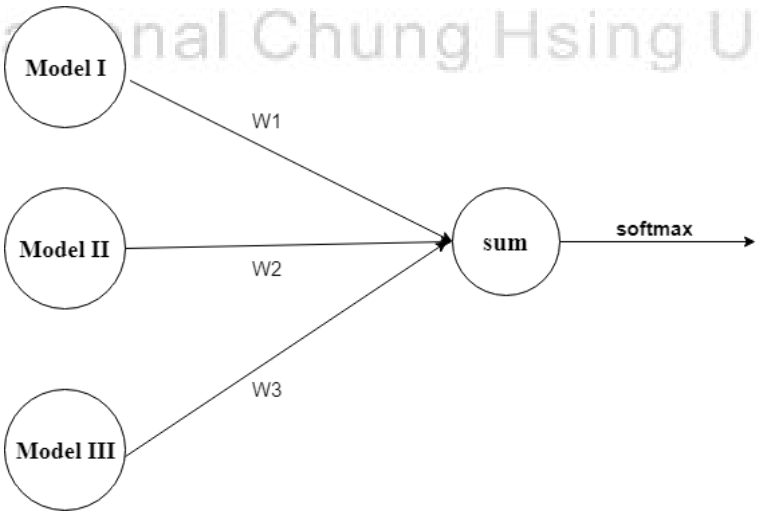

圖 12 模型投票

圖 12 中的權重值，亦是可訓練的。本論文為降低複雜度，w_1、w_2、w_3均取 1，意即不加權直接平均。

第四章　實驗流程與結果

此章節將說明實驗的整個流程以及使用到的工具，比較幾種不同想法的CNN 模型設計，並探討實驗所得到的結果。

4.1 實作軟體

本篇論文中的實作均使用 Python 程式語言搭配 GPU 版本的 Tensorflow 套件。Tensorflow 是由 Google 公司推出的機器學習開源框架，支援 C++、Python 等多個程式語言。其使用自由度高，適合實作各種機器學習、統計模型。GPU 版本也支援 cuda 架構的 GPU 運算，在效率、準確率上都相當優良。

4.2 資料來源

本篇論文中所採用的資料庫為二十位不同語者所錄製之語音，共 1391 個中文單音，語者每一個中文單音錄製十次。其中，第四次錄製的單音作為測試音，其餘為訓練音。

經一系列前處理後，得到每一個單音的 MFCC 特徵值，是一個53×39的矩陣，其中每一列(row)是一個音框，一共有 53 列、長度為 39 的音框。其中前 24 列為子音音框、後 29 列為母音音框。
建立模型時，大致分為幾種辨識目標:

(1) 單音辨識:1391 類別。
(2) 分聲調母音辨識:160 類別。
(3) 不分聲調母音辨識:35 類別。
(4) 不分聲調子音辨識:36 類別。

實際輸入模型時，依目標不同需再作音框挑選:

(1) 分聲調、不分聲調母音辨識時，取後 29 個音框、再刪去頭尾各 2 個音框降低雜訊。也就是取第 27~51 列的音框，最後得到25×39的母音特

徵值矩陣。

(2) 不分聲調子音辨識時，取前 24 個音框、再刪去頭尾各 2 個音框降低雜訊。也就是取第 3~22 列的音框，最後得到20×39子音特徵值矩陣。

(3) 單音辨識時，同前面 2 者共刪去 8 個音框，也就是取第 3~22、第 27~51 列音框，最後得到45×39子音特徵值矩陣。

4.3 主要模型設計

中文有 35 個母音、36 個子音以及 5 種聲調，總共組合出 1391 個單音。本論文的目標是最佳之單音辨識率模型，但是單音分類供達 1391 種，直接分類效果不一定最好；因此試著以三種不同想法，將單音辨識分拆成母音辨識、子音辨識等幾種不同想法的 CNN 模型並作比較，並利用投票的方式結合每個模型的優點，試著達成最佳單音辨識。

(1) 模型 I、全單音分類辨識

如圖 13，輸入子音、母音共 45 個音框，輸出長度為 1391 的 one-hot 編碼，即直接將全部 1391 個單音作分類辨識。

圖 13 主要模型 I

(2) 模型 II、分聲調母音與全子音辨識

如果直接將全部 1391 個分類的單音，由於類別太多，效果不一定好。故模型 II 分成兩個部分:

- 35 類母音、5 類聲調結合成 160 類「分聲調母音」。
- 36 類子音。

分別訓練出 2 個 CNN 模型，再根據 2 個模型的辨識結果，組合出單音類別作辨識。如圖 14。

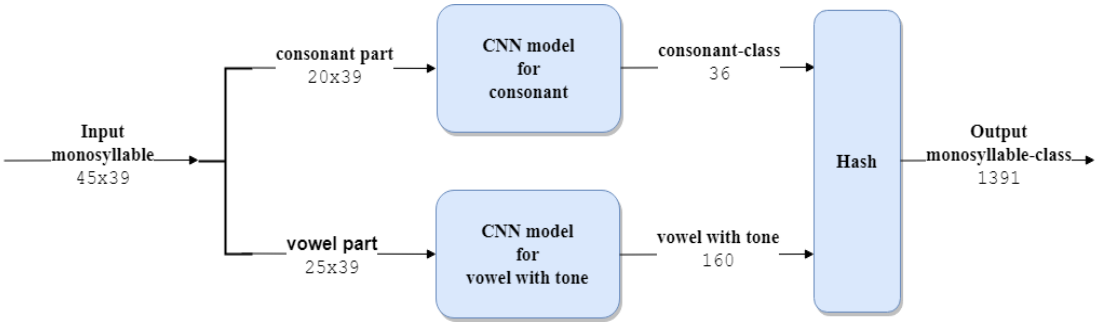

圖 14 主要模型 II

(3) 模型 III、分聲調母音下子音辨識

與模型 II 相似，但差別在預測子音時，將母音的類別考慮進來。事先訓練好在 160 個母音之下的子音分類器，先做分聲調的母音辨識，再從該預測類別下，可能的子音做辨識。這種作法的好處是，相較模型 II，對子音的分類目標更清楚；壞處則是，每一個分類的樣本數大幅降低，訓練難度增加，初始值的選擇也更重要。如圖 15。

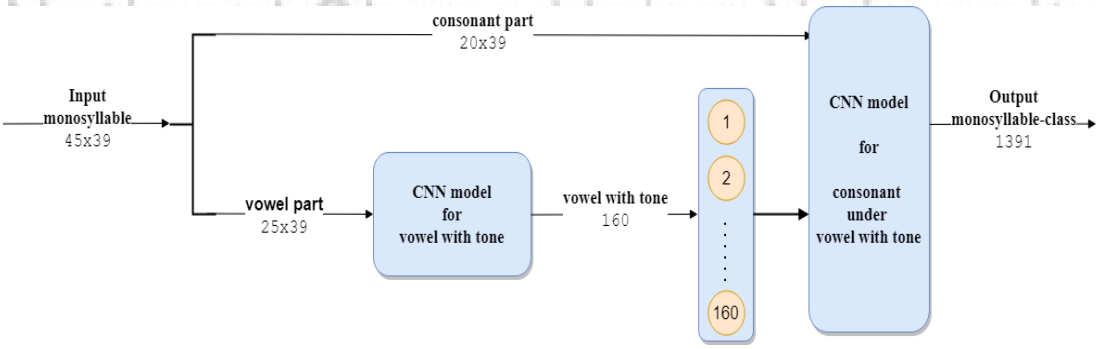

圖 15 主要模型 III

4.4 超參數最佳化

超參數(hyperparameter)指的是相對於可以被訓練的學習參數，那些無法被最佳化方法訓練的參數，如活化函數的選擇、學習率大小、神經元個數、神經

層的層數等等，必須透過適當的實驗找尋最佳的超參數配置。本論文嘗試的超參數重點如下:

(1) 卷積深度: 即卷積層的層數，實驗範圍 2~4 層。
(2) 卷積厚度: 即每層卷積的神經元個數，實驗範圍大約在 8~1024 個神經元(卷積核)。

母音的波型通常較為規律，在本論文中使用不分聲調的母音(160 類別)CNN 模型做做超參數的最佳化，其他分類模型如單音、子音模型，則以同樣超參數設計類推。

一般來說，隨著深度、厚度的提昇，模型的擬合能力也會有所提高。但隨著參數量的上昇，計算量、訓練難度也會大大提升。所以本論文在同一個深度之下，慢慢提昇厚度並觀察準確率的變化，找出最兼具辨識率與效率的組合。

4.5 實驗流程

本論文之實驗流程依序為錄製語音、語音訊號前處理、多層感知機模型訓練、測試語音辨識、輸出結果、計算錯誤率、錯誤率比較，實驗流程圖如圖 16。

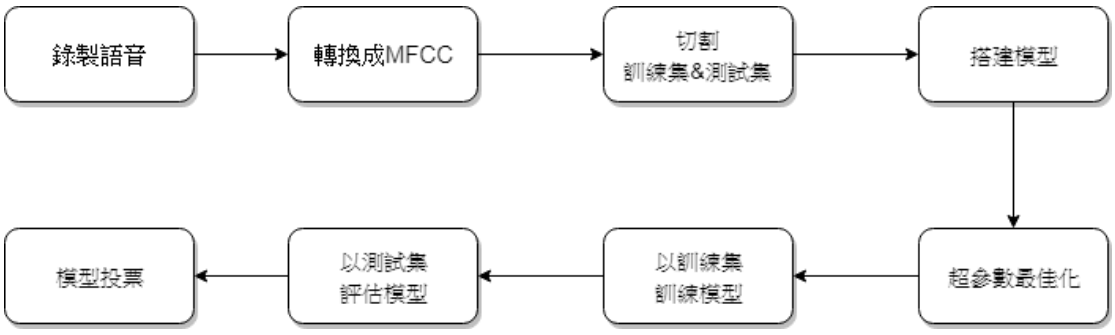

圖 16 實驗流程

4.6 實驗結果

4.6.1 超參數結果

以分聲調的母音模型做測試，訓練、測試準確率詳見 3.4.1。不同於 MLP，本篇論文 CNN 模型中的全連接層部分，僅視為將特徵值分開的工具，因此不特別注重全連接層的神經元個數、層數。原則上只要從全連接層的輸入到輸出，各層間維度變化不太過劇烈即可。表 1 為卷積深度為 2 時，模型在不同神經元數量下的訓練結果:

表 1: 2 層卷積 CNN 模型的分聲調母音辨識率

卷積層	訓練準確率	測試準確率
(8, 16)	0.8715	0.7909
(16, 8)	0.8321	0.7559
(16, 16)	0.8987	0.8176
(16, 32)	0.9010	0.8285
(32, 16)	0.8898	0.8136
(32, 32)	0.9132	0.8411
(32, 64)	0.9069	0.8468
(64, 32)	0.9225	0.8546
(64, 64)	0.9201	0.8561
(64, 128)	0.9419	0.8756
(128, 64)	0.9268	0.8641
(128, 128)	0.94335	0.8807
(128, 256)	0.949936	**0.88956**

備註: 卷積層欄位中，括號長度為卷積層數，括號內數字依序為各層神經元個數。如(8, 16)為 2 層卷積層，第 1 層 8 個神經元、第 2 層 16 個神經元。

可以看到在(8, 16)時，參數量顯然太少，連訓練集無法收斂的更好；而隨著厚度的提升，訓練集準確率顯著提昇；同時測試集準確率也慢慢爬升，說明在厚度的增加，也提高了模型的泛化能力。

另外在這邊我們也想知道，前寬厚窄、前窄厚寬等設計，是否對辨識率有影

響。可以看到的是，在總參數量相同的情況下(例如(16, 8)與(8, 16))，前窄後寬的確表現的必較好，符合一般的認知。因此後續實驗，僅嘗試前窄厚寬的設計。在此結論下，嘗試深度為 3 的情形，如表 2:

表 2: 3 層卷積 CNN 模型的分聲調母音辨識率

卷積層	訓練準確率	測試準確率
(8, 16, 32)	0.8837	0.7969
(16, 32, 64)	0.9394	0.8569
(24, 48, 96)	0.9467	0.8725
(32, 64, 128)	0.9482	0.8796
(64, 128, 256)	0.9603	0.8954
(128, 256, 512)	0.9538	0.9013
(256, 512, 1024)	0.9775	**0.9231**

備註: 卷積層欄位中，括號長度為卷積層數，括號內數字依序為各層神經元個數。如(8, 16, 32)為 3 層卷積層，第 1 層 8 個神經元、第 2 層 16 個神經元、第 3 層 32 個神經元。

比起 2 層卷積，最佳辨識率多了 3%左右，多了一層的卷積層，顯著的提升了辨識率。接著嘗試深度為 4，如表 3:

表 3: 4 層卷積 CNN 模型的分聲調母音辨識率

卷積層	訓練準確率	測試準確率
(8, 16, 32, 64)	0.8556	0.7942
(16, 32, 64, 128)	0.8830	0.8249
(32, 64, 128, 256)	0.9243	0.8690
(64, 128, 256, 512)	0.9298	**0.8821**

備註: 卷積層欄位中，括號長度為卷積層數，括號內數字依序為各層神經元個數。如(8, 16, 32, 64)為 4 層卷積層，第 1 層 8 個神經元、第 2 層 16 個神經元、第 3 層 32 個神經元、第 4 層 64 個神經元。

在(64, 128, 256, 512)這個組合時，參數量已經超越了 3 層卷積時的(64, 128, 256)組合，但辨識結果卻反而較差，因此不繼續往下嘗試。

綜合以上的結果，選擇(256, 512, 1024)為最佳的卷積超參數配置，後續子音、單音也以相同的設計訓練模型，如圖 11。

4.6.2 主要模型結果

經過 4.6.1 超參數最佳化之後，採用(256, 512, 1024)卷積超參數組合的 3 個主要模型結果如表 4:

表 4：主要模型辨識率

	訓練準確率	測試準確率
模型 I	0.9196	0.8289
模型 II	0.9505	0.8276
模型 III	0.9402	0.8046
模型 I+II+II	0.9626	**0.8405**

備註: 模型 I+II+II 為 3.4.7 節中模型投票之結果。

模型 I、II、III 詳見 4.3 節。由於模型 I 高達 1391 種分類，輸入、輸出的維度均較大非常多，模型收斂情況也較差，相較其他主要模型，訓練準確率只有 91.96%；而模型 II、III 採兩層式分類，維度上的負擔較小，訓練準確率分別能達到 95.05%、94.02%。

模型 I+II+II 則是希望取 3 個模型的優點，整合成一個大模型起，提高整體辨識率。整合之後，訓練集提高了 1.21%、訓練集提高了 1.16%。

第五章　結論

5.1 總結

本篇論文是利用 MFCC 特徵值與 CNN 類神經模型，對 1391 個中文單音進行辨識。而挑選、訓練模型 CNN 模型的目標在於，盡可能的讓卷積層對資料做更好的開採、提取。

1. 不管是母音、子音或是單音，加入了卷積層的 CNN 模型，辨識率確實高於一般的 MLP，說明語音資料的確具有一定的結構性，透過適當的疊加卷積層，能夠開採出更具代表性的特徵。

2. 活化函數的選擇對中相當重要，但根據資料的特性不同，最適合的活化函數也不一定相同。在本論文的資料集上，Tanh 相對於 Sigmoid 修正的較快、較好，辨識率差距不大；而 Relu 則較前兩者更容易發生梯度消失，但透過批標準化層解決梯度問題後，辨識率、計算速度都優於前兩者。

3. 從 4.6.1 的結果可以看出，隨著卷積層的神經元數量不斷增加，模型辨識率可以說是只升不降；而當總參數量大到一定程度時，辨識率的上升也趨緩，尤其是測試集。或許可以解釋成: 隨著特徵數量的增加，模型對既有資料的解釋比例也只會不斷上升。然而這未必代表卷積的深度、厚度，越大越好、沒有限制。特徵維度過高不僅容易造成過擬和而失去泛化能力，也會大幅增加計算量。用過高的計算量，換取準確率的些微提昇，並不明智。

4. 本論文中的 3 個主要模型，最佳辨識率分別為: 82.89%、82.76%、80.46%，藉由不加權投票的方式合成一個更大的模型，能夠將辨識率提高至 84.05%。只要個別模型的辨識率不要太差，這樣的方式的確可以讓辨識率獲得提高。

5.2 改善與展望

1. 特徵值的提取對機器學習的結果至關重要，而 MFCC 特徵一直是傳統語音辨識領域中，最常用的特徵提取方法，本論文亦是基於此特徵下建模。然而

CNN 類神經模型特別的地方在於，其模型包含的卷積層構造，已具有自動提取特徵的效果。未來或許可以嘗試，不經由 MFCC，直接讓卷積層對原始資料提取特徵，是否會有更好的結果。

2. 許多機器學習相關研究指出，初始值的選取對分類結果、訓練效率，都有很大的影響，甚至可以解決梯度相關問題；然而本論文僅使用簡單的常態隨機初始值，搭配批標準化解決模型深度太大、難以訓練的問題。未來或可導入一些現在已有的初始值估計方法做預訓練(pre-training)。

3. 本論文模型僅考慮單音情況，假定每個單音都是相互獨立的變數。然而許多真實應用場景中，單音都來自於有意義的語句之中，而非單獨的被讀出來。在模型之中加入一些具備「遞迴」性質的層，像是「LSTM cell」。如此一來，就能夠在考慮當下語境、前後文的情況下，縮小可能的單音類別，應該能對辨識率有很大的提昇。

4. 在本論文的 3 個主要模型中，模型 II 與模型 III 均是將單音拆解成母音、子音，最後再整合其結果。在實驗前的預想中，模型 III 是在給定母音的情況下，縮小了可能子音的範圍後，再做分類，照理說辨識率應該會至少比模型 II 好，但結果卻不然。是超參數的配置無法類推?又或許是限縮子音類別後樣本數下降，影響了訓練的結果?值得做更進一步的探討。

5. CNN 在影像辨識應用中，常將影像資料作旋轉、反白、增加躁點等不影響人眼識別的處理，以增加新樣本，稱為「資料增強(data augmentation)」。此技巧可擴增資料集的樣本數、多樣性，並在一定程度上降低過擬和。而在語音資料中，如果能夠更加了解什麼樣的語音干擾常出現，但卻不影響人耳識別聲音，或許能夠使的訓練出來的模型更加穩健。

參考文獻

[1] 蘇木春、張孝德。2003。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。

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

[3] Alex Graves, Abdel-rahman Mohamed and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013.

[4] Abdul Ahad, Ahsan Fayyaz and Tariq Mehmood, "Speech recognition using multilayer perceptron." in *Students Conference, 2002. ISCON '02. Proceedings* , vol. 1, pp.103-109, IEEE, 2002.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet classification with deep convolutional neural networks", in *NIPS*, 2012.

[6] Diederik P. Kingma and Jimmy Lei Ba," ADAM: A method for stochastic optimization" in *ICLR*, 2015.

[7] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." in *ICLR*, 2015.

[8] LeCun, Y., Bottou, L., Orr, G. B. and Muller, K-R., "Efficient backProp." in *Orr, G. B. and Muller, K-R. (Eds), Neural Networks: Tricks of the trade*, Springer, 1998.

[9] Neelima Rajput and S.K. Verma, "Back propagation feed forward neural network approach for speech recognition," in *2014 3rd International Conference on Reliability, Infocom Technologies and Optimization (ICRITO 2014) - (Trends and Future Directions)*, pp. 1-6, IEEE, 2014.

[10] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional neural networks for speech recognition" in *Transactions on audio, speech, and Language Processing*, vol. 22, no. 10, IEEE/ACM, october 2014.

[11] Yanmin Qian, Mengxiao Bi, Member, Tian Tan and Kai Yu, "Very deep convolutional neural networks for noise robust speech recognition" in *Transactions on audio, speech, and Language Processing*, vol. 24, no. 12, IEEE/ACM, december 2016.

[12] Xu Tian, Jun Zhang, Zejun Ma, Yi He, Juan Wei, Peihao Wu, Wenchang Situ, Shuai Li and Yang Zhang, "Deep LSTM for large vocabulary continuous speech recognition" arXiv:1703.07090, 2017.