108學年度專題報告競賽

題目：英文數字0~9的語音辨識

系所班別：統計系 四年級&三年級

姓名學號：黎　薇(410578022)

許月華(410578044)

張崴智(410578047)

王勃淵(410578053)

姜孝軒(410578054)

報告日期：2020/06/05

英文數字 0~9 的語音辨識

摘要

現代的語音相關研究，多是以類神經網路進行辨識，包含 CNN、LSTM 等神經網路。本次研究的任務為，對固定長度的語音資料進行辨識，並與傳統統計模型進行比較，最後挑選出最合適的模型。研究首先對語音資料進行了 MFCC 轉換，得出 39 維的特徵，接著分別架構了 CNN+LSTM 與 Encoder-Decoder 模型，再進一步利用兩個模型進行預測。而 CNN+LSTM 與 Encoder-Decoder 模型的準確率分別達到 96.9%及 97.37%。使用 CPU 對 CNN+LSTM 進行訓練，其訓練時間為 10.37 分鐘；使用 GPU 對 Encoder-Decoder 模型進行訓練，費時 348.23 分鐘。根據時間及準確率的評估，得出本次任務最合適的模型為 CNN+LSTM 模型；若探討模型的可發展性，則 Encoder-Decoder 模型對於連貫的單字或語句，可以進行連續的預測，故有更好的可發展性。

Abstract

Modern researches related to speech are mostly conducting with neural network, including convolutional neural network, long short-term memory, etc. The task of this research is to recognize speech data of fixed length, we also build some statistic models for comparison, and try to find out the best model for the task. First of all, we apply Mel-frequency Cepstral Coefficients to speech data, and gain 39-dimension features of speech data. Secondly, we construct CNN+LSTM model and Encoder-Decoder Model separately, and then predict the label of the speech by them. Furthermore, CNN+LSTM model and Encoder-Decoder model achieve accuracy of 96.9% and 97.37% respectively. By CPU, we train CNN+LSTM model and get the training time which is 10.37 minutes. On the other hand, we train the Encoder-Decoder model with GPU, which is 348.23 minutes. According to the training time and accuracy, we find out the best model for our task is CNN+LSTM. However, for the purpose of developability, we convince that Encoder-Decoder model will have a better opportunity due to the ability to predict sequential words, continuously.

第一章、前言

語言（language）長久以來是人與人之間最自然且最方便的溝通方式，而語音（speech）指的是藉由口述來表達語言的內涵。隨著數位電子科技的蓬勃發展以及無線通訊與網際網路的創新普及，傳統的鍵盤、滑鼠已不能滿足現代人們的需求，而語音控制成為現在熱門的前端技術之一。舉凡：1）語音輸入 2）聲紋辨識 3）聲音合成，在人類與機器間的溝通上，越來越被廣泛應用。在絕大部分的情況下，語音比肢體語言更能明確表達語者所要傳遞的訊息。當語音和語言被正確地轉換時，我們才能理解語者所要傳達的內容或概念。因此，語音辨識技術對語言、語音之間的轉換扮演著相當重要的角色。

本研究的資料來源為 Google 於 Kaggle 平台提供的多個英文單字語音資料，該語音資料集中的每個單字皆為不同語者，每個音檔的時長皆在一秒以內，且每一個單字都包含兩千筆以上的資料。而本研究著重於英文數字的語音辨識，故僅使用英文數字 0~9 作為資料集進行研究。

第二章、文獻回顧

卷積神經網路（Convolutional Neural Network, CNN）在近代的類神經網路當中，多被應用於圖像辨識，例如 LeCun 於 1998 年所發表的[1]，奠定了 CNN 的重要基礎；Alex 所提出的多層 CNN 架構 AlexNet[2]，奪得了 2012 年的 ImageNet 冠軍；以及考慮了殘差，進而誕生的 ResNet[3]，更是終結了 ImageNet 這場行之有年的圖像辨識比賽。雖說如此，由於 CNN 的根本概念是將資料點 p 的資訊與其周圍 k 個資訊納入考量，此一特性若運用於時間數列資料上，亦會考慮時間數列在時間點 t 與前後 n 個時間點彼此之間的時間相關性，故使用 CNN 對語音資料進行處理的研究亦不在少數。在 CNN 尚未十分成熟時，便有[4]所提出的類 CNN，運用於語音辨識上的實際例子，近幾年也有利用 CNN 對語音進行辨識的研究[5]。

由於考慮到時間前後的相關性，亦有多位研究者投入使用循環神經網路（Recurrent Neural Network, RNN），作為時間數列資料預測模型的核心技術。雖說 RNN 能夠考慮時間數列資料的時間性，但缺點在於其記憶時間十分短暫，也因此長短期記憶網路（Long Short-Term Memory, LSTM）[6]，作為 RNN 的改良版，更加廣泛的被應用於實務上。多層的神經網路所堆疊的架構[7, 8]，往往能帶來更為優良的結果，這在 ImageNet 比賽上已獲得了證明[2, 3, 9]，因此近代的 LSTM，常會使用多層的 LSTM 架構[10]，而其往往也會有比較好的結果。此外，雙向的 LSTM（Bidirectional LSTM, BiLSTM）[11]，亦為 LSTM 的重要架構，其於 Seq2Seq 上的應用[12]，一再地表明了同時使用向前與向後的時間數列資料，能使得模型更加準確。

若談論到自然語言處理（Natural Language Processing, NLP），不得不提到

Encoder-Decoder 模型[13, 14]。Encoder-Decoder 模型在提出後，迅速獲得高度關注，由於其可以將一不固定長度的時間數列資料投射至一固定長度的時間數列資料上[10]，且能透過梯度下降法（Gradient Descent）同時進行模型訓練，因此相對於以往的 RNN，Encoder-Decoder 更能使時間數列資料有更多發展性。隨後，Attention 模型[13, 15]隨著 Encoder-Decoder 模型的問世，因運而生。Attention 模型架構使自然語言處理獲得了重大的躍進，該架構先是使用了 Encoder-Decoder 模型作為基礎，再將 Encoder 的各個輸出的資訊集中彙整，並作為輸入的一部份使用於 Decoder 之中，解決了 LSTM 在過長的時間數列上，訊息消失的問題。日後，Attention 被應用於[16]，並且很好的對齊了語音資料與各個英文字母，結果十分顯著，再次檢證了 Attention 模型在 Encoder-Decoder 模型上的重要性。

第三章、研究方法

本研究探討語音辨識系統的語音前處理、音素（phoneme）對齊和字詞（lexicon）辨識。所有辨識系統皆對前處理有相當大的倚重，語音更是如此。由於語音訊號（signal）會受到諸多因素影響，例如：語者聲調、說話語速、外在雜訊等等，使得語音的處理相對地複雜與困難。普遍上，語音的處理以著名的梅爾倒頻譜係數（Mel-frequency Cepstral Coefficients, MFCC）作為主流的語音特徵擷取方法，該方法以人的聽覺感知頻率作為語音特徵擷取的對象，配合其一階與二階時間軸導數（Time Derivatives），得出 39 維的 MFCC 語音特徵，可以有效提升後續辨識的準確率。

本研究分三部分，如圖 1。壹、使用頻譜圖（Spectrogram）進行主成分分析（Principal Component Analysis, PCA），擷取特徵並利用傳統統計模型進行辨識。貳、以 MFCC 和 PCA 做前處理，接著透過傳統統計模型進行辨識。參、架構類神經網路（Neural Network），並以 MFCC 處理後的資料作為輸入進行辨識。

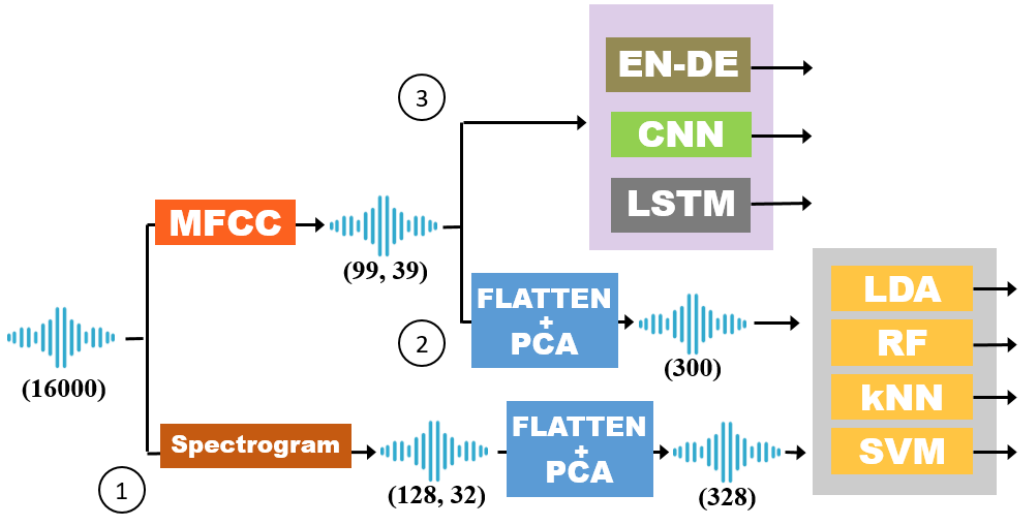

圖 1 研究流程圖

3.1 梅爾倒頻譜係數（Mel-frequency Cepstral Coefficients, MFCC）

MFCC 主要有七個對語音訊號的運算過程，包括預強調（pre-emphasis）、音框化（framing）、窗化（windowing）、快速傅立葉轉換（FFT）、梅爾濾波器組（Mel-triangular filter banks）、對數能量（log energy）及離散餘弦轉換（DCT）。

3.1.1 預強調

在發聲的過程中，聲帶和嘴唇產生高頻衰減的效應，所以我們聽到的聲音就像是經過了一個低通濾波器，利用一個a參數對聲音訊號z進行高頻補強，來補償高頻衰減的部分，此濾波器表示為

$$H(z) = 1 - az^{-1} \tag{3.1}$$

將時間點 t 的輸入訊號 s[t]通過高通濾波器後，在時域表示為

$$s_P[t] = s[t] - as[t-1] \tag{3.2}$$

其中 α 介於 0.95 和 0.98 之間，$s_P[t]$為預強調後的訊號。

3.1.2 音框化

由於語音訊號變化快速，而短時間內的語音訊號通常相對穩定，因此通常將數個語音訊號作為一個觀測單位，這個單位稱為音框（frame），再根據音框內的訊號進行分析，為了避免相鄰的音框變化過大，因此會讓相鄰的音框有一段重疊區域。本研究使用 0.025 秒的音框，每次移動 0.01 秒，得出 99 個時間段的 MFCC 特徵。

3.1.3 窗化

為了減少音框左右兩端信號的不連續，及保留語音訊號的完整性，會將每一個音框乘上一個視窗，本研究使用的是漢明窗（Hamming Window），其會凸顯音框中間的資訊，並縮減音框兩端的資訊，使得每個音框與下一個音框的資訊較為連續，漢明窗表示為

$$w[n] = \begin{cases} 0.54 - 0.46\cos(\frac{2n\pi}{N-1}), & 0 \le n \le N\text{-}1, \\ 0 & ,otherwise, \end{cases} \tag{3.3}$$

其中 N 為音框內的訊號個數。

3.1.4 快速傅立葉轉換

由於語音訊號在時域（time domain）上變化快速，不容易看出訊號的特性，所以通常會將它轉到頻域（frequency domain）上觀察能量分布，不同的能量分布，就擁有不同的語音特性。所以在窗化後，每個音框經過快速傅立葉轉換來得到頻域上分布。

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi nk}{N}}, 0 \le k \le N-1. \quad (3.4)$$

將歐拉公式（Euler formula）$e^{\pm j2\pi nk} = \cos(2\pi kn) \pm jsin(2\pi nk)$代入式(3.4)，可以得到

$$X[k] = \sum_{n=0}^{N-1} x[n] \left[cos\left(\frac{2\pi kn}{N}\right) - jsin\left(\frac{2\pi kn}{N}\right) \right] \quad (3.5)$$

將式(3.5)的實部與虛部取平方後，相加得到頻域能量S_k。

3.1.5 梅爾濾波器組

人耳對不同頻率的感受程度，並非在所有頻域都是一樣的。一般來說，人耳對低頻有比較高的敏感度，也就是在低頻時可以分辨較細微的頻率差異。因此，若是聲音在某個範圍變動下，人耳感覺不出差異，這個範圍就稱為臨界頻帶（critical band）。梅爾刻度（Mel scale）是根據人的聽覺系統所模擬出的線性轉換，在梅爾刻度中使用數個範圍不等的三角濾波器，梅爾刻度與頻率的關係如式(3.6)。

$$M(f) = 2595 \times log_{10}\left(1 + \frac{f}{700}\right) \ , f = frequency \quad (3.6)$$

3.1.6 對數能量

人耳對聲音大小的感知跟頻率同樣非呈直線關係，當聲音能量放大10倍時，人耳只會感覺放大 1 倍，而我們所得的語音訊號，可能因為語者的說話音量或背景的雜訊大小，影響語音訊號的數值變動。因為聲音的能量變動劇烈，通常會以對數形式表示其能量大小，也就是將通過梅爾濾波器組的訊號平方後取對數值，如式(3.7)。

$$energy = log(\sum m^2) \ , m = mel \ \ signals \quad (3.7)$$

3.1.7 離散餘弦轉換

將通過梅爾濾波器組的訊號取對數值後，再做離散餘弦轉換到時域，這就是梅爾頻率倒頻譜係數，其中包含了 13 個倒頻譜係數和 1 個對數能量，DCT 表示為

$$c_x[n] = \frac{1}{M} \sum_{m=1}^{M} \log(S[k]) \, cos\left(\frac{\pi n\left(m-\frac{1}{2}\right)}{M}\right), 0 \le n \le 12 \quad (3.8)$$

其中$c_x[n]$為梅爾頻率倒頻譜係數（MFCC），$S[k]$為經過梅爾濾波器組的訊號。

3.2 主成分分析（Principal Component Analysis, PCA）

PCA 主要用來擷取特徵維度，它將高維度資料投射至一個較低維度的空間，

並讓資料特徵在新空間中為正交(Orthogonal)，且保留對變異量貢獻最大的特徵。PCA 的作法為使用所有訓練資料 $X=[x_1,x_2,x_3,\ldots x_N]$， N 為資料的總數，每個 x_i 為 n 維向量，來統計訓練資料的整體共變異矩陣（Covariance Matrix）：

$$\Psi=\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{X})(x_i-\bar{X})^T \tag{3.9}$$

其中 $\overline{X}$ 為整體的平均向量，所以Ψ為 $n\times n$ 維的共變異數矩陣。

$$\bar{X}=\frac{1}{N}\sum_{i=1}^{N}x_i \tag{3.10}$$

對Ψ求特徵向量分解（Eigenvectors Decomposition），以特徵值（Eigenvalues）最大的前 p 個特徵向量（Eigenvectors）當成基底矩陣 θ_p 的行向量，最後所有資料可以利用求得的轉換矩陣投影到新特徵空間 $Y=[y_1,y_2,y_3,\ldots y_N]$。

3.3 傳統統計模型

3.3.1 k-最近鄰居法（k-Nearest Neighbors, kNN）

kNN 計算距離的方法是採用歐式距離法（Euclidean Distance），並依據距離遠近給予其對應權重，距離越近的，權重越大，而分數越高的，即為其分類。

而在模型的選擇中，k 是一個很重要的參數，我們觀察 k 個最近距離的鄰居類別，以多數決的方式決定該筆輸入的類別，在本研究，以準確率最高的 k=10 作為我們的預測模型。

3.3.2 隨機森林（Random Forest, RF）

RF 是一個由多棵決策樹（Decision Tree）所構成的模型組合，森林中的每棵決策樹都是沒有關聯的，而樹的每個節點皆為某個隨機抽取的樣本特徵，來將資料做區分。由於我們在此用的是二分法，即成功（Success）或失敗（Failure）的分類，所以我們以 Gini 係數（Gini Index）作為節點的最佳分類能力指標，其值越小，代表其為較佳的分類變數。

在建立 RF 的過程中，最重要的是決策樹的數量 N，通常 N 越大分類準確度越高，分類能力越好，在反覆實驗中，發現當 k＝900~1500 時，正確率達穩定，所以我們以 k＝900~1500 作為本研究的決策樹數量。

3.3.3 支持向量機（Support Vector Machine, SVM）

SVM 主要用來解決線性不可分（linearly non-separable）的問題。其可以通過事先選擇的非線性投影（non-linear projection），將 X 空間的向量 x 映射到一個高維度的特徵空間 Z，在這個空間中尋找最佳分類超平面（hyper-plane），而該超平面能使類別間的邊界距離達到最大，使分類更精準。本研究所使用的是一對一的方法進行分類，它可以分辨樣本是否屬於某個類別的 SVM。在 K 個類別

中，共有$\frac{K(K-1)}{2}$個 SVM 模型，也就是說，任兩種類別間都會有一個 SVM，去辨別輸入的資料是屬於哪個類別。

在本研究中，由於資料為非線性關係，所以我們使用高斯核函數（Radial Based Function），將資料映射到一高維度的特徵空間，並找到可以作為分類的最佳超平面。並且我們使用 soft-margin SVM，也就是容許一些分類錯誤的存在，避免 overfitting 的問題。而 C 值（cost）就是一個懲罰係數或是容錯項，他會給予分類錯誤的資料懲罰，當 C 的值越大，容錯的能力越小。本研究實驗發現，取 C=7 時有最佳的模型。

3.3.4 線性判別分析（Linear Discriminant Analysis, LDA）

LDA 是一個降維及分類兼具的分析方法，在本研究中，我們主要用於分類。在 LDA 中，最重要的參數為 w，使我們可以將樣本投影到 [1, w-1] 維空間內，找到具有最佳可分離性的維度。而 w 要由資料樣本空間的類別數來做決定，由於本研究中共有 10 個類別，因此我們在 [1, 9] 維空間內，找到最佳維度 9 並進行分類。

3.4 類神經網路

3.4.1 卷積神經網絡（Convolutional Neural Network, CNN）

現代的 CNN 層通常是以卷積層（Convolutional Layer）、激活層（Activation Layer）、池化層（Pooling Layer）所組成，最後會以卷積網路層所萃取出的特徵資訊做為分類依據，透過全連接層（Fully-connected Layer）進行分類。CNN 層是一組可平行運算的網路架構，通過卷積核（kernel）中的每個權重 $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ 與輸入的值 $X = (\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_k})$ 相乘後的加總，作為該次卷積核中心的特徵值 y，如式(3.11)。LeCun 於[17]提出了的權重共享（weight sharing）概念，不同於以往每次移動卷積核後，便以不同的權重進行運算的方式；權重共享的卷積核，在提升準確度的同時，亦能減少需要更新的權重數量，故現代卷積核通常使用權重共享的方式進行。依據核的移動步幅（stride）大小，依次以每個核的中心進行運算，重複此動作直至對所有輸入皆進行過一次後，代表一層 CNN 運算的完成，輸出結果稱為 Y，如式(3.12)。若設定的移動步幅較大，也就意味著該次的 CNN 層包含較少的運算，且也會輸出較少的特徵維度，同時也隱含著訊息遺失的可能。

$$y = \boldsymbol{\alpha x_i} \ , \ x_i \in \mathbb{R}^n, i = 1, 2, \dots, k \tag{3.11}$$

$$Y = \alpha[x_1 \ \ x_{1+s} \ \ x_{1+2s} \dots x_k] \ , \ s = size \ \ of \ \ stride \tag{3.12}$$

現今的 CNN 經常使用 ReLU 激活層[18]，如式(3.13)，其出眾的點在於：能

有效避免梯度消失問題，且在反向傳播法（Backpropagation）[17]的運算上，由於其一階導函數相對容易求得，因此成為現代主流的激活函數。

$$f(x) = \max(0, x) \tag{3.13}$$

池化層為現代 CNN 中另一個常被使用的層，除了多種不同形式的非線性池化函式以外，屬最大池化（Max Pooling）[19]和平均池化（Average Pooling）[3, 9]最為常見。而池化可以將輸入劃分為若干個區域，而取每個區域的特徵值作為輸出，也可透過調整移動步幅，決定輸出維度，通常會使用不重疊的池化層進行池化，由於重疊的池化層通常不會增進模型結果。此舉旨在進行維度縮減和特徵擷取，在一定程度上改善了過擬合的問題。

本研究輸入的特徵矩陣為 99 個時間點，每個時間點有 39 個特徵的 (99, 39) 矩陣，故我們採用 Convolution 1D 來做進一步的特徵擷取。首先，對於橫向時間點採用 kernel = 8，並將原本的 39 維重新取出 filter = 64 維不同權重的特徵，形成維度為 (92, 64)，並隨機將 3 成的神經元刪去可改善過擬合[2]的問題。再重複一次上述過程並微調參數後，會形成 (8, 32) 的維度，為了搭配類神經網路的全連接層，在這裡將特徵展開（Flatten）為 $8 \times 32 = 256$的向量，成為 Dense 層的輸入，首層 Dense 取輸出為 64，第二層 Dense 取輸出為 10，以第二層的輸出作為分類依據，如圖 2。

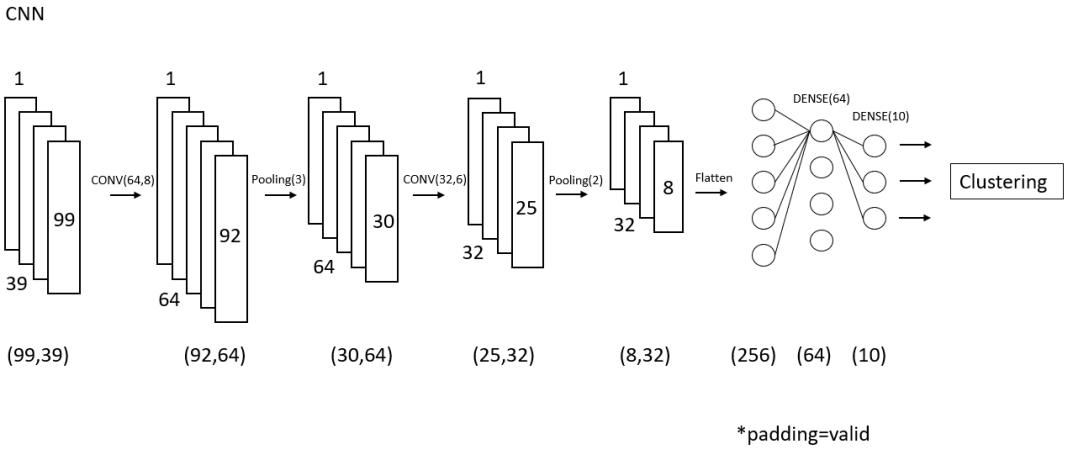

圖 2　CNN 架構圖

3.4.2 長短期記憶網路（Long Short-Term Memory, LSTM）[6]

LSTM 改善了 RNN 的缺陷，也就是無法記憶長期的時間數列資料[20]。LSTM 與 RNN 最大的不同，莫過於在每個 Cell 的輸出（hidden state）以外，新增加了細胞狀態（cell state）這一個連通所有 Cell 的資訊流。並透過 3 個 Gate 控制每次的更新，分別為 Input Gate、Forget Gate 與 Output Gate。Gate 採用的函數為 sigmoid 函數，也就是 $sigmoid(x) = \frac{1}{1+e^{-x}}$ ，sigmoid 函數的值域介於 0 和 1 之間，這同時也代表更新的幅度。透過輸入 Gate 的資訊，Gate 會輸出資訊的更新權重，若

Gate 的輸出值為 0，則代表著候選資訊完全無法通過；1 則代表可以完全通過。tanh 的值域介於 -1 與 1 之間，tanh 能透過輸入的資訊輸出候選資訊，以供細胞狀態的更新使用。而 LSTM 的最後，會以細胞狀態的輸出值作為候選資訊，透過 Output Gate 決定要輸出多少細胞狀態的資訊作為輸出(hidden state)，如圖 3。

Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

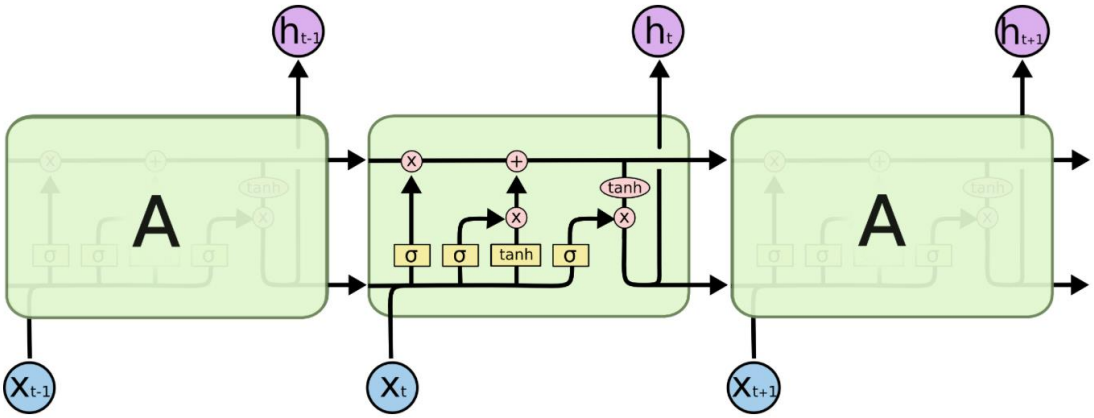

圖 3　LSTM 架構圖

3.4.3 CNN+LSTM 模型架構

本研究進一步使用 CNN+LSTM 模型架構，資料的 convolution 特徵擷取部分與前述的 CNN 模型並沒有做太大的改變；而明顯不同的地方在於，針對 CNN 的 Flatten 步驟，我們多建了一層 LSTM，改用該層的輸出值作為全連接層的輸入，建構出綜合 CNN 特徵擷取優勢及 LSTM 的循環前後記憶優勢，建構出我們的最佳模型，如圖 4。

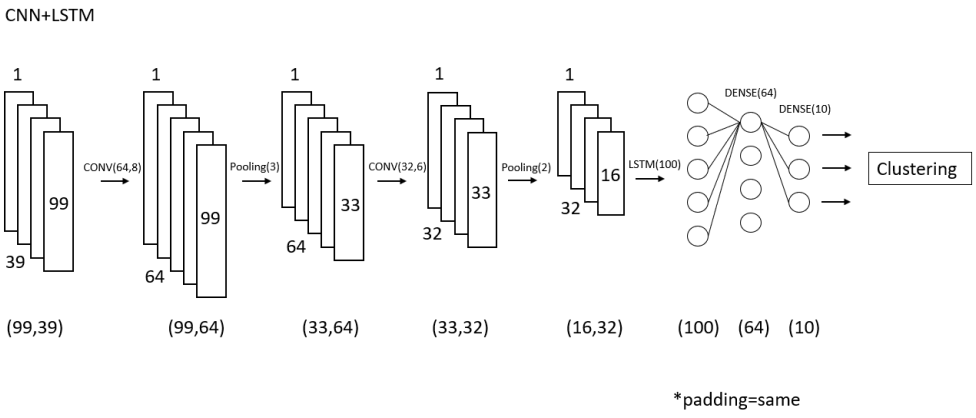

圖 4　CNN+LSTM 架構圖

而對於模型的參數選擇，除了 kernel、filter、dropout rate…函數內建參數的嘗試和選擇外，在建模的同時，我們發現平均池化會比最大池化效果來的佳，另外，機器學習時，因大量的循環及運算，可能造成梯度下降或爆炸的問題，而導致辨識率下降，因此我們在模型加入 batch normalization 也能適當的提升模型準

確率，並強化模型。最後在模型的訓練上，我們觀察到 val_loss 和 val_accuracy 的上下震盪問題，發現可能是 learning rate 造成的跳動問題，並進行調整，有效解決了模型的收斂問題和辨識率的再提升，如圖 5。

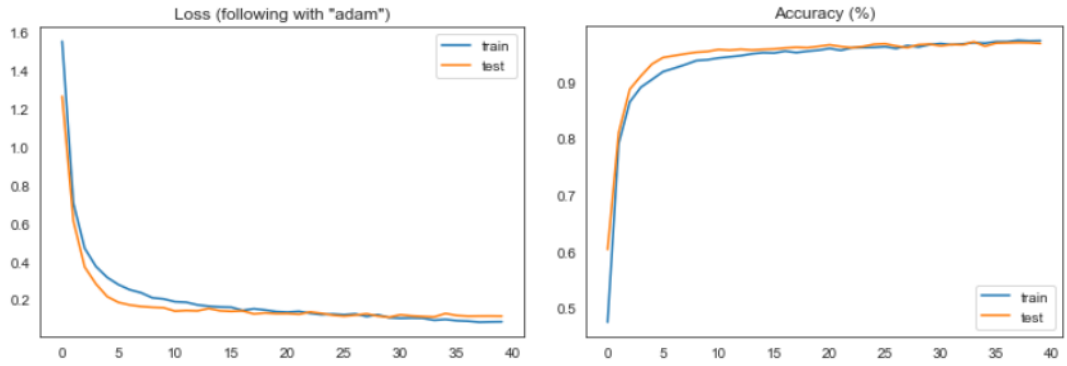

圖 5　CNN+LSTM Loss（左）　Accuracy（右）

3.5 Encoder-Decoder Attention 模型

現存的諸多模型皆存在著 Encoder-Decoder 的架構，例如：Seq2Seq、語音辨識、機器翻譯等領域，都以 Encoder-Decoder 模型進行相應的類神經網路領域研究[10, 12, 14]。Encoder 可以視為利用時間數列資料的時間點 t-1 進行預測，並輸出一個時間點 t 的結果，可表示成 $(\hat{e}_1, \hat{e}_2, \dots, \hat{e}_t) = Encoder(e_0, e_1, \dots, e_{t-1})$，其中 Encoder 的輸入稱為 Input，通常是欲預測的資料。而 Decoder 亦是利用時間數列資料的時間點 T-1 去預測時間點 T，可以表示成 $(\hat{d}_1, \hat{d}_2, \dots, \hat{d}_T) = decoder(d_0, d_1, \dots, d_{T-1})$。訓練完成後，Encoder 的最終輸出 $\hat{e}_t$ 即包含 $(e_0, e_1, \dots, e_{t-1})$序列的資訊，而其會作為輸入 Decoder 的一部份，放入 Decoder 模型當中，得出 Decoder 在每一個時間點的輸出。本模型的 Encoder 輸入層為經過 MFCC 轉換過後的一秒英文數字檔案，為 99 個時間點的 39 維 MFCC 特徵，而 Decoder 的輸入為該音檔的音素（Phoneme）。以 Decoder 在每個時間點的輸出作為特徵，透過全連接層（Fully-connected）辨識該時間點的音素為何。Attention 可視作嫁接 Encoder 與 Decoder 的橋樑，由於目前的 RNN 皆存在梯度消失以及梯度爆炸的問題，而 Attention 為一個十分簡單的全連接層架構，會對每一個時間點的 RNN 輸出進行特徵提取，並得出該輸出的注意力分配。使用 RNN + Attention 模型能夠過濾、提取出，對於輸出來說重要且具有時間相關性的特徵。

3.5.1 Encoder

我們的 Encoder 分為三大部分，如圖 6。分別為全連接層、ResNet 層[3]，以及 LSTM 層。全連接層先對每個時點的 39 維特徵進行線性投射至高維度，突顯該時點的特徵。經由三個 ResNet 層將鄰近的時間也納入考量，此處的 ResNet 將模型輸入經由三層 CNN 層進行特徵擷取，且將輸入與第三層 CNN 的輸出結

果相加，再透過 ReLU 輸出，可以有效避免模型的梯度消失。而為避免梯度爆炸問題，所以在每個CNN層的後面，都加上了一層Batch Normalization，如式(3.14)。將第一與第二層的 ResNet 每三個時間點的特徵進行連接，使時間長度 t 壓縮成 t/3，也就是每個新的時間點，皆包含 3 個時間長度的資訊，如式(3.15)，此金字塔架構（Pyramidal Structure）是參考[16]的 LSTM 金字塔架構，但為了加速模型訓練，而使用了可平行運算的 CNN 作為替代。時間的壓縮，有助於減少模型於 LSTM 層的運算時間，如此一來便可以加速模型訓練的進行。而最後一層的 ResNet 輸出，會作為 Encoder 的 LSTM 層的每一個時間點的輸入$(e_0, e_1, \dots, e_{t-1})$得到$(\hat{e}_1, \hat{e}_2, \dots, \hat{e}_t)$。

$$R_0, R_1, \dots, R_{t-1} = ResNet(I_0, I_1, \dots, I_{t-1}),\ I_i = Input, i = time\ \ step \quad (3.14)$$

$$I'_0 = [R_0\ \ R_1\ \ R_2], I'_1 = [R_3\ \ R_4\ \ \ R_5], \dots, I'_{\frac{t}{3}-1} = [R_{t-3}\ \ R_{t-2}\ \ R_{t-1}] \quad (3.15)$$

3.5.2 Attention

Attention可以簡單分為兩個全連接層，與一個輸出，稱為Context Vector（C）。依據[13, 15]所述，將 Encoder 的輸出$(\hat{e}_1, \hat{e}_2, \dots, \hat{e}_{t-1})$稱作 query（Q），而$\hat{e}_t$稱為 value（V）。透過兩個不同的全連接層 f 與 g，分別對 query 與 value 進行轉換過後，以逐點（element-wise）的方式相加，最後透過 tanh 進行轉換，得出第一層的結果（output, O），如式(3.16)。接著再次通過一層僅有一個結果的全連接層 h，便可得出 Encoder 輸出所形成的 score（S），如式(3.17)。而對 score 取 softmax 轉換後的結果，為該 Attention 的 weights（W），如式(3.18)。將 weights 與 value 以逐點的方式相乘並加總過後，稱為 Context Vector，如式(3.19)。

$$O = \tanh(f(Q) + g(V)) \quad (3.16)$$

$$S = h(O) \quad (3.17)$$

$$W = softmax（S） \quad (3.18)$$

$$C = \sum \mathrm{WV} \quad (3.19)$$

3.5.3 Decoder

Decoder 主要為一層 LSTM，其輸入為音素序列。音素序列首先會先通過一個 Embedding 層，將每個音素投射至指定維度的空間，使得音素不會因編號的數值大小影響輸出結果，音素序列（P）前後會加入序列開始（Start of sequence）與序列結束（End of sequence）的信號，亦會同時透過 Embedding 層進行轉換，稱經過 Embedding 的音素序列為音素向量（P'），如式(3.20)。接著將音素向量與 Decoder 的開始信號和 Attention 模型所輸出的 C 連接起來，作為 LSTM 層的輸入值，再對開始信號之後的每一個時間步進行預測，得出結果𝕝，如式(3.21)，直到預測的結果出現序列結束信號，又或者達到設定的預測個數上限時，才會停止。

最後，將該預測的結果通過全連接層進行分類，進而得到每個時間點所預測的音素。

$$P' = Embedding(P) \tag{3.20}$$

$$\mathbb{d} = Decoder([P' \ \ C]), \mathbb{d} = (\hat{\mathrm{d}}_1, \hat{\mathrm{d}}_2, \dots, \hat{\mathrm{d}}_\mathrm{T}) \tag{3.21}$$

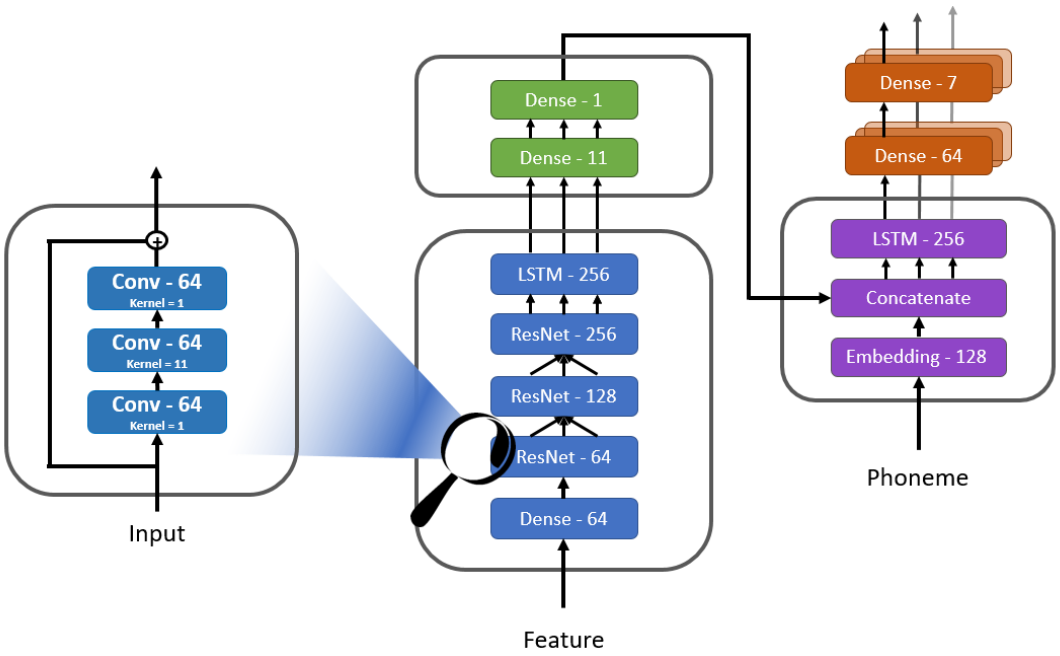

圖 6　Encoder-Decoder 架構圖

3.5.4 Encoder-Decoder 模型訓練

訓練 Encoder-Decoder 模型時，共讀取 20000 筆語音資料，每種類別皆讀取 2000 筆，並預留 1000 筆語音資料作為驗證集，進行訓練。每筆語音資料皆進行過 MFCC 轉換，並伴隨著相對應的標籤與音素。模型訓練是以四筆語音資料作為一個批次（Batch）進行，在一次 Epoch 中，總共有 4750 個批次，而模型每次的 Epoch 約需耗時 15 分鐘。經過反覆的測試過後，最終決定採用 Adam 優化器 [21]，並將模型的學習率（Learning rate）設定為 0.0001，Epoch 的最高上限訂為 20 次，當訓練集的損失小於 0.02 時則訓練終止。在聲音模型（Acoustic Model）的訓練完成之後，以該結果進一步訓練字詞模型（Lexicon Model），並以字詞模型所輸出的結果與其他模型進行比較。

第四章、研究結果

本研究中關於英文數字 0~9 的語音辨識，建立了多個模型來進行模型之間整體表現上的比較，如表 1。以準確率來說，不論以 MFCC 或頻譜圖（Spectrogram）進行辨識，後續以傳統統計模型辨識的準確率最高約為 90%，相對於類神經網路

的辨識率來得較低。但經過多次測試，發現在準確率並無明顯落差的情況下，用 PCA 取特徵值，能使模型配適時間縮短，故我們在傳統統計模型中皆使用 PCA 進行降維。

CNN 和 LSTM 模型在以往的語音辨識中，皆有不錯的表現。LSTM 模型訓練時間為 90 分鐘，CNN 訓練時間為 5 分鐘，模型準確率分別為 94%及 96%，不論是時間上或是準確率上，CNN 皆有較好的表現。本研究透過合併使用 CNN 與 LSTM 模型，不僅取得 CNN 可平行運算的優勢，且能結合 LSTM 考量時間相關的特性，以達到更好的效果。CNN+LSTM 模型訓練的花費時間僅比 CNN 模型多了 5 分鐘，而準確率卻提升了將近 1%。因此，在上述所有模型中，CNN+LSTM 模型在整體表現上為最好的模型。

本研究的 EN-DE 模型以 GTX1050 GPU 進行訓練，訓練 Epoch = 12 後，所花費的時間為 198 分鐘，模型準確率為 96%；而 Epoch = 20 時，總訓練時間為 348 分鐘。以準確率來說，EN-DE(20)的準確率超過 97%，是所有模型中準確率最高的;但其訓練時間及預測時間都相當費時，若以本研究的目標作為判斷依據，此模型較不適當。

在本次英文數字 0~9 的語音辨識上，考慮了準確度與模型訓練的時間，加以評估模型的整體表現。從準確度的面向來看，EN-DE(20)的準確率與 CNN+LSTM 模型相去不遠；而從模型訓練的時間剖析，EN-DE(20)所花費的時間大幅超越其他模型所需花費的時間；反之，CNN+LSTM 模型僅需 10 分鐘，故本次研究之最佳模型為 CNN+LSTM 模型。

表 1 模型結果比較

Model	**MFCC**		**Spectrogram**	
	Acc(%)	Time (m)	Acc(%)	Time (m)
LDA	62.76	0.01	65.45	0.01
RF	79.13	6.08	80.89	2.37
kNN	86.08	0.64	82.82	0.49
SVM	90.51	1.51	88.74	0.59
LSTM	94.50	90.69		
CNN	96.10	5.21		
CNN + LSTM	96.90	10.37		
EN-DE (12)	96.58	198.87		
EN-DE (20)	97.37	348.23		

第五章、結論與建議

本次研究探討了多種模型架構對於語音辨識的預測能力，分別架構了多種傳

統統計模型與 LSTM 模型、CNN 模型及 CNN+LSTM 模型。在本次英文數字 0~9 的語音辨識任務中，傳統統計模型之中，表現最佳的 SVM 模型，其準確率相比起類神經網路，仍有相當大的進步空間；而類神經網路當中，表現最佳的為 CNN+LSTM 模型，該模型綜合準確率與訓練時間，皆有較出眾的表現，因此，在固定長度的單字語音辨識中，CNN+LSTM 的模型架構可以取得較優良的成果。

EN-DE 模型在本次語音辨識任務中，訓練時間雖較長，但在準確率上，取得十分優異的成績。而本次研究著重於固定長度的單字語音辨識，所以雖說整體上 EN-DE 模型並非最為合適的模型，但在語音辨識的發展性上，因其可以對非固定長度的語音，也就是對連貫的單字或語句，進行連續的預測。若以將來的可發展性做為考量依據，則本研究相當看好 EN-DE 模型的發展及其應用。

參考文獻

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278-2324, 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[4] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing,* vol. 37, no. 3, pp. 328-339, 1989.

[5] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing,* vol. 22, no. 10, pp. 1533-1545, 2014.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," 9, Neural computation, 8, 1997.

[7] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," in *Advances in neural information processing systems*, 2013, pp. 190-198.

[8] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *arXiv preprint arXiv:1312.6026,* 2013.

[9] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the*

IEEE conference on computer vision and pattern recognition, 2015, pp. 1-9.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104-3112.

[11] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," *arXiv preprint arXiv:1508.01991,* 2015.

[12] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078,* 2014.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473,* 2014.

[14] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259,* 2014.

[15] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025,* 2015.

[16] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211,* 2015.

[17] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural computation,* vol. 1, no. 4, pp. 541-551, 1989.

[18] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807-814.

[19] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *International conference on artificial neural networks*, 2010: Springer, pp. 92-101.

[20] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks,* vol. 5, no. 2, pp. 157-166, 1994.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

附錄

組員分工表

組員	工作分配
黎薇	書面彙整
許月華	統計模型
張崴智	看影片、與教授 meeting
王勃淵	EN-DE 模型
姜孝軒	CNN+LSTM 模型