國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

基於專注式類神經網路之端對端口述詞彙偵測

End-to-End Spoken Term Detection Based On Attention-based Nerual Network

敖家維

Chia-Wei Ao

指導教授：李宏毅 教授

Advisor: Hung-Yi Lee, Ph.D.

中華民國一百零六年六月

June, 2017

國立臺灣大學碩士學位論文
口試委員會審定書

基於專注式類神經網路之端對端口述詞彙偵測

End-to-End Spoken Term Detection Based On Attention-based Nerual Network

本論文係敖家維君 (R04942094) 在國立臺灣大學電信工程學研究所完成之碩士學位論文，於民國 106 年 6 月 xx 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：

_____ (簽名)

(指導教授)

_____ _____

_____ _____

_____ _____

_____ _____

系主任、所長 _____ (簽名)

誌謝

摘要

Contents

圖目錄

表目錄

第一章　導論

1.1　研究動機

隨著網路跟電腦的普及，電腦資訊愈來愈豐富，使得我們尋找所需的資料變成一個大問題。因此，我們需要一套好的檢索系統（Retrieval System）幫助我們快速地瀏覽資訊，並從中找到有用的部分，在過去已有許多文字檢索系統與演算法被開發出來並應用於產業中，如 Google Search、Microsoft Bing Search、Yahoo Search等。隨著錄影音設備的普及，語音文件量正蓬勃地增加當中，隨著線上影片、會議錄音、線上課程等網站的興起，語音資料量越來越多，因此如何在其中找到使用者感興趣的資料便成為重要的議題，即為語音數位內容檢索（Spoken Content Retrieval） [4,5]。相較於文字資訊檢索，語音資訊檢索面臨到更多的挑戰，如辨識錯誤、辨識訓練資料不足等問題，使得此問題更形困難。

更由於行動裝置的出現，使用者可以不受地形時間限制，隨時取得資訊，促使許多網路公司一一推出了用語音輸入來檢索文字資訊的系統，如Google 公司推出的語音檢索功能即可讓使用者在手機或瀏覽器的介面上以語音輸入，由 Google 將其辨識成文字後再於 Google 的搜尋引擎上檢索資訊。 Apple 公司推出的個人語音助理 Siri，也讓使用者能以十分自然的方式對 Siri說出想要查詢的查詢詞（Query），由 Siri 辨識後在網路上檢索，並將檢索結果分門別類整理好後呈現給使用者看。如上述所說的這類檢索系統是用語音輸入的查詢詞去檢索大量的文字資訊，此方法稱為人聲檢索（Voice Search），和本論文所探討的語音數位內容檢索（Spoken Content Retrieval）完全不同。

本論文所探討的語音數位內容檢索，是指由於網路上有大量的多媒體文件，如線上影片、會議錄音、線上課程、電視連續劇、演講等，而使用者也有搜尋這

些多媒體文件的需求，此類允許使用者用文字或聲音輸入查詢詞並搜尋語音數位內容（Spoken Content）的系統稱為語音數位內容檢索，如 TED（美國著名的演講網站）會將網站上的演講內容轉寫（Transcript） 成為文字，並允許使用者於網站上輸入文字檢索這些影片的文字稿。 Youtube 也會於離線時將其網站上的影片辨識成文字，但目前尚不支援直接輸入查詢詞檢索影片轉寫的方式，可以期待未來 Youtube 會開放這方面的功能。只是上述兩個例子仍要倚賴人工的轉寫，要完全只靠機器自動辨識仍不容易做到。這種語音數位內容檢索將是本論文主要的研究主軸。

語音資訊檢索系統的效能取決於其前端語音辨識系統的效能，只要可以發展出完美的語音辨識系統，能夠完全正確的把聲音轉寫為文字，那需要將文件資訊檢索的方式套用到語音辨識的文字輸出上就解決語音資訊檢索這個問題了。依此邏輯來看，與其研究語音檢索，不如去研究如何提升語音辨識的正確率，而當完美的語音辨識系統被創造出來，語音檢索這個問題也就沒什麼好研究的了。因此TREC SDR track [6]在2000年的時候即宣稱語音檢索在廣播新聞上（BroadcastNews）是一個「已解決的問題」(solved problem)，而在當時廣播新聞的辨識正確率已經到達90%以上，但在自發性（Spontaneous）語音上，語音辨識正確率往往不到50%，進而增加了在自發性語音上檢索的難度 [7,8]，即使最近很流行的深度學習（Deep Learning）大幅提升語音辨識系統的能力，語音辨識仍然是個困難的問題。短時間內語音辨識錯誤似乎是不可避免的，辨識錯誤對語音檢索所帶來的傷害幾乎在這個領域的每一篇論文的導論都會提到，用以彰顯語音檢索仍是個值得研究的問題，然而卻沒有太多方法能真正突破語音辨識錯誤所帶來的限制。並且傳統的語音數位內容之語意檢索系統主要的實作方法為先將語音文件辨識為文字檔後，對文字做檢索，但在辨識之中，會遇到如詞典外詞彙

（Out Of Vocabulary）、辨識錯誤等情況，更甚者，許多語音中珍貴的資訊如韻律（Prosody）、語速（Speaking Rate）和語者特徵（Speaker Characteristic）等在辨識後就消失了，十分地可惜。

本論文因此將語音辨識的部分移除，直接在語音訊號本身進行搜索，以類神經網路直接學習語音訊號的相似性，有別於傳統的聲學模型訓練强調辨識正確率的提升，本論文所提出的移除語音辨識，直接由語音訊號進行搜索提升語音檢索的效能。

1.2 研究方向

本論文之研究方向為使用類神經網路强化檢索系統之語音資訊檢索，主要包含以下幾點:

- 傳統的語意檢索系統是先將語音文件辨識為文字後，將輸入的文字查詢詞進行檢索，語音辨識系統的影響會直接反應在搜尋結果上，但自動語音辨識系統（Automatically Speech Recognition） 的訓練是很昂貴的，需要大量標注完善的語料才能訓練出很好的聲學與語言模型。因此，利用類神經網路擁有有良好的推廣性（Generization） 和適應性（Adaptability）强的優點，直接對於語音信號本身搜索，避免語音辨識的錯誤。
- 更進一步地，由於語音文件無法跟文字文件一樣清楚分段，對於模型來說計算量是很龐大，且充斥著雜訊。因而，引進了專注式機制（Attention Mechanism），來使模型能夠將目光注目特定的位置上，減少雜訊的干擾，進而學習出更好的向量表示（Vector Representation）
- 由於學習的向量表示，會過度貼近（Overfit）訓練資料，導致推廣性的下

降。使用自動編碼器（Auto-encoder） [9–11] 產生文字向量（Word2vec），語音向量（Audio2vec） [12–15]，進而進行搜索。

- 再者，利用了成對學習法 （Pairwise Learning）使得原本的分類問題加入了排序（Ranking）的概念，使得系統能夠更加精準的分出正確資料與錯誤資料間的差異，進而提升檢索系統。
- 最後，將上述的模型統整在一起，將語音向量當作正規化（Regularization），利用畫重點機制排除多餘雜訊，提升模型的效能，使搜尋結果更加進步。

1.3 章節安排

本論文之章節安排如下：

- 第二章：介紹本論文相關背景知識。
- 第三章：介紹如何以類神經網路實現口述詞彙偵測。
- 第四章：介紹如何以畫重點機制類神經網路實現口述詞彙偵測。
- 第五章：介紹如何產生語音向量跟文字向量，並應用於口述詞彙偵測
- 第六章：介紹如何以成對學習法訓練檢索系統。
- 第七章：介紹將畫重點機制跟語音向量同時應用在口語詞彙偵測。
- 第八章：本論文之結論與未來研究方向。

第 二 章　背景知識

2.1　資訊檢索與語音資訊檢索

2.1.1　資訊檢索

資訊檢索（Information Retrieval,IR）是指從資料庫中擷取出與某個主題（Topic）相關之檢索對象（Object）的行為。基本使用流程為：使用者輸入了一段查詢對象 Q（Query），系統能夠從資料庫中傳回和這個主題相關的檢索對象（Object），並進一步依據檢索對象和主題的相關程度進行排序（Ranking），以方便使用者瀏覽並取得所需的檢索對象。如圖2.1。

檢索對象 x 指的是系統所檢索出的東西，例如：在文字搜尋中檢索對象指的是文章、在網路檢索中指的是網頁、在圖像檢索指的是圖片、在語音檢索指的是口語片段（Spoken Segment）等，同時，查詞對象 Q 也可以為圖片、文字或聲音。 本篇論文中所使用的查詢詞 Q 形式有二：文字構成的字串、口述形式。本篇論文使用 x 則都是語音文件。

當有了查詢對象，系統的主要任務就是去計算相關性（Relevance），其定義為 $P(R|x,Q)$ ，然後依此機率大小對檢索對象進行排序，這個排序方式 被稱為機率排序原則（Probability Ranking Principle,PRP）[16]。 這個相關性函式通常是按照檢索系統的需求去決定的，可以是用機器學習（Machine learning）的方法學習出來，也可以是由系統設計者決定。上述的方法可以視為一個分類問題（Classfication），另一種方法為排序學習（Learning to Rank），此時系統不在需要計算 $P(R|x,Q)$ ，而是尋找一個排序函式（Ranking Function） $S(x,Q)$ ，能夠將 $S(x_T,Q) > s(x_F,Q)$ ， x_T 為相關的檢索對象、 x_F 為不相關的檢索對象。也就

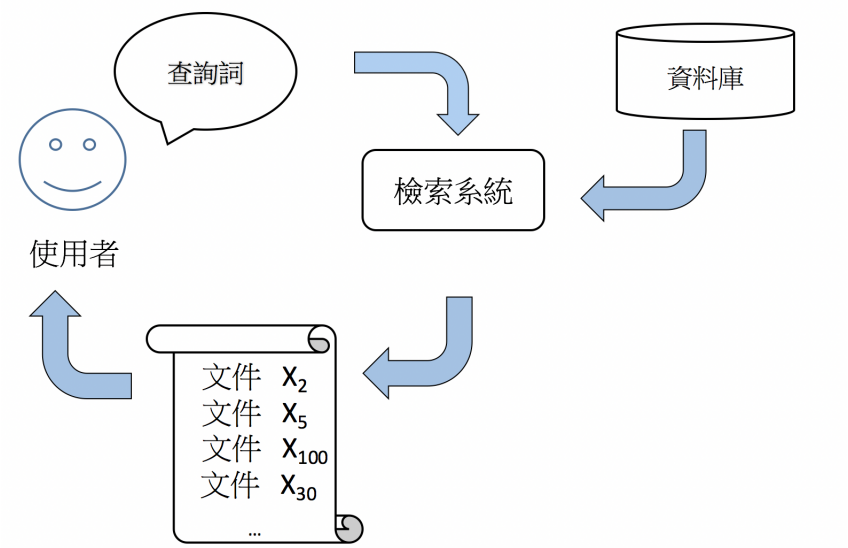


圖 2.1: 資訊檢索系統基本架構

是說，所有相關的檢索對象，都能夠超過不相關的檢索對象。

如何檢驗一個資訊系統的好壞也是重要的課題首先我們必須要準備測試集，其中包含文件資料庫、查詢詞、以及與每個查詢詞對應的相關文件的標註。藉由評估系統在測試集上於「正確度」、「速度」、以及「佔用空間」等數個面向的表現，以了解此系統與其他系統的優劣比較。

文字檢索會議（The Text REtrieval Conference,TREC），自1992年起每年的居舉行會議，並在多個測試項目上提供大規模之標準測試集，藉此讓來自世界各地的參加團隊評估自己的系統效益同時與其他團隊相互討論切磋。如今，TREC除英文的檢索測試集外，更提供了非英文的大規模檢索測試集（如西班牙文與中文）、語音檢索測試集、以及跨語言檢索測試集。另外在測試項目上也有了多樣的發展，包括引入了開放領域自動問答（Open-Domain Question Answering）以及基於內容的數位影音檢索（Content-based Retrieval of Digital Video）等。資訊檢索已是一門受到關注的領域。

2.1.2 語音資訊檢索

目前語音檢索可以分為兩類，「口述語彙偵測 （Spoken Term Detection）」與「語意檢索 （Semantic Retrieval）」，本論文著重在口述詞彙偵測，在此將兩者簡介如下：

口述語彙偵測

口述語彙偵測的目的是檢索出所有包含查詢詞的語音文件，著重於逐詞比對（Literal Term Matching），主要有兩種檢索情境。第一種是先將語音文件辨識為詞圖（Lattice）後，當使用者輸入文字的查詢詞後，系統會在詞圖上進行查詢詞的檢索。這種檢索情境需要有訓練好的自動語音辨識系統（Automatic Speech Recognition,ASR），由於語音辨識系統並無法保證在所有情況下都有很好的辨識率，因此由辨識錯誤所導致的檢索性能下降是在此最需要解決的問題。過去的文獻有用聲學特徵如梅爾倒頻譜係數 （Mel-Frequency Cepstral Coefficients （MFCC））幫助分類器 （Classifier） 在分類一篇語音文件是否相關時的判斷，也有利用相關回饋 （Relevance Feedback）、圖論 （Graph）與隨機漫步 （Random Walk） 解決這些問題。

第二種是非監督式 （Unsupervised），的方法，查詢詞與語音文件都是語音形式的，也有人稱之為依例查詢 （Query-by-Example），系統直接利用如動態時間校準（Dynamic Time Warping） 等方法在信號上比對語音文件中是否有某一段聲音與查詢詞很相像，過去的方法通常是為了解決不同文件間語速上的差異提出如有斜率限制的動態時間校準（Slope-Constraint DTW），或是為了解決動態時間校準逐一比對所有文件庫所花時間過多的問題。計算此情境下的 $S(Q, x)$ 為利用片段式動態時間規劃 （Segmental Dynamic Time Warping），簡介於 2.1.3。

語意檢索

語意檢索的目的為回傳概念上相關的語音文件，而不一定要出現查詢詞。希望查詢結果與輸入的查詢詞是概念上匹配（Concept Matching）的，而不是只回傳有出現查詢詞的文件，如當使用者查詢「美國總統」，文件中包含「美國總統」、「美國白宮」，而不是與查詢詞完全一樣的文件，通常也是使用者想要的。文字領域的資訊檢索已經有「概念匹配」的做法來達到語意檢索，但是由於文字領域的資訊檢索是在文字為完全正確的狀況下，不像語音數位文件的檢索往往會有辨識錯誤的問題。對於語音文件，一個較常見的實作方法通常是使用相關回饋（Relevance Feedback） [17] 與查詢詞擴展（Query Expansion) [18,19]。

2.1.3 片段式動態時間校準 （Segmental DTW）

口述語彙偵測的目的為搜尋整個語料庫後，找出其中有出現查詢詞的語音文件，同時找出這些語音文件中可能有出現查詢詞的假設區域 (Hypothesized Region) ，並給這個假設區域一個相關分數，最後系統再根據相關分數進行排序後回傳給使用者 (分數較高為較相關)。

這裡考慮的口述語彙偵測是查詢詞也是語音形式的情境。此時所用的方法為片段式動態時間校準 (Segmental Dynamic Time Warping) [20,21]。假設輸入的查詢詞的特徵為 $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_{|\mathbf{X}|})$ ，語音文件的特徵為 $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_{|\mathbf{Y}|})$ ，有了這兩者之後，就可以建立一個距離表格 (Distance Table) $D(i, j) = \rho(\mathbf{x}_i, \mathbf{y}_j)$ ，通常這裡使用的特徵是 高斯事後機率 （Gaussian Posteriorgrams） [2]或是梅爾倒頻譜係數 （MFCC）。如果使用高斯事後機率的話，兩個音框之間的距離為 $\rho(\mathbf{x}_i, \mathbf{y}_j) \equiv -\log(\mathbf{x}_i \cdot \mathbf{y}_j)$ 。如果使用梅爾倒頻譜係數的話，兩個音框之間的距離為兩點之間的歐幾里得距離（Euclidean Distance），即 $\rho(\mathbf{x}_i, \mathbf{y}_j) \equiv \sqrt{|\mathbf{x}_i - \mathbf{y}_j|^2}$ 。

動態時間校準的目的是要在 $D(i,j)$ 上找一條距離總合最短的路徑從 $(1,s)$ 到 $(|\mathbf{X}|,e)$ 表示從 $\mathbf{X}$ 對應到 $(\mathbf{y}_s,...,\mathbf{y}_e)$ (因為查詢詞一定要被完全對應，而語音文件不一定要被完全對應到)。由於假設區域可以出現在語音文件中的任何地方， 因此片段式動態時間校準將距離表格 $D(i,j)$ 切成數個重疊的對角片段 (寬度為 R)，所以說每個片段的起始點分別為 $(1,1),(1,1+R),(1,1+2R),...$ 如圖 2.2 所示，每個對角片段都代表了一個可能出現查詢詞的區域，所以要在每個對角片段上找出距離總合最短的路徑，即這個對角片段上的假設區域。片段式動態時間校準會從每個片段中找出一段距離總合最短的路徑，在每個片段中所有對應的路徑 (Warping Path) 都必須要完整地待在片段內，不可超出片段。

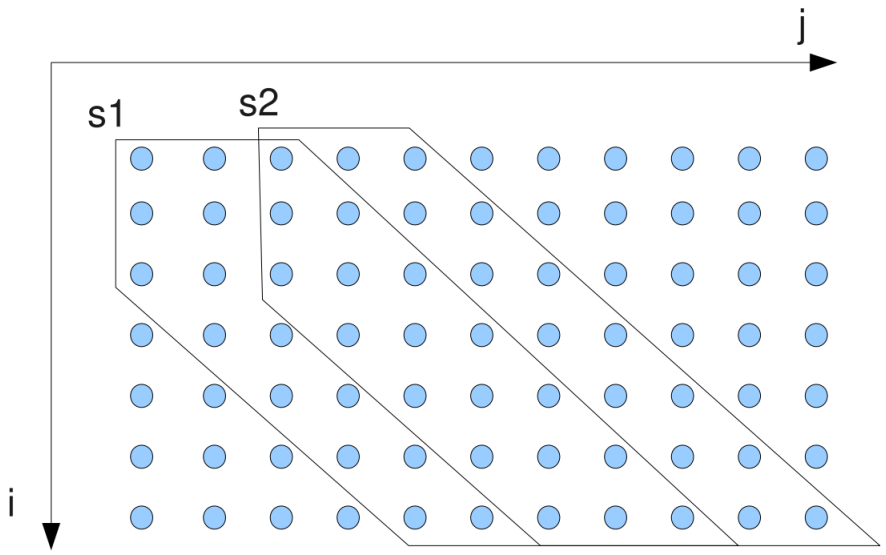


圖 2.2: 片段式動態時間校準示意圖 [2]

假設在每個片段中的對應路徑為：

$$\phi = (i_t, j_t), t = 1, ..., |\phi|$$

代表著如下的對應關係：

$$\mathbf{x}_{i_1} \leftrightarrow \mathbf{y}_{j_1}, \mathbf{x}_{i_2} \leftrightarrow \mathbf{y}_{j_2}, ..., \mathbf{x}_{|\phi|} \leftrightarrow \mathbf{y}_{|\phi|},$$

而邊界條件是 $i_1 = 1, i_{|\phi|} = |\mathbf{X}|$ ， j_1 為 $1 + kR$ ，對應路徑中所有的點都要在片段內，即：

$$|(i_t - i_1) - (j_t - j_1)| <= R$$

而片段式動態時間校準的目標是要在每個片段內找到一條路徑 ϕ 使得下式的距離總和最小：

$$C_\phi(\mathbf{X}, \mathbf{Y}) = \sum_{t=1}^{|\phi|} \rho(\mathbf{x}_{i_t}, \mathbf{y}_{j_t}) \tag{2.1}$$

每個片段中使得 $C_\phi(\mathbf{X}, \mathbf{Y})$ 最小，即使得相關分數 $-C_\phi(\mathbf{X}, \mathbf{Y})$ 最大的那條 ϕ 即為每個片段中的假設區域 $(\mathbf{y}_{j_1}, ..., \mathbf{y}_{j_{|\phi|}})$ ，而在每個片段中找到最大相關分數的路徑可以使用動態規劃 (Dynamic Programming) 求解。圖 2.3中顯示了兩個對角片段與其對應的最大相關分數的路徑與假設區域。圖中上半部中的 $(\mathbf{y}_1, ..., \mathbf{y}_8)$ 代表在第一個對角片段中找到的假設區間，圖中下半部中的 $(\mathbf{y}_3, ..., \mathbf{y}_8)$ 則是在第二個對角片段中找到的假設區域。 找到所有假設區域後，將假設區域按照其與查詢詞的相關分數進行排序後，即為口述語彙偵測的結果。在一篇語音文件中找到所有可能的假設區域需要 $O(|\mathbf{X}||\mathbf{Y}|)$ 的計算量來計算點與點之間的距離 ρ ，並需要 $O(|\mathbf{X}||\mathbf{Y}|)$ 的計算量來找到相關分數最大的路徑。

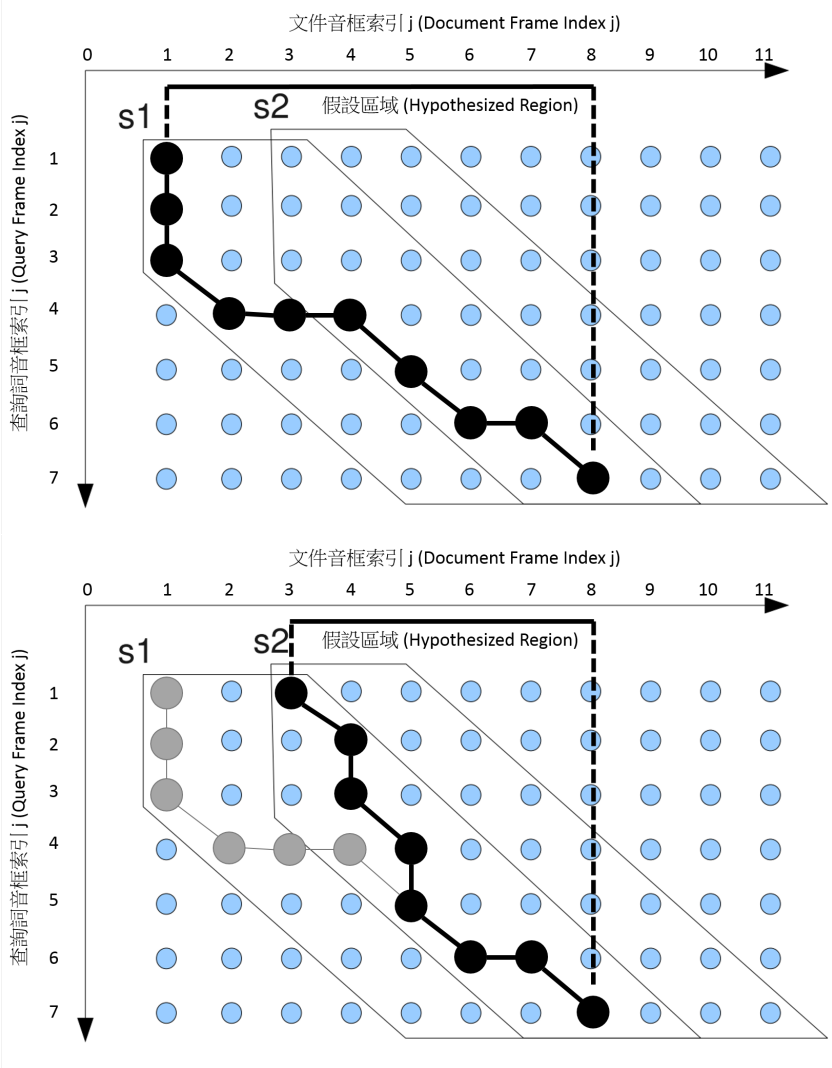


圖 2.3: 示意如何從兩個對角片段中找到相關分數最大的假設區域，其中 $R = 2$

2.1.4 資訊檢索評估機制

為了有效比較彼此系統，制訊檢索評估機制的標準是很重要的一環，本節將對一些常見以及本論文所使用的評分標準一一做介紹。

準確率(Precision)與召回率(Recall)

檢索系統找出的所有可能相關檢索對象中，真正相關的比例稱之為準確率，準確率高代表所找出的檢索結果的可信度高； 而所有真正的相關檢索對象有多少比例

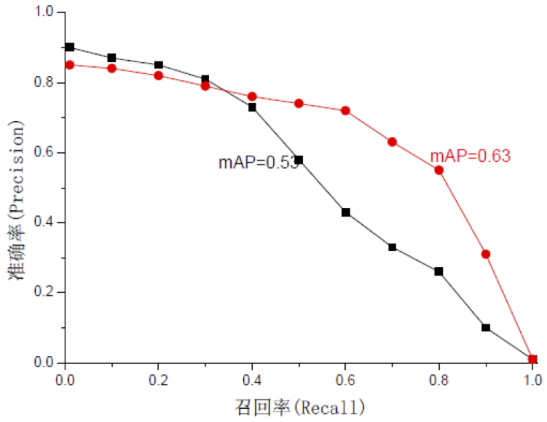


圖 2.4: 準確率、召回率和平均準確率之關係

被系統檢索出來，我們稱之為召回率，召回率高代表系統找回越多相關的檢索目標。通常會為系統設定一個閥值(Threshold)，文件的分數若高於閥值，則視為相關，反之若文件的分數低於閥值，則視為不相關。準確率和召回率的定義如下：

$$\text{準確率} = \frac{\text{檢索到的相關檢索對象數}}{\text{檢索到的檢索對象數}}$$

$$\text{召回率} = \frac{\text{檢索到的相關檢索對象數}}{\text{所有的相關檢索對象數}}$$

通常這兩個值彼此之間的關係為負相關。調高閥值的話準確率會上升，但召回率則會下降；反之若調低閥值，準確率會因此下降，召回率則會很高。可以考慮一個極端例子：當閥值非常低時，幾乎所有的文件都是相關文件，此時的召回率相當於1，但準確率就會很低了。因此單看準確率或召回率是無法準確地評估系統的優劣的，必須要兩者一起評估。

P@N

通常使用者最重視的是檢索系統傳回的前幾名結果，所以就發展出了 $P@N$ 這個

評估機制。 $P@N$ 就是只看前N個檢索結果的正確率。例如：前五個檢索結果中有一個是相關的，那 $P@5$ 就是20%。

$P@N$ 的定義如下：

$$P@N = \frac{\text{前N個文件裡的相關文件數}}{N}$$

然而由於不同的查詢詞其相關的檢索對象數不同，因此 $P@N$ 有時候並非公平的評比方式， 舉 例來說，如果整個資料庫裡面相關的檢索對象只有5個， $P@10$ 最高就只能到 0.5。

平均準確率 [22]

因為準確率和 $P@N$ 都需要事先決定，當查詢詞和條件不同時，很難準確地評估兩個系統的效能。因此有人提出了平均準確率(Mean Average Precision, MAP)的概念，如圖 2.4，平均準確率就是準確率和召回率曲線下面積的平均值。平均準確率的定義如下：

$$MAP = \frac{1}{|Q|}\sum_{Q}\frac{\sum_{d\in D^R} precision(d)}{|D^R|} \tag{2.2}$$

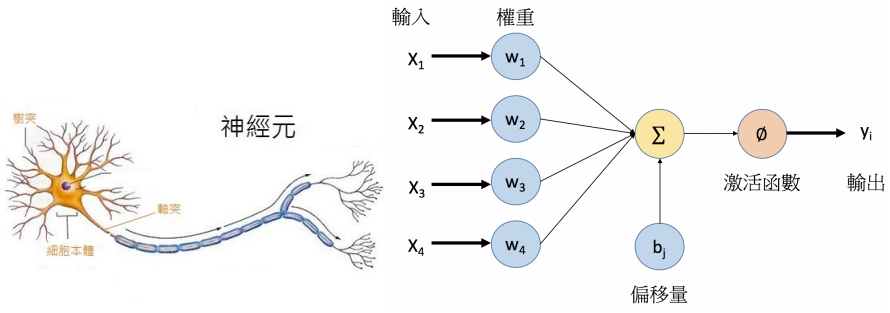
其中Q代表查詢詞的集合， $|Q|$ 為查詢詞的總數， D^R 為和查詢詞Q相關的文件d的集合， $|D^R|$ 代表和查詢詞Q相關的文件數量。precision(d)代表系統檢索出文件d時的準確率。

2.2 深層類神經網路（Deep Neural Network, DNN）

2.2.1 簡介

深層類神經網路發想於生物的神經系統結構。神經系統由非常多的神經元組成，彼此以樹突、軸突與突觸連結，每顆神經元自己有活化閾值來決定激發 態與否。

根據此仿生的觀察，便產生一種數學模型（如圖 2.5a與 2.5b），將其層層疊起（如圖 2.6），即為深度類神經網路的全貌。



(a) 生物學上的單顆神經元結構　(b) 數學上的單顆神經元結構（感知器）

圖 2.5: 類神經網路的單顆神經元

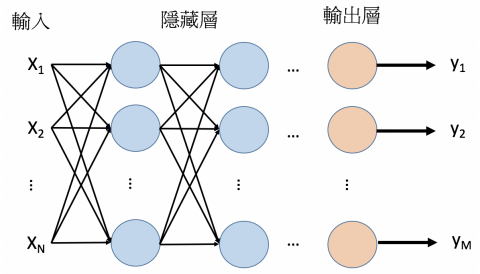


圖 2.6: 基本深度類神經網路架構圖

深度類神經網路是目前機器學習重要的一個分支，曾經在1980年代蓬勃發展，但因計算代價太高，且同時有其他簡單的模型如支撐向量機（Support Vector Machine）的興起而沒落。深層學習的概念由辛氏（Geoffrey Hinton）於2006重新推出 [23]，並使用一種新型的訓練方法，能夠有效提升訓練的執行速度；加上圖形處理器（Graphics Processing Unit，GPU）大幅提高了數值和矩陣運算的速度，使得深層類神經網路重新成為機器學習領域中的重要角色。

2.2.2 運作原理

類神經網路的架構如圖 2.6，其結構由感知器（Perceptron) 圖2.5b串接而成，因此深層類神經網路又稱為多層感知器（Multi-layer Perceptron，MLP）。每一層根據所在位置可分為三類：

- 輸入層(Input Layer): 即為類神經網路輸入特徵向量（Feature Vector）$\mathbf{X} = [x_1, x_2, x_3, x_4..., x_N]^T$
- 隱藏層(Hidden Layer): 類神經網路的主要部分，可以有很多層，且每層的感知器（神經元）數目可以不同。
- 輸出層(Output Layer): 跟隱藏層很相似，不過會依據模型的目的有所改變。譬如回歸問題，此時輸出層就不會經過活化函數。對於分類問題，輸出層的數量會跟標記種類（Class）相同。

每個感知器都包含了一組加權係數跟偏移量(Bias)，跟非線性的活化函數（Activation Function）。數學式如下：

$$y_i = \phi(\sum_{i=1}^{M} w_{ij} x_i + b_j) \tag{2.3}$$

其中 ϕ 是活化函數， w_{ij} 是第 j 個感知器中，對應到第 i 個輸入 x_i 的加權係數， b_j 是第 j 個感知器的偏移量。 ϕ 括號內的特徵轉換稱為仿射變換（Affine Transformation）。可將之想像為一個從 M 維實數空間映射到 N 維實數空間的函數 $A : R^M \rightarrow R^N$ ，其中 M 和 N 分別是輸入向量 x 與輸出向量 y 的維度。由於仿 射變換可看成是一個線性矩陣的轉換，複雜度並不夠高，因此我們引入 ϕ 這個非線性的活化函數。常見的活化函數分別為 S 型函數（Sigmoid）和整流線性單元

（Rectified Linear Unit, ReLU）。

$$
\begin{aligned}
sigmoid(x) &= \frac{1}{1+e^{-x}} \\
ReLU(x) &= max(0,x)
\end{aligned}
\tag{2.4}
$$

2.2.3 訓練類神經網路

深層類神經網路常見的訓練方法為反向傳播演算法（Back Propagation）[24]，通常要搭配一些最佳化演算法（Optimization Method）例如梯度下降法（Gradient Descent）。其概念為模型會計算出當下錯誤的程度，並依照可以減少錯誤的方向更改參數。為了定義錯誤的程度，需借用到損失函數（Loss Function）來定義錯誤，其訓練的目標可以規劃為以下的最佳化問題（Optimization Problem）:

$$
\min_{\theta} \sum_{n=1}^{N} L(\mathbf{x}_n, \mathbf{y}_n, \theta) \tag{2.5}
$$

損失函數通常是要能夠反應在你預想的輸出跟模型的輸出的真實距離。損失函數的值越大，代表模型的輸出結果與期望目標相差越遠，也因此訓練的目標為最小化損失函數，所以定義一個好的損失函數是很重要的。

以多類別分類器為例，分類器的輸出 $\hat{\mathbf{y}}$ ，會將每一個維度對應到一個分類標籤，共有 C 種類別。屬於正確標籤的 $\mathbf{y}$ 表示為一個獨一餘零（1-hot）的向量 $[0,0,...,0,1,0...,0]^T$ ，只有正確類別 l 的值是1，其餘為0。訓練此種分類器時，損失函數常為交叉熵（Cross Entropy, CE），定義為：

$$
L_{CE}(\mathbf{x},\mathbf{y},\theta) = KL(\mathbf{y}||f_\theta(\mathbf{x})) = KL(\mathbf{y}||\hat{\mathbf{y}}) = \sum_{i=1}^{C} y_i \log \frac{y_i}{\hat{y}_i} = -\log \hat{y}_l \tag{2.6}
$$

其中 $KL(p||q)$ 代表的是克雷散度（Kullback–Leibler Divergence），用以衡量兩組機率分佈的距離，值越大則代表兩組分佈越不相似。 $L_{CE}(\mathbf{x},\mathbf{y},\theta)$ 計算了分類器的正確標籤 $\mathbf{y}$ 與通過模型參數後的預測向量 $\hat{\mathbf{y}}$ 的克雷散度，但由於 C 種正確標籤

中，只有一個類別 l 的值為1，也因此 $L_{CE}(\mathbf{x}, \mathbf{y}, \theta)$ 可以簡化成公式2.6最右邊的簡單型態，最小化正確分類標籤的負對數可能性（Negative Log Likelihood, NLL）。

以多類別回歸器而言，則更廣義地希望將回歸器的輸出向量 $\hat{\mathbf{y}}$ 與目標向量 $\mathbf{y}$ 的距離拉近。訓練此種分類器時，距離的衡量通常為均方差（Mean Square Error, MSE），定義為：

$$L_{MSE}(\mathbf{x}, \mathbf{y}, \theta) = ||\mathbf{y} - f_\theta(\mathbf{x})||_2 = ||\mathbf{y} - \hat{\mathbf{y}}||_2 = \sum_{i=1}^{C}(y_i - \hat{y}_i)^2 \tag{2.7}$$

當設計者決定好損失函數後，下一步是選擇解決最佳化問題的演算法。理想上，理想參數應能夠最小化損失函數：

$$\theta^* = \arg\min_\theta \sum_{n=1}^{N} L(\mathbf{x}_n, \mathbf{y}_n, \theta) \tag{2.8}$$

但因類神經網路中間含有非線性活化函數的緣故，最小化損失函數通常沒有解析解（Analytical Solution），或稱封閉解（Close-form Solution）。實務面上，通常採用迭代式（Iterative）演算法，給定一個參數起點，一步一步減少損失函數。 最簡單的迭代式最佳化演算法為統計式梯度降低（Stochastic Gradient Descent, SGD）演算法，給定某一組參數 θ ，損失函數沿著該參數上的梯度方向更新，下降的速度最快，可表達成：

$$\begin{aligned} \theta_{k+1} &\leftarrow \theta_k - \eta \Delta\theta_k \\ \Delta\theta_k &= \left.\frac{\partial L}{\partial \theta}\right|_{\theta=\theta_k} \end{aligned} \tag{2.9}$$

其中 η 為學習率（Learning rate），調控最佳化的速度與精細度， k 為更新的迭代次數，隨著 k 的增加，可期望損失函數的值越來越小，模型越接近理想模型。

在使用SGD訓練深度類神經網路的時候，通常是使用反向傳播（Back Propagation）演算法，在模型完成順向預測（Forward Prediction）計算出 $\hat{\mathbf{y}}$ 後，會先得到最後一層參數的梯度值，再根據鏈鎖律（Chain-rule），將梯度反向傳播回輸入層，取得每一層的參數的梯度值，從而完成更新。

2.2.4 類神經網路的困難

- 局部最佳解（Local Optimum）

  使用最佳化演算法訓練類神經網路的時候，沒有任何方法可以保證損失函數是下凹（Convex）的。在損失函數下降的過程中，並不保證會下降到最佳解（Global Optimum） 上，很有可能會停在局部最佳解上。因此在訓練之初通常使用隨機初始化，並且擴增模型的隨機性，以避免掉落到結果較差的局部最佳解上。

  同時梯度的計算牽扯到函數的一次微分，對於雜訊的反應較不穩健，訓練上不穩定。更進階的訓練方式包含了慣量（Momentum），使得每次SGD更新的時候，包含了前次更新的梯度方向，可表達為：

$$\Delta\theta_k = \mu\Delta\theta_{k-1} + \left.\frac{\partial L}{\partial\theta}\right|_{\theta=\theta_k} \tag{2.10}$$

  其中 μ 為慣量係數，用以調控慣量的比例，來降低卡在局部最佳解上的機率。

- 過度貼合（Overfitting）

  機器學習常會遇到過度貼合（Overfitting）的問題，亦即模型在訓練資料表現越來越好，但在測試資料中損失函數越變越差。模型為了在訓練資料中表現好，而是將所有訓練資料背誦，無法學習到真正幫助分類與回歸的本質，導致因此模型的概括化（Generization）下降。常見的避免過度貼和的方法為正規化（Regularization）跟丟棄演算法（Dropout） [25]。

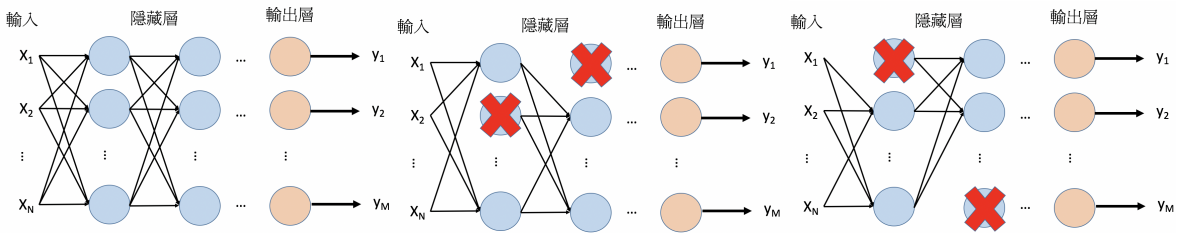
  1. 正規化的目的在於減弱模型的複雜度，也就是對於模型參數大小做限制，

即是是在原本的損失函數中，加入 $||\theta||_p$ 如2.11：

$$\min_{\theta} \sum_{n=1}^{N} L(\mathbf{x}_n, \mathbf{y}_n, \theta) + \lambda ||\theta||_p^p$$
$$||\theta||_p = (\sum_{\theta_i \in \theta} |\theta_i|^p)^{\frac{1}{p}} \tag{2.11}$$

其中 $||\theta||_p$ 稱為 L^p 範數 (L^p norm) ， λ 則是一個介於 0 到 1 之間的係數。當 $p = 1$ 時，稱為一次正規化（L1 Regularization）。而 $p = 2$ 時，為二次正規化（L2 Regularization） 。一次正規化使模型 θ 中傾向出現較多的 0 ，故常稱為稀疏解（sparse solution）。二次正規化對於模型 θ 絕對值較大的參數給予較多的懲罰，避免其中的 θ_i 值太大或太小，來減少模型的複雜度來避免避免過度貼合現象。由於類神經網路中， θ 多半是層與層之間的權重矩陣 **W** ，故二次正規化也常被稱為權重衰減 （weight decay）。

2. 丟棄演算法（dropout），在每次訓練中，模型內每個神經元會有 p 的機率直接被丟棄，使得輸出值為0，如圖2.7b和2.7c。藉由隨機丟棄，可以讓模型中的各種參數自力更生，使模型學到更概括化的能力。



(a) 未經丟棄的DNN (b) 隨機丟棄的DNN (c) 隨機丟棄的DNN

圖 2.7: 丟棄演算法

丟棄演算法同時也可以視為隨機整合（Random Ensemble）模型。整合模型（Ensemble Model）在機器學習領域已經被證明是非常强大的模型，藉由多個模型的多樣性（Diversity）各司其職，提昇模型强度。類比在丟棄演算法

中，根據不同的丟棄情況，有不同的神經元組合互相整合而成。隨機性造就了多樣性，又同時控制調適，使得模型更不容易過度貼合。

2.3 遞迴式類神經網路（Recurrent Neural Network,RNN）

2.3.1 簡介

類神經網絡已經是非常强大的模型，但對它而言，每次輸入都是獨立的，它並不記得曾經有過什麼樣的 輸入。這樣的類神經網絡對於有時序性（Sequential），且有上下文關係（Context Dependency）的輸入時，例如語音辨識（Speech Recognition）和自然語言理解（Naturl Language Understanding），便無法表現得太好。因此讓類神經網路加上記憶的能力，即為遞迴式類神經網路（Recurrent Neural Network） [26]。

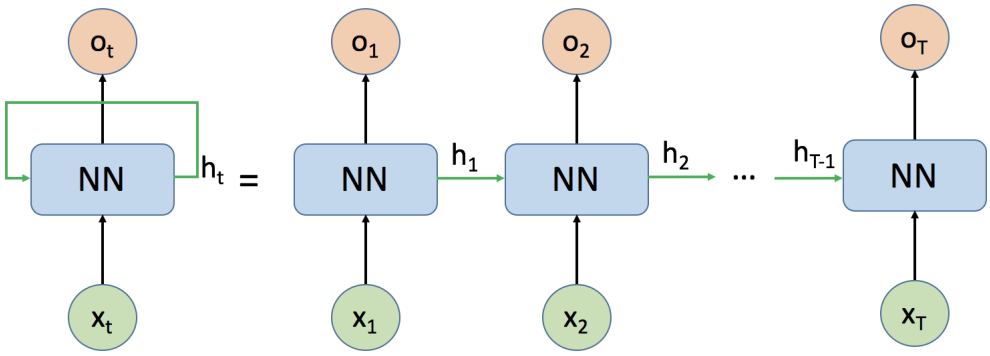


圖 2.8: 遞迴式類神經網路架構圖

2.3.2 運作原理

圖 2.8為模型架構， NN 為一組類神經網路， x_t, o_t, h_t 為時間點t的輸入、輸出跟記憶。透過記憶細胞，類神經網路能夠將之前輸入的資訊往後傳遞，並且影響後

面的輸出。如下式2.12。

$$h_t = \phi_h(W_h x_t + U_h h_{t-1} + b_h)$$
$$o_t = \phi_o(W_o h_t + b_o) \tag{2.12}$$

記憶 h_t 會跟時間 t 的輸入 x_t 與上個時間點的記憶 h_{t-1} 影響。 其中 W_h, W_o, U_h, b_o, b_h 為遞迴式類神經網路的參數， ϕ_h, ϕ_o 為活化函數。

2.3.3 沿時間反向傳播演算法

沿時間反向傳播演算法 (back propagation through time) [27]的概念跟反向傳播演算法類似，其唯一不同的是每個時間的梯度（Gradient)會沿著時間往前傳，如圖2.9紅色的梯度和綠色的梯度都會一直往前傳，更新模型的參數。實務上，須根據所設定的展開時間層數，往前傳播，且訓練後個時間點的隱藏層與隱藏層間的權重矩陣將取平均，以讓個時間點的權重相同。

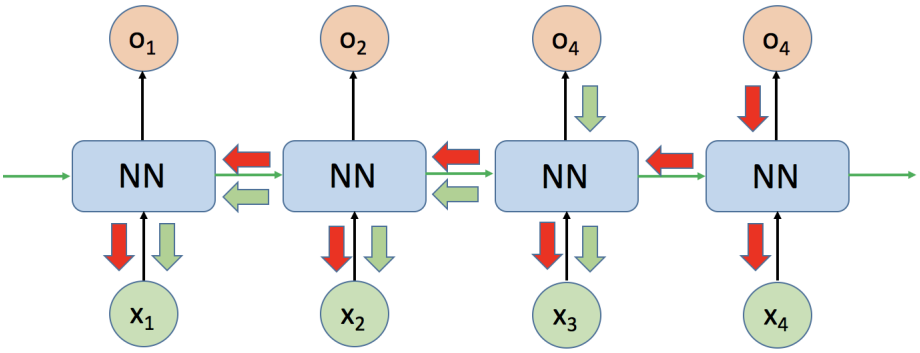


圖 2.9: 沿時間反向傳播演算法的示意圖

2.3.4 長短期記憶神經網路

隨著輸入序列的長度加長，遞迴神經網路無法記得一開始的資訊，同時跟人類一樣，我們不需要記憶每個時間點的資訊，只需要記憶關鍵的資訊。因此更近一步

提出一種進階形式：長短期記憶網路（Long Short-term Memory Network） [28]，是一種特殊的遞迴神經網路，能夠彌補之前的缺點。 長短期記憶網路，比一般遞迴神經網路來得複雜，將原本的類神經元由長短期記憶細胞（Long Short-term Memory Cell）取代，如圖2.10，在每個時間點，這組神經網路會有一個單元狀態（Cell State），並使用三個被稱作門限（Gate）的機制來輔助管理、移除、以及新增資訊。以下將會詳細說明各門限及其運作方式。

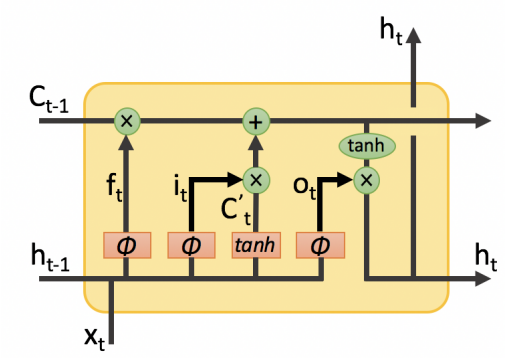


圖 2.10: 長短期記憶細胞的示意圖 [3]

- 遺忘門限（Forget Gate） f_t

  負責決定哪些資訊需要從單元狀態中丟棄。 如式子2.13跟圖2.11，依照前一個時序的輸出以及現在的輸入，產生一個值介於 0 到 1 的向量，藉由跟前一單元狀態 C_{t-1} 進行逐點乘積（Pointwise Product）時，可以決定哪些值要被捨棄多少。其中 ϕ 表S型函數、 x_t 為第 t 個時刻的輸入、 h_{t-1} 則為前一時刻的輸出。

$$f_t = \phi(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{2.13}$$

- 輸入門限（Input Gate） i_t

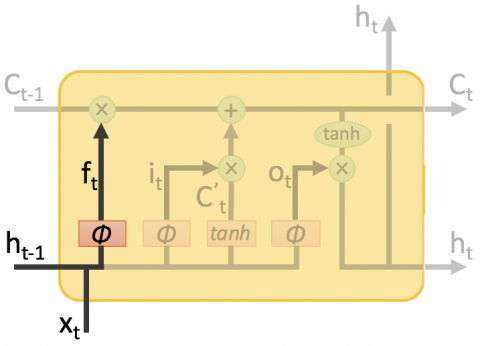


圖 2.11: 遺忘門限的運作圖 [3]

負責控制輸入訊息，假如模型認為是此時輸入為雜訊，可以將輸入門限設為0，反之設為1。另一個非線性函數雙曲正切（Hyperbolic Tangent，tanh）將負責產生此時間點儲存資訊的候選值 C'_t 。如式2.14，圖2.12所示。

$$
\begin{aligned}
i_t &= \phi(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
C'_i &= tanh(W_{xC}x_t + W_{hC}h_{t-1} + b_C)
\end{aligned}
\tag{2.14}
$$

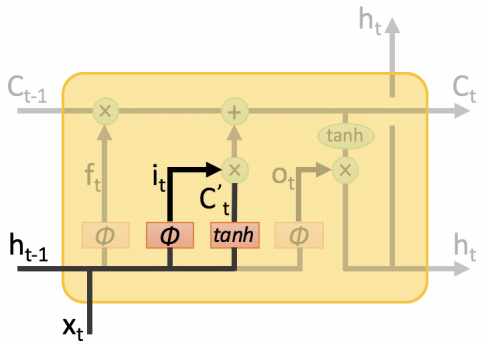


圖 2.12: 輸入門限的運作圖 [3]

- 更新單元狀態（Cell State)

基於上面幾個運算，現在可以更新單元狀態，如式2.15，圖2.13。新的單元狀態為上一個時間點的單元狀態 C_{t-1} 乘上遺忘門限 f_t 與此時的單元狀態候

選 C'_t 乘上輸入門限 i_t 相加。

$$C_t = f_t * C_{t-1} + i_t * C'_t \tag{2.15}$$

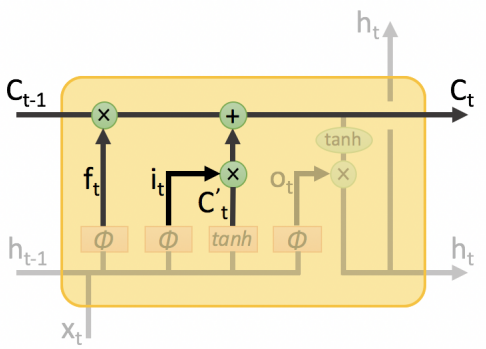


圖 2.13: 更新單元狀態的運作圖 [3]

- 輸出門限（Output Gate） o_t

  輸出門限 o_t 控制模型的輸出結果，其運作方式為與將通過雙曲正切函數的單元狀態 C_t 產生出 h_t ，進行逐點乘積，如此使模型能夠輸出想要的部分。如式2.16，圖2.14所示。

$$\begin{aligned} o_t &= \phi(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{2.16}$$

因上述的門限控制，可以使新的類神經網路，控制自己想要記憶的東西，而不是每個時間點的模型都會記憶下來，使其較久的資訊也能夠被模型記住。

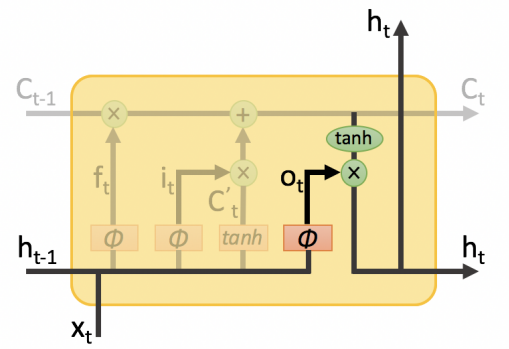


圖 2.14: 輸出門限的運作圖 [3]

2.4 本章總結

本章介紹了資訊檢索的背景，包含了基礎的資訊檢索架構、口述語彙偵測與語意檢索的差別，以及以動態時間校準做為口述語彙偵測的方法，再來介紹了類神經網路的基本原理，訓練方式及類神經網路常見的困難。最後介紹了遞迴式類神經網路，且介紹其中一種名為長短期記憶網路的運作方式。

第 三 章　以遞迴類神經網路之口語詞彙偵測

3.1　簡介

傳統的語音資訊檢索，基本上都要經過語音辨識系統，轉變成文字，在進行搜尋，然而這種做法的缺點是辨識過程中，不可避免地會出現辨識錯誤、辭典外詞彙而導致辨識結果不準確，進而影響到檢索結果。同時語音文件本身帶有的語音資訊如音高、聲調等等，經過語音辨識系統後，隨即消失了且再也找不回來。

本章中，口述詞彙偵測會轉化為一個二元分類的問題（Binary Classification）。此問題又可被稱為序列分類（Sequence Classification），因系統會將每一份語音文件跟語音查詢詞，以聲學特徵序列之型式作為輸入，最後輸出其判斷的類別。倘若運用 深層學習中的遞迴神經網路（Recurrent Neural Network），將每個聲學特徵視為一個時間點上的輸入，便能夠妥善考慮序列元素之間彼此的關係，將整串序列壓縮為一有意義的向量並最大程度地保留所有資訊。如此一來，更能進行準確的分類。本章想討論的為利用遞迴類神經網路，分別將語音文件跟查詢對象抽取它們的代表向量，再藉由類神經網路判斷查詢詞是否出現在語音文件中。希望系統能夠給予一個介於 0～1 的分數（機率值），對於查詢對象確實出現在語音文件中的情況給予高分，反之亦然。

3.2　利用遞迴式神經網路的特徵向量表示法

3.2.1　抽取聲學特徵

對於語音文件，必須先抽取相關的特徵以便進行語音檢索，梅爾倒頻譜係數

（Mel-Frequency Cepstrum Coefficient, MFCC）為目前語音辨識最常使用的聲學特徵參數，其抽取流程如圖3.1。 以下為各步驟簡略說明：

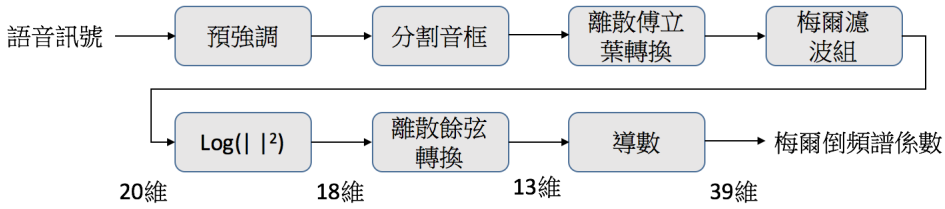


圖 3.1: 梅爾倒頻譜係數流程圖

1. 預强調

   語音訊號中高頻成分較低頻成分能量微弱許多，而語音的高頻成分蘊藏許多能辨別語音的資訊，因此將語音訊號做預强調，增加其高頻成分的比重，幫助語音訊號有較好的辨識效果。實作方法為將語音訊號通過一個高通濾波器(High-pass Filter)，以提高其高頻成分的能量。

2. 分割音框（Frame）

   語音訊號為時變訊號，其訊號特徵跟隨時間改變。為了分析語音訊號，將語音訊號分割成許多固定長度且彼此重疊的音框，並假設在音框內訊號特徵是穩定不變的。一般而言，音框的長度為20ms，而音框重疊的長度為10ms。且音框通常會乘上漢明窗 (Hamming Window) 以增加音框之間的連續性。

3. 離散傅立葉轉換 (Discrete Fourier Transform, DFT)

   在時間軸進行分析並不容易，所以將語音訊號分割成音框後，對每個音框內的訊號進行離散傅立葉轉換，將每個音框內的時軸訊號轉為以頻率軸來表示。就訊號儲存的觀點來說，儲存完整的訊號是非常耗費資源的，且完整的

訊號包含太多跟語音無關的資訊，其中只需要儲存有用的特徵即可。因此以下步驟大部份是為了壓縮訊號，提取關鍵訊號資訊為目的。

4. 梅爾濾波組 (Mel-Filter Bank)

梅爾濾波組由數組重疊的三角形濾波器所組成，如圖3.2所示，其概念是模仿人類的聽覺神經。由於人類的聽覺神經對低頻部份較敏感，因此在低頻部份濾波器分佈較密集，高頻部分則較鬆散。取通過各濾波器的訊號能量做為參數。假設梅爾濾波組由20組三角濾波器組成，則經過濾波組後，每一個音框會被壓縮成一個20維的向量。再者，因為聽覺神經對音量的敏感度與訊號能量取對數成正比，為了模擬聽覺神經，我們對經過濾波組後得到的訊號能量取對數。

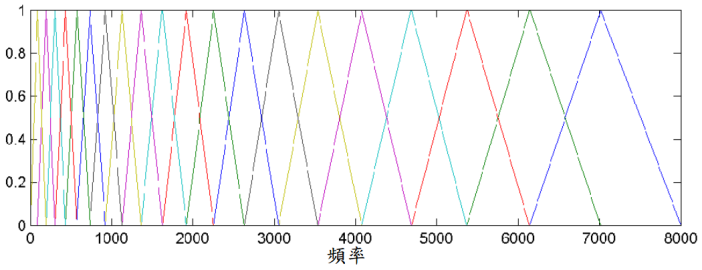


圖 3.2: 梅爾濾波組

5. 離散餘弦轉換 (Discrete Cosine Transform, DCT)

梅爾濾波組是由彼此重疊的濾波器所組成，可以想像抽出來的參數其維度及維度之間有很大的相關性(Correlation)。為了進一步達到訊號壓縮的效果，將向量透過離散餘弦轉換，投射至維度彼此正交(Orthogonal)的軸上。忽略一些對訊號影響較小的維度後，此向量從20維降至12維， 最後再將音框中總體訊號能量列入，每個音框以13維的向量來表示。由於離散餘弦訊號可以視為一種反離散傅立葉轉換 (Inverse Discrete Fourier Transform,IDFT) ，此步

驟相當於將頻率軸的訊號又轉回類似時軸的空間中，而經過梅爾濾波組及取對數等的處理後，反離散傅立葉轉換後不能稱為時軸，學者便將其稱為倒頻譜 (Cepstrum) ，即為梅爾倒頻譜係數名稱的由來。

6. 取導數 (Derivatives)

至離散餘弦轉換為止，梅爾倒頻譜係數只考慮音框內的訊號資訊，然而音框與音框之間的關係也蘊藏有許多語音的訊息。因此，我們對13維梅爾倒頻譜係數取一階及二階導數，並將其與本來的係數串接為39維的向量，其目的在於將音框之間的資訊也列入聲學特徵的考量中。

產生出39維的梅爾倒頻譜係數後，接下來，使用倒頻譜平均數與變異數正規化法（Cepstral Mean and Variance Normalization,CMVN），將每一維度的參數平均值變為 0 ，變異數變為 1 ，如此能夠抵抗雜訊的干擾。數學式如3.1：

$$X_{CMVN}[n] = \frac{X[n] - \mu_x}{\sigma_x}, n = 1, 2, ..., N$$

$$\mu_x = \frac{1}{N}\sum_{n=1}^{N} X[n], \sigma_x = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(X[n] - \mu_x)^2} \tag{3.1}$$

其中 $X[n], n = 1, 2, ..., N$ ，為音框的倒頻譜係數，經過正規化的係數為 $X_{CMVN}[n], n = 1, 2, ..., N$ ， μ_x 為音框參數的平均數， σ_x 為音框參數的標準差。

藉由上述的方法，可以將語音文件跟語音查詢詞，抽出其對應的聲學特徵。

3.2.2 序列對序列模型（Sequence-to-Sequence Model） [1]

遞迴式類神經網路因其有記憶性，每個時間點的輸出，會依據之前記憶跟現在時間點的輸入改變。所以依照其特性，可以將語音文件一一給入模型中，在最後時

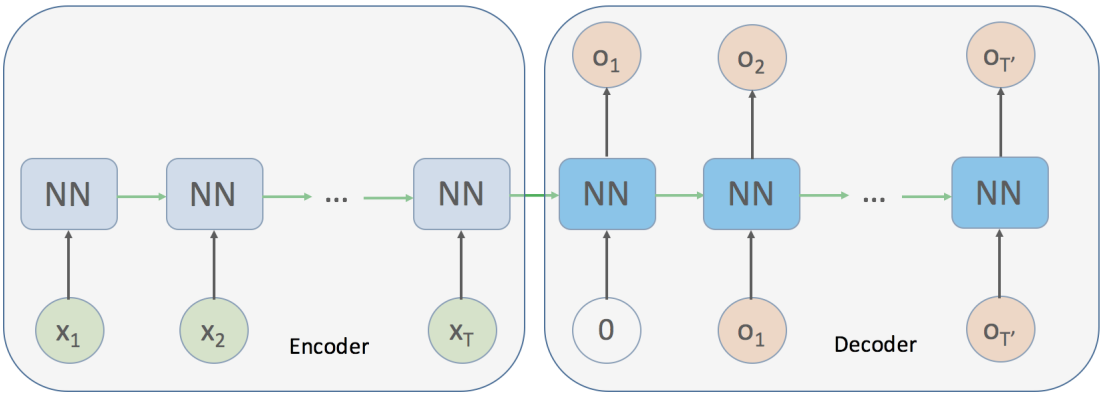


圖 3.3: 序列對序列模型

間點的輸出，可以當做模型已經看過整個語音文件，產生的語音特徵向量。此想法跟序列對序列模型的概念相同，如圖 3.3 所示。

序列對序列模型有兩個部分，分別為編碼器（Encoder）跟解碼器（Decoder），由兩個不同的遞迴類神經網路所組成，編碼器會先將輸入 X 依序讀過，跟先前的遞迴神經網路有點不同，會忽略編碼器的輸出，因為其目的是為了將輸入 X 依序看過轉變成為一個維度固定的隱藏狀態（Hidden State）。解碼器則將編碼器的隱藏狀態當做初始，在每個時間點會依據隱藏狀態跟上個時間點的輸出產生出此時的輸出，產生解碼器此時的輸出，在最一開始的輸入，解碼器的輸入為零向量（Zero Vector），如圖上的 0 。此種模型在機器翻譯（Machine Translation）和自動摘要（Summarization）裡是很常見。在機器翻譯中，會先將欲翻譯的文字先經編碼器轉換成向量，再透過解碼器產生翻譯的文字。在自動摘要中，會將整篇文章先經過編碼器變成向量，利用解碼器輸出文章的摘要內容。

藉此可以將查詢詞跟語音文件藉由編碼器將原本為一連串的序列轉變成單一個隱藏狀態，再藉由最後一個時間點的輸出來代表文件或查詢詞的特徵向量表示，如圖3.4。

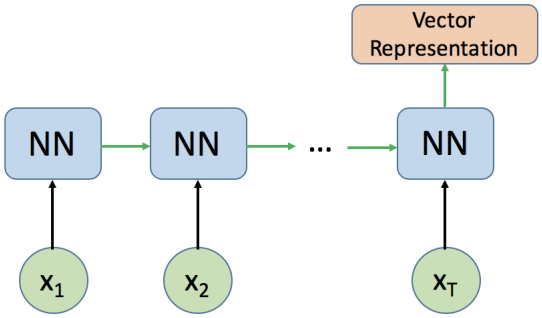


圖 3.4: 遞迴類神經網路之向量表示

3.3 系統架構

3.3.1 系統概觀

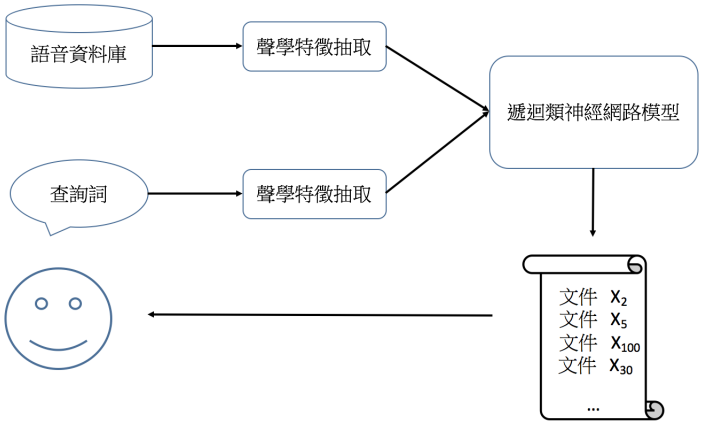


圖 3.5: 系統概念圖

本章節所提出的口語詞彙架構如圖3.5所示。主要為兩個部分，第一部分為特徵抽取，將查詢詞跟語音文件利用前述的方式進行聲學特徵抽取，抽取出 39 維梅爾倒頻譜係數。第二部分為遞迴類神經網路模型，將聲學特徵給入模型，最後模型會給予每個文件分數，來判定文件中是否出現查詢詞，此模型將在3.3.2做介紹。

3.3.2 遞迴類神經網路模型

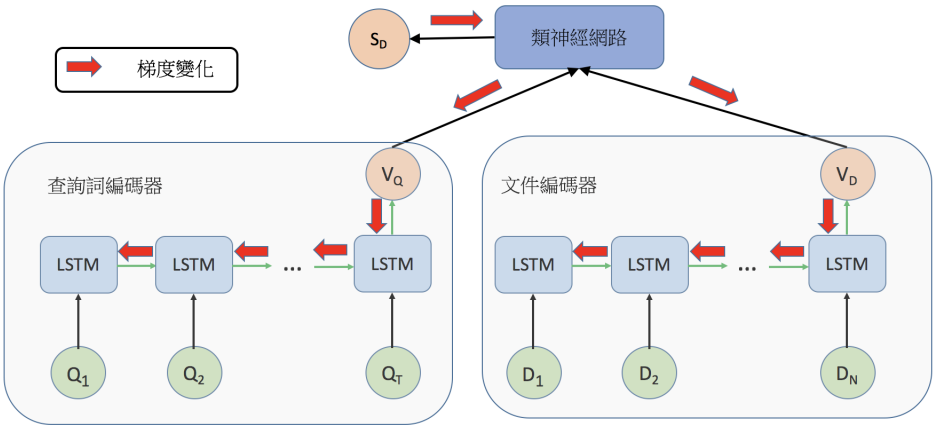


圖 3.6: 遞迴類神經網路模型圖

圖3.6為整個模型的架構圖，分為兩個部分，一個為遞迴類神經網路的編碼器，一個為類神經網路的分類器。編碼器負責將查詢詞跟語音文件分別由藉由它們的聲學特徵編碼成代表的向量特徵，類神經網路藉由向量特徵給予此文件一個分數。編碼器為前述2.3.4的長短期記憶網路(LSTM)，且圖上的兩個編碼器參數是相同的。因此時口述詞彙偵測問題被視為分類問題，模型最後的輸出為兩維，一維代表查詢詞出現在文件當中的分數，一維則是未出現的分數，最後會在經過正規化，使兩維分數相加為 1 。

3.3.3 訓練方式

訓練類神經網路，需要先定義出損失函數。損失函數使用先前2.2.3章提到交叉熵，來進行訓練。簡化的交叉熵為式子3.2

$$L_{CE}(\mathbf{x}, \mathbf{y}, \theta) = KL(\mathbf{y}||\hat{\mathbf{y}}) = -\log \hat{y}_l \tag{3.2}$$

其中KL為克雷散度， $\hat{y}_l$ 為模型給正確標籤的分數。在最小化損失函數的同時，也就將正確標籤的分數提高，使模型能夠分類越準確。整個模型的訓練採用端對端

訓練（End-to-End Training），亦即不需要獨立訓練編碼器的部分，直接由輸出端的梯度變化，向後傳遞藉此訓練編碼器，如圖3.6紅色箭頭所表示，則不須費心在編碼器產生的向量品質好壞，完全交由模型自動訓練跟判斷。

3.4 實驗結果與分析

3.4.1 實驗設定

實驗語料採用LIBRISPEECH [29]的英文語料，這是使用LibriVox的應用程式介面（LibriVox's API)收集參加讀者的閱讀訊息、音頻及閱讀書籍的章節。語料庫的內容大小不同，利用華爾街日報（The Wall Street Journal, WSJ） [30]語料庫裡的Si-84當作訓練語料，訓練出語音辨識模型，根據此模型測試的詞錯誤率（Word Error Rate,WER）大約分成三個子集合，分別為100小時、360小時與500小時。前兩組語料為詞錯誤率較小的集合，故以clean稱之，其中每位講者時間限制為25分鐘以避免權重不平衡，此兩個語料集合口音較接近美式英語，而最後500小時為詞錯誤率較高則以other稱之。表3.1提供了Librispeech的集合資訊。

表 3.1: Librispeech 集合列表

集合	時間（hr）	講者時數（min)	女性講者人數	男性講者人數	總講者
train-clean-100	100.6	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1166

本篇論文將train-clean-360 當作訓練語料，將train-other-500 當作測試語料。利用TF-IDF（Term Frequency–inverse Document Frequency）分別對訓練和測試語

料做排序，訓練語料中選出500個查詢詞，109,220筆的訓練資料，測試語料30個查詢詞，2,000筆測試資料。且限制語音文件的長度為15秒，語音查詢詞的長度為2秒。

3.4.2 基準實驗

本章所使用的基準實驗為前述2.1.3章所提到的片段式動態時間校準，利用動態時間校準可以計算出查詢詞聲學特徵跟語音文件聲學特徵的最相關的距離。依照此距離來將文件進行排序，計算平均準確率。基準實驗在測試語料上獲得的平均準確度為 0.6173 。

3.4.3 實驗結果與分析

表3.2為本章的實驗結果，比較了各模型與基準實驗的表現，使用平均準確率來做衡量標準。在本章的編碼器採用了兩種遞迴類神經細胞，一種為2.3.4章介紹的長短期記憶細胞（LSTM），另一種記憶細胞為門閘遞迴單元（Gated Recurrent Unit, GRU） [31]，是長短期記憶細胞的簡化版。輸入門限和遺忘門限是連動的，當遺忘門限關閉清除儲存內容時，輸入門限才會開啓儲存新的資料。編碼器的記憶細胞數量為128，也就是語音文件跟查詢詞經由編碼器產生出128維度的向量表示。類神經網路分類器的部分採用了各種結構如表3.2上所示，以128-64-32-2 來說，類神經網路的架構即為第一層128個類神經元，第二層64個，第三層32個，最後一層為2個。模型的學習率皆為0.001，並使用二次正規化來避免模型過度貼合。

從表中可以看出類神經網路128-2的表現，相比於其他128-128-128-2、128-64-32-2結構，少了2%的平均準確率，因深層的模型可以有效提高分類正確率。對於不同類型的遞迴神經網路GRU跟LSTM的表現是差不多的，且模型的層數也對平

表 3.2: 遞迴類神經網路實驗結果

模型架構		平均準確率
基準實驗		0.6173
編碼器	類神經網路結構	平均準確率
兩層LSTM	128-2	0.5753
	128-64-32-2	0.5935
	128-128-128-2	0.6076
三層LSTM	128-64-32-2	0.5950
	128-128-128-2	0.6025
兩層GRU	128-64-32-2	0.6080
	128-128-128-2	0.6070
三層GRU	128-64-32-2	0.6010
	128-128-128-2	0.5965

均準確率影響不大。綜觀所有模型，平均準確率大約落在0.6，仍無法贏過基準實驗的0.6173。因模型僅僅簡單的將語音文件轉化成一個向量，可能無法充分表示此語音文件，語音文件不只有一個詞而是一整段句子，語音文件向量會遺失掉某些詞的資訊。即使分類器在强大，仍無法準確檢索出來。

3.5 本章總結

本章使用了遞迴類神經網路來產生出語音文件跟查詢詞的單一特徵向量表示，由於遞迴類神經網路有著序列順序前後關係及空間的不變性，可以有效的將序列的語音文件，描述成單一向量表示。產生代表語音文件跟查詢詞的向量表示後，再藉由多層類神經網路分類器依據向量表示來判斷查詢詞出現與否。本章提出了一個基本的檢索模型，無須經過語音辨識系統，能夠直接在聲學特徵上進行檢索。雖其表現都輸給基準實驗一小段差距，不過平均準確率也大致能夠達到0.6，顯示出利用遞迴類神經網路做檢索是可行的，並非完全失敗，之後將以此模型為雛形進行改良。

第 四 章　專注式類神經網路之口述詞彙偵測

4.1　簡介

在第3章中，嘗試了直接將語音文件跟語音查詢詞編碼成向量，進行口述詞彙偵測。雖然表現不如非監督式的動態時間規劃，但至少實作出可以捨棄語音辨識系統的口述詞彙偵測。第3章中，直接將語音查詢詞變成一個向量是很合理的，因為語音查詢詞只包含完整的查詢詞，並無多餘的部分，但語音文件為一整個段落，包含了許多並非查詢詞的資訊，導致產生出來的語音文件向量喪失了查詢詞的特徵。即使後面分類器在複雜，但因語音文件向量中喪失了查詢詞特徵，所以無法準確的分辨出來。本章將專注式機制引入，專注式機制可以使模型專注在某個地方，將多餘的資訊濾除。藉由專注式機制使模型能夠關注在查詢詞的部分，可以有效保留著查詢詞特徵在產生出來的語音文件向量中，則不會分心在其他非查詢詞的部分。所以本章藉由專注式機制產生出較好的語音文件向量，可以保留著查詢詞的部分，不會被其他多餘的詞彙影響，以提升口述詞彙偵測的效能。

4.2　專注式機制

專注式機制，在近年漸漸受到大家重視，專注式機制最早被應用於機器翻譯的領域上，該系統為先前3.2.2章提到的編碼器-解碼器遞迴神經網路，能提供端對端機器翻譯。而該系統的特點為能夠根據前一個時刻的輸出，去側重當今時刻的資料中重要的部分；如此這種根據狀況進行「挑選」的技能是先前的其他類神經網路所無法比擬的。 現已出現一些模擬專注力的神經網絡架構，如記憶網絡（Memory Network） [32]、類神經圖靈機（Neural Turing Machine） [33]、

動態記憶網路（Dynamic Memory Network） [34]等等，這類模型統稱專注式模型（Attention-based Model）。

專注式模型常應用在問答（Question Answering,QA）系統上，讓機器可以針對提出的問題，回答正確的答案。當使用者輸入問題時，專注式模型可以幫助機器從資料庫中選擇相關資料，而將專注力放在相關的資料上，避免其他無關資訊干擾，藉此得到問題的答案。問答系統甚至已經從文字擴展到多媒體，機器可以依據圖片回答相關問題（如：圖中的人穿什麼顏色的衣服） [35]，而專注式機制可以幫助模型將重心放在圖片中人的衣服上，來判斷衣服的顏色，不會受到圖片中其他物件影響。不只如此，專注式模型也被應用在自動新聞摘要 [36]、自動產生圖片／影像說明 [37]、語音辨識 [38] 與機器自動翻譯等任務上 [39]。

受到這些啓發，將專注式機制應用於口述詞彙偵測。在第3章中，編碼器最主要的難處是當輸入序列相當長時，可能會有比較多雜訊在其中、或者許多部分是與整體資訊無關的，這些或多或少都會影響編碼出來的語音文件向量，進而影響分類器判斷結果的正確率。因此我們希望透過專注式機制的特性，建立在原本長短期記憶神經網路就能處理序列問題的基礎上，再自動從序列中挑出重要的部分並且忽略其餘跟查詢詞無關的部分，使其語音文件向量能夠包含著查詢詞的資訊，使平均準確率能夠提升。

4.3 模型架構

4.3.1 系統架構簡介

圖4.1為本章節之系統架構圖。系統輸入部分包含了語音查詢詞跟語音文件，分別會經由聲學特徵抽取，轉變為39維的梅爾倒頻譜係數。首先，語音查詢詞的聲學

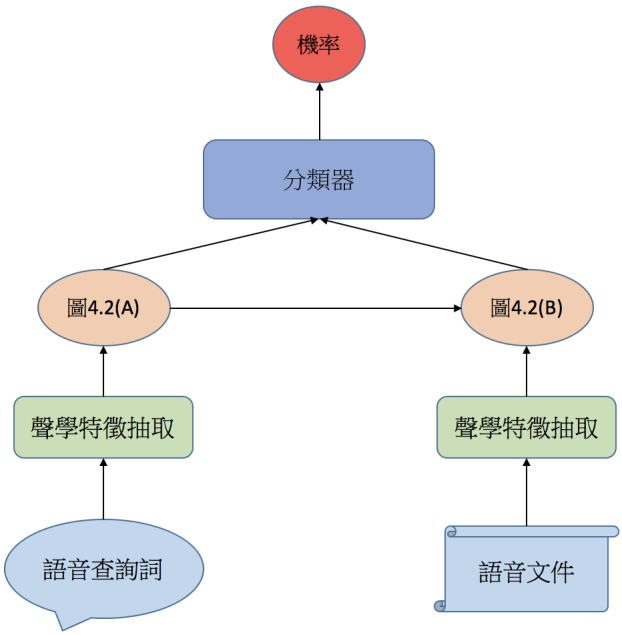


圖 4.1: 模型架構圖

特徵序列如同第3章一樣被壓縮編碼並表示成為一向量，此向量稱之 V_Q ，此部分會在4.3.2章在做介紹。產生出了 V_Q ，於4.3.3章中將使用專注式機制找出跟語音查詢詞有關的部分，產生出較好的語音文件向量 V_S 。最後，將 V_Q 跟 V_S 藉由分類器去預測查詢詞出現在文件中的機率，將在4.3.4章中作介紹。

4.3.2 語音查詢詞之向量表示法

圖4.2（A）為抽取 V_Q 向量的介紹，將一連串的聲學特徵序列轉換為 V_Q 的過程。輸入的查詢詞為 一長度T的序列，$C_1, C_2, ..., C_T$，其中 C_i 均為39維梅爾倒頻譜係數。使用雙向式長短期記憶網路（Bidirectional Long Short-term Memory Network）作為編碼器的模型；其在輸入語音序列時，一個時間點只會讀取一個音框進去。在圖4.2(A)中的第t個時間點時，正向（forward）長短期記憶網路的隱藏層輸出表示為 y_t^f 、而反向（backward）長短期記憶網路之隱藏層輸出表示為 y_t^b 。在讀入所有的輸入序列之後，可在正向長短期記憶網路的最後一個時間點獲得一組隱藏

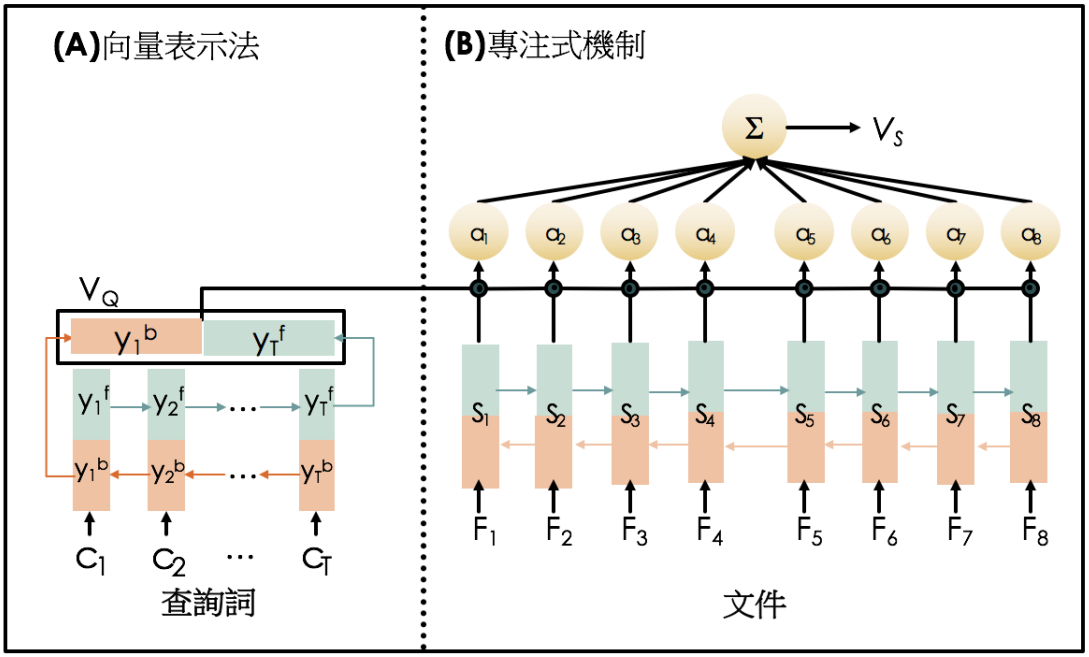


圖 4.2: 專注式機制流程圖

層輸出向量 y_T^f 、在反向長短期記憶網路的第一個時間點獲得 y_1^b ，並把他們串聯起來作為查詢詞的向量表示法 V_Q ， $V_Q = [y_T^b || y_1^f]$ 。

4.3.3 專注式機制與語音文件表示法

在圖4.2（A）中獲得查詢詞的向量表示法 V_Q 之後，之後將利用畫重點機制配合遞迴神經網路來對於語音文件的序列進行編碼，得到向量 V_S ，如圖4.2（B）所示，語音文件的內容其實是一相當長的聲學特徵序列，但圖中我們簡化為八個音框，我們同樣使用雙向式長短期記憶網路來作為編碼的工具，其中在第t個時間點時，輸入詞彙的向量表示 S_t 為正向長短期記憶網路與反向長短期記憶網路隱藏層輸出的串聯，$S_t = [y_t^f || y_t^b]$。 接著我們引入專注式機制，來衡量查詢詞向量 V_Q 與語音文件內音框的關聯性高低，第 t 個時間點的語音文件向量 S_t 之專注式權重 α_t 為 V_Q 與 S_t 的相關程度，$\alpha_t = S_t \odot V_Q$，其中⊙表示兩向量之間的相關程度運算。

相關程度的運算自己定義的，可以是簡單的歐式距離（Euclidean Distance）、餘弦相似性（Cosine Similarity），也可使用類神經網路來計算，將兩向量丟入類神經網路去計算相關程度。 在本章中採用餘弦相似性當作相關程度，如式子4.1所示:

$$\alpha_t = S_t \odot V_Q = \frac{S_t \cdot V_Q}{|S_t||V_Q|} \tag{4.1}$$

接下來，對於所有的專注式權重 α_t 進行正規化，成為 $\hat{\alpha}_t$ ，正規化的方式有兩種。

- 尖銳（Sharpening）正規化

  我們將專注式權重透過軟式最大化（Softmax）之活化函數進行正規化：

$$\hat{\alpha}_t = \frac{exp(\alpha_t)}{\sum_{t=1}^{T} exp(\alpha_t)} \tag{4.2}$$

  由於可以確實降低資料的雜訊，此方式已被現存其餘專注式機制的相關研究所廣泛使用。

- 平滑（Smoothing）正規化

  尖銳正規化偏好於注意一個點，因此一整個專注式權重集 $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_T)$ 可能只會有某個向量 α_t 的權重特別高，其餘資訊皆被捨去。這樣的特性有可能降低口述詞彙偵測的表現。因此另外設計了一種正規化表示法，能夠考慮更多重要的點，此種方式也讓模型的多樣性因而增加。

  我們將原本算式中的指數函數改成S型函數（Sigmoid Function）σ:

$$\hat{\alpha}_t = \frac{\sigma(\alpha_t)}{\sum_{t=1}^{T} \sigma(\alpha_t)} \tag{4.3}$$

最後對於所有的語音文件向量 S_t 進行加權平均，使用的權重便為正規化後的專注式權重 $\hat{\alpha}_t$ ，得到代表語音文件的向量 V_S，$V_S = \sum_t \hat{\alpha}_t S_t$。

4.3.4 分類器

最後模型的輸出分數由分類器來決定的。分類器有很多種模型，常見的有支撐向量機（Support Vector Machine）、類神經網路、貝氏分類器（Bayes classifier）、決策樹（Decision Tree）等等。在此章中，以多層類神經網路當作分類器，另外還使用語音文件向量 V_S 跟語音查詢詞向量 V_Q 的餘弦相似度當作輸出分數為兩種產生分數的方式。

4.4 實驗與分析

4.5 本章總結

第 五 章　非監督式學習語音向量之口語詞彙偵測

5.1 簡介

5.2 語音向量

5.3 模型架構

5.4 實驗與分析

5.5 本章總結

參 考 文 獻

[1] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[2] Yaodong Zhang and James R Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 398–403.

[3] 沈昇勳, "藉助線上課程之自動結構化、分類與理解以提升學習效率," 2016.

[4] Ciprian Chelba, Timothy J Hazen, and Murat Saraçlar, "Retrieval and browsing of spoken content," *Signal Processing Magazine, IEEE*, vol. 25, no. 3, pp. 39–49, 2008.

[5] Lin-shan Lee and Berlin Chen, "Spoken document understanding and organization," *Signal Processing Magazine, IEEE*, vol. 22, no. 5, pp. 42–60, 2005.

[6] "Text retrieval conference," Website, `http://trec.nist.gov/`.

[7] Murat Saraclar and Richard Sproat, "Lattice-based search for spoken utterance retrieval," *Urbana*, vol. 51, pp. 61801, 2004.

[8] Jonathan Mamou, David Carmel, and Ron Hoory, "Spoken document retrieval from call-center conversations," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 51–58.

[9] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[10] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky, "A hierarchical neural autoencoder for paragraphs and documents," *arXiv preprint arXiv:1506.01057*, 2015.

[11] Pierre Baldi, "Autoencoders, unsupervised learning, and deep architectures.," *ICML unsupervised and transfer learning*, vol. 27, no. 37-50, pp. 1, 2012.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[14] Jeffrey Pennington, Richard Socher, and Christopher D Manning, "Glove: Global vectors for word representation.," in *EMNLP*, 2014, vol. 14, pp. 1532–1543.

[15] Yu-An Chung, Chao-Chung Wu, Chia-Hao Shen, Hung-Yi Lee, and Lin-Shan Lee, "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," *arXiv preprint arXiv:1603.00982*, 2016.

[16] Stephen E Robertson, "The probability ranking principle in ir," .

[17] Ian Ruthven and Mounia Lalmas, "A survey on the use of relevance feedback for information access systems," *The Knowledge Engineering Review*, vol. 18, no. 02, pp. 95–145, 2003.

[18] Ellen M Voorhees, "Query expansion using lexical-semantic relations," in *SIGIR'94*. Springer, 1994, pp. 61–69.

[19] Jinxi Xu and W Bruce Croft, "Query expansion using local and global document analysis," in *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1996, pp. 4–11.

[20] Chun-an Chan and Lin-shan Lee, "Unsupervised spoken-term detection with spoken queries using segment-based dynamic time warping.," in *INTERSPEECH*, 2010, pp. 693–696.

[21] Timothy J Hazen, Wade Shen, and Christopher White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 421–426.

[22] John S Garofolo, Cedric GP Auzanne, and Ellen M Voorhees, "The trec spoken document retrieval track: A success story.," .

[23] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[24] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, pp. 1.

[25] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] Jeffrey L Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[27] Paul J Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[28] Felix Gers, *Long short-term memory in recurrent neural networks*, Ph.D. thesis, Universität Hannover, 2001.

[29] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[30] Douglas B Paul and Janet M Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[31] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[32] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al., "End-to-end memory networks," in *Advances in neural information processing systems*, 2015, pp. 2440–2448.

[33] Alex Graves, Greg Wayne, and Ivo Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[34] Ankit Kumar and Ozan Irsoy, "Ask me anything: Dynamic memory networks for natural language processing," .

[35] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C Lawrence Zitnick, Devi Parikh, and Dhruv Batra, "Vqa: Visual question answering," *International Journal of Computer Vision*, pp. 1–28.

[36] Alexander M Rush, Sumit Chopra, and Jason Weston, "A neural attention model for abstractive sentence summarization," *arXiv preprint arXiv:1509.00685*, 2015.

[37] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[38] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[39] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.