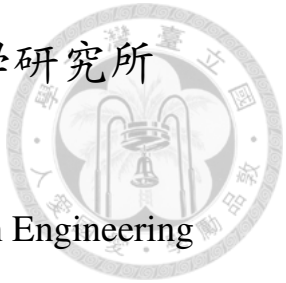


國立臺灣大學電機資訊學院資訊工程學研究所

碩士論文

Graduate Institute of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

分段式語音詞向量：將語句信號自動表示爲語音詞向量序列

Segmental Audio Word2Vec: Representing Utterances as Sequences of Audio Word Vectors

王育軒

Yu-Hsuan Wang

指導教授：李琳山 教授

Advisor: Lin-shan Lee, Ph.D.

中華民國107年1月

January, 2018

doi:10.6342/NTU201801733

誌謝

這致謝寫在口試結束半年後，飛往美國念書的三個禮拜前。在語音實驗室待了將近三年，轉眼間也準備邁入下一個人生階段。

首先最感謝的人當然就是我的指導教授李琳山老師。老師不只在研究所，在我大學時就已經對我十分照顧。當初在大三下決定要出國唸書時，首先就要決定大四專題要找哪位教授。此時腦海中想到當初大三上修語音處理概論時，那位授課教授好像蠻喜歡我寫的報告的，甚至還請我們某些修課學生吃過飯，可能代表我們調性很相符，於是就決定跟著老師做專題。之後大四請老師寫推薦信時十分戰戰兢兢，因為聽說老師的推薦信不好拿，就我所知當時身邊就至少兩位同學請老師寫推薦信被拒絕，沒想到老師爽快地答應幫我寫推薦信，讓我感念至今。然而當時由於成績不理想，申請結果即使有老師的推薦信也是神仙難救。由於不服輸，所以想要再全力拼搏一次，因此厚顏地請老師收我做研究生，讓我用研究所的經歷說服國外一流大學收我作學生。老師也爽快答應，就這樣促成了這段語音實驗室的緣分。這三年來，老師從來不用嘴巴說他心中對研究生的要求；相反地，老師用身教將一個從事研究人員該有的精神表露無遺：大家都吃完晚餐準備回家時，老師仍然在辦公室處理公務；深夜時間仍然批改著學生準備投稿的論文；無視週末假日，照常到學校辦公。看不明白的人可能覺得老師帶研究生的方法相對其他教授而言較為寬鬆因此鬆懈，然而看明白的人則會對自己時時警惕，不敢有任何懈怠。這段時間透過投稿論文和論文口試，跟老師學習到了許多關於研究和呈獻成果的方法。這些都是老師濃縮了數十年功力而成的精華，在未來行走江湖的路上必定將這些心法牢記在心。

謝謝李宏毅教授宏毅哥，在研究所這段時間在許多地方都麻煩宏毅哥照顧。只要有問題找宏毅哥幫忙，永遠都會獲得一個義不容辭相挺的答案。另外研究上

doi:10.6342/NTU201801733

卡住的時候請教宏毅哥也都能學到一些奇招將問題迎刃而解。也由於宏毅哥默默的到處賣臉，讓語音實驗室能有許多資源可以擴增戰艦、增加運算資源等等。雖然宏毅哥因為過於强大所以常常到處開會，超級霹靂忙，但是只要有需要向宏毅哥請教的時候，宏毅哥永遠都願意為我們從已經被榨乾的時間中再擠出時間和我們討論問題，語音實驗室的研究產能如此强大宏毅哥是最大功臣！

謝謝陳縕儂教授儂大，儂大强勢回歸臺大教書成為我申請留學的最後一塊拼圖，讓我成功一圓留學夢。申請時只要有問題都會厚臉皮的巴著儂大不放問問題，根本把儂大當成留學顧問。在CMU和JHU間搖擺不定時，儂大以過來人的經驗給了第一手的情報。而我去CMU參加Open House時，儂大所建立的好名聲給了我一個跟那邊教授聊天時很好的起頭，進而相談甚歡。往後在CMU唸書時期許自己也能成為和儂大一樣前人種樹後人乘涼，讓往後來CMU的學弟妹輕鬆給教授好印象。

謝謝實驗室助理彤恩姐，幫我這雷人解決了各式各樣千奇百怪的問題。在尚未入學時需要用臨時工的身份領取津貼，每次都需要勞煩彤恩姐計算各項計畫經費，用最適合的方式給我，給彤恩姐添了不少麻煩。另外出國開會科技部補助有問題時也幫我打電話詢問出了什麼事情。彤恩姊總是幫實驗室衆處理各種疑難雜症，讓大家可以專注在自己的研究和課業上，根本苦海明燈。

謝謝帶我專題的學長，鍾承道學長。很幸運能有博班學長帶我，給了我很多研究上的insights。想當初非督導式學習實驗室只有學長在做，現在大家都在做，想來學長的眼光果然十分準確。發表第一篇論文時要不是學長在精神上的各種支持，我可能撐不到最後。謝謝我B99的同學呂相弘，從我一進實驗室就十分照顧我，帶著我融入實驗室的環境，讓我沒有所謂的適應期。也為了促進實驗室感情揪了各種活動，這些活動確實讓實驗室的大家感情熱絡。謝謝廖宜修，從肌肉到

doi:10.6342/NTU201801733

研究都十分紮實的男人，讓我了解到什麼叫實力紮實的研究生。在心中默默把宜修哥當偶像，提醒自己世界上有人就是這麼硬派，所以也不能對自己放水。

謝謝我的實驗室同學：小豪、俊哥、永哲、家翔、Poyu、YD、資偉、賢進、朗祺。好在有你們一起在實驗室討論想法、閒聊打屁，不然研究所的生活實在難以想像的可怕。謝謝實驗室的碩二8+1衆：Roy、水靜、致緯、邦齊、家宏、瓊之、球哥、佩宏(Best Pei)＋舜博沒有排擠我讓我變邊緣人，在許多地方十分照顧小弟我，不勝感激。

我相信在語音實驗室的這三年帶給我人生十分深遠的影響：如果我未來還待在學術界的話自然不用說，從一篇論文從點子的發想，到實驗的設計與實作到最後的論文寫作我都有了深刻的體會，這些年在大師們的身邊邊看邊學也算是學到了一點點的皮毛。另一方面，若未來選擇待在業界，研究所訓練出的查資料、思考問題解決問題的能力在工作上也是不可或缺。總而言之，如同老師當初在信中跟我說的，語音實驗室確實給了我的人生更高的高度，這也是我將會一直十分感激的一件事。

doi:10.6342/NTU201801733

摘要

在自然語言處理中，詞向量（Word2Vec）可以用於將一個詞表示為一個一定維數（Dimensionality）的實數向量並帶有語意資訊（語意接近的詞在向量空間中會接近），這些向量所帶的語意並在向量空間上具有向量運算的可平移特性。另一方面，語音詞向量（Audio Word2Vec）則能使用一定維數的實數向量表示語音詞（一個詞的語音訊號，Spoken Word），並帶有音素結構的資訊。前人所提出的語音詞向量雖然可以在非督導式學習的框架下訓練，然而訓練語料之音訊需要事先標註好詞邊界。

在本論文中，我們將語音詞向量由語音詞的層級提升至整句語句的層級。在本論文所提出的模型中，同時針對語音詞切割與語音詞向量訓練進行訓練，讓此兩者能夠相互增强。藉由引入一切割門限至序列對序列自動編碼器，本論文提出全新的分段式序列對序列自動編碼器（Segmental Sequence-to-Sequence Autoencoder, SSAE），並用深層强化學習（Deep Reinforcement Learning）加以訓練。藉由此一方法，一語句能夠被自動切割為一系列的語音詞，再轉化為一系列之語音詞向量。本論文之實驗使用詞切割與口述語彙偵測來探討所提出的分段式序列對序列自動編碼器之效能，並在四種語言上（英文、捷克文，法文與德文）進行實驗，實驗結果顯示此模型具有比以往方法更佳的效能。

除了分段式序列對序列自動編碼器外，本論文亦分析一種遞迴式類神經網路內部之訊號：門限激發訊號；並發現此訊號在非督導式學習框架下與輸入音訊中語音特性之邊界（如音素邊界）具有强烈關聯，因此可以廣泛應用於所有非督導式學習下的遞迴式類神經網路模型中。

doi:10.6342/NTU201801733

Contents

doi:10.6342/NTU201801733

doi:10.6342/NTU201801733

圖目錄

doi:10.6342/NTU201801733

doi:10.6342/NTU201801733

doi:10.6342/NTU201801733

doi:10.6342/NTU201801733

表目錄

doi:10.6342/NTU201801733

doi:10.6342/NTU201801733

符號	意涵	符號	意涵
$\mathbf{x}$	模型之輸入資料	$[\bullet]^T$	轉置矩陣運算
$exp(\bullet)$	自然指數函數運算	$\mathbf{W}$	神經元權重矩陣
$\mathbf{b}$	神經元閾值偏移	σ	激發函數
$\boldsymbol{\alpha}$	神經元輸出	$\mathbf{y}$	針對輸入資料所標註之答案
$\hat{\mathbf{y}}$	模型所輸出之答案	Θ	模型參數集
$\mathbf{J}$	向量維度大小	$\|\| \bullet \|\|^2$	方均運算
ζ	邏輯子	η	學習率
ξ	慣量係數	$\mathbf{c}_t$	遞迴式類神經網路於時間點t時之隱藏狀態
$\mathbf{h}_t$	遞迴式類神經網路於時間點t時之輸出	$\mathbf{U}$	遞迴式類神經網路隱藏狀態之權重
$\hat{\mathbf{x}}$	自動編碼器所重建之輸入資料	$\mathbf{e}$	自動編碼器用於重建輸入資料之編碼
D	語音文件	Q	語音查詢指令
$\mathbf{s}$	在強化學習中環境之狀態	$\mathbf{a}$	強化學習中代理人所執行的動作
π	強化學習中之策略	r	強化學習中之獎勵
$\mathbb{E}[\bullet]$	期望值運算	∇	梯度運算
V	辭典大小	p	機率大小
γ	獎勵折扣係數	$\mathbf{f}_t$	時間點t時之忘卻門限激發訊號
$\mathbf{i}_t$	時間點t時之輸入門限激發訊號	$\mathbf{o}_t$	時間點t時之輸出門限激發訊號
$\tilde{\mathbf{c}}_t$	時間點t時之候選隱藏狀態	$\mathbf{z}_t$	時間點t時之更新門限激發訊號
$\mathbf{r}_t$	時間點t時之重設門限激發訊號	$tanh(\bullet)$	雙曲正切函數運算
$\odot$	以單元為單位之矩陣相乘	$\mathbf{g}_t$	時間點t時之一特定門限之門限激發訊號
δ	閾值	E_t	遞迴式預測模型在時間點t時的錯誤訊號
w	線性內插之權重	λ	減損函數中需要調整之超參數
ψ_t	切割門限在時間點t時的輸出	μ	高斯函數中之平均值
v	高斯函數中之變異數		

表 1: 符號對照表

doi:10.6342/NTU201801733

第一章　導論

1.1　研究背景及研究動機

伴隨著網路時代的來臨以及智慧電子產品的普及，全世界每分每秒都有巨量的資料產生並且發佈在網路上向全世界的人展示，如影音社群YouTube中的影音，社群網路Facebook上的照片及貼文等等。任何人只要連上網際網路都能夠輕易地獲得這些資料，也因此有許多人將現在稱作為巨量資料（Big Data）的時代。這些大量的資料為科技提供了進步的能量，因此近年來人們不管在是語音處理，自然語言處理或是影像辨識上皆有了突破性的發展。

另一方面，曾經沈寂一時的類神經網路（Neural Networks）在近年重新嶄露頭角，於這些突破性發展中扮演著不可或缺的角色。在過去，使用多層類神網路模型進行訓練十分曠日費。然而由於硬體設備的進步，讓模型訓練時間大幅縮短。又因多層的類神經網路模型的效能顯著優異於淺層的模型，因此把模型疊「深」成為這波類神經網路發展的主要共識，因此這波類神經網路革命也稱為深層學習（Deep Learning）。

伴隨著因深層不斷加深而模型複雜度不斷上升的模型，深層學習模型的訓練需要大量的資料。巨量資料時代的來臨恰好提供了讓深層學習大放異彩的能量，也因此深層學習發展出了各式各樣的架構：針對抽取圖像特徵的卷積類神經網路（Convolutional Neural Networks） [4]、能夠獲取時間資訊的遞迴式類神經網路（Recurrent Neural Networks, RNN） [5] 等等。其中遞迴式類神經網路與語音科技密切相關。對於語音的應用而言，時間資訊十分重要，每個時間點的資訊皆彼此息息相關。遞迴式類神經網路藉由隱藏狀態（Hidden State）的傳遞來捕捉時間點間彼此的關係，也因此遞迴式類神經網路成為了處理語音資料時最常被使用的一

doi:10.6342/NTU201801733

種類神經網路 [6]。而在遞迴式類神經網路的架構上，又發展出了各式各樣的神經網路。其中最著名的為加入了門限（Gate）的機制，使表現更加突出的長短期記憶神經網路（Long Short-term Memory Networks, LSTM） [7]。

目前成效最顯著的深層學習模型是經由督導式學習（Supervised Learning）而得 [8] [9]。所謂的督導式學習，是給予模型經過標註（Annotation）的資料並以其訓練之，令訓練模型使其在測試資料上也能判斷出正確的答案。以語音辨識為例，會給予模型一段語音和這此段語音經由專家標註而得的音素轉寫（Phoneme Transcription），訓練模型由聲音判斷出對應的音素轉寫。督導式學習模型雖然表現卓越，但其致命傷是標註訓練資料的成本高昂；縱然網路上有成千上萬的資料，但有經過標註可供督導式學習使用的資料只是冰山一角。因此，近年來人們也試著利用未標註的資料進行非督導式訓練（Unsupervised Learning） [10] [11] [12]。相較於督導式訓練，非督導式需要從沒有標註的資料中擷取出有意義的資訊，以往普遍被認為是相當困難的問題。然而非督導式訓練並非沒有成功的例子。人們在學習語言的過程中就有很大的一部分是非督導式訓練；嬰幼兒在成長過程中接受了來自於父母及週遭人們大量的說話聲音，而且幼兒並不會每接收到一段聲音就有人告訴他這段聲音是什麼音素，或者是什麼詞。然而在上述情況下，大多數的幼兒依舊能夠從這些大量未經標註的聲音中自然地學習語言 [13]。近年來使用類神經網路來模擬人腦的研究風氣日盛，模擬嬰幼兒學習語言的非督導式學習也被認為具有極大的潛力和進步空間。

語音領域中若能在應用非督導式學習上獲得成功，將能讓此領域邁向另一個新的里程碑。首先，需要大量專業語言知識的標註工作可以省去，語音技術便可以推廣至缺乏系統性專業語言知識的語言，如地方方言。再者，非督導式學習能與巨量資料時代相互配合，網路上大量未標註的資料都能被其利用。機器學習的

doi:10.6342/NTU201801733

概念中，資料量常與模型表現呈現正相關，因此可以期待大量的資料能夠帶給非督導式學習模型更佳的效能表現。

近年來許多非督導式學習演算法已經被使用在訓練深層學習的模型上，並且在包含語音的眾多領域中獲得十分重要之研究成果 [14] [15] [16] [17]。 在這眾多的結果中，最具有突破性發展的莫過於輸入特徵的抽取 [18] [19] [20]。以語音而言，以往抽取音訊特徵的方式為應用聲學知識，創造出近似人耳聽覺系統的特徵：梅爾倒頻譜係數（Mel Frequency Ceptral Coefficients, MFCCs） [21]。然而此種由人為所定義之音訊特徵受限於人類聲學知識之發展，要定義出能顯著提升效能之新音訊特徵並非易事。另一方面，使用深層模型所抽取之音訊特徵完全由資料本身決定，資料越多則模型所能學習之資訊也越多，所獲得之特徵也越佳。因此在巨量資料的時代下，使用深層學習模型可以獲得比以往人為定義法更佳的特徵 [22]。

使用深層學習模型來抽取特徵時，可以針對不同領域與應用之需要設計深層學習模型，進而抽取出各式各樣的特徵。在電腦視覺上，深層學習模型可從輸入模型之圖片抽取出重要資訊，進而顯著提升圖片分類的效能 [23]；在文字領域的自然語言處理上，深層學習模型可透過閱讀大量文章，從中學習出每個詞之間的關係，進而學習出每個詞最適切的特徵為何，此特徵稱為詞向量（Word2Vec） [1]。語音方面，近年來重要的突破莫過於能保留輸入音訊中音素資訊之語音詞向量（Audio Word2Vec） [2]。與詞向量概念相同，只是此時深層學習的模型輸入不是文字，而是許多以詞為單位的音訊。深層學習模型可以透過聆聽大量音訊，藉由每個詞中不同的音素進而學習出每個音訊間的關係。學習到此關係後，模型便可將一個詞的音訊以一個特徵向量表示，稱為語音詞向量。如同詞向量被廣泛應用在眾多文字領域的應用中，語音詞向量可以被使用在許多語音

doi:10.6342/NTU201801733

領域的應用中，如口述語彙偵測（Spoken Term Detection）。

雖然語音詞向量能夠透過非督導式學習的深層學習所獲得，然而在語音領域中非督導式學習的框架下所能獲得之輸入音訊通常為語句，語句中的詞邊界為一不可得之資訊。理想狀況下的語音詞向量應是模型能將一語句轉化為語音詞向量序列，序列中每個語音詞向量所代表的是語句中的一個詞的語音。有鑒於語音詞向量之應用潛力，如何將語音詞向量的輸入由詞的層級提升為語句層級，使其成為一完整之非督導式學習架構成為一十分吸引人之研究方向。

1.2 研究方向

本論文主要研究向為探討如何在無任何標註的情境下，將一輸入語句轉化為語音詞向量序列，每個語音詞向量所代表的是語句中的一個詞之語音，主要包含以下兩點：

- 第一部份主要探討深層學習模型內部訊號與語音中邊界的關聯，並探討將其應用於切割輸入語句之可能性
- 第二部份則是探討如何訓練深層學習模型，使其能將一輸入語句轉化為語音詞向量序列

1.3 相關研究

以往語音領域的非督導式學習旨在透過隱藏馬可夫模型（Hidden Markov Model, HMM）來將語音中不同等級的音訊，如次詞（Subwords）與音素（Phonemes），模擬為不同的音型（Tokens），藉此將一輸入語句以音型序列來表示 [24] [25]。此音型序列與用來模擬不同音型的隱藏馬可夫模型參數則被使用

doi:10.6342/NTU201801733

於後續的語音應用上。

而今在深層學習的框架下，隱藏馬可夫模型被深層學習模型所取代，所輸入之語句直接透過深層學習模型抽取特徵，並用於後續的語音應用上。由於深層學習模型能夠從資料中抽取出抽象概念的特徵（如一張手寫數字圖片其所代表的數字） [23]，因此在後續的應用上能夠獲得更佳的效能。

詞向量（Word2Vec）是由米氏（Thomas Mikolov）等人所發展出的一套使用深層學習模型抽取文字特徵的方法 [26] [1] [27]。將一篇文章中的每個詞輸入模型並要求模型必須判斷出此詞的上下文（Context），藉此讓模型學習出詞與詞彼此間的關係。此模型針對各個詞所抽取出之向量表示（Representations）稱為詞向量，為模型考量了詞與詞間彼此的關聯所產生之表示法，因此帶有語意（Semantics）的資訊。語意的資訊在處理文字的自然語言處理中具有十分重要的地位，因此詞向量被廣泛應用於後續許多自然語言處理相關的研究與應用中 [28] [29]。

另一方面，語音詞向量（Audio Word2Vec）則是由鍾氏（Yu-An Chung）等人所發表之用來表示一段音訊之新方法 [2]。在訓練深層學習模型時，模型首先需要將輸入之語音以一特徵向量表示，此特徵向量接著會被用來產生輸入模型之語音訊號。透過此種方式，模型在使用一特徵向量表示輸入語音時，此特徵向量必須含有關於輸入語音之重要資訊，否則無法憑此特徵向量便能產生輸入模型之語音訊號，而語音中最重要的資訊莫過於音素的資訊。由於輸入的音訊為各個詞之語音，因此每段語音中包含許多音素。鍾氏（Yu-An Chung）等人發現透過此法訓練模型，模型能夠從大量的音訊中學習出語音中音素的結構，進而將語音中音素結構之資訊有效地以一特徵向量表示。在後續應用中，此一特徵向量即可代表輸入語音，進而大量減少運算與記憶體用量。

doi:10.6342/NTU201801733

然而語音詞向量的輸入音訊為以詞為單位之語音訊號，在非督導式學習的框架下，輸入語音通常為語句且詞邊界為不可得之資訊。因此在完整的非督導式學習框架下，機器需要能夠從輸入語句中自動判斷出詞邊界，進而將輸入語句以語音詞向量序列表示。近年已有學者發現語音中邊界的判斷可以仰賴深層學習模型之內部訊號。在語音合成（Speech Synthesis）的應用中，吳氏（Zhizheng Wu）等人發現深層學習模型中用來溝通內部各元件的訊號變化與輸入音訊之音素邊界（Phoneme Bounadries）具有強烈的關聯 [3]。因此如何利用深層學習模型之內部訊號來作為詞邊界的判斷標，讓機器能夠自動決定語句中之詞邊界為一十分具有價值之研究方向。

1.4 研究貢獻

本論文主要研究貢獻為在前人所提出之語音詞向量的基礎上，將其進行延伸：讓輸入音訊從詞變為語句，使語音詞向量不再需要詞邊界之資訊，成為一完整之非督導式學習架構下之語音模型，在本論文中將此延伸後的語音詞向量稱之為分段式語音詞向量（Segmental Audio Word2Vec）。分段式語音詞向量透過兩種實驗進行效能評估，詞切割與口述語彙偵測。在這兩個實驗上分段式語音詞向量皆具有比傳統方法具有更佳之效能。

另外本論文也分析非督導式學習下之一存在於深層學習中的內部訊號與輸入語音之音素邊界間的強烈的關聯性，此訊號在本論文中稱為門限激發訊號（Gate Activation Signals）。其關聯透過音素切割（Phoneme Segmentation）實驗來展示。在音素切割實驗中，我們發現使用門限激發訊號能夠比傳統的遞迴式類神經網路的方法更加準確及穩健（Robust）的切割音素。

上述實驗結果並不侷限在英文，另外在其他語言如捷克文，法文與德文上皆

doi:10.6342/NTU201801733

有相似實驗結果，說明本論文前述之貢獻能夠概括化得應用在不同語言上，符合非督導式學習之精神。

1.5 章節安排

本論文接下各章節簡述如下：

- 第二章：首先介紹深層學習中非督導式學習的遞迴式類神經網路架構與基本訓練方法。第二部份介紹强化學習：一個同樣不需標註來訓練模型之學習方式。在第三部份及第四部份則分別介紹音訊切割與口述語彙偵測和其效能評量方式。在最後部份針對語音詞向量做深入介紹。
- 第三章：探討遞深層學習模型中門限激發訊號與音素邊界的關聯性，並利用音素切割實驗量化其關聯和强健性
- 第四章：介紹分段式語音詞向量之基本模型架構，並探討使用端對端訓練之可能性
- 第五章：介紹如何使用强化學習訓練分段式語音詞向量，並以詞切割及口述語彙偵測來評估模型效能
- 第六章：總結本論文的研究成果與未來展望

doi:10.6342/NTU201801733

第二章　背景知識

2.1　基於非督導式學習的遞迴式類神經網路

2.1.1　類神經網路

類神經網路是由生物的神經系統結構得到靈感所發展出的一套數學模型。在生物的神經系統中，神經元（Neurons）彼此由樹突、軸突連結。每個神經元的激發與否由閾值（Threshold）決定。在類神經網路的數學模型中，一個神經元會接受多個輸入（Input）的刺激，將這些刺激的總和通過一個激發函數來做為這個神經元的輸出（Output）。由於神經元彼此間的連結程度不同，來自不同神經元的輸入會分別乘以不同權重（Weights）。另外每個神經元各自不同的閾值則使用閾值偏移（Threshold Bias）來表示。在使用這類神經網路的模型時，我們會將一個神經元的所有輸入以一個向量**x**表示，$\mathbf{x} = [x_1, x_2, x_3, x_4..., x_n]^T$，稱之為輸入向量。給定一輸入向量，一個神經元的數學式可表示如下：

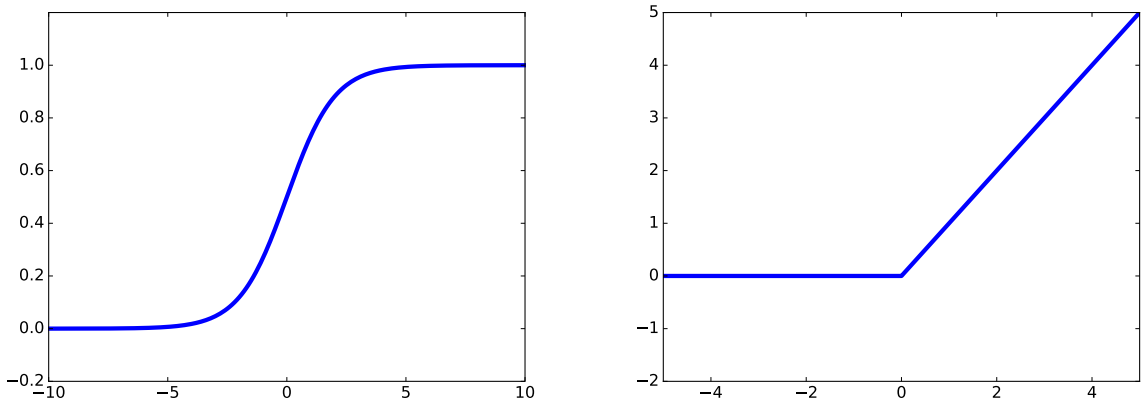
$$\alpha = \sigma(\sum_{i=0}^{n} w_i x_i + b) \tag{2.1}$$

其中w_i為此神經元對第i個輸入的權重，b代表閾值偏移而α則為此神經元最後的輸出。σ為激發函數，通常為一非線性函數。

若沒有使用激發函數σ，類神經網路的訊號傳遞為線性轉換。使用非線性激發函數能將類神經訊號的傳導轉變為非線性轉換，增加模型的複雜度。最常見的激發函數為S型函數（Sigmoid Function）：

$$sigmoid(x) = \frac{1}{1 + exp(-x)} \tag{2.2}$$

doi:10.6342/NTU201801733



(a) S型函數　　(b) 整流線性單元

圖 2.1: 不同種類的激發函數

S型函數的優點為處處可微分。而另一經常使用的激發函數為整流線性單元（Rectified Linear Unit, ReLU）[30]：

$$ReLU(x) = max(0, x) \tag{2.3}$$

此種激發函數在輸入訊號大於0時直接將訊號無損輸出；然而當輸入訊號小於0時則將此訊號壓縮為0。

深層類神經網路模型通常是將神經元規劃成層狀結構，每層均有數個神經元，每層神經元的輸出會傳遞給下一層的神經元作為輸入。整個類神經網路模型的輸入向量由輸入層（Input Layer）傳入，經過許多的隱藏層(Hidden Layer)，由最後一層神經元，稱為輸出層（Output Layer）輸出此模型的輸出向量，如圖2.2。事實上，對於處於同一層的神經元來說，由於它們輸入向量相同，所以此層的輸出可以利用矩陣乘法進行運算，如下式：

$$\boldsymbol{\alpha} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{2.4}$$

doi:10.6342/NTU201801733

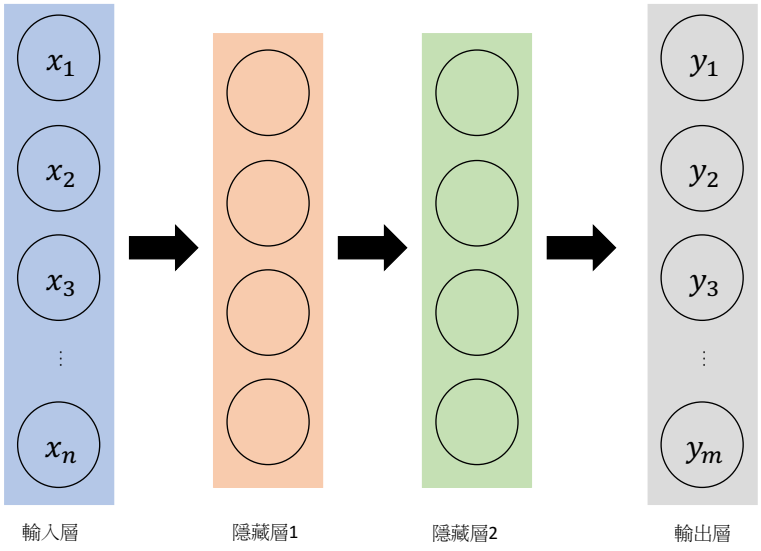


圖 2.2: 一具有兩層隱藏層的層狀深層類神經網路模型

其中$\mathbf{W} = [\mathbf{w_1 w_2 w_3 ... w_n}]^T$，而$\mathbf{w_i}$，$i = 1, ..., n$ 代表這層中n個神經元對於輸入訊號各自不同的權重。而神經元各自的閾值偏移則由向量$\mathbf{b}$表示。此層神經元的輸出由$\boldsymbol{\alpha}$表示。

我們若將此層神經元的所有輸出視為一個特徵向量，我們便可以此特徵向量作為下一層神經元的輸入，如此不斷迭代堆疊，此類神經網路便可不斷疊深。多層的類神經網路模型我們稱為深層類神經網路（Deep Neural Networks, DNN）。

2.1.2 類神經網路訓練

對於每一個輸入$\mathbf{x}$，我們希望類神經網路模型能夠給予我們所希望的答案$\mathbf{y}$。以語音辨識為例，給予一個語音的特徵向量，我們希望類神經網路模型能夠正確判斷此語音是屬於哪一個音素。我們透過訓練類神經網路模型使其能夠完成這件事情。要了解如何訓練類神經網路模型之前，我們需要先對模型的參數進行分類。所有參數可分為兩類，第一類為可以調整的參數，像是類神經網路模型中每個神經元對於輸入的權重，閾值偏移等等。這種參數會在訓練模型的過程中不斷地被

doi:10.6342/NTU201801733

調整。然而第二類的參數是必須在進行訓練前就設定好，而且在訓練過程中會一直保持不變的參數，稱為超參數（Hyper Parameters）。在類神經網路模型的訓練中，超參數通常為隱藏層的層數，一個隱藏層含有多少神經元等等。我們以Θ表示類神經網路模型中所有第一類參數所形成的集合。

類神經網路的訓練目標是希望其輸出，以$\hat{\mathbf{y}}$表示，能讓減損函數（Loss Function）最小化。所謂的減損函數，是由此輸入$\mathbf{x}$，和模型參數集Θ以及此筆輸入的答案$\mathbf{y}$所計算而得。減損函數代表的是此模型的效能表現，我們通常將減損函數設計為越低表示效能越好。類神經網路的訓練過程即尋找一組能夠最小化減損函數的模型參數Θ^*。

$$\Theta^* = arg \min_{\Theta} loss(\mathbf{x}, \mathbf{y}; \Theta) \tag{2.5}$$

上式中的$loss(\bullet)$就是減損函數，，因此在式2.5中我們所要尋找的是能夠使減損函數最小化的模型參數。

根據模型目標的不同，所設計出之減損函數也會不同。在處理迴歸問題（Regression Problems）時，最常使用的減損函數分別為均方差（Mean Squared Error, MSE），數學式表示分別為：

$$loss(\mathbf{x}, \mathbf{y}; \Theta)_{MSE} = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 \tag{2.6}$$

其中$\hat{\mathbf{y}}$表示將$\mathbf{x}$輸入類神經網路後所得到的輸出，而$\| \bullet \|^2$為方均運算，為方均根（Root Mean Square, RMS）之平方。均方差減損函數是以答案$\mathbf{y}$和模型之輸出$\hat{\mathbf{y}}$間的歐幾里德距離（Euclidean Distance）作為評估，其精神為統計學上的迴歸分析（Regression Analysis）。

另一方面，在處理分類問題（Classification Tasks）時則是最常使用交叉熵

doi:10.6342/NTU201801733

（Cross Entropy, CE）減損函數。若總共可能的類別有J種，此種問題所給予的答案$\mathbf{y}$會是一個維度為J之獨一餘零的向量（One-hot Vector），如$[0, 1, 0, 0..., 0]^T$。此種向量只會有一個維度為1其餘為0。為1的那個維度所對應的類別就是該輸入的正確類別，以前述中的獨一餘零向量為例，第二個維度所對應到的類別就是該輸入的正確類別。而類神經網路模型之輸出則同樣是一個J維的向量，每一個維度的值對應到將輸入$\mathbf{x}$分至某一類別的信心指數，最後會以信心指數最高的類別作為此類神經網路模型的判斷類別。

在使用交叉熵減損函數前，我們會需要將前述代表各類別信心指數的J維向量通過軟性最大化轉換（Softmax），轉變成機率分佈的輸出作為整個模型的最後輸出$\hat{\mathbf{y}}$，也就是各維度總和為1且各維度皆為一不小於0的實數。軟性最大化運算如下式：

$$\hat{y}_j = \frac{exp(\zeta_j)}{\sum_{j=1}^{J} exp(\zeta_j)} \tag{2.7}$$

ζ_j表示第j類的信心指數，這些透過軟性最大轉換前的信心指數又被稱為邏輯子（Logits）。最後使用模型輸出$\hat{\mathbf{y}}$與答案$\mathbf{y}$所計算之交叉熵之減損函數如下式：

$$loss(\mathbf{x}, \mathbf{y}; \Theta)_{CE} = \frac{1}{J} \sum_{j=1}^{J} -y_j \, log(\hat{y}_j) \tag{2.8}$$

以獨一餘零向量所表示的答案$\mathbf{y}$配合軟性最大轉換，便可發現交叉熵減損函數的精神在類神經網路模型將預測正確類別的機率予以最大化。式2.8中，只有正確類別的機率會被交叉熵減損函數所評估，然而在最大化這個類別的機率的同時，由於軟性最大化轉換的緣故，其他類別的機率也相對的被降低了，因此交叉熵減損函數和軟性最大化轉換的這個組合在分類問題的應用上十分有效。

針對不同問題定義好減損函數後，接著需要找出一組能夠獲得最小減損函

doi:10.6342/NTU201801733

數的模型參數。實務上在訓練類神經網路模型時，我們通常會使用梯度下降法（Gradient Descent），以迭代（Iterative）的方式進行訓練。所謂的梯度下降法是計算減損函數的梯度，接著將模型參數往此梯度的反方向做更新，以此來降低減損函數。藉由不斷迭代更新參數，減損函數便會不斷往最低點前進。具體而言，對於一個模型參數θ，我們以下式將θ更新為θ'：

$$\theta' = \theta - \eta \frac{\partial loss(\mathbf{x}, \mathbf{y}; \Theta)}{\partial \theta} \tag{2.9}$$

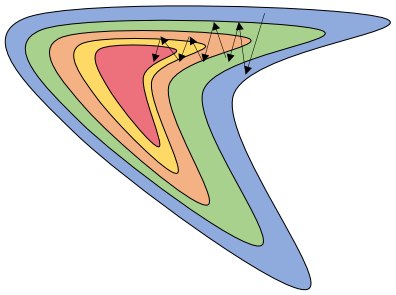
其中η被稱為學習率（Learning Rate），表示依據梯度更新的幅度大小。

學習率越大表示更新的幅度越快，通常減損函數下降的速度也越快，然而容易遇到無法收斂在最低點的狀況。學習率小雖然減損函數下降的速度比較慢，但是因為更新參數的方式較為精細，最終收斂的減損函數值通常比學習率大的更新方式為低。

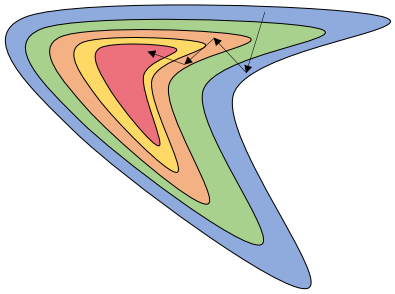
由於一個類神經網路的參數眾多，對每一個參數都各別以式2.9進行計算和更新不免曠日費時。反向傳播演算法（Backpropagation Algorithm）是個對類神經網路模型而言具有高度效率的更新參數方式。首先一個N層的類神經網路模型會由輸入端給予輸入，順向傳播（Feedforward Pass）至第N層的神經元，也就是輸出端產生輸出。將此輸出和答案計算減損函數後，便可以得到第N層神經元參數的梯度。利用鏈鎖率，我們可以快速得到第$N-1$層神經元參數的梯度。接著不斷重複，直到計算出第一層的神經元參數的梯度。整個過程就好像將梯度訊號從輸出層反向傳播回輸入層一樣，因此稱之為反向傳播演算法。

依照模型學習演算法的不同，類神經網路模型的訓練方式十分有彈性。最基本的梯度下降演算法為每次迭代都計算一次梯度，並依照式2.9更新參數。此種訓練方式在更新參數時並不考慮從前的梯度。相反地，加入了慣量（Momentum）

doi:10.6342/NTU201801733



(a) 無慣量之梯度下降法



(b) 具備慣量之梯度下降法

圖 2.3: 用慣量加速梯度下降法之一例。圖中不同顏色區塊表示減損函數值在不同參數下之大小，由外而內遞減。圖中之箭頭表示模型參數更新之方向，箭頭數目表示模型參數更新之次數。可以發現圖(b)的例子中使用慣量訓練法後可以加速模型參數更新至減損函數最小處的過程。

觀念的慣量訓練法則是會考慮從前的梯度資訊 [31]。對模型的某一參數θ，最基本的慣量訓練法可以用下式表示：

$$\Delta\theta_{t+1} = \xi\Delta\theta_t + \left.\frac{\partial loss(\mathbf{x}, \mathbf{y}; \Theta)}{\partial\theta}\right|_{\theta=\theta_t} \tag{2.10}$$

其中ξ為慣量係數，用以調控慣量的比例。由於慣量的觀念是考慮所有計算過的梯度，因此模型在更新參數時能保有更多資訊來決定梯度方向，因而加速模型參數的收斂。如圖2.3所示，圖中不同顏色區塊表示減損函數值在不同參數下之大小，由外而內遞減。圖中之箭頭表示模型參數更新之方向，箭頭數目表示模型參數更新之次數。可以發現圖2.3(b)的例子中使用慣量訓練法後可以加速模型參數更新至最佳點（Optimal Point），也就是減損函數最小處的過程。慣量的觀念衍生出許多不同的慣量訓練法，如ADAM訓練法 [32]。這些訓練法們針對慣量的調控各自有許多更為精細方法，在實務上都有很不錯的效果。

doi:10.6342/NTU201801733

2.1.3 遞迴式類神經網路

根據輸入資料的種類不同，深層類神經網路也發展出不同的變型。對於語音這種序列式資料（Sequential Data）而言，每一個時間點的資料並非彼此獨立，而是息息相關的。以語音辨識為例，在辨識第t個時間點的音素時，若能夠有此時間點之前$t-1$個時間點的資料，辨識結果一定能夠更為準確。也因此，遞迴式類神經網路就是設法將從前的輸入資料中重要的資訊以隱藏狀態（Hidden State）的形式保留下來，在處理當前輸入資料時便能夠擁有更多的資訊。

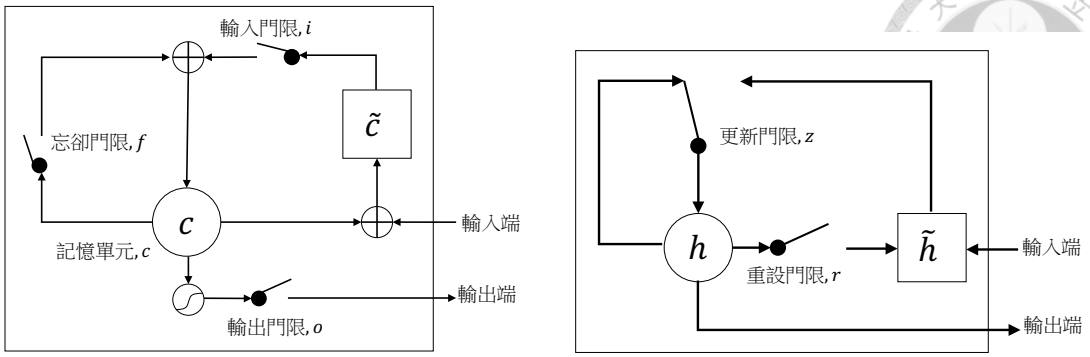
一個最簡單的遞迴式類神經網路模型在處理第t個時間點的輸入時可以用以下數學式表示 [33]：

$$\mathbf{c}_t = \sigma_h(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{c}_{t-1} + \mathbf{b}_c) \tag{2.11}$$

$$\mathbf{h}_t = \sigma_h(\mathbf{W}_h\mathbf{c}_t + \mathbf{b}_h) \tag{2.12}$$

其中$\mathbf{c}_t$與$\mathbf{h}_t$分別表示時間點t時此遞迴式類神經網路的隱藏狀態與輸出，$\mathbf{U}$則是運算過程中考量隱藏狀態之權重，下標c與下標h則分別表示在計算隱藏狀態與神經網路輸出時所使用的不同參數。然而，此種遞迴式類神經網路容易遇到梯度消失（Gradient Vanishing）的問題 [34]。具門限（Gate）機制的遞迴式類神經網路（Gated Recurrent Neural Networks）因此被提出，其中最著名的架構為長短期記憶神經網路（Long Short-term Memory Networks, LSTM） [7]。一個長短期記憶神經網路的神經單元由記憶單元（Memory Cells）、忘卻門限（Forget Gate），輸入門限（Input Gate）以及輸出門限（Output Gate）所組成。記憶單元用以儲存輸入過的資訊。忘卻門限則是控制是否要將記憶單元中的資訊清除。輸入門限以及輸出門限則分別控制輸入以及輸出資訊的多寡。門限機制的加入讓梯度訊號可以經由門限傳遞，避免了梯度消失的問題。另一著名的遞迴式類神經網路架構則為門

doi:10.6342/NTU201801733



(a) 長短期記憶神經網路單元　　(b) 門限遞迴單元

圖 2.4: 遞迴式類神經網路神經單元

限遞迴單元（Gated Recurrent Unit, GRU） [35]。門限遞迴單元可視為長短期記憶神經網路的簡化版。雖然同樣具有門限機制，但門限遞迴單元中隱藏狀態**c**即為神經網路輸出**h**且是門限遞迴單元只有兩個門限，分別為更新門限（Update Gate）與重設門限（Reset Gate），因此參數相較之下少了許多。因為參數量較少的關係，訓練起來也簡單許多。兩種具門限機制的神經單元架構如圖2.4所示。

2.1.4 自動編碼器

自動編碼器（Autoencoder）是一種資料壓縮（Data Compression）的深層類神經網路模型 [22] [14]。此深層類神經網路模型通常由一編碼器（Encoder）和一解碼器（Decoder）相接而成。編碼器將輸入壓縮為一個編碼（Code），此編碼通常維度會比輸入要小。而解碼器則是需要利用此編碼來還原（Reconstruction）出原來的輸入。由於解碼器在進行解碼時的依據只有編碼器所產生的編碼，因此此編碼被視為輸入端資料的一種壓縮後的表示法（Representations）。自動編碼器的訓練並不需要任何標註，因此是最常被使用在非督導式學習的類神經網路模型。自動編碼器通常使用均方差減損函數來訓練，此減損函數主要是評估輸入資料**x**和解碼

doi:10.6342/NTU201801733

器還原出的輸入資料$\hat{\mathbf{x}}$之間的均方差：

$$loss(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \tag{2.13}$$

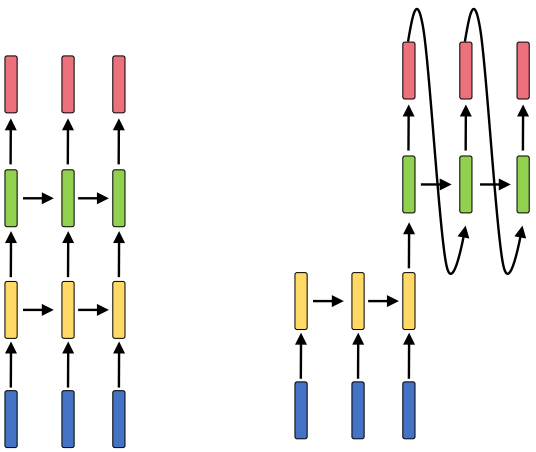
和深層類神經網路的自動編碼器相比，基於遞迴式類神經網路的自動編碼器在編碼時能夠考慮從前輸入過的資訊，同樣地在解碼時也能考慮從前看過的編碼，此種特性使其在訓練方式上也更加富有彈性。訓練方式主要分為兩種，分別為序列標註式訓練（Sequence Labeling Training）以及序列對序列式訓練（Sequence to Sequence Training） [36]，如圖2.5。序列標註式訓練與深層類神經網路訓練方式相同，在每個時間點皆會輸出一個編碼，而解碼器利用此編碼解碼出該時間點的輸入。由於遞迴式類神經網路能夠保留從前的資訊，因此在進行時間t的解碼時，能夠保有時間$t = 1$至$t = t - 1$的資訊：

$$\hat{\mathbf{x}}_t = RNN(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{t-1}) \tag{2.14}$$

序列對序列式訓練是較為特殊的一種訓練方式，此種訓練方式是將所有的輸入資料皆輸入遞迴式類神經網路的編碼器，過程中不輸出編碼。直到最後一筆輸入資料輸入後，將此時類神經網路中的記憶單元取出，當作是這一整個序列資料的編碼$\mathbf{e}$。解碼器需要利用此編碼解碼出所有時間點的輸入資料。在解碼時間點t時，會將解碼器所解碼出的前一個時間點的資料和編碼串接（Concat）後當作此時間點解碼器的輸入。此種訓練方式下的自動編碼器被稱作遞迴式自動編碼器（Recurrent Autoencoder），其解碼可用下式表示：

$$\hat{\mathbf{x}}_t = RNN(\mathbf{e}, \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, ..., \hat{\mathbf{x}}_{t-1}) \tag{2.15}$$

doi:10.6342/NTU201801733



(a) 序列標註式訓練　　(b) 序列對序列式訓練

圖 2.5: 遞迴式類神經網路訓練法。上圖中不同顏色分別代表：藍色表示輸入訊號；黃色表示編碼器；綠色表示解碼器；紅色表示輸出訊號。在序列對序列式訓練中，前一時間點的輸出訊號會成為下一個時間點的輸入訊號。

2.2 强化學習

2.2.1 馬可夫決策過程

强化學習（Reinforcement Learning）可用來解決馬可夫決策過程（Markov Decision Process, MDP）的問題。所謂馬可夫決策過程係為一代理人（Agent）需在一環境中（Environment）連續做出許多決策的過程，由下列五個成分所組成：

- 狀態（State, s）：指環境目前的狀態，也是代理人能掌握到的資訊。例如在圍棋中，狀態可以指目前的盤面。然而狀態的訂法不只一種，同樣以圍棋為例，可以指目前的盤面，也可以指前十步所有的盤面狀況。通常而言，當給予代理人的狀態包含資訊越多，則代理人能夠習得越佳的表現。
- 動作（Action, a）：每個狀態下代理人所能執行的動作。以圍棋而言，可以做的動作為盤面上尚未落子之處。

doi:10.6342/NTU201801733

- 策略（Policy, π）：為代理人之行為模式，也就是其在每個狀態下採取各種應對動作的選擇或方法，此成分需要由代理人與環境互動進而學習出一套策略。
- 轉移機率（Transition Probability）：當代理人在狀態s下，做出動作a時，狀態會轉移到s′的機率。以圍棋作為例子而言，在某盤面的情況下，代理人做出了一個動作，也就是在某處落子，則下次輪到代理人落子時的各盤面機率則為此二盤面間的轉移機率。此項目可以代表環境的動態變化。
- 獎勵（Rewards, r）：獎勵為系統設計者根據系統目標，針對代理人所定義出的一套回饋機制，旨在透過獎勵讓代理人了解每個執行動作的好壞，進而摸索出能夠達到系統目標的動作。當代理人在狀態s下，執行了動作a後轉移至狀態s′，系統會根據此過程給予代理人一個獎勵，代表代理人所執行的動作是否符合系統目標。越大的獎勵代表代理人所執行的動作越符合系統目標，而代理人的學習目標便是盡力獲得越大的獎勵。

在轉移機率已知時，可以應用動態規劃（Dynamic Programming）來解決馬可夫決策問題。然而在多數的現實問題中，轉移機率為不可知資訊，因此無法使用動態規劃來處理馬可夫決策問題，需要應用强化學習。

2.2.2 强化學習簡介

在現實生活中的馬可夫決策問題通常複雜度高而且狀態間的轉移機率為未知資訊，因此無法使用動態規劃求出最佳解，需要轉而使用機器學習的方法來處理。機器學習中的强化學習即是為了處理馬可夫決策問題所發展出的學習演算法。圖2.6為使用强化學習之學習流程。首先在環境當前的狀態s（圖中例子為環境中有一杯水）會被作為輸入資料被輸入代理人，在强化學習中此代理人即為需要進

doi:10.6342/NTU201801733

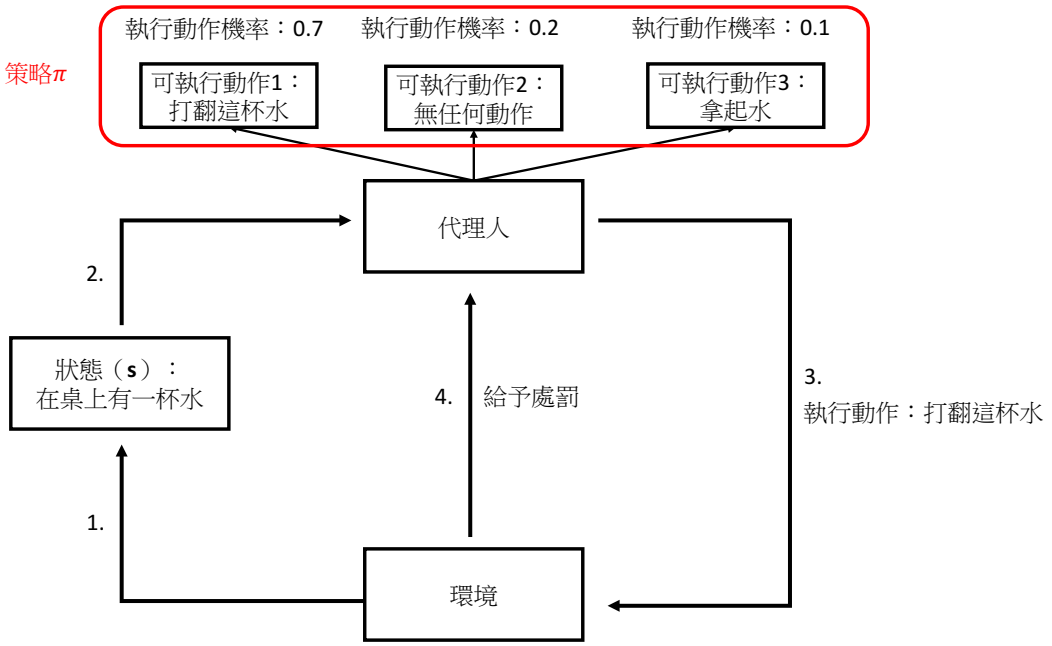


圖 2.6: 强化學習流程圖，箭頭旁之數字表示流程順序

行訓練之强化學習模型。在獲得了當前的狀態s後，代理人會對於所有可執行動衡量出執行之機率，這些動作之機率分佈稱作策略。代理人接著從所有動作中依照策略來挑選動作執行，在圖中的例子為代理人選擇打翻環境中的這杯水。環境可給予代理人所執行之動作一個獎勵。若打翻水為一不被允許之動作，則此獎勵將轉變為處罰，藉此讓代理人學習到不應該打翻水。

透過上述過程，强化學習模型不斷在各種狀態下執行動作，與環境互動獲得獎勵。而設計環境之系統設計者對於强化學習模型有一個希望其表現之行為模式，也可稱作系統目標。藉由獎勵大小來告訴强化學習模型其執行之動作是否符合系統目標，藉由不斷互動來修正模型的行為模式，使其逐漸符合系統目標。

雖然乍看之下强化式學習似乎是一種督導式學習（Supervised Learning），只是把預測答案的正確與否改由獎勵大小表示。然而實際上强化式學習與監督式學習本質上有許多差異。督導式學習在進行訓練之前，對於每一筆輸入資料需要有

doi:10.6342/NTU201801733

完整標註好的答案；强化學習只需要定義好計算獎勵的準則，根據每一筆輸入資料便可計算出獎勵以進行模型訓練。

隨著時間的演進，强化學習也發展出了各式各樣的學習演算法。這些演算法主要可以分為兩類：基於價值（Value-based）的强化學習與基於策略（Policy-based）的强化學習。基於價值的强化學習旨在訓練一個評分員（Critic）來根據當前狀態s為不同策略（Policy）π打分數，因此可以可以藉由此評分員來選出最適合的策略。另一方面基於策略的强化學習則是利用獎勵直接針對代理人的策略進行調整，並無評分員的存在。在本論文中所使用的强化學習演算法為後者，因此在後續章節中僅針對基於策略的强化學習作介紹。

2.2.3 基於策略的强化學習

基於策略的强化學習可以想像是希望尋找一個策略函數$\pi(\mathbf{s})$將每個狀態s對應至一個最適切的動作a：

$$a = \pi(\mathbf{s}) \tag{2.16}$$

在簡單的問題中，此策略函數可以使用人為制定的規則（Rule-based）來將每一個狀態對應至一個動作，然而此方法並不適用於高複雜度的問題上。近年來深層學習也被應用於基於策略的强化學習，策略函數之輸出由深層類神經網路（Neural Network, NN）所決定。實作上類神經網路會依照輸入狀態s來為每個動作決定出一個分數ζ。假定所有可執行之動作共有J種，類神經網路便會輸出一個J維大小的向量$\boldsymbol{\zeta}$，其中每個維度為對應動作之分數ζ_j。而這些分數會通過一個軟性最大化（Softmax）後得到執行每一個動作之機率p_j，最後策略函數之輸出為具有最大機率的動作a。上述過程可以使用式2.17至式2.19表示。

doi:10.6342/NTU201801733

$$\boldsymbol{\zeta} = NN(\mathbf{s}) \tag{2.17}$$

$$p_j = \frac{exp(\zeta_j)}{\sum_{j=1}^{J} exp(\zeta_j)} \tag{2.18}$$

$$a = arg \ \max_{j}[p_1, p_2, ..., p_J] \tag{2.19}$$

目前强化學習已發展出許多演算法來訓練前述之類神經網路 [37] [38] [39]，其中主流的訓練法為策略梯度（Policy Gradient）訓練法 [37]，本文中亦使用此技術，因此在接下來的文字中將介紹此訓練方法。為求說明精簡，我們接下來使用式2.20來表示式2.17至式2.19之過程。

$$a = \pi^{(\Theta)}(\mathbf{s}) \tag{2.20}$$

其中Θ為類神經網路模型之參數集。

使用策略梯度訓練法時，其訓練目標為尋找一組前述之類神經網路參數Θ來最大化代理人所獲得之獎勵的期望值。因此定義目標函數$J(\Theta)$如下：

$$J(\Theta) = \mathbb{E}_{\pi^{(\Theta)}}[r] \tag{2.21}$$

右式期望值$\mathbb{E}$之下標$\pi^{(\Theta)}$表示此為代理人根據策略函數$\pi^{(\Theta)}$所獲得之獎勵期望值。

針對每一個由策略函數$\pi^{(\Theta)}$所決定之動作a，環境會給予一個獎勵r，此獎勵將被用在更新參數集Θ上。

在說明更新參數集Θ的數學式之前，我們首先需要說明獎勵r如何被用來更新參數。由於强化學習所處理之馬可夫決策過程（Markov Decision Process, MDP）為一連續決策問題，因此在此過程中代理人需要與系統連續執行若干動作。每當

doi:10.6342/NTU201801733

代理人執行一個動作a後，其會從系統設計者所設計之環境獲得一個獎勵r代表該動作是否符合系統目標。亦即若代理人持續與環境互動T次，則獎勵也會是一個長度為T的序列，$[r_1, r_2, ..., r_T]$。為了完整考慮時間點t時所執行之動作a_t在馬可夫決策過程中之好壞，我們會將該時間點t後所有獲得之獎勵納入考慮，因此實際上對於時間點t時所執行之動作a_t的獎勵估計為：

$$\tilde{r} = \sum_{i=t}^{T} \gamma^{i-t} r_i \tag{2.22}$$

其中r_i為時間點i時代理人所獲得之獎勵。γ為折扣係數，為一小於1之正實數。為求符號簡潔與一致，以下不特別列出代表時序的符號t，僅需將$\tilde{r}$理解為環境所給予代理人執行動作a之一個含有未來資訊的獎勵即可。

在將參數集Θ更新為Θ'時，首先利用似然比值技法（Likelihood Ratio Trick）計算目標函數之梯度 [40] [41]：

$$\nabla_\Theta J(\Theta) = \mathbb{E}_{a\sim\pi^{(\Theta)}}[\tilde{r}\nabla_\Theta log(p_a)], \tag{2.23}$$

式中的$\tilde{r}$為前述之環境根據代理人所執行的動作a所給予的獎勵，右式期望值之下標$a \sim \pi^{(\Theta)}$表示此動作由策略函數$\pi^{(\Theta)}$所決定。p_a表示式2.18中由策略函數$\pi^{(\Theta)}$所輸出之動作a的機率。由上式不難看出若獎勵$\tilde{r}$為一大於零之實數，則代理人所執行的動作a之機率將會被放大，反之則是縮小。其物理意義為若環境認為代理人所執行之動作a符合目標行為模式，其給予的獎勵便為正值，藉此鼓勵代理人執行此動作；反之給予的獎勵則為負值，讓代理人在未來避免執行該動作。

計算完目標梯度後，接著使用梯度上升法（Gradient Ascent）來更新參數集Θ，藉此將目標函數最大化：

doi:10.6342/NTU201801733

$$\Theta' = \Theta + \nabla_{\Theta} J(\Theta) \tag{2.24}$$

另外由於在强化式學習下，代理人所拿到的只有針對每個動作所得到的獎勵，其並不知道是否存在能夠獲得更佳獎勵的動作存在。因此在强化學習的訓練上，代理人會需要做許多的探索（Exploration）。所謂的探索即是去嘗試沒有做過的動作，透過這些探索試圖去找出能夠獲得最大獎勵的動作。要能獲得最大之獎勵最好的情況是代理人能夠將所有的動作都探索過，如此才能找出獲得最大獎勵的動作。

然而此探索過程可能過於耗時，造成訓練時間過長，亦或是探索空間過大，代理人不可能在所有狀態都窮舉所有動作並進行探索。因此實作上在訓練代理人時，代理人在面對狀態s所執行的動作a為從該時間點的策略函數$\pi^{(\Theta)}(\mathbf{s})$中取樣（Sample）之後的結果，因此將式2.19中的取最大值之索引值（arg max）運算改為從各動作之機率所形成之多項分佈（Multinomial Distribution）中進行取樣：

$$a \sim \pi^{(\Theta)}(\mathbf{s}) = [p_1, p_2, ..., p_J] \tag{2.25}$$

其中J為可執行之動作數目且p_j所表示的是執行第j個動作的機率。

2.2.4 以遞迴式類神經網路進行之强化學習

如同前節所提到的，强化學習所處理之馬可夫決策過程（Markov Decision Process, MDP）為一連續決策問題，因此在此過程中代理人需要與系統連續執行若干動作。在每個時間點t，代理人執行一個動作a_t後都會獲得一個獎勵r_t代表該動作是否符合系統目標。若代理人持續與環境互動T次，則獎勵也會是一個長度為T的序列。然而此種獎勵方式存在著許多問題，如有些動作可能短期內看不出效果，是

doi:10.6342/NTU201801733

影響較為長遠的，如對弈中常見的佈局。或者是決策過程中的動作無法與系統目標存在著能夠人為定義出來的獎勵方式，只有在整個決策過程結束才有辦法給予獎勵，如聊天機器人或者是類神經網路模型架構之探索 [42]。

針對上述問題，近年來許多强化學習的研究皆使用遞迴式類神經網路配合策略梯度訓練法，只在結束互動時計算出一個獎勵r，並以獎勵r來作為給予每個時間點動作的獎勵r_t，利用各時間點隱藏狀態的傳遞來調整各時間點之動作，藉此解決上述問題，示意圖如圖2.7。

由於遞迴式類神經網路之結構，其可以考慮所有輸入過的狀態來決定每個時間點各個動作的分數ζ_t^j（下標t表示時序，上標j表示第j種動作，以下亦同），並透過軟性最大轉換將其轉換成該動作之機率p_t^j。而每個時間點由遞迴式類神經網路模擬之策略函數$\pi^{(\Theta)}$輸出為具有最大機率的動作a_t。此使用遞迴式類神經網路所模擬之策略函數$\pi^{(\Theta)}$的決策過程可由式2.26至式2.28表示。

$$\boldsymbol{\zeta}_t = \mathit{RNN}(\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_t) \tag{2.26}$$

$$p_t^j = \frac{exp(\zeta_t^j)}{\sum_{j=1}^{J} exp(\zeta_t^j)} \tag{2.27}$$

$$a_t = arg \max_j [p_t^1, p_t^2, ..., p_t^J] \tag{2.28}$$

在進行參數更新時，系統會以與環境互動結束後所獲得之獎勵r來作為給予每個時間點動作的獎勵r_t，因此參數集Θ之梯度的計算方式由式2.23將轉變為式2.29：

$$\nabla_\Theta J(\Theta) = \mathbb{E}_{a_t \sim \pi^{(\Theta)}}[r \nabla_\Theta \sum_{t=1}^{T} log(p_t^{a_t})], \tag{2.29}$$

doi:10.6342/NTU201801733

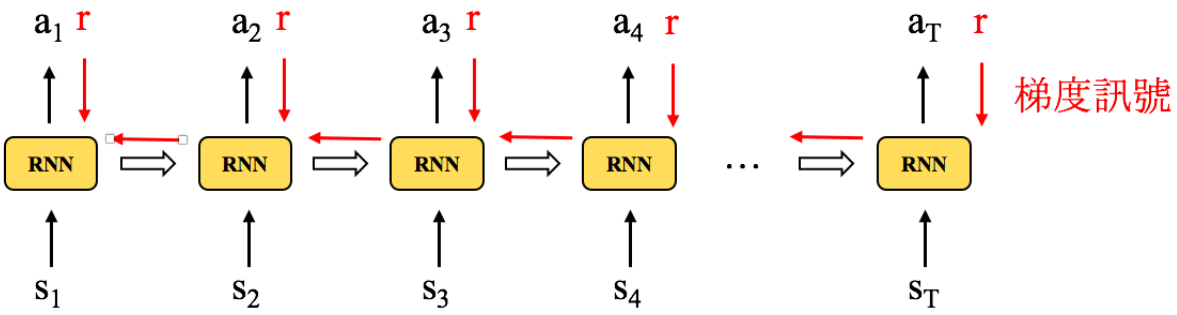


圖 2.7: 使用遞迴式類神經網路以策略梯度演算法進行訓練

右式期望值之下標$a_t \sim \pi^{(\Theta)}$表示動作a_t由策略函數$\pi^{(\Theta)}$所決定。$p_t^{a_t}$表示式2.27中由策略函數$\pi^{(\Theta)}$所輸出之動作a_t的機率。本式之概念與前節之式2.23雷同，只是由於現在改為遞迴式類神經網路之架構，因此在更新參數時會將整個互動過程（$t = 1 \sim T$）列入考慮。計算完參數集Θ之梯度後同樣使用梯度上升的方式用式2.24來更新參數集以求將目標函數$J(\Theta)$最大化。

由於各時間點間隱藏狀態的傳遞，使用反向時間傳播演算法（Back Propagation Through Time, BPTT）更新遞迴式類神經網路的參數時，各時間點皆會考慮到此動作對於後續動作的影響，因此一個具有長遠影響的動作對後續時間點所造成之影響便能以此被納入考慮。

2.3 音訊切割

語音訊號和文字最大的差別是語音訊號是連續的，並非如同文字是以詞為單位斷開。然而人類的語意表達是以詞為單位，同屬一個詞的語音訊號應在處理時能夠同時被考慮，如此才能產生有意義的辨識或是更進一步的應用。如何將這連續的語音訊號以詞為單位切割成不同片段的語音便成為一重要的問題。

然而一段語音是由許多詞組成，而每個詞又能拆成許多音素。因此

doi:10.6342/NTU201801733

語音訊號切割又可分為詞切割（Word Segmentation）與音素切割（Phoneme Segmentation）。詞切割之目標在於將音訊以詞為單位進行切割，每個切割出之音訊片段為一個詞之音訊；而音素切割則是以希望每個切割出之音訊片段為一音素。兩者差異在於切割出之語音片段長度（詞的長度較長而音素長度較短），其餘概念相同。為避免內容重複，以下只針對音素切割進行說明。

音素切割的目標在於找出語音中音素間的邊界。實作上，一段語音訊號會先被切割成在時間軸上許多部分重疊的音框（Frames）。針對每個音框都抽取出其梅爾倒頻譜係數（Mel-frequency Cepstral Coefficients, MFCCs）作為其語音特徵。在做音素切割時，我們挑選部分時間點作為認定的音素邊界（Phoneme Boundaries），且以距離該認定音素邊界最近的音框作為正確答案。在此框架下，評估效能的方法是將其視為資訊檢索（Information Retrieval）的問題，最常使用的準確率（Precision Rate）、召回率（Recall Rate）以及F1分數（F1-Score）的方式進行評估 [43] [44]。 準確率是將從所有認定的音素邊界的音框中，計算正確發掘出之音素邊界數和認定的邊界數的比值所計算而成得：

$$Precision = \frac{N_{hit}}{N_f} \tag{2.30}$$

其中N_{hit}表示正確被發掘出的邊界數量，N_f則是認定的邊界數量。而召回率則是計算被正確發掘出的邊界數目與所有認定的邊界數目的比值：

$$Recall = \frac{N_{hit}}{N_{ref}} \tag{2.31}$$

N_{ref}表示所有認定的邊界數目。最後，F1分數為準確率和召回率的調和平均：

$$F1\ score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.32}$$

doi:10.6342/NTU201801733

實際評估效能時，由於此框架下所使用之認定的音素邊界為一近似結果，且為了能有效比較各方法之優劣，我們常會使用一容忍窗（Tolerance Window）來作為我們容許的音素邊界誤差。在音素切割的文獻中以20毫秒（Miliseconds）的容忍窗最為常用 [45]。換言之，只要機器所發掘出來的音素邊界與真實的音素邊界差距在20毫秒以內，我們便認定此為一正確發掘之音素邊界。

對一段轉換為若干個音框的語句而言，其中若干個音框代表的是音素邊界。而音素邊界切割的目標便是將這若干個音框找出。如圖2.8(a)，表示一段轉換為若干個音框的語句，藍色音框表示音素邊界。圖2.8(b)所表示的是一音素切割之實驗結果，有著紅色框線之音框表示被模型認定為音素邊界的音框。在圖2.8(b)的例子中，左邊所認定之音素邊界（兩個紅框者中左邊的那一個）雖然沒有與語料所提供之音素邊界相符（Hit），但由於和真實音素邊界的差距小於容忍窗的大小，因此認定兩者相符，所以圖2.8(b)的音素切割結果，其認定之邊界與真實邊界之相符數量（Number of Hits, $\#hits$）為2。圖2.8(c)和圖2.8(d)的相符數量則分別為1和0。

音訊切割在督導式學習的框架下有許多方法已被提出，如以模型為基礎的隱藏馬可夫模型（Model-based Hidden Markov Models），或是以類神經網路模型為基礎，應用結構式減損函數（Structure Loss Function）所訓練的模型 [46]。在非督導式學習的框架下，音訊切割常是藉著分析語音訊號的能量變化或是頻譜資訊（Spectral Information）的變化來進行 [47]。也有以分群（Clustering）的觀點所進行的音素邊界切割，如階層聚合式分群法（Hierarchical Agglomerative Clustering, HAC） [48] [49]。

在近年，深層學習在語音領域中逐漸展露頭角。因此即使非督導式學習下的語音切割在先前已有了十分出色的結果，學者們仍試著將類神經網路模型引入非督導式學習的語音切割，如遞迴式預測模型（Recurrent Predictor Model） [45]。

doi:10.6342/NTU201801733

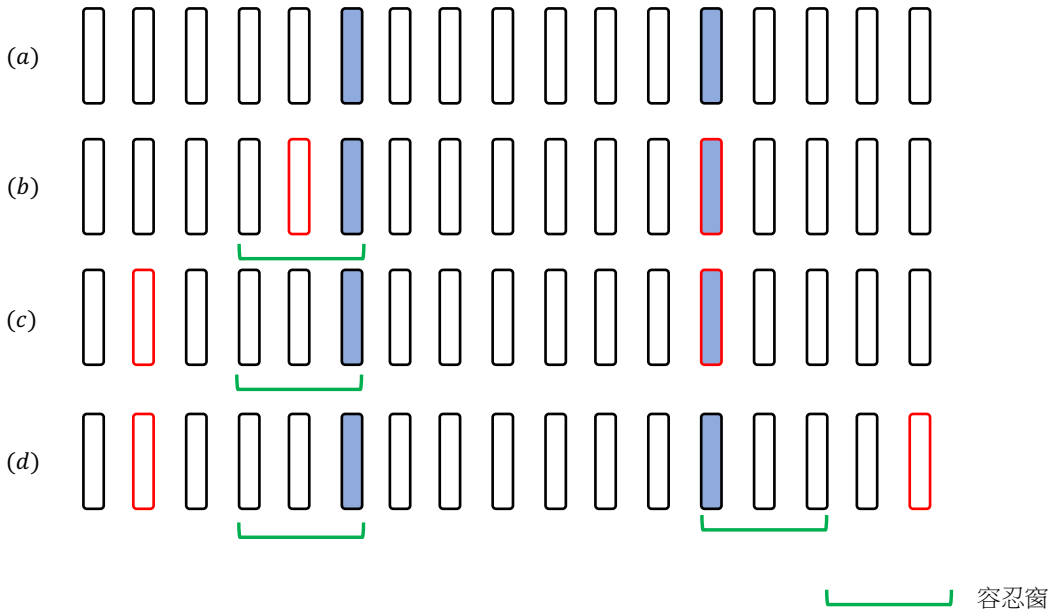


圖 2.8: 音素切割效能評估示意圖。圖(a)表示一段轉換為若干個音框的語句，藍色音框表示音素邊界。圖(b)，圖(c)與圖(d)所表示的是三組音素切割之實驗結果，有著紅色框線之音框表示被模型認定為音素邊界的音框。綠色框線表示容忍窗之大小，20毫秒之容忍窗為兩個音框距離大小，因此只要被模型認定為音素邊界的音框與音素邊界相距在兩個音框以內即認定該音素邊界被成功發掘。因此在圖(b)，圖(c)與圖(d)之音素切割結果中，與真實邊界之相符數量分別為2，1和0。

doi:10.6342/NTU201801733

2.4 口述語彙偵測

按例查詢（Query-by-Example, QbE）下的口述語彙偵測（Spoken Term Detection）問題是給予一個語音查詢指令（Spoken Query）和一個語音文件（Spoken Document）的資料庫，希望從這資料庫中檢索出含有此語音查詢指令的語音文件。在督導式學習的框架下，通常是使用自動語音辨識將語音查詢指令以及語音文件都轉換成文字後再進行匹配 [50]。然而此種做法不僅需要大量資源來訓練一個語音辨識模型，另外也存在辭典外（Out-of-Vocabulary, OOV）詞彙和辨識錯誤（Recognition Errors）的問題。在非督導式學習的框架下，通常直接從訊號匹配的角度切入，不僅節省資源也避免了辭典外詞彙的問題。傳統方法上以動態時間校準（Dynamic Time Warping, DTW）為基礎，針對由語音查詢指令以及語音文件兩者所抽取出的語音特徵進行相似度比對 [12]。此種類方法的缺點除了十分耗時外，使用語音特徵的相似度比對常常因為語者的不同或是錄音環境影響而失準。為了克服相似度的問題，也有學者嘗試利用以模型為基礎的隱藏式馬可夫模型來進行相似度的衡量。此種方法先將語音訊號轉換以一系列的音型（Tokens）模型表示，用音型模型序列間的相似度取代語音特徵的相似度 [24]。

口述語彙偵測的效能我們通常使用平均準確平均值（Mean Average Precision, MAP）來表示。對於一個語音查詢指令而言，我們的模型都給予語音文件資料庫中的每一個語音文件一個相關分數（Relevance Score），並希望含有此語音查詢指令的語音文件們的相關分數能夠比其他語音文件的分數都要高。將所有語音文件，假設總共有N_D個，依照相關分數由高至低依序排列為$D_1, D_2, ..., D_{N_D}$。對排序後位於第i位置的語音文件D_i，我們可以根據$D_1, D_2, ..., D_i$計算出一個準確率$Precision(i)$。一個語音查詢指令在語音文件資料庫中的平均準確率（Average Precision, AP）的計算方式為：

doi:10.6342/NTU201801733

$$AP = \frac{\sum_{i=1}^{N_D} Precision(i) * Relevance(i)}{N_Q} \tag{2.33}$$

其中N_Q表示含有此語音查詢指令的語音文件數量，$Relevance(i)$則為一個二元函數，表示在第i位置的語音文件是否包含此語音查詢指令：若包含則$Relevance(i) = 1$反之則為0。而將所有語音查詢指令的平均準確率做平均後就能得到平均準確平均值。

2.5 語音詞向量

由米氏（Thomas Mikolov）等人在2013年所發表之詞向量（Word2Vec）保留詞與詞間的關係，進而給予每一個詞專屬的特徵向量來表達其所含有之語意。受此啓發，在2016年由鍾氏（Yu-An Chung）等人發表了語音版之詞向量，稱作語音詞向量（Audio Word2Vec）。本節將針對此二者進行詳細介紹。

2.5.1 詞向量簡介

人類在書寫文字進行溝通時，其表達語意之基本單位為詞。因此若希望機器能夠藉由讀取文字為人類進行服務，其在了解一段文字之意涵時需要充分掌握每個詞之語意。因此對於每一個詞，如何有效表示其所含之語意便成為自然語言處理（Natural Language Processing）一個重要的研究問題。傳統上最簡單的表示法為使用章節2.1.2中所介紹之獨一餘零向量（One-hot Vector）來表示一個詞。此獨一餘零的向量維度大小為所有詞的數目，即辭典大小，而唯一為1的維度所對應的就是該詞。換句話說，此法假設每一個詞所含的語意皆不相同，獨一餘零向量中的每個維度都代表一個潛藏的語意。此法之優點為簡單，不需要任何知識與訓練便

doi:10.6342/NTU201801733

可使用。此外學者也發現此表示法與一些機器學習模型搭配起來能夠具有不錯的效能，因此至今許多研究都會首先嘗試此表示法。

然而獨一餘零表示法也存在許多顯而易見之問題。首先由於詞的數量十分巨大，為數萬至數十萬之量級。在此數量級下此表示法十分稀疏（Sparse），因此在運算上效率十分低落，也容易遇到維數災難（Curse of Dimensionality）的問題造成訓練時間過久或是效能不彰。另一個問題是此法認定每個詞之語意皆不相同，顯然與真實情況相悖。在人類的語言中，兩個不同文字具有相同語意，互為彼此之同義字（Synonym）為十分常見之情形。使用此表示法無法表示出此現象。

有鑒於上述之限制，辛氏（Geoffrey E. Hinton）在1986年提出分佈式表達（Distributed Representation）的概念 [51]。每個詞不再使用一個獨一餘零向量表示，而是使用一維度遠小於詞數目的特徵向量來表示，且每個維度的值不再受限於原本的0或1，而是任意實數。此種表達方式稱為詞嵌入（Word Embeddings）。使用詞嵌入來表示詞時，詞所代表的語意之相似度便可以用歐式距離（Euclidean Distance）或是餘弦相似度（Cosine Similarity）來表達。因此前段所述獨一餘零向量表示法之缺點便可由此克服。

使用詞嵌入來表示文字的語意已有許多模型被提出，如考慮不同文件中各詞頻間的關係並將其使用利用奇異值分解（Singular Value Decomposition）對文字空間進行降維的潛藏語意分析（Latent Semantic Analysis, LSA）。在近年，亦有使用類神經網路來為每個詞產生詞嵌入 [5] [52]。在使用類神經網路訓練語言模型時，每個詞首先會被隱藏層（Hidden Layer）投射至向量空間產生詞嵌入，這些詞嵌入接著被用來訓練語言模型（Language Model），整個訓練過程中語言模型與詞嵌入為共同訓練（Jointly Training）。由於訓練目標為語言模型，因此這種訓練

doi:10.6342/NTU201801733

方法下之詞嵌入為語言模型之副產物。若只是需要為每個詞找一個最佳的表示法而非訓練語言模型，此種訓練方式十分缺乏效率 [26]。

為解決前述缺點，在2013年米氏（Thomas Mikolov）等人提出了一個更有效率的方法來產生詞嵌入，學界將使用此方法所產生之詞嵌入稱為詞向量（Word2Vec） [26]。米氏（Thomas Mikolov）等人所提出之模型共有兩種：連續詞袋模型（Continuous Bag-of-Words, CBOW）與跳躍文法模型（Skip-gram）。此兩種模型為兩互相對稱結構：連續詞袋模型是利用前後文來預測出當前的詞；跳躍文法模型則是利用當前的詞來預測前後文之內容。由於此兩者為互相對稱結構，我們僅以跳躍文法模型作為說明，如圖2.9。跳躍文法模型架構為三層的類神經網路：輸入層，隱藏層與輸出層。輸入層為當前的詞$\mathbf{x}_t$，使用獨一餘零向量（One-hot Vector）來表示，其維度大小為辭典大小V。位於輸入層之後的隱藏層則將此獨一餘零向量投影至一維度為J的向量空間產生詞$\mathbf{x}_t$的詞向量$\mathbf{e}_t$。而最後的輸出層則會將維度為J的詞向量$\mathbf{e}_t$再轉換為一個維度大小為辭典大小V之獨一餘零向量來分別預測當前詞$\mathbf{x}_t$在一個範圍大小C內的前後文：$\mathbf{x}_{t-\frac{C}{2}}$、$\mathbf{x}_{t-\frac{C}{2}+1}$、...，$\mathbf{x}_{t+\frac{C}{2}}$。在此訓練方法下模型即可學習到當前詞與其前後文之關係，進而給予當前詞一個最佳的詞向量表示法來表示其語意。

詞向量最大的優點在不需要任何標註，只需要讓機器閱讀大量文章機器即可自動學習出詞與詞間的語意關聯，為一非督導式學習（Unsupervised Learning）訓練法。詞向量用途甚廣，在文字領域中可以被使用在詞性標註（POS Tagging）、句法分析（Syntactic Parsing）、語意分析（Semantic Analysis）、情感分析（Sentiment Analysis）等等應用上。若單純要衡量詞向量之好壞，也可以利用詞向量間的餘弦相似度來做同義詞搜尋，量化其搜尋效能 [53]。除了量化分析外，詞向量也具有良好的可解釋性（Interpretability），對於定性分析時十分有

doi:10.6342/NTU201801733

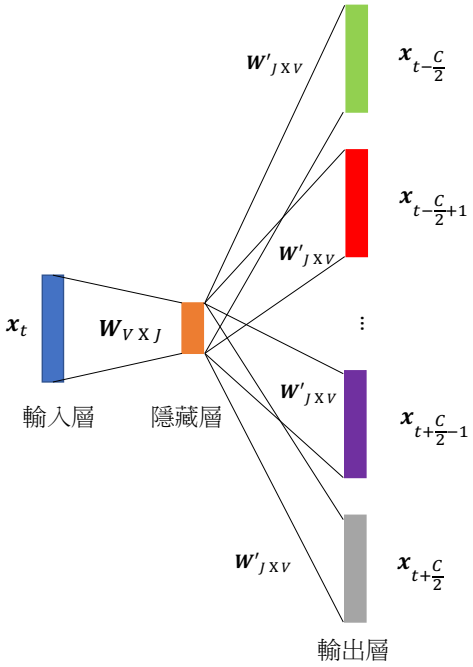


圖 2.9: 跳躍文法模型。當前詞$\mathbf{x}_t$透過隱藏層轉換為詞向量$\mathbf{e}_t$，再使用詞向量分別預測當前詞$\mathbf{x}_t$在一個範圍大小C內的前後文：$\mathbf{x}_{t-\frac{C}{2}}$、$\mathbf{x}_{t-\frac{C}{2}+1}$、...，$\mathbf{x}_{t+\frac{C}{2}}$

doi:10.6342/NTU201801733

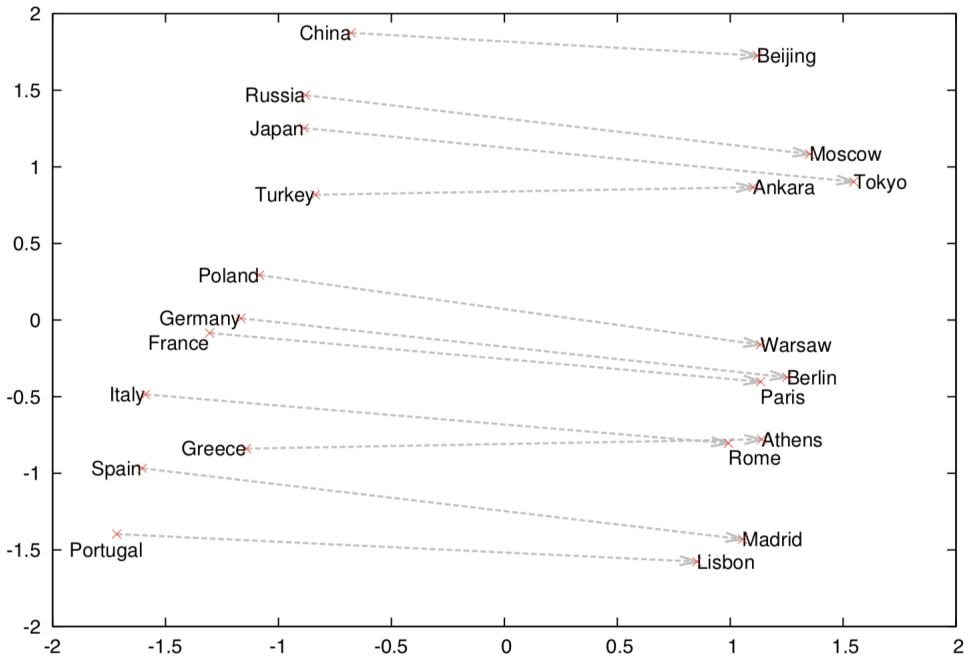


圖 2.10: 將100維之詞向量利用主成份分析（Principal Component Analysis, PCA）投射至2維平面上之詞向量。位於左半部的各個國家之名稱往相同方向移動可以得到其首都名稱。取自參考文獻 [1]

幫助。米氏（Thomas Mikolov）等人將100維之詞向量利用主成份分析（Principal Component Analysis, PCA）投射至二維平面上，形成圖2.10。圖中可以看出位於左半部的各個國家之名稱往相同方向移動可以得到其首都名稱，表示詞向量包含了語意中「國家-首都」之關聯，並且將其用向量空間中的某個移動方向表示。

2.5.2 語音詞向量簡介與應用

由於詞向量在文字領域的應用中展現無比的潛力，學者們也將詞向量的觀念引入語音領域中。鍾氏（Yu-An Chung）等人在2016年提出語音詞向量（Audio Word2Vec），將一個不定長度（Variable-length）的詞之音訊由一固定維度的特徵向量（Fixed Dimensionality Vector）所表示 [2]。

與詞向量（Word2Vec）相同，語音詞向量的訓練過程不需要任何標註，為非

doi:10.6342/NTU201801733

督導式學習。然而在詞向量中所使用的神經網路模型並無法套用在語音的輸入上面，因為語音是由一段不定長度之音訊所組成，並非如同文字一樣可以使用獨一餘零向量來表示每個詞。為了符合音訊之不定長度的特性，鍾氏（Yu-An Chung）等人使用序列對序列自動編碼器（Sequence-to-Sequence Autoencoder）來訓練語音詞向量 [36]。圖2.11為語音詞向量之訓練過程。圖中輸入音訊$\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_4$的長度為4，輸入一基於遞迴式類神經網路的編碼器後，取編碼器在最後一個時間點的輸出，也就是第4個輸出作為語音詞向量$\mathbf{e}$。此語音詞向量接著會被輸入一同樣基於遞迴式類神經網路的解碼器進行反向解碼，目標是要使解碼器能夠完整重建輸入音訊。由於輸入音訊之重建只仰賴語音詞向量$\mathbf{e}$的資訊，因此編碼器必須將輸入音訊有效壓縮成特徵向量，才能使此向量含有足夠資訊完整重建輸入音訊。訓練此序列對序列自動編碼器的減損函數為計算解碼器的輸出$\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, ...\hat{\mathbf{x}}_4$與輸入音訊間之均方差而得，如式2.34。

$$loss(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{T}\sum_{t}^{T} \left\|\mathbf{x_t} - \hat{\mathbf{x}}_t\right\|^2, \tag{2.34}$$

其中T為輸入音訊之長度，在上述例子中$T = 4$。

鍾氏（Yu-An Chung）等人發現使用前述模型所訓練之語音詞向量含有輸入音訊之音素結構（Phonetic Structure）資訊 [2]。由於輸入的音訊為詞而每個詞都具有自已特有的音素結構，機器能夠從大量的音訊中學習出輸入語音中音素的結構並將其用語音詞向量表示。鍾氏（Yu-An Chung）等人發現具有相似音素結構之詞其對應之語音詞向量會具有較高之餘弦相似度（Cosine Similarity）。對於兩個由不同音素所組成的詞之語音，其音素結構的差異可由此二語音之音素序列（Phoneme Sequence）間的編輯距離（Edit Distance）所表示：音素序列間的編輯距離越大表示此二音訊之音素結構差異越大。表2.1表示兩個詞音訊之音素序列間

doi:10.6342/NTU201801733

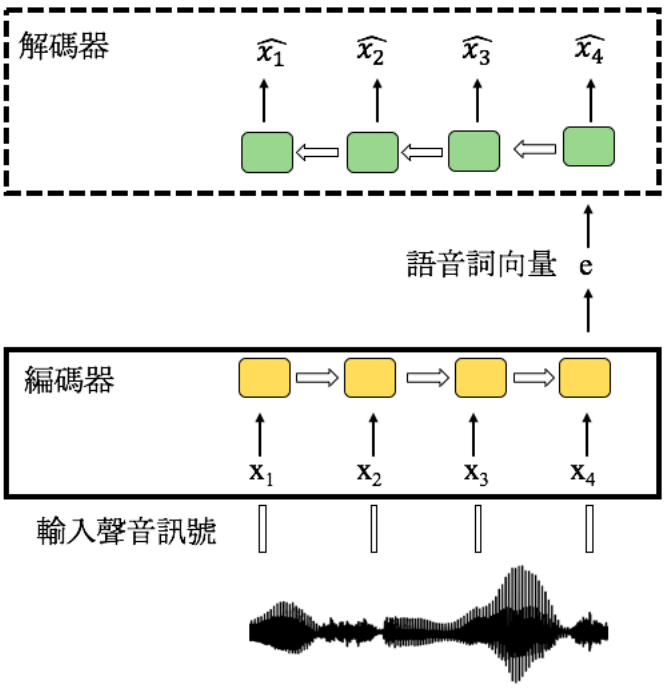


圖 2.11: 使用序列對序列式自動編碼器所抽取之語音詞向量

的編輯距離與其語音詞向量的餘弦相似度的關係。由表可以看出當兩個詞之音素結構越相近（其音素序列間的編輯距離越小），其語音詞向量間的餘弦相似度越高。

語音詞向量的特點除了音素結構相近者具有較高之餘弦相似度外，與詞向量相呼應的是在音素結構上也被發現具有向量運算的特性。在詞向量中，不同語意被包含在向量空間中的不同方向上（如每個國家的名稱往相同方向移動後便可得到其對應首都名稱） [54]。 而在語音詞向量中，鍾氏（Yu-An Chung）等人發現此特性同樣能夠反應在音素結構上。圖2.12所表示的是一個使用語音詞向量進行向量運算之例子：當輸入聲音訊號的第一個音素由f轉變為n時，可以看到投射在二維平面上的語音詞向量皆往相同方向移動。此種特性為語音詞向量所攜帶之音素結構提供了極佳的可解釋性（Interpretability）。

帶有音素結構之語音詞向量具有顯而易見之優點：將一段不定長度之音訊改

doi:10.6342/NTU201801733

音素序列間之編輯距離	平均餘弦相似度
0 (兩音訊為同一詞之語音)	0.4847
1	0.4016
2	0.2674
3	0.0835
4	0.0255
5 或以上	0.0051

表 2.1: 兩個詞音訊之音素序列間的編輯距離與其語音詞向量的餘弦相似度的關係。當兩個詞之音素結構越相近（其音素序列間的編輯距離越小），其語音詞向量間的餘弦相似度越高。取自參考文獻 [2]

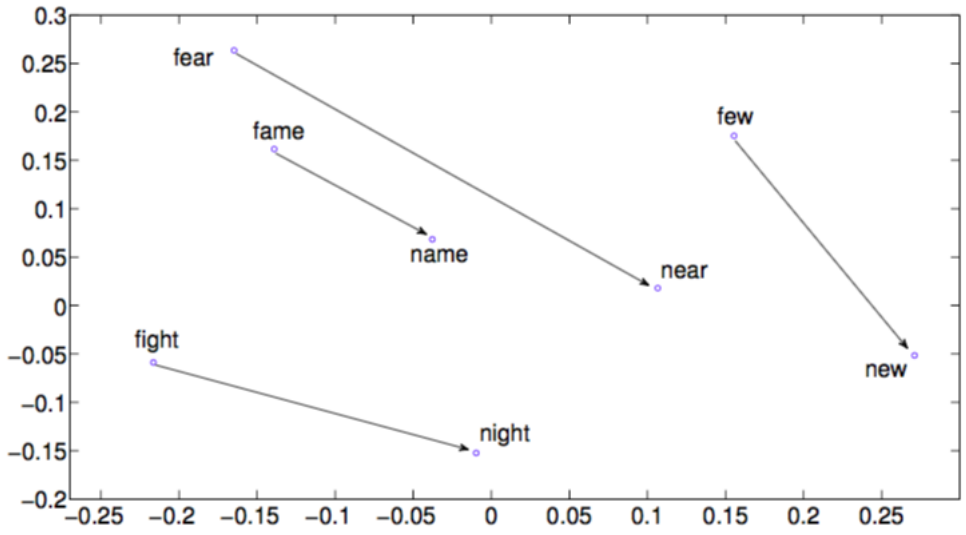


圖 2.12: 將100維之語音詞向量降維至2維平面上。當第一個音素由f轉變為n時，每個語音詞向量皆往相同方向移動。取自參考文獻 [2]

doi:10.6342/NTU201801733

口述語彙偵測方法	平均準確平均值
動態時間校準	0.181
語音詞向量	0.278
使用除噪型自動編碼器所訓練之語音詞向量	0.302

表 2.2: 比較語音詞向量與動態時間校準在口述語彙偵測上之效能。語音詞向量效能比動態時間校準要出色許多。另外使用除噪型自動編碼器來訓練語音詞向量可以更進一步提升效能。取自參考文獻 [2]

用一固定維度特徵向量表示，不論在記憶體的用量或是在後續應用的計算量上都能有顯著的降低。另外此向量為了要能夠有效壓縮音訊，機器會學習出音訊中真正具有重要意義之資訊而忽略掉音訊中之雜訊，進而獲得較佳之效能。鍾氏（Yu-An Chung）等人首先將語音詞向量應用於按例查詢（Query-by-Example, QbE）下的口述語彙偵測（Spoken Term Detection）上。在他們的設定中，語音查詢指令（Spoken Query）與語音文件（Spoken Document）皆為詞之音訊。在線下（Offline）作業時，每個語音文件會被壓縮為一個語音詞向量。在線上（Online）作業時，每當系統收到一個語音查詢指令，其輸入音訊會被轉換成為一語音詞向量，並且與語音文件之語音詞向量計算餘弦相似度作為其相關分數（Relevance Score）。鍾氏等人發現使用語音詞向量進行口述語彙偵測時，其效能比傳統的動態時間校準（Dynamic Time Warpping, DTW）要出色許多，其實驗結果如表2.2。不僅如此，使用更進階的除噪型自動編碼器（Denoising Autoencoder）來訓練語音詞向量時，可以更進一步提升效能 [20]。不只效能優異，在線上作業時的計算量方面也比動態時間校準要低上許多 [55]。

鍾氏（Yu-An Chung）等人所發表之語音詞向量的輸入音訊為以詞為單位之

doi:10.6342/NTU201801733

語音訊號。然而在非督導式學習的框架下，輸入語音通常為語句且詞邊界為不可得之資訊。因此在完整的非督導式學習框架下，機器還需要能夠從輸入語句中自動判斷出詞邊界，進而將輸入語句以語音詞向量序列表示。在後續段落中，本論文將介紹如何完成這件事。

doi:10.6342/NTU201801733

第 三 章　門限激發訊號與音素邊界之關聯分析

若要機器自動將輸入語句以詞為單位做分段來產生語音詞向量，機器需要能掌握語句中之詞邊界。語音中詞邊界與音素邊界密切相關，在本章節中將介紹一個存在於遞迴式類神經網路內部與音素邊界具有強烈關聯之訊號：門限激發訊號。非督導式學習下的遞迴式類神經網路模型之門限激發訊號能夠提供語音中音素邊界之資訊，無疑是一個能讓機器在無任何標註的情形下學習為語句做有意義分段之重要訊息。

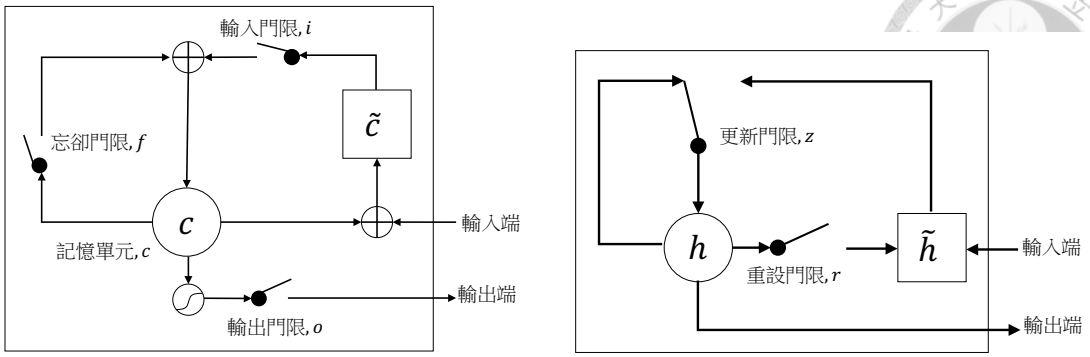
3.1　具門限機制之遞迴式類神經網路

具門限（Gate）機制之遞迴式類神經網路（Gated Recurrent Neural Networks）利用門限避免了梯度消失（Gradient Vanishing）的問題 [34]，因此能夠擁有比普通遞迴式類神經網路 [5]還要穩定而優異的表現。 長短期記憶神經網路（Long Short-term Memory Networks, LSTM） [7]以及門限遞迴單元（Gated Recurrent Unit, GRU） [35]為具門限機制之遞迴式類神經網路中最被廣泛使用的兩種結構，他們的神經元結構如圖3.1。

處理時間點t資訊時的長短期類神經網路的數學模型可以式3.1～式3.6所表示，分別表示忘卻門限激發訊號$\mathbf{f}_t$、輸入門限激發訊號$\mathbf{i}_t$、候選隱藏狀態$\tilde{\mathbf{c}}_t$、隱藏狀態$\mathbf{c}_t$、輸出門限激發訊號$\mathbf{o}_t$以及神經元輸出$\mathbf{h}_t$：

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \tag{3.1}$$

doi:10.6342/NTU201801733



(a) 長短期記憶神經網路單元　　(b) 門限遞迴單元

圖 3.1: 遞迴式類神經網路神經單元

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \tag{3.2}$$

$$\widetilde{\mathbf{c}_t} = tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c) \tag{3.3}$$

$$\mathbf{c}_t = \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t\widetilde{\mathbf{c}_t} \tag{3.4}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o) \tag{3.5}$$

$$\mathbf{h}_t = \mathbf{o}_t tanh(\mathbf{c}_t) \tag{3.6}$$

其中$\mathbf{U}$為運算過程中考量隱藏狀態之權重。$\mathbf{W}$、$\mathbf{U}$與$\mathbf{b}$之各下標則分別表示在計算各門限，隱藏狀態與神經網路輸出時所使用的不同參數，以下亦同。

另一方面，在門限遞迴單元中隱藏狀態$\mathbf{c}$即為神經元輸出$\mathbf{h}$，兩者在門限遞迴單元的數學式中皆以$\mathbf{h}$表示。式3.7～式3.10為門限遞迴單元運算時的數學式，分別代表更新門限訊號$\mathbf{u}_t$、候選隱藏狀態$\tilde{\mathbf{h}}_t$、重設門限訊號$\mathbf{r}_t$和隱藏狀態$\mathbf{h}_t$。相較於長短期記憶類神經網路，門限遞迴單元少了一個門限，參數量上少了許多也因此可被視為簡易版的長短期記憶類神經網路。在本論文中，所謂的門限激發訊號（Gate Activation Signals, GAS）乃指由門限所輸出的訊號向量。在長短期記憶類神經網路中，各門限激發訊號分別由式3.1，3.2以及3.5計算得之。而在門限遞迴

doi:10.6342/NTU201801733

單元則是由式3.7, 3.9計算得之。

$$\mathbf{z}_t = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{3.7}$$

$$\widetilde{\mathbf{h}}_t = tanh(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \tag{3.8}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1} + \mathbf{b}_r) \tag{3.9}$$

$$\mathbf{h}_t = (\mathbf{1} - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t\widetilde{\mathbf{h}}_t \tag{3.10}$$

其中⊙為以單元為單位之矩陣相乘（Element-wise Multiplication）。

觀察上述式子，可以知道兩種類神經網路神經單元的門限各有其功能。忘卻門限和更新門限控制神經單元中儲存的記憶是否需要被捨棄；輸入門限和重設門限控制能夠進入神經單元元件的輸入訊號大小。而長短期記憶類神經網路的輸出門限則是控制將多少比例的隱藏狀態輸出，門限遞迴單元則無此門限，將神經單元的內容完全輸出 [35]。

3.2 門限激發訊號與音素邊界

目前學界已經有許多學者針對遞迴式類神經網路的門限激發訊號進行研究。李氏（Fei-Fei Li）等人在字母層級（Character Level）的語言模型上，對門限激發訊和隱藏狀態進行分析 [15]，發現遞迴式類神經網路的學習表現與這些訊號們的表現十分相關，可藉此瞭解類神經網路是如何學習的以及其學習極限。語音的研究上門限激發訊號也被發現具有重要的意義。在語音合成的研究中，吳氏（Zhizheng Wu）等人發現長短期記憶類神經網路的忘卻門限激發訊號的變化與音素邊界有強烈的關係存在 [3]，如圖3.2。然而語音合成為一督導式學習的框架，本論文繼續將語音中之門限激發訊號的探討從督導式學習延伸至非督導式學習的框架下。

doi:10.6342/NTU201801733

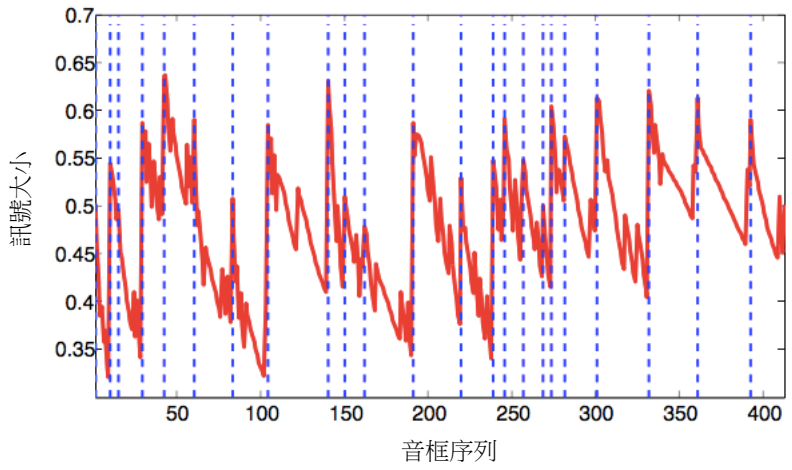


圖 3.2: 在語音合成中音素邊界與長短期類神經網路忘卻門限激發訊號之關係。圖中藍色虛線表示音素邊界之位置，而紅色曲線表示忘卻門限激發訊號。橫軸為音框序列，縱軸表示訊號的大小。取自參考文獻 [3]

3.2.1 模型概述

本章中所使用的模型為基於遞迴式類神經網路的自動編碼器，訓練方式是使用序列標註式（Sequence Labeling）訓練法。模型如圖3.3。在每個時間點t，輸入訊號$\mathbf{x}_t$會通過先通過編碼器（Encoder）中的全連接層（Fully Connected Layer，FC），接著進入編碼器中的遞迴式類神經網路。遞迴式類神經網路的輸出即被視為此輸入訊號的編碼並被送入解碼器（Decoder）進行解碼。解碼器為編碼器的對稱結構，編碼先被送入遞迴式類神經網路再進入全連接層。全連結層的輸出通過一個線性轉換得到經解碼器還原後的輸入訊號$\hat{\mathbf{x}}_t$。此模型中全連接層的激發函數我們使用整流線性單元。我們希望每個遞迴式類神經網路單元在產生編碼及解碼時不要依賴彼此，藉此學習到概括化資料的能力，因此在訓練遞迴式類神經網路時使用丟棄演算法輔助 [56]。在本章的實驗中，我們的全連接層使用64個神經元，遞迴式類神經網路使用32個神經元。訓練模型的方法使用慣量演算法中的ADAM訓練法 [32]，初始的學習率設定為0.0008。

doi:10.6342/NTU201801733

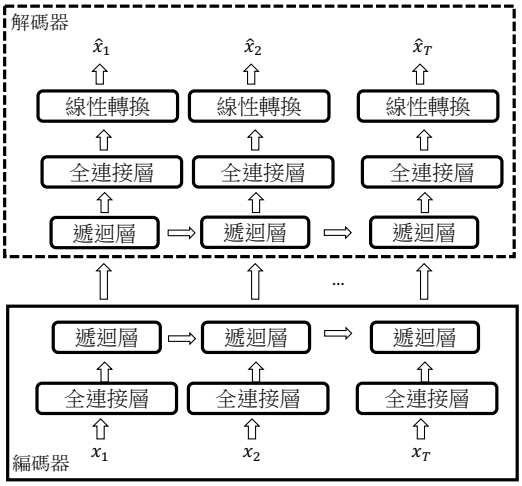


圖 3.3: 使用序列標註式訓練的自動編碼器，由一般類神經網路的全連接層（Fully Connected Layer），遞迴式類神經網路的遞迴層與線性轉換（Linear Transform）所組成。

3.2.2 實驗設計

本章實驗分為兩個階段，分別為第一階段為預備實驗及第二階段的音素切割實驗。第一階段的預備實驗之目的為定性地比較不同的門限激發訊號與音素邊界的關聯。觀察在非督導式學習的框架下，門限激發訊號與輸入語音之音素邊界是否具有關聯性。第二階段的音素切割實驗為將門限激發訊號與音素邊界關聯的量化實驗。

在兩階段的實驗中，我們使用TIMIT作為本章的實驗語料。雖然本實驗的模型訓練為非督導式學習，訓練模型過程並無使用任何關於音素邊界的標註，我們依舊將訓練模型的資料和評估模型效能的資料分開。我們使用TIMIT訓練集中的3696個語句來訓練模型，並且在測試集上的192個語句進行效能評估 [45]。使用的語音特徵為梅爾倒頻譜係數（Mel-frequency Cepstral Coefficients, MFCCs）並且加上其一階與二階導數，計算出總計39維度的特徵向量，接著使用以語句為單位

的倒譜頻平均變異數正規化（Utterance-wise Ceptral Mean Variance Normalization, CMVN），調整一個語句中所抽出的語音特徵們各維度數值，使其平均為0，變異數為1。正規化能夠降低不同語句在錄音時環境不同所造成的差異性，將不同語句都能投射到一個共同的輸入空間上供類神經網路模型進行訓練。

3.3 預備實驗

3.3.1 門限激發訊號與門限激發訊號均值

我們首先定義本論文中所使用的門限激發訊號為何。在時間點t時，一層具有J個神經元的遞迴式類神經網路，每一個神經元的某一種門限皆會輸出一個實數g_t。因此針對某一種門限，比方說長短期記憶類神經網路的忘卻門限，在時間點t便可獲得一個J維的向量，$\mathbf{g}_t = [g_1, g_2, ...g_J]^T$，我們定義此向量為忘卻門限的門限激發訊號，其中任一維度的值為某一神經元的忘卻門限之輸出值。

在預備實驗中，我們首先對於非督導式架構下的門限激發函數與音素邊界進行定性的比較。如同吳氏（Zhizheng Wu）之研究 [3]，我們將時間點t時的門限激發訊號之所有維度的訊號計算其平均值後得到門限激發訊號均值$\bar{g}_t$，觀察其與音素邊界間的關聯。本論文中的門限激發訊號均取自於編碼器的遞迴層。

3.3.2 實驗結果

針對同一段語音，將長短期類神經網路和門限遞迴單元中的門限激發訊號均值與音素邊界作圖，可以得到如圖3.4的結果。圖3.4(a)(b)(c)為長短期記憶類神經網路的門限激發訊號，分別屬於忘卻門限，輸入門限和輸出門限。而圖3.4(d)(e)則為門限遞迴單元的門限激發訊號，分別為更新門限和重設門限。觀察圖中不同門限激

doi:10.6342/NTU201801733

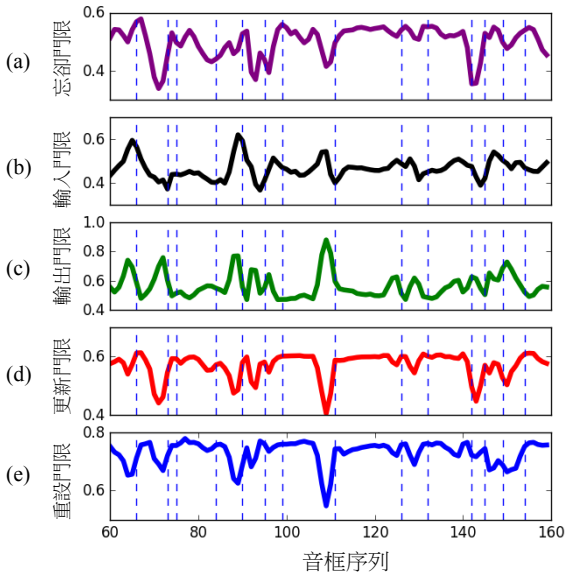


圖 3.4: 不同門限激發函數與音素邊界之關係。橫軸為音框序列，縱軸為門限激發訊號的平均值$\bar{g}_t$。藍色虛線表示音素邊界。

發訊號的趨勢，可以發現長短期記憶類神經網路的輸出門限激發訊號與輸入門限激發訊號的趨勢明顯與其他三個不同，呈現反相的情形。忘卻門限的門限激發均值的變化比更新門限和重設門限要來得劇烈。更新門限和重設門限的們限激發均值趨勢十分相似，但更新門限在某些時間點的變化幅度比重設門限要來得大。

由吳氏（Zhizheng Wu）研究指出 [3]，忘卻門限為長短期記憶類神經網路中與時間訊息最息息相關的門限，我們觀察圖3.4(a)，可以發現忘卻門限激發訊號均值的陡升之處與音素邊界（圖中藍色虛線處）具有强烈的關聯性，此現象與吳式在語音合成上所得到的結果相符，可以想像是在由一個音素切換到另一音素之際，前一個音素的相關資訊需要適度出清，轉而容納下一個音素的相關資訊，忘卻門限激發訊號因而需要變大。而和忘卻門限同樣控制隱藏狀態中儲存的資訊是否該被捨棄的更新門限，其門限激發訊號均值具有與忘卻門限相同的趨勢。有趣的是，我們觀察到皆為控制輸入訊號比例的輸入門限和重設門限，它們竟擁有近

doi:10.6342/NTU201801733

乎反相的門限激發訊號均值。可能原因為在門限遞迴單元的架構中（如圖3.1(b)）重設門限能夠掌握隱藏狀態中的資訊，因此讓兩者的門限激發訊號產生如此不一樣的特性 [35]。

3.3.3 門限激發訊號均值變化量

藉由上述觀察，我們定義門限激發訊號均值變化量（Difference GAS）$\Delta\bar{g}_t$作為觀察對象。定義時間點t時的門限激發訊號均值變化量如式3.11。我們依據門限激發均值是否產生陡升之情形來判斷是否為音素邊界。

$$\Delta\bar{g}_t = \bar{g}_{t+1} - \bar{g}_t \tag{3.11}$$

除了門限激發均值變化量，我們也可以觀察每一個遞迴式類神經網路的神經元的門限輸出變化量，作為更進一步之分析，如下式：

$$\Delta\bar{g}_t^j = \bar{g}_{t+1}^j - \bar{g}_t^j \tag{3.12}$$

其中上標j表示第j個遞迴式類神經網路神經元。

圖3.5描繪長短期類神經網路中忘卻門限之門限激發訊號均值變化量（式3.11中的$\Delta\bar{g}_t$以及式3.12中的$\Delta\bar{g}_t^j$）與音素邊界的關聯。由圖3.5(a)可以看出變化量大之處與音素邊界具有高度相關。另外圖3.5(b)表示各個神經元的門限輸出變化量與音素邊界的關聯（為求圖片清晰，以其中8個神經元做為代表）。在多數神經元產生的門限輸出變化劇烈之處與音素邊界具有高度相關。換言之，門限輸出變化量與音素邊界間的關聯存在於所有神經元而非少數的特定神經元 。

doi:10.6342/NTU201801733

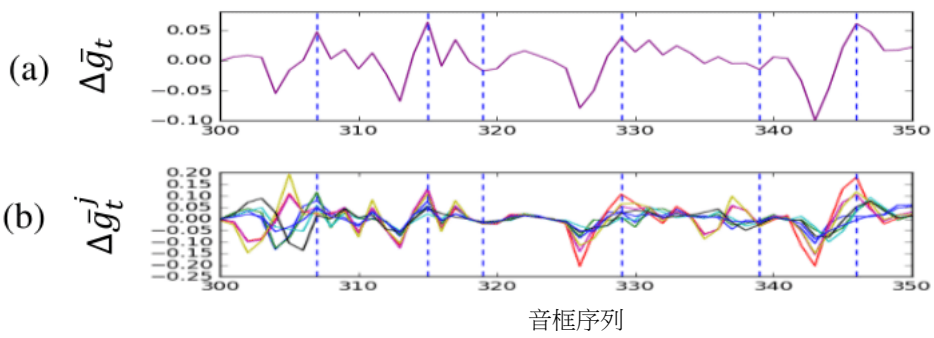


圖 3.5: (a)長短期記憶類神經網路中忘卻門限之門限激發訊號均值變化量（式3.11中的$\Delta\bar{g}_t$），(b)各神經元的門限輸出變化量（式3.12中的$\Delta\bar{g}_t^j$）與音素邊界之關係。橫軸為音框序列。藍色虛線表示音素邊界。

3.4 音素切割實驗

3.4.1 門限激發訊號在音素切割之應用

由預備實驗我們發現門限激發訊號均值變化量與音素邊界具有高度相關，然而此相關的程度為何我們仍需透過實驗來量化它，我們選用音素邊界切割實驗來量化其關聯。我們使用門限激發均值變化量（式3.11）作為判斷音素邊界的指標分數。若時間點t時的門限變化均值高於我們所選定的閾值δ（$\Delta\bar{g}_t > \delta$），而且為局部最大值時（$\Delta\bar{g}_t > \Delta\bar{g}_{t-1}$且$\Delta\bar{g}_t > \Delta\bar{g}_{t+1}$），我們便挑選時間$t$為我們認定的音素邊界 [45]。

3.4.2 遞迴式預測模型

遞迴式類神經網路已經有前人將其應用於非督導式的音素切割上，稱為遞迴式預測模型（Recurrent Predictor Model, RPM）。此種模型之訓練目標為預測下一個

doi:10.6342/NTU201801733

時間點的語音特徵。在每個時間點t，遞迴式預測模型會根據已經輸入的語音特徵：$\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_t$，來預測下一個時間點的語音特徵，$\mathbf{x}_{t+1}$。減損函如式3.13：

$$loss = \frac{1}{T-1}\sum_{t}^{T-1} \|\mathbf{x}_{t+1} - \mathit{RPM}(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t)\|^2 \tag{3.13}$$

此種模型訓練法希望能夠藉由預測下一個時間點的資訊使模型能夠捕捉到輸入語音的特性 [57] [45]。遞迴式預測模型在音素切割實驗的應用為尋找最難預測下一個語音特徵的時間點t，也就是遞迴式預測模型產生巨大預測錯誤的時間點。由於處於音素邊界的語音特徵之間的連結並不如單一音素內語音特徵來的強，因此在音素邊界變容易產生預測錯誤。遞迴式預測模型邊利用這個錯誤訊號進行音素邊界切割，也因此此模型在音素切割的應用上也被稱作錯誤訊號模型（Error Signal Model）） [45]。

在進行音素邊界切割時，遞迴式預測模型使用式3.14計算時間點t時的錯誤訊號E_t作為判斷音素邊界的指標分數。如同前一節使用門限激發均值變化量偵測音素邊界的方法，若時間點t時的錯誤訊號高於一個選定的閾值δ（$E_t > \delta$），而且為局部最大值時（$E_t > E_{t-1}$且$E_t > E_{t+1}$），我們便挑選時間t為我們認定的音素邊界。

$$E_t = \|\mathbf{x}_{t+1} - \mathit{RPM}(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t)\|^2 \tag{3.14}$$

3.4.3 結合門限激發訊號之遞迴式預測模型

門限激發訊號存在於所有使用遞迴式類神經網路的模型中，因此遞迴式預測模型中也存在此訊號，所以我們可以在不增加任何參數的情況下使用門限激發訊號來作為額外的資訊。結合式3.14與式3.11，我們可以使用一介於0至1之間的權重w來

doi:10.6342/NTU201801733

進行線性內插，獲得一結合遞迴式預測模型與門限激發訊號兩者資訊的指標分數I_t：

$$I_t = (1-w)E_t + w\Delta\bar{g}_t \tag{3.15}$$

3.4.4 效能評估

如同章節2.3所述，音訊的切割通常使用F1分數（F1 Score）來作為效能評估。然而近年來有些學者認為F1分數並不適合作為音素邊界切割的效能評估方法 [58]。學者發現F1分數在音素邊界切割的應用上會偏好高召回率的切割結果，造成過度切割（Over Segmentation）現象，因此無法有效反映切割結果之優劣。相較於使用準確率（Precison）與召回率（Recall）之調和平均所計算之F1分數，另一種較為精細評估切割結果的方式為R值（R-value） [58]，同樣使用準確率（Precison）與召回率（Recall）進行計算，其計算方式如下：

$$OS = (\frac{Recall}{Precision} - 1) \tag{3.16}$$

$$\textit{R-value} = 1 - \frac{\sqrt{(1-Recall)^2 + OS^2} + |\frac{Recall-1-OS}{\sqrt{2}}|}{2} \tag{3.17}$$

式3.16藉由召回率與準確率之比值來衡量過度切割情形的嚴重程度，在計算R值時會懲罰有過度切割情形的切割結果，藉此避免了F1分數所具有的偏差（Bias）。

我們使用週期預測模型（Periodic Predictor）在音素切割的實驗結果來說明為何為何R值比F1分數更為適合作為效能評估之方式。週期預測模型為一週期性地認定音素邊界的模型：假設一個週期為20毫秒之週期預測模，其每隔20毫秒便認

doi:10.6342/NTU201801733

定一個音素邊界。此種模型在操作上並不考慮輸入訊號之資訊，也不具備任何語音知識。在我們的實驗中，若使用F1分數對週期預測模型的音素切割結果進行評估，一個週期為40毫秒之週期預測模型可以得到71.07之F1分數。相對地，文獻中針對輸入語音的訊號進行分析所提出的方法也只獲得73的F1分數 [43]。當我們對週期預測模型的切割結果進行分析，發現準確率只有0.5513，而召回率為0.9999。此種現象稱為過度切割現象（Oversegmentation Phenomenon） [58]，其原因為模型不斷的判斷輸入音框為音素邊界，藉此來提高召回率和F1分數 [58]。因此使用F1分數來評估切割結果時便會因過度切割現象而造成評量失準。

而在R值的評量過程中，會使用式 3.16來針對過度切割現象進行適當的處理，計算出之R值為一越高越好的評量分數。將前述之週期預測模型之切割結果的召回率與準確率換算為R值，得到0.3053，而前述文獻將中記載之實驗結果換算為R值，得到0.76 [43]，因此R值確實可以反應出音素切割結果之優劣。綜合以上討論，本章中使用R值來作為音素切割結果的效能評量。本章的實驗中由於音素切割所使用的閾值為一人為設定之超參數，不同閾值會產生不同的切割結果。本章實驗結果所回報之R值為該模型在不同閾值下之最佳的切割結果所計算出之R值。

3.4.5 不同門限實驗結果

在本節中，我們比較來自不同門限的門限激發訊號均值變化量的音素邊界切割效能。這些門限分別為：長短期記憶網路的忘卻門限，輸入門限和輸出門限以及門限遞迴單元的更新門限以及重設門限。表 3.1分別列出使用各門限之激發訊號作為音素切割的效能。實驗結果與文獻中對於各門限的闡述相吻合 [3] [35] [59]。長短期記憶網路的門限中，忘卻門限明顯地比另外兩個門限更能夠捕捉到輸入語音的

doi:10.6342/NTU201801733

在時間上的結構。對於具有時序性的序列式資料而言，此種結構表示所有時間點的輸入資料在時間上的關係。以語音而言，音素的邊界便是一種時間結構。另外文獻中的實驗結果顯示，能否捕捉到此種結構的訊息對於模型效能有很大的影響。前人在不同領域的實驗中對長短期記憶類神經網路的門限進行討論。透過移除部分門限並觀察模型效能的變化，藉此觀察各門限之重要性。不同的文獻中皆指出忘卻門限為三者中最重要之門限，對於模型效能表現有著十分重要的影響。相較之下，輸出門限的有無則是在不同文獻中一致性地對於模型效能並無產生顯著影響。

神經單元	門限	最佳R值
長短期記憶類神經網路	忘卻門限	0.7915
	輸入門限	0.7075
	輸出門限	0.6197
門限遞迴單元	更新門限	0.8254
	重設門限	0.7894

表 3.1: 不同門限之音素切割結果。忘卻門限與更新門限皆為決定記憶單元中的資訊是否應該被繼續保留，在訓練過程中會學習去捕捉輸入語音的在時序上的結構，因此在音素切割的實驗中具有比同神經單元中的其他門限更佳之效能

在門限遞迴單元方面，其與長短期記憶類神經網路的門限彼此具有對應關係。長短期記憶類神經網路中的忘卻門限對應到門限遞迴單元中的更新門限，兩者皆決定記憶單元中的資訊是否應該被繼續保留。長短期記憶類神經網路中的輸出門限在門限遞迴單元中並無類似機制，門限遞迴單元會將隱藏狀態中的資訊完全輸出而非使用門限機制控制輸出比例。最後，輸入門限和重設門限皆用於控制

doi:10.6342/NTU201801733

隱藏狀態與候選隱藏狀態間的訊號傳遞（在門限遞迴單元中，由於沒有輸出門限的機制，因此隱藏狀態與神經元輸出是相同的）。與長短期記憶類神經網路相同的是，最佳的效能表現是由更新記憶單元的更新門限所獲得。

在本節的實驗中，使用門限遞迴單元能夠獲得比使用長短期神經網路更佳的表現。雖然在模型的複雜度上，長短期記憶類神經網路較門限遞迴單元複雜，理應獲得更佳的效能表現，但本節的實驗結果與之相反。在訓練模型的實務上，具有較高複雜度的長短期記憶類神經網路也較難訓練，因此造成其在測試時的表現輸給了較為單純的門限遞迴單元模型。前人亦有針對長短期記憶類神經網路以及門限遞迴單元兩種架構進行比較，發現兩種架構在不同任務（Tasks）中的表現互有優劣，構造較為簡單的門限遞迴單元模型其效能表現並不必然劣於長短期記憶類神經網路 [35]。

3.4.6 不同模型實驗結果

除了比較不同門限間的效能，我們也將門限激發訊號與其他模型比較其音素切割的效能，藉以了解門限激發訊號與音素邊界間的關聯是否具有應用價值。我們使用前一節中具有最佳結果的門限遞迴單元自動編碼器架構，且本節後文中所提及之門限激發訊號均為更新門限之門限激發訊號。門限遞迴單元自動編碼器直接的比較對象為同樣基於遞迴式類神經網路的遞迴式預測模型。本節中的遞迴式預測模型具有兩種架構，第一種是只有兩層類神經網路的遞迴式預測模型，本節中稱為雙層遞迴式預測模型，其架構同門限遞迴單元自動編碼器中的編碼器。由於門限激發訊號在測試時所使用到的參數只有編碼器的參數，因此與雙層遞迴式預測模型在測試時的參數相同。第二種遞迴式預測模型則為具有四層類神經網路，其架構與門限遞迴單元自動編碼器相同，稱之為四層遞迴式預測模型。雖然在測

doi:10.6342/NTU201801733

試時四層遞迴式預測模型使用比門限遞迴單元自動編碼器還要多的參數，但他們在訓練時的參數是相同的，我們藉此比較訓練時的參數量是否會對結果產生影響。在這兩種遞迴式預測模型中，我們可以從其架構中的遞迴式類神經網路擷取門限激發訊號，將其與遞迴式預測模型本身之錯誤訊結合得到一綜合訊號（式3.15）。我們的實驗也藉此比較是否此門限激發激發訊號能夠增進遞迴式預測模型的效能。

除了上述基於遞迴式類神經網路的方法外，我們也與傳統的音素切割方法，階層聚合式分群法（Hierarchical Agglomerative Clustering, HAC）進行比較 [48] [49]。此方法旨在建立一棵分類樹，給定一閾值δ將訊號相近的輸入分成一群，藉此進行音訊切割。最後一個比較基準是週期預測模型，此模型所認定之音素邊界間隔為80毫秒（使用間隔80毫秒之週期預測模型能比使用其他間隔獲得更高之R值），為一完全不考慮輸入訊號的模型。除了原本的語料外，我們也將其加入SNR-6dB的白雜訊（White Noise），比較不同模型的抗噪程度及穩健性。

在進行音素切割時，選定不同的閾值各模型都會產生不同的切割結果，我們可以由這些切割結果畫出準確率-召回率曲線。圖3.6所表示的是所有模型在未加入白雜訊的音訊上，由不同閾值所產生的切割結果，可視作模型整體的效能表現。對單一模型而言，我們從這些結果中選取最高的R值來作為此模型在音素切割上的效能表現，實驗結果如表3.2。

從表中我們可以看出四層遞迴式預測模型具有最佳的效能表現，而門限遞迴單元自動編碼器表現次佳。 另外門限激發訊號能夠顯著地增進遞迴式預測模型的效能（表3.2列(a)比列(b)，列(c)比列(d)），尤其是在加入噪音的語料上，加入門限激發訊號的資訊可以讓遞迴式預測模型的表現更加穩健。

我們更近一步分析雙層遞迴式預測模型與門限遞迴單元自動編碼器在無噪狀

doi:10.6342/NTU201801733

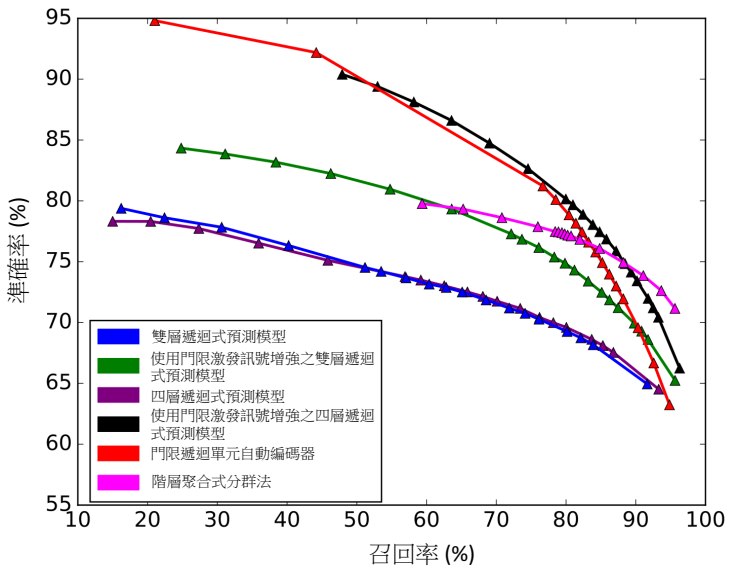


圖 3.6: 不同模型之的準確率-召回率曲線。各曲線為同一模型下選定不同閾值所產生之不同切割結果作圖而成，可視作模型整體的效能表現。曲線上不同的標記代表同一模型使用不同的閾值之結果

模型	無噪	SNR-6dB
（a）雙層遞迴式預測模型	0.7602	0.7370
（b）使用門限激發訊號增强之雙層遞迴式預測模型	0.7994	0.7916
（c）四層遞迴式預測模型	0.7610	0.7365
（d）使用門限激發訊號增强之四層遞迴式預測模型	0.8316	0.8154
（e）門限遞迴單元自動編碼器	0.8254	0.8122
（f）階層聚合式分群法	0.8161	0.8014
（g）週期預測模型	0.6217	0.6217

表 3.2: 不同模型間音素切割結果，表中數據為R值。使用門限激發訊號增强之四層遞迴式預測模型具有最佳的效能表現，而門限遞迴單元自動編碼器表現次佳。另外門限激發訊號能夠顯著地增進遞迴式預測模型的效能

doi:10.6342/NTU201801733

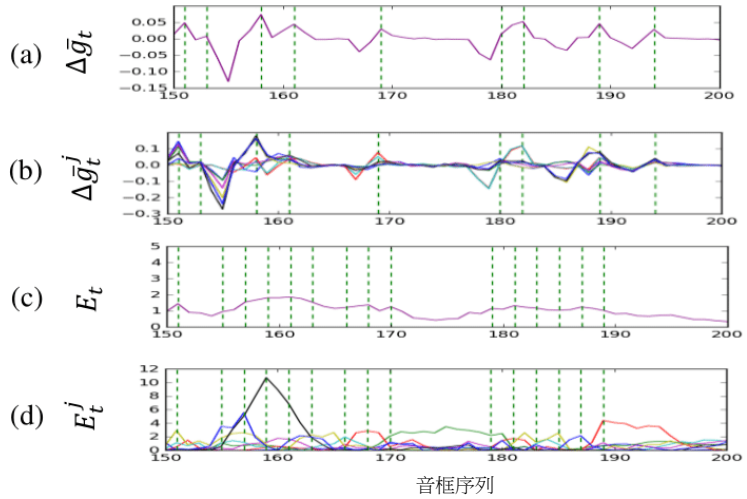


圖 3.7: 門限均值變化量$\Delta\bar{g}_t$，錯誤訊號E_t與其組成成分與音素切割結果之關係圖。圖中的綠色虛線表示機器所認定之音素邊界。錯誤訊號在每個時間點都有某一成分具有特別大的值，因此錯誤訊號E_t的曲線相較平滑，進而造成過度切割。門限均值變化量各成分數值變化相對而言十分一致，因此避免了機器有認定過多音素邊界的情形

況下有著最高R值的切割結果。針對門限遞迴單元自動編碼器，我們觀察門限均值變化量$\Delta\bar{g}_t$（式3.11）與各別神經元的門限變化量$\Delta\bar{g}_t^j$（式3.12）；針對雙層遞迴式預測模型，我們分析其產生的錯誤訊號E_t與錯誤訊號成分E_t^j。E_t^j所表示的是式3.13中第j維度上的方差（Squared Error）。我們將上述的資訊作圖成圖3.7，為求清晰，此圖只列出$J = 1, ...8$的成分。由圖我們發現錯誤訊號的曲線比門限均值變化量的曲線要平滑許多，造成機器認定過多的音素邊界。觀察錯誤訊號的成分後可以發現錯誤訊號在每個時間點都有某一成分具有特別大的值，使得錯誤訊號的整體曲線十分平滑，進而造成過度切割。另一方面，門限均值變化量各成分數值變化相對而言十分一致，因此避免了機器有認定過多音素邊界的情形。

此外增加模型的參數對於遞迴式預測模型而言並無幫助（表3.2列(c)比

doi:10.6342/NTU201801733

模型	捷克文	德文	法文
（a）雙層遞迴式預測模型	0.7466	0.8034	0.7782
（b）使用門限激發訊號增強之雙層遞迴式預測模型	0.7794	0.8179	0.8018
（c）四層遞迴式預測模型	0.7456	0.7987	0.7736
（d）使用門限激發訊號增強之四層遞迴式預測模型	0.7759	0.8090	0.8259
（e）門限遞迴單元自動編碼器	0.7709	0.8110	0.8107

表 3.3: 不同語言上之實驗結果，表中數據為R值。雖然同一模型之音素切割的效能在不同語言上各有差異，但是使用門限激發訊號之模型皆具有顯著優異之效能

列(a)）。由於具備更多參數的遞迴式預測模型在能夠將下一個音框的預測的更好，因此被用來作為音素邊界指標的錯誤訊號便會減弱，因此這個副作用削減了增加模型複雜度後所能帶來的好處。然而我們發現門限激發訊號不僅能提升遞迴式預測模型的效能，隨著模型複雜度的增加，門限激發訊號能帶來更大的效能提升表3.2列(b)比列(d)）。

為了檢驗前述門限激發訊號在其他語言上是否也能增進遞迴式預測模型之效能，我們也在其他語言上進行音素切割實驗，這些語言包含捷克文，德文與法文。實驗結果如表3.3。我們發現雖然同一模型之音素切割的效能在不同語言上各有差異，但是使用門限激發訊號之切割結果皆具有顯著優異之效能（表3.3列(b)(d)(e)比列(a)(c)），表示門限激發訊號確實在不同語言上都掌握了音訊中音素邊界的重要資訊。

3.5 本章總結

在本章節中介紹了一個位於遞迴式類神經網路內部之訊號：門限激發訊號。在一

doi:10.6342/NTU201801733

個使用非督導式學習的遞迴式類神經網路模型中，此訊號與輸入語句中之音素邊界具有强烈關聯。不同的門限所產生的門限激發訊號與音素邊界間關聯的强弱也有不同，透過實驗發現各門限訊號之强弱與前人研究相符：掌控更新模型記憶單元的門限（忘卻門限與更新門限）具有與音素邊界最强的關聯。另外在音素切割實驗中，門限激發訊號不僅比前人之方法具有更佳的强健性，也可以在不增加參數的情況下提昇前人方法的效能。後續章節將介紹如何結合門限激發訊號與語音詞向量，將輸入語句表示為語音詞向量序列。

doi:10.6342/NTU201801733

第 四 章　分段式語音詞向量之初步研究

在前一章節中我們介紹了一個在使用非督導式學習架構中與音素邊界息息相關的訊號：門限激發訊號，本章將介紹如何將其運用到詞切割上。在非督導式的學習框架下，音素邊界與詞邊界在並無意義上之不同，皆為語音中重複出現之模式，只是音素的長度較短而詞的長度較長。由於是非督導式學習，機器並不知道該邊界是音素邊界還是詞邊界，只知道其為一重複出現模式之邊界。因此當機器可以藉由聆聽含有許多詞之大量語音來學習出詞的重複模式，門限激發訊號同樣可以提供詞邊界之資訊。在本章節中我們介紹如何將門限激發訊號與語音詞向量結合，提出一個可以將語句以一語音詞向量序列表示之方法，我們稱之為分段式語音詞向量。

4.1　分段式序列對序列自動編碼器

4.1.1　分段式語音詞向量

在章節2.5.2中所介紹之語音詞向量雖然能夠透過非督導式學習所得到，但是輸入為詞之音訊。然而在非督導式學習的框架下，輸入機器之音訊通常為一語句，而語句中的詞邊界為未知資訊。因此在完整的非督導式學習框架下，機器還需要能夠從輸入語句中自動判斷出詞邊界。利用機器將輸入語句以語音詞向量序列表示，其中每個詞向量所代表的是語句中一特定詞的語音詞向量，我們將此法所產生之語音詞向量稱為分段式語音詞向量（Segmental Audio Word2Vec）。

doi:10.6342/NTU201801733

4.1.2 切割門限

我們在章節2.5.2所介紹之用來產生語音詞向量的序列對序列式自動編碼器中引入一個二元值的（Binary）切割門限（Segmentation Gate）ψ_t，此切割門限與該時間點編碼器的輸出$\hat{\mathbf{e}}_t$進行相乘運算。當該時間點為詞邊界時，此切割門限將會打開，其值為1，此時編碼器的輸出$\hat{\mathbf{e}}_t$將會作為輸入音訊的語音詞向量$\mathbf{e}_t$；反之，切割門限關閉時的值為0，因此編碼器的輸出為$\mathbf{0}$，如式4.1與式4.2。如何訓練切割門限判斷詞邊界則會在後面章節進行說明。

$$\psi_t = \begin{cases} 1, & \text{如果 t 為詞邊界} \\ 0, & \text{其他} \end{cases} \tag{4.1}$$

$$\mathbf{e}_t = \begin{cases} \mathbf{0}, & \text{如果 } \psi_t = 0 \\ \hat{\mathbf{e}}_t, & \text{其他} \end{cases} \tag{4.2}$$

4.1.3 重設機制

藉由切割門限，編碼器（Encoder）只有在詞邊界才會輸出語音詞向量。為了使所產生之每個語音詞向量只包含該對應詞的音訊資訊，在每次編碼器輸出語音詞向量後，編碼器之遞迴式類神經網路的狀態便會重設（Reset）至初始狀態。因此即使編碼器是使用具有記憶功能的遞迴式類神經網路進行編碼，該語音詞向量也只會含有目前音訊片段的資訊。第n個音訊片段的語音詞向量$\mathbf{e}_n$的產生可用式4.3表示：

$$\mathbf{e}_n = \mathbf{e}_{t_2} = \mathit{Encoder}(\mathbf{x}_{t_1}, \mathbf{x}_{t_1+1}, ...\mathbf{x}_{t_2}) \tag{4.3}$$

其中t_1和t_2分別表示第n個音訊片段的開始時間點與結束時間點。

doi:10.6342/NTU201801733

相同的道理，在解碼器（Decoder）解碼時，在使用不同語音詞向量對一段音訊做反向解碼前 [36]，解碼器的狀態也會重設至初始狀態，因此位於第n個音訊片段的時間點t，此時間點的重建音訊特徵$\hat{\mathbf{x}}_t$可以用式4.4表示。

$$\hat{\mathbf{x}}_t = Decoder(\hat{\mathbf{x}}_{t_2}, \hat{\mathbf{x}}_{t_2-1}, ...\hat{\mathbf{x}}_{t+1}, \mathbf{e}_n) \tag{4.4}$$

藉由此重設的機制，語句中各片段資訊不流通，此序列對序列式自動編碼器可視作同時進行不同區段的序列對序列式訓練，本文中將稱其為分段式序列對序列自動編碼器（Segmental Sequence-to-Sequence Autoencoder，SSAE）。

4.1.4 分段式序列對序列式訓練

圖4.1所表示的是一分段式序列對序列自動編碼器。輸入一長度為T之語句$[\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_T]$，切割門限在若干時間點打開，使用編碼器（圖中之方框ER）之輸出$\mathbf{e}$作為目前輸入音訊之語音詞向量。一長度為T之輸入語句便可由此轉化一長度為N之語音詞向量序列，解碼器（圖中之方框DR）再使用此向量序列重建出原本的輸入語句。前段所述之重設機制以含有斜線之箭頭表示。由於此重設機制，每個色塊間的資訊不流通，因此可視做在一語句內同時進行若干獨立的序列對序列式訓練。

4.2 端對端訓練下之分段式語音詞向量

4.2.1 端對端訓練

近年來深層學習崇尚端對端訓練（End-to-End Training），根據目標設計一減損函數後，便針對此減損函數調整模型參數 [60] [61]。以往一個完成複雜工作的

doi:10.6342/NTU201801733

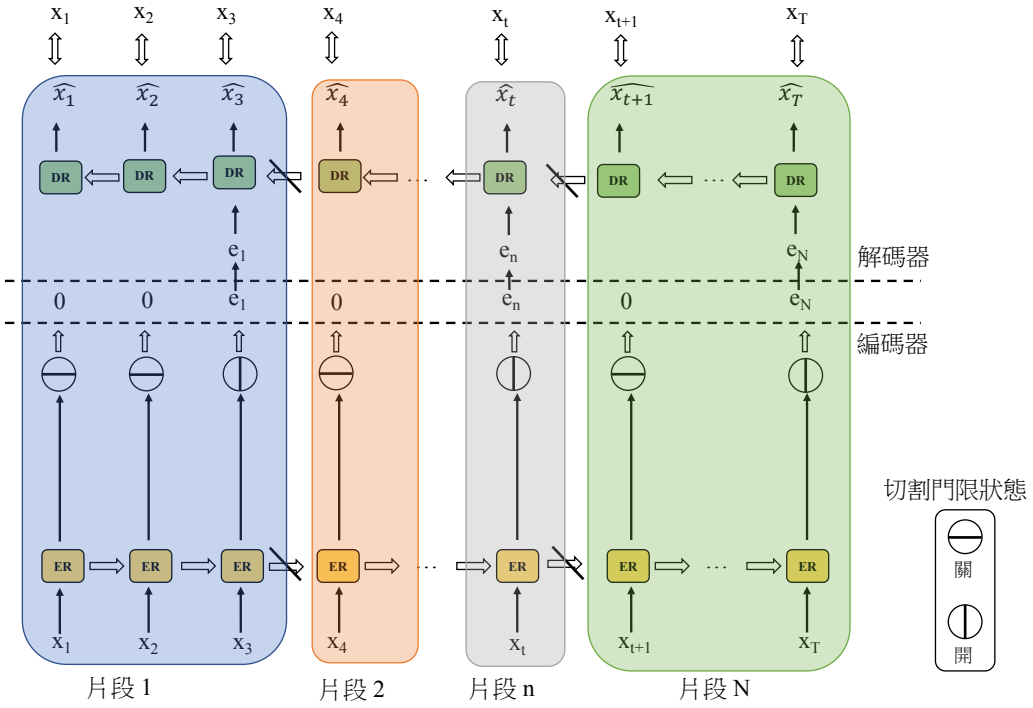


圖 4.1: 分段式序列對序列式自動編碼器。切割門限在若干時間點打開，使用編碼器（圖中之方框ER）之輸出**e**作為目前輸入音訊之語音詞向量。一長度為T之輸入語句便可由此轉化一長度為N之語音詞向量序列，解碼器（圖中之方框DR）再使用此向量序列重建出原本的輸入語句。重設機制以含有斜線之箭頭表示。由於此重設機制，每個色塊間的資訊不流通，因此可視做在一語句內同時進行若干獨立的序列對序列式訓練

模型需要仰賴事先訓練好各不同功能的子模組（Sub-Module），將眾多子模組組裝起來進而達成目標。然而在深層學習中，端對端訓練利用反向傳播演算法將所有子模組一起訓練，一起調整參數，減少子模的各自局部最佳化（Local Optimization）進而達到更佳的效能。端對端訓練的優點為直接針對與目標直接相關的減損函數來整體全面考量並訓練整個模型中的各個參數。

鑑於上述優點，本章後文將探討將端對端訓練應用於分段式語音詞向量之可能性。在這邊我們將章節3所發現之門限激發訊號應用於前節所述之切割門限中。我們在每個時間點t觀察編碼器的門限激發訊號均值$\bar{g}_t$，當發現其有陡升之情形，且陡升幅度超過一閾值δ，我們便認定此時間點為詞邊界。此種設定下，我們的切割門限為一單位階梯函數（Step Function），當門限激發訊號均值變化量$\Delta\bar{g}_t$超過閾值δ時輸出1，反之則輸出0：

$$\Delta\bar{g}_t = \bar{g}_t - \bar{g}_{t-1} \tag{4.5}$$

$$\psi_t = Step(\Delta\bar{g}_t - \delta) \tag{4.6}$$

4.2.2 直通評估器

在端對端訓練的精神下，我們希望閾值δ為一由模型自行決定之參數而非由人為設定，但δ屬於式4.6中單位階梯函數的參數且單位階梯函數為一不可微分之函數，因此一般的反向傳播演算法並不適用。在訓練需要處理不可微分問題的模型時，班氏（Yoshua Bengio）提出直通評估器（Straight Through Estimator）來解決其不可微分之問題 [62]。

圖4.2所表示的是一個直通評估器。直通評估器的精神在於在反向傳播時，將

doi:10.6342/NTU201801733

圖 4.2: 直通評估器。在順向傳遞時為單位階梯函數；在反向傳遞時，將單位階梯函數視作恆等函數

單位階梯函數視作恆等函數（Identity Function）來計算梯度，因此流至單位階梯函數的梯度將原封不動往下傳遞，藉此進行反向傳播演算法。因此切割門限之輸出ψ_t在順向傳遞與反向傳播時會被視作不同的運算，其數學式如下：

$$\psi_t = \begin{cases} Step(\Delta\bar{g}_t - \delta), & \text{順向傳遞} \\ Identity(\Delta\bar{g}_t - \delta) = \Delta\bar{g}_t - \delta, & \text{反向傳播} \end{cases} \tag{4.7}$$

4.2.3 減損函數設計

分段式語音詞向量之訓練目標如同語音詞向量，希望重建音訊與輸入音訊之間的均方差（Mean Squared Error）越小越好。但顯而易見的是，若只單純考慮重建音訊與輸入音訊之間的均方差，則切割門限會選擇在每個時間點都讓編碼器輸出語音詞向量。這與分段式語音詞向量的目標不符，因為我們希望模型所輸出之每個語音詞向量其代表的是一個語音詞。故我們需要將用來表示輸入語句之語音詞向量數量作限制。

在設計減損函數時，發揮端對端訓練的精神，直接將前述目標化成減損函數：

doi:10.6342/NTU201801733

$$loss = \frac{1}{T}\sum_{t}^{T}\{\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2 + \psi_t\} \tag{4.8}$$

其中切割門限的輸出ψ_t在所有時間點的總和等同於輸出之語音詞向量數目。

此減損函數精神為最小化重建錯誤的同時也最小化使用的語音詞向量數目，避免切割門限選擇在每個時間點都讓編碼器輸出語音詞向量。

4.3 實驗

4.3.1 實驗設計

本節實驗設計與章節3.2.2大致相同，我們使用TIMIT作為本章的實驗語料。使用的語音特徵為梅爾倒頻譜係數（Mel-frequency Cepstral Coefficients, MFCCs）並且加上其一階與二階導數，計算出總計39維度的特徵向量，接著使用語句倒譜頻平均變異數正規化（Utterance-wise Ceptral Mean Variance Normalization, CMVN），調整一個語句中所抽出的語音特徵各維度數值，使其平均為0，變異數為1。序列對序列自動編碼器之編碼器與解碼器各由一層100個神經元的長短期記憶類神經網路所組成 [2]。所有的參數更新使用慣量（Momentum）演算法中的ADAM訓練法進行更新 [32]。由於與音素相比詞為長度較長之音訊片段，因此在進行評估詞邊界切割效能時，會使用較大之容忍窗（Tolerance Window）來進行有效評估，本論文中詞切割之容忍窗大小設為40毫秒 [44]。

4.3.2 實驗結果與討論

圖4.3中顏色較淡之曲線為模型產生之所切割出之音訊的平均長度的趨勢。然而為了能更清楚顯示趨勢，我們將原始資料使用平滑處理（Smoothing）後繪製成深色

doi:10.6342/NTU201801733

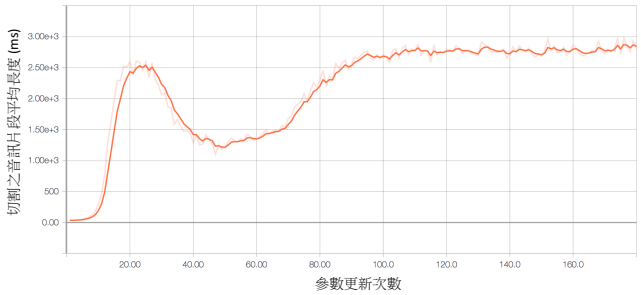


圖 4.3: 模型產生之所切割出之音訊的平均長度的趨勢。淡色曲線為原始資料，深色曲線為將資料經過平滑處理而繪製。模型所切割出之音訊的平均長度在訓練初期略有起伏，但是在中後期開始所切割出之音訊的平均長度約略等於一個語句長度，表示對於每一個語句只使用一個語音詞向量來代表

線條。平滑處理的方式為將每個時間點的原始資料與其鄰近之10個資料點的資料做平均。圖4.3之縱軸為所切割出之音訊的平均長度，單位為毫秒（ms），橫軸則為參數更新次數。從圖中可以看出模型所切割出之音訊的平均長度在訓練初期略有起伏，但是在中後期開始所切割出之音訊的平均長度約略等於一個語句長度，表示對於每一個語句只使用一個語音詞向量來代表。

為探究原因，我們同樣將閾值δ之趨勢化成圖4.4。可以發現閾值δ呈現一單調上升的趨勢。由於閾值δ不斷上升，因此模型在語句中的門限激發均值變化量$\Delta\bar{g}_t$始終無法超越閾值δ，因此無法在語句中產生語音詞向量，造成只能在句末產生語音詞向量來代表整個語句。

更深入分析造成閾值δ呈現一單調上升的趨勢的原因，我們將所使用之減損函數（式4.8）對閾值δ的進行微分，得到式4.9。可以看出若利用原有的減損函數對閾值δ進行梯度下降法更新，閾值δ只會往正的方向進行更新，因此只會遞增而呈現一單調上升的趨勢。

doi:10.6342/NTU201801733

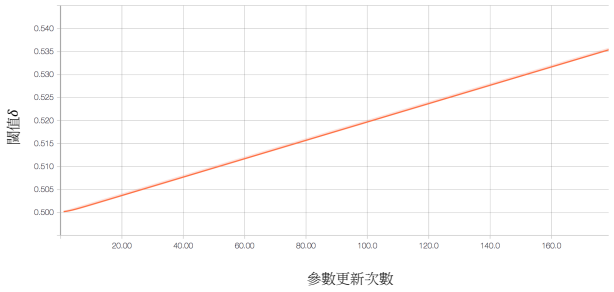


圖 4.4: 閾值δ之趨勢，呈現一單調上升的趨勢

$$\frac{\partial loss}{\partial \delta} = \frac{1}{T}\sum_{t}^{T}\frac{\partial \psi_t}{\partial \delta} = \frac{1}{T}\sum_{t}^{T}\frac{\partial(\Delta\bar{g}_t - \delta)}{\partial \delta} = -\frac{\partial \delta}{\partial \delta} = -1 \tag{4.9}$$

考慮上述原因後，我們將一針對閾值δ的控制調適（Regularization）項加入式4.8的減損函數中，將其改寫為式4.10。

$$loss = \frac{1}{T}\sum_{t}^{T}\{\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2 + \psi_t\} + \lambda\delta \tag{4.10}$$

其中λ為一表示控制調適强弱效果之超參數，在接下來的實驗中λ設為0.01。

在此減損函數下，模型的訓練除了需要考慮原本的訓練目標，也會盡量將閾值δ壓低。我們希望引入此控制調適（Regularization）項能夠解決閾值δ呈現一單調上升之問題。

引入此控制調適項後之實驗結果如圖4.5與圖4.6。由圖4.5可以發現，引入控制調適項後，模型所切割出之音訊的平均長度明顯小於尚未引入之前的長度，模型對於每一語句不再只使用一語音詞向量表示它。圖圖4.6也顯示引入控制調適項後閾值δ之趨勢也不再呈現單調遞增，其在不斷震盪後逐漸收斂至某一定值。因此引入控制調適項後確實解決了上述遇到之問題。另外我們也可以從圖4.5看出模型所切割出之音訊的平均長度在約380毫秒的長度間震盪，表示模型了解到其所切割出之音訊必須大約為此長度。

doi:10.6342/NTU201801733

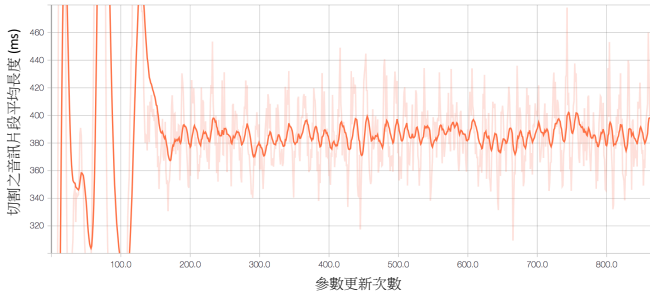


圖 4.5: 引入控制調適項後模型產生之所切割出之音訊的平均長度的趨勢。淡色曲線為原始資料，深色曲線為將資料經過平滑處理而繪製。模型所切割出之音訊的平均長度在約380毫秒的長度間震盪

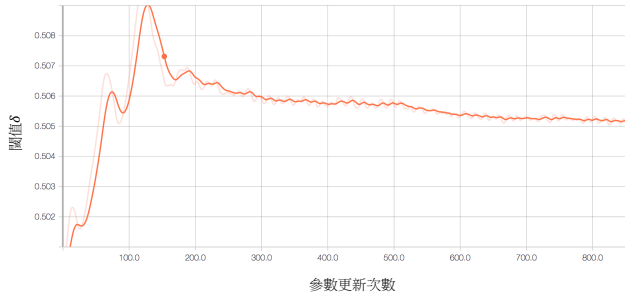


圖 4.6: 引入控制調適子後閾值δ之趨勢。淡色曲線為原始資料，深色曲線為將資料經過平滑處理而繪製。閾值δ不斷震盪後逐漸收斂至某一定值，不再呈現單調遞增

doi:10.6342/NTU201801733

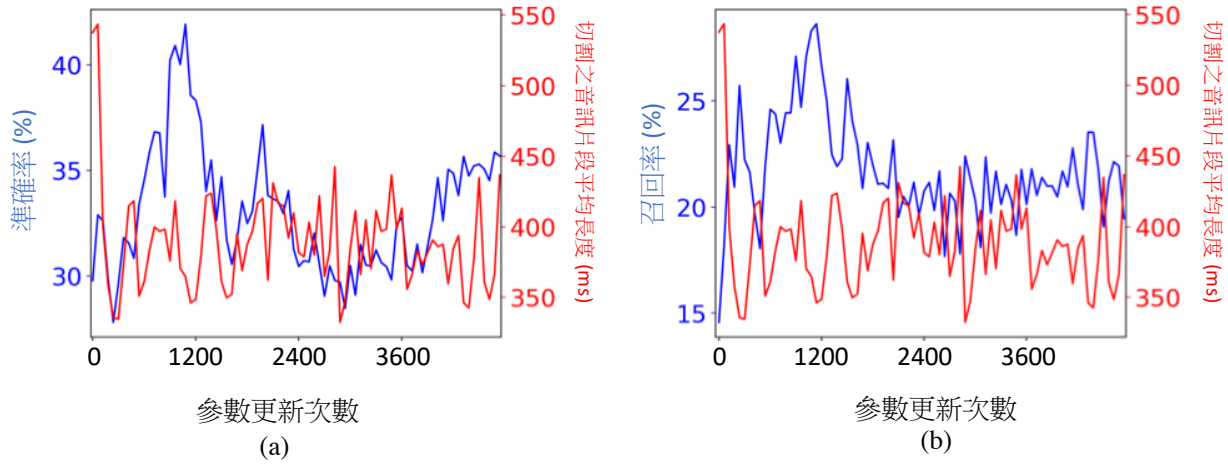


圖 4.7: 使用準確率（Precision）與召回率（Recall）所表示之詞切割效能曲線。詞切割的效能並無隨著訓練過程而增進

我們希望模型在了解到其所切出之音訊在長度上的限制後可以在此限制下找出最佳的詞切割點。因此我們進一步針對其詞切割的效能進行檢視，如圖4.7。圖中之橫軸為參數更新次數，縱軸有二曲線：紅色曲線為模型所切割出之音訊的平均長度（單位為毫秒），藍色曲線則為：圖(a)準確率（Precision）；圖(b)召回率（Recall）。從圖中我們發現詞切割的效能並無隨著訓練過程而增進。推測其原因為切割出之各音訊片段間資訊無法流通，因此在訓練時無法藉由前後片段的資訊來調整切割方式，進而無法提升詞切割之效能。故本章之構想基本上未能達成所設定的目標。

4.4 本章總結

本章介紹了分段式語音詞向量之概念與分段式序列對序列自動編碼器模型架構。分段式序列對序列自動編碼器的目標是在一語句中可以同時對多個片段進行序列

doi:10.6342/NTU201801733

對序列訓練，因此可以視為擴充版本之序列對序列自動編碼器，希望能將一輸入語句轉化為一語音詞向量序列。本章探討以端對端訓練之概念訓練此模型並使用直通評估器來解決減損函數中不可微分之問題。在本章實驗中探討減損函數之設計算產生之問題，並藉由修改減損函數改善。然而修改過後之減損函數依舊無法使詞切割效能隨著訓練過程而提升。事實上在處理具有不可微分問題時，除了直通評估器外的另一作法為強化學習。在下一章中我們將介紹如何使用強化學習來訓練分段式語音詞向量。

doi:10.6342/NTU201801733

第 五 章　基於强化學習之分段式語音詞向量

在訓練分段式語音詞向量時，我們必須將模型所產生之語音詞向量數量控制在合理的數量，其為一不可微分問題。在訓練需要處理不可微分問題的模型時，除了可以使用直通評估器進行端對端訓練，另一方法為使用强化學習（Reinforcement Learning）。由於强化式學習是代理人（Agent）透過獎勵（Reward）進而摸索出系統目標，而實作上獎勵的計算可以獨立於訓練模型參數的計算圖（Computational Graph）之外，因此强化式學習為一處理減損函數中具有不可微分項問題之替代做法。本章中我們將介紹如何使用强化學習來訓練分段式語音詞向量。

5.1　以强化學習訓練之分段式語音詞向量

從章節2.2中我們知道强化學習的應用情境為一代理人（Agent）需在一環境中（Environment）連續做出許多決策的問題。代理人，也就是强化學習模型，會藉由目前環境的狀態（State）與其策略（也就是其行為模式，Policy）作出一個動作（Action）來與環境互動並從環境獲得獎勵（Reward）。藉由不斷與環境互動並獲得不同的獎勵，使用進而摸索出系統目標。

我們將詞切割的問題規劃成一個代理人需要連續做出許多決策的問題，藉此使用强化學習來解決詞切割的問題。在我們的設定中，章節4.1.2中所描述之切割門限視為代理人，其行為模式（也就是策略π）由一遞迴式類神經網路所模擬。在每個時間點t ，切割門限都會根據給定的狀態 $\mathbf{s}_t$來執行一個動作a_t。在詞切割的問題中，切割門限只會輸出兩種動作，「切割」（Segment）或是「繼續」（Pass）。每當切割門限在時間點t所執行的動作為「切割」時，此時間點便會被

doi:10.6342/NTU201801733

强化學習元素	詞切割問題中之對應元素
狀態（State, s）	切割門限之輸入
動作（Action, a）	切割門限之輸出：「切割」或是「繼續」
策略（Policy, π）	由切割門限之遞迴式類神經網路所模擬之行為模式
獎勵（Rewards, r）	由分段式序列對序列自動編碼器中的編碼器與解碼器之表現所計算

表 5.1: 章節2.2中所介紹之强化學習的元素與本章中詞切割問題間的對應關係

視為一個詞邊界，切割門限便會打開並使用編碼器之輸出作為該聲音片段之語音詞向量。最後，我們設定分段式序列對序列自動編碼器中的編碼器與解碼器為環境，每當切割門限對語句做完詞切割之後，我們會藉由編碼器與解碼器的表現給予切割門限獎勵，使其學習到如何進行正確之詞切割。我們可以將章節2.2中所介紹之强化學習的元素與本章中詞切割問題間的對應關係製作成表5.1。

在這問題下使用一遞迴式類神經網路控制切割門限的最大好處為其能夠擁有輸入語句完整的資訊。分段式序列對序列自動編碼器中的編碼器與解碼器由於要避免所產生之語音詞向量混雜了1個以上的語音詞資訊，在語句中詞的分段點處皆需要使用重設機制將資訊清除，因此無法掌握整個語句之資訊。然而這邊所使用來控制切割門限之遞迴式類神經網路只用來決定語句中詞的分段點，並不參與語音詞向量之生產，因此不需要使用重設機制清除資訊，故能擁有輸入語句完整的資訊。

從章節2.2中可以了解，狀態的資訊含量若越豐富，則模型越能夠透過强化學習來學習到符合系統目標的動作。因此我們的狀態也是由許多不同的資訊串接而成，其中包含：

- 輸入的音訊特徵 $\mathbf{x}_t$

doi:10.6342/NTU201801733

- 門限激發訊號（Gate Activation Signal, $\mathbf{g}_t$）。在章節3.2中所發現之門限激發訊號與輸入語句的音訊邊界具有强烈關聯，可視作一輸入語音在時間上的訊號特徵。這裡的門限激發訊號取自一個預先訓練好的（Pre-trained）基於門限遞迴單元之自動編碼器中更新門限的訊號 [63]。

- 前一個時間點的執行動作 a_{t-1}

我們將時間點t時的狀態$\mathbf{s}_t$用下式定義為：

$$\mathbf{s}_t = [\mathbf{x}_t || \mathbf{g}_t || a_{t-1}] \tag{5.1}$$

其中||表示串接運算（Concatenate）。

控制切割門限的遞迴式類神經網路根據時間點t時的狀態$\mathbf{s}_t$ 輸出訊號$\mathbf{h}_t$，此訊號會經過一個由參數($\mathbf{W}^p$,$\mathbf{b}^p$) 所決定的線性轉換（Linear Transform），和軟性最大化（Softmax）的非線性轉換，得到時間點 t 時各動作的機率$\mathbf{p}_t$ 。此$\mathbf{p}_t$為一個2維的向量，兩個維度分別代表執行動作「切割」和動作「繼續」的機率。此過程可用式5.2 和式5.3 表示：

$$\mathbf{h}_t = \mathit{RNN}(\mathbf{s_1}, \mathbf{s_2}, ..., \mathbf{s_t}) \tag{5.2}$$

$$\mathbf{p}_t = [p_t^1, p_t^2] = \mathit{Softmax}(\mathbf{W}^p \mathbf{h}_t + \mathbf{b}^p) \tag{5.3}$$

其中p_t^1之上標1表示時間點t時第1種動作的機率。

如同章節2.2所述，為了探索各種動作的獎勵以求進行有效的强化學習，在訓練時的每個時間點t，切割門限會從由$\mathbf{p}_t$中的機率所形成的一個多項分佈（Multinomial）中取樣出該時間點所執行的動作a_t；而在測試時就直接選取機率

doi:10.6342/NTU201801733

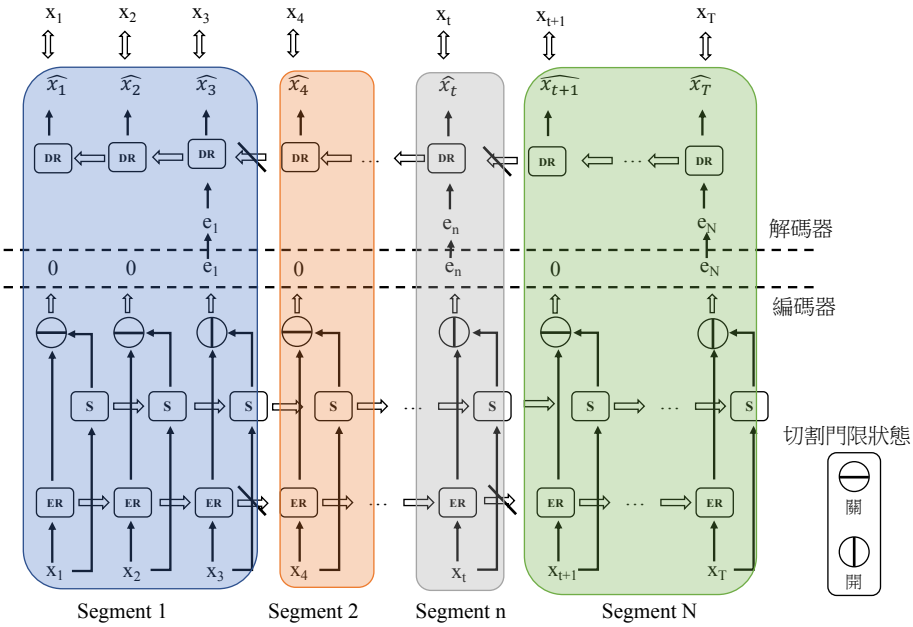


圖 5.1: 使用遞迴式類神經網路模擬之切割門限下的分段式序列對序列自動編碼器。與章節4.1中圖4.1架構大致相同，但切割門限改由一遞迴式類神經網路所控制（圖中之方框S）

最高的動作作為該時間點所執行的動作，並非經過取樣決定。上述之模型架構如圖5.1。與章節4.1中圖4.1最主要的不同的是本章中之切割門限是由一遞迴式類神經網路所控制（圖5.1中的方框S）。

5.2 訓練分段式語音詞向量之獎勵

5.2.1 獎勵設計

分段式語音詞向量之訓練目標如同語音詞向量，希望重建音訊與輸入音訊之間的均方差（Mean Squared Error）越小越好。但顯而易見的是，若只單純考慮重建音訊與輸入音訊之間的均方差，則切割門限會選擇在每個時間點都讓編碼器輸出語音詞向量。這與分段式語音詞向量的目標不符，因為我們希望模型所輸出之每個

語音詞向量其代表的是一個詞。故我們需要將用來表示輸入語句之語音詞向量數量作限制。

獎勵機制的目的是對於詞切割較好的語句給予較高的獎勵，反之則給予較低的獎勵。藉由獎勵，强化學習下的代理人，也就是分段式序列對序列自動編碼器中的切割門限，就可以學習到如何將一個語句進行正確的詞切割。如同前面章節所述，我們利用分段式序列對序列自動編碼器中的編碼器與解碼器來作為環境，根據其表現來給予切割門限獎勵。

我們假設一個語句經由越是正確的詞切割後，例如產生的音訊片段邊界與真實詞邊界十分接近，分段式序列對序列自動編碼器中的編碼器與解碼器能夠獲得越低的重建錯誤。因為這些較為正確的音訊片段會在詞切割後的訓練語料中出現較為多次，因此它們的語音詞向量可以被訓練得較好進而讓編碼器與解碼器可以獲得較低之重建錯誤。我們以在之前章節中出現過很多次的均方差（Mean Squared Error），也就是數學式5.4來描述此獎勵：

$$r_{MSE} = -\frac{1}{T}\sum_{t}^{T} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2 \tag{5.4}$$

其中$\mathbf{x}_t$與$\hat{\mathbf{x}}_t$分別為輸入之音訊特徵與重建之音訊特徵。

然而直覺上此假設的成立有待商榷，因為重建錯誤的大小或許並非與詞邊界有關，而是只與產生出的語音詞向量數目有關。意即針對同一語句，使用較多的語音詞向量必定能夠獲得較小的重建錯誤，未必與詞切割的正確與否有關。針對此一論述，我們將以預備實驗來探討此假設的成立與否。

另一方面，我們的獎勵機制必須考慮切割出的音訊片段的數量是否合理，否則為了獲得最低的重建錯誤，我們的切割門限會儘可能地產生越多音訊片段。因為每一個音訊片段越短，就越容易正確的重建出來，或重建錯誤會越小。因此這

doi:10.6342/NTU201801733

方面的獎勵設計為若切割門限產生出越少音訊片段，其就能獲得越高的獎勵。針對一個長度為T 個音框的語句，它若被切割為N個音訊片段，我們設定這方面的獎勵為：

$$r_{N/T} = -\frac{N}{T} \tag{5.5}$$

亦即為平均每個音訊片段的音框數的倒數取負值。

用於强化學習的整體的獎勵r，其需要同時考慮上述兩種獎勵，因此針對某一語句的切割，切割門限所獲得的獎勵為對上述兩個獎勵取最小值作為切割門限的獎勵：

$$r = min(r_{MSE}, \lambda r_{N/T}) \tag{5.6}$$

其中λ是個需要調整的超參數，用於設定切割出的音訊片段數量在一個合理的數量的目標相對於重建錯誤的重要性。其物理意義可視作語句壓縮率的一種估計。越大的λ表示我們希望能將語句壓縮成越少的語音詞向量，因此產生的語音詞向量數目就會越少。在我們的實驗中，我們發現比起將上述兩種獎勵用線性內差（Linear Interpolation），取兩者中最小值作為獎勵的作法能夠讓模型學到更佳的詞切割。

5.2.2 獎勵基準

獎勵基準（Reward Baseline）為一强化學習之技巧。强化學習仰賴獎勵的大小來訓練模型，在更新參數時，獎勵為正之動作其機率會被放大，反之為負者其機率會被壓低。然而許多時候在設計獎勵機制時難以同時考慮此演算法之性質，如本章節前述之獎勵設計，切割門限所能獲得之獎勵永遠為負值，換句話說，所有執

doi:10.6342/NTU201801733

行過的動作之機率都會被縮小，顯然無法提供有效之學習。因此獎勵基準之精神在於提供模型所獲得之獎勵一比較基準，若該動作具有比獎勵基準大之獎勵，則放大該動作之機率，反之則縮小該動作之機率。獎勵基準可以為一負數，因此即使所有動作所獲得之獎勵為負值，只要比獎勵基準大之動作其機率就能被放大，進而達到有效之學習。

不同於多數論文使用一個單位訓練集（Training Batch）中所有獎勵的平均作為獎勵基準，我們使用以語句作為單位（Utterance-wise）所估計的獎勵基準來移除不同語句間的偏差（Bias），藉此對真實的獎勵基準有更準的估計。換句話說，有些語句在先天上其所包含的詞比較少（式5.5所計算的值比較大）或是語句內容本身就容易造成較大之重建錯誤，因此獎勵基準的設計若以各語句為單位而非以單位訓練集為單位，則可獲得較為準確的估計。

在訓練過程中，每一個語句會被切割門限取樣出M組詞邊界。我們可以針對每一組詞邊界都估計出獎勵 r_m，這M個獎勵的平均就是此語句獎勵基準r_b，如式5.7：

$$r_b = \frac{1}{M}\sum_{m=1}^{M} r_m \tag{5.7}$$

5.3 兩步驟之迭代式訓練法

雖然在本章節中所描述的所有模型參數可以同時被調整，我們發現迭代式訓練法可以使整個訓練過程更加穩定，因此我們使用一兩步驟之迭代式訓練法來訓練我們的模型參數，兩個步驟不斷交替執行直到模型表現收斂。此兩步分別為：

1. 使用重建特徵與輸入特徵間的重建錯誤來訓練我們的編碼器與解碼器。在這步中，切割門限的參數是固定不動的。

doi:10.6342/NTU201801733

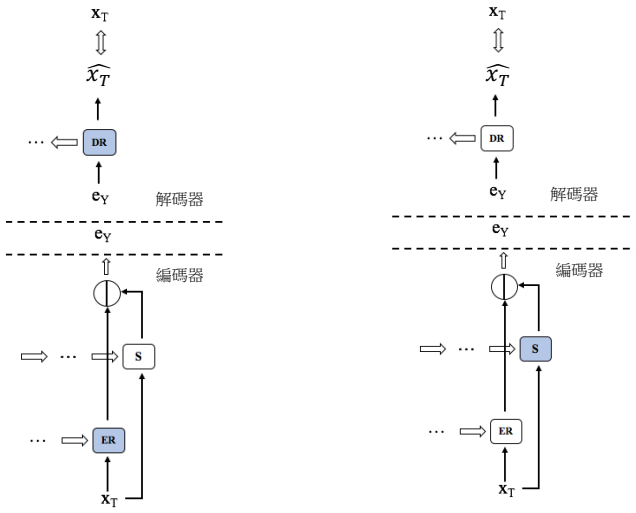


圖 5.2: 本章所使用之迭代式訓練法，藍色方框表示被更新的參數。圖(a)第一步為使用重建特徵與輸入特徵間的重建錯誤來訓練我們的編碼器與解碼器。圖(b)第二步為使用由編碼器與解碼器所計算之獎勵來更新切割門限的參數

2. 使用由編碼器與解碼器所計算之獎勵來更新切割門限的參數。在這一步中固定編碼器與解碼器的參數。

對應本章節之模型圖5.1，迭代式訓練法可以圖5.2來說明。圖5.2(a)表示此訓練法之第一步，固定住切割門限的參數並針對編碼器與解碼器的參數進行更新（以藍色方框表示被更新的參數）。而圖5.2(b)則是第二步，反過來只針對切割門限的參數進行更新。

具體而言，在迭代式訓練法之第一步的訓練過程如圖5.3，輸入語句會透過切割門限被切割成許多音訊片段，而這些音訊片段將會被用於訓練編碼器與解碼器，其減損函數為一般自動編碼器經常使用之均方差函數（式5.8），此過程中切割門限之參數保持不變

doi:10.6342/NTU201801733

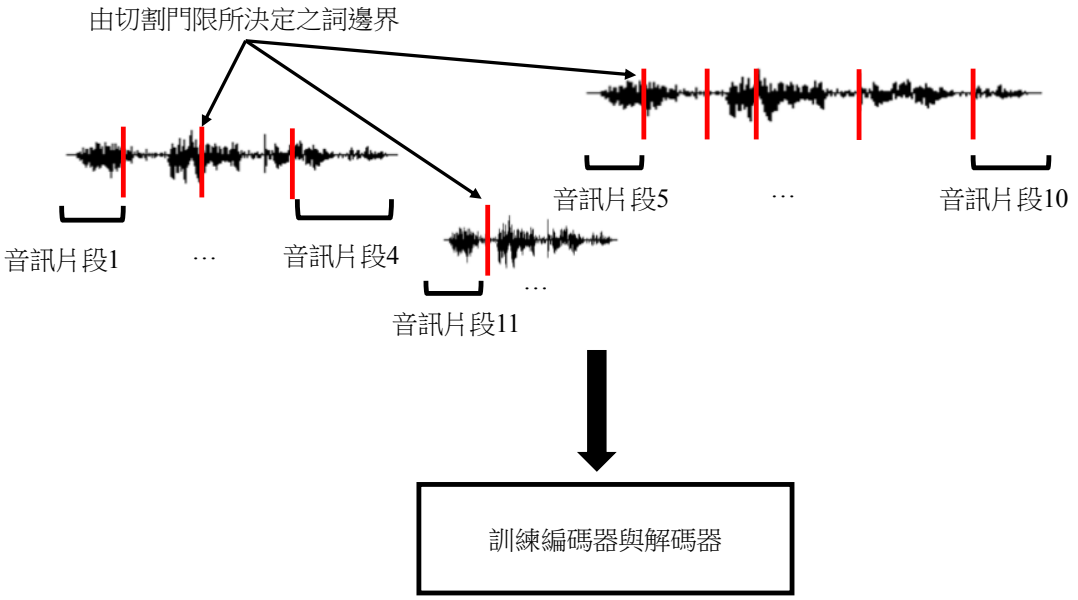


圖 5.3: 使用切割門限所決定出之音訊片段訓練編碼器與解碼器

$$loss = \frac{1}{T}\sum_{t}^{T} \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 \tag{5.8}$$

而在第二步的訓練中，則是利用編碼器與解碼器在重建輸入語句上的表現來給予切割門限關於詞邊界的訊息。若切割門限對一語句的詞切割正確，則根據我們在章節5.2.1中的假設，此語句的重建錯誤會小於那些詞切割不正確的語句，因此獲得更大的獎勵。在第二步中我們使用强化學習來更新切割門限遞迴式類神經網路的參數，所使用的强化學習演算法為章節2.2.4所介紹之策略梯度演算法，並將獎勵基準的技巧用於式2.29，進而得到以下更新式：

$$\nabla_{\Theta} J(\Theta) = \mathbb{E}_{a_t \sim \pi^{(\Theta)}}[(r - r_b)\nabla_{\Theta} \sum_{t=1}^{T} log(p_t^{a_t})], \tag{5.9}$$

其中Θ為切割門限遞迴式類神經網路的參數集。右式期望值之下標$a_t \sim \pi^{(\Theta)}$表示動作a_t由切割門限遞迴式類神經網路所模擬之策略$\pi^{(\Theta)}$所決定。$p_t^{a_t}$表示式5.3中動作a_t的機率。$J(\theta)$之定義與章節2.2相同，如式2.21之定義。r與r_b則是如式5.6與

doi:10.6342/NTU201801733

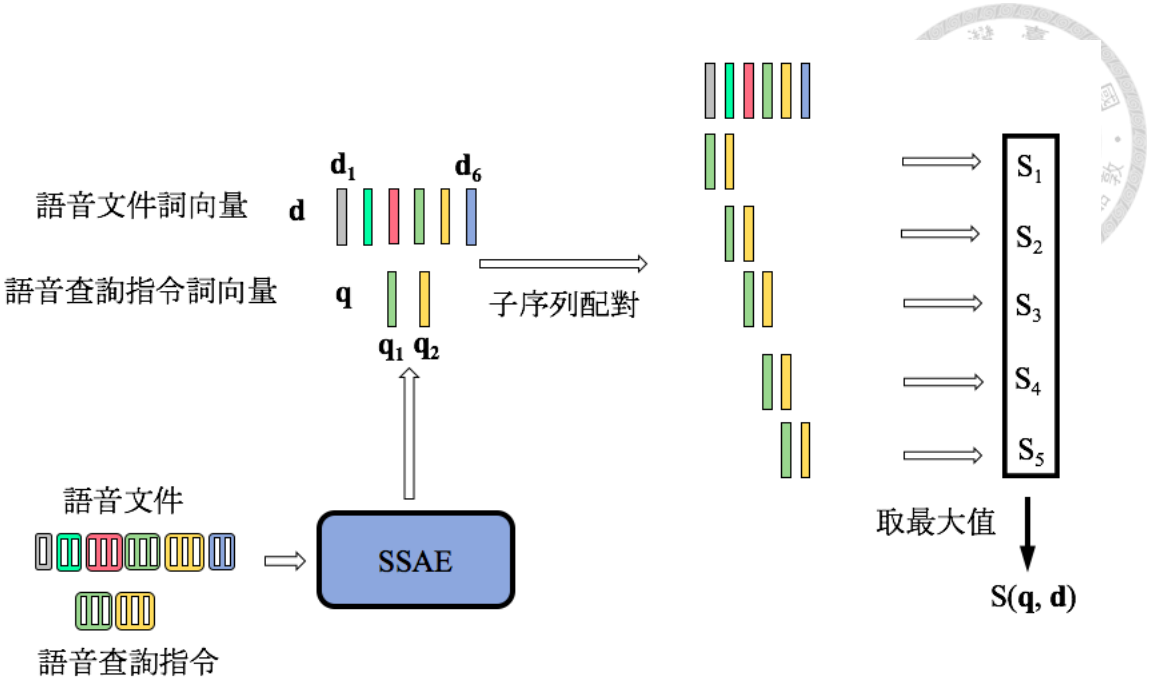


圖 5.4: 使用應用分段式語音詞向量與子系列配對所進行之口述語彙偵測。首先使用分段式序列對序列自動編碼器將語音查詢指令和語音文件分別轉換成語音詞向量序列**q**與**d**，接著對兩者使用子序列配對來評估此語音查詢指令和語音文件的相關分數

式5.7之定義。

5.4 應用分段式語音詞向量於口述語彙偵測

我們的模型可被應用於許多不同的應用上，我們在這邊將此模型應用於非督導式的按例查詢（Query-by-Example, QbE）下的口述語彙偵測（Spoken Term Detection）。此應用之目的是在不使用語音辨識的前提下，從大筆的語音文件（Spoken Document）中找出含有語音查詢指令（Query）的文件。將分段式語音詞向量應用於口述語彙偵測的整個過程可以用圖5.4描述。

首先使用分段式序列對序列自動編碼器（Segmental Sequence-to-Sequence

doi:10.6342/NTU201801733

Autoencoder, SSAE）將語音查詢指令和語音文件分別轉換成語音詞向量序列，假設語音查詢指令和語音文件的語音詞向量序列分別為：$\mathbf{q} = \{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_{N_q}\}$和$\mathbf{d} = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_{N_d}\}$。$N_d$，$N_q$表示分別由語音文件和語音查詢指令所轉換出之語音詞向量數目。

有了語音詞向量序列後，子序列配對（Subsequence Matching）便可應用於評估此語音查詢指令和語音文件的相關分數（Relevance Score），表示為$S(\mathbf{q}, \mathbf{d})$：

$$S(\mathbf{q}, \mathbf{d}) = max(S_1, S_2, ..., S_{N_d - N_q + 1}) \tag{5.10}$$

其中$S_1, S_2, ..., S_{N_d - N_q + 1}$為各子序列配對之分數，在我們的應用中我們取其中之最大值來作為對一語音查詢指令與語音文件之估計。

在進行各子序列配對時，我們使用兩序列向量間的相似度（Similarity）之乘積作為其分數之評估：

$$S_n = \prod_{m=1}^{N_q} Sim(\mathbf{q}_m, \mathbf{d}_{m+n-1}) \tag{5.11}$$

其中相似度的估計所使用的是正規化至0到1區間的餘弦相似度（Cosine Similarity）。

整個子序列配對的過程可用圖5.4右半部表示，$S_1 = Sim(\mathbf{q}_1, \mathbf{d}_1) \cdot Sim(\mathbf{q}_2, \mathbf{d}_2)$、$S_2 = Sim(\mathbf{q}_1, \mathbf{d}_2) \cdot Sim(\mathbf{q}_2, \mathbf{d}_3)$等等。如式5.10，一個語音查詢指令和語音文件的相關分數是取所有的S_n的最大值。此種以區段為單位（Segment-based）的子序列比對方法能夠取代以音框為單位（Frame-based）進行比對的方法，如動態時間校準（Dynamic Time Warping, DTW），並且節省大量的運算。

doi:10.6342/NTU201801733

5.5 實驗

5.5.1 實驗設計

我們的實驗語料包涵四種語言：英文、捷克文、法文與德文。英文是使用TIMIT語料，而其他三種語言是使用GlobalPhone語料 [64]。英文的詞邊界是由TIMIT語料所提供的詞邊界資訊，其他語言之詞邊界的資訊是使用强迫對齊法（Forced-alignment）所得到的。分段式序列對序列自動編碼器（Sequence-to-Sequence Autoencoder, SSAE）中編碼器與解碼器各別由一層100個長短期記憶類神經網路單元（Long Short-term Memory, LSTM）所組成 [2]。切割門限則是由兩層256個長短期記憶類神經網路單元所組成。切割門限所使用之門限激發訊號為一預訓練好之門限遞迴單元自動編碼器之更新門限，其設定與章節3.2相同。所有的參數更新使用慣量（Momentum）演算法中的ADAM訓練法進行更新 [32]。在評估獎勵基準時所使用的式5.7的M值為5。我們可以藉由估計語音詞之平均長度來調整式5.6中的λ。在本章中所有分割語句之模型之參數皆調整為其分割出語音詞之平均長度與真值相符，在本章實驗中λ值為5。强化學習演算法方面，我們使用一進階版本之策略梯度演算法：近似策略最佳化（Proximal Policy Optimization） [39]。

我們在本章將進行三種實驗：預備實驗、詞切割實驗與口述語彙偵測實驗。預備實驗旨在驗證章節5.2.1中，我們的針對詞切割的正確與模型之重建錯誤之間關聯的探討。詞切割實驗與口述語彙偵測實驗則是分別對於語句切割的正確與否和所產生出之語音詞向量的品質進行探討。

在進行評估詞邊界切割效能時，所使用的容忍窗（Tolerance Window）的大小與章節4.3.1相同，設定為40毫秒。所使用的音訊特徵為39維的梅爾倒頻譜係

doi:10.6342/NTU201801733

數（MFCC），搭配語句倒頻譜平均變異數正規化（Utterance-wise Cepstral Mean Variance Normalization, CMVN）。

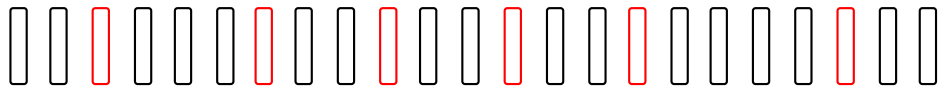
5.5.2 預備實驗

在章節5.2.1中，我們假設一個語句經由越是正確的詞切割後，可以獲得較低的重建錯誤，因此獎勵設計為較低的重建錯誤可以獲得較高的獎勵。然而此假設的成立與否有待商榷，因為直覺上一個語句切割成越多區段，也就是使用較多的語音詞向量來代表此語句，則每個語音詞向量所需要學習到的區段長度就會較小，因此就能獲得越低的重建錯誤。簡言之，直覺上而言一個語句的重建錯誤未必直接反應該語句之詞切割正確與否，因此使用此獎勵來訓練模型的正確性需要利用實驗證明。本章節所使用之語料為TIMIT。

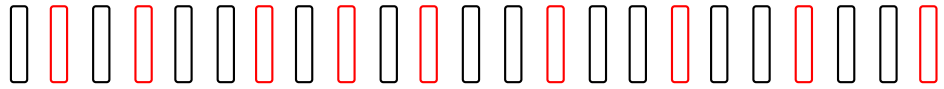
本節之實驗為驗證上述直覺，因此我們的實驗希望可以顯示出使用較少數量但切割正確的語音詞向量，其重建錯誤比隨機切割下但使用較多語音詞向量之重建錯誤是否更低。本節之實驗設計為針對每一語句，除了使用正確詞邊界進行切割來讓自動編碼器進行訓練外，我們也比較使用不同數量的詞邊界來訓練自動編碼器。假設一語句的詞邊界數量為N，則我們使用1.5N個詞邊界（也就是使用1.5N個語音詞向量來代表該語句）來訓練另外一個自動編碼器，然而此1.5N個詞邊界為隨機切割之結果。

圖5.5為針對同一具有6個詞邊界之語句，產生不同數量的詞邊界之示意圖。圖中的方框表示音框序列，而紅色方框表示用於切割語句之詞邊界。圖5.5(a)所表示的是真實詞邊界之數量與位置。圖5.5(b)所表示的是隨機產生1.5N，也就是9個詞邊界的示意圖，而圖5.5(c)則表示隨機產生1.0N，也就是6個詞邊界的示意圖。圖5.5(a)與圖5.5(c)的詞邊界數量雖然都是6個，但圖5.5(c)之詞邊界為隨機

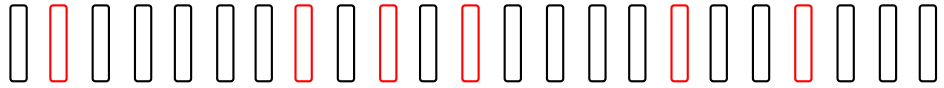
doi:10.6342/NTU201801733



(a) 真實語音詞邊界，N = 6



(b) 隨機語音詞邊界，1.5N = 9



(c) 隨機語音詞邊界，1.0N = 6

圖 5.5: 使用不同數量之語音詞邊界示意圖。圖中的方框表示音框序列，而紅色方框表示用於切割語句之詞邊界

產生之詞邊界。除了使用1.5N個隨機產生的詞邊界外，我們也比較使用數量為1.0N與1.2N之隨機產生詞邊界。

除了比較使用不同數量的語音詞向量對於重建錯誤的影響，我們也比較在相同數量下詞切割邊界正確與否對於重建錯誤的影響。對於語句中每一個真實詞邊界，我們會將其隨機移動，產生最大為一誤差窗大小的偏移。舉例而言，若我們設定的誤差窗為50毫秒，則經移動過後的詞邊界可能與原本的詞邊界距離10毫秒、30毫秒甚至0毫秒等等，完全隨機決定，然而此偏移的距離最大為50毫秒，不可能產生60毫秒之偏移。換句話說，當誤差窗越小時，表示語句中經過移動後的詞邊界距離真實詞邊界越近，也就是語句被詞切割的越準確；相反地，當我們設定的誤差窗越大，表示語句中的詞邊界可能偏移到離真實詞邊界越遠的時間點。

藉由比較上述切割結果，我們將可檢驗章節5.2.1中之假設的正確性。實驗結果如表5.2，表中之重建錯誤為更新自動編碼器參數後在驗證集（Validation Set）上使用式5.8所計算而得。實驗結果可以看出當詞邊界皆為隨機產生時，其重建錯

doi:10.6342/NTU201801733

誤確實與詞邊界數量呈現反相關係：越多的詞邊界，意即使用越多語音詞向量來代表一語句，可以獲得越低之重建錯誤。比較使用隨機產生1.2倍詞邊界的重建錯誤與使用真實與音詞邊界所獲得之重建錯誤，可以發現使用真實詞邊界的重建錯誤顯著低於使用隨機產生1.2倍詞邊界的重建錯誤。即使使用真實詞邊界的後所產生的語音詞向量數量較少，其依舊能獲得較低之重建錯誤。推測是因為這些經由正確切割的音訊片段會在詞切割後的訓練語料中出現較為多次，因此它們的語音詞向量可以被訓練得較好進而可以獲得較低的重建錯誤。然而若產生更多數量的語音詞向量，如使用隨機產生1.5倍詞邊界，則能夠獲得比使用真實詞邊界更低之重建錯誤，因此我們需要限制產生的詞邊界數目，否則模型會想辦法產生越多的詞向量來壓低重建錯誤。

詞邊界之描述	重建錯誤
隨機產生1.0倍詞邊界	0.9210
隨機產生1.2倍詞邊界	0.9002
隨機產生1.5倍詞邊界	0.8540
10毫秒誤差窗	0.8912
20毫秒誤差窗	0.9395
50毫秒誤差窗	0.9434
真實語音詞邊界	0.8651

表 5.2: 產生不同數量的詞向量來訓練編碼器與解碼器所獲得之重建錯誤。使用真實詞邊界的重建錯誤顯著低於使用隨機產生1.2倍詞邊界的重建錯誤，但當隨機產生之詞邊界過多時，如1.5倍詞邊界數目，其能獲得比使用真實詞邊界更低之重建錯誤。另外當誤差窗越大時所造成的重建錯誤越大

doi:10.6342/NTU201801733

在詞切割之正確性的比較方面，我們可以發現當誤差窗越大時所造成的重建錯誤越大。換句話說，在同樣數量下詞切割的正確與否會直接反應在重建錯誤上，越正確的詞切割能夠獲得越低之重建錯誤。

藉由本節實驗，我們驗證了在章節5.2.1中所提出之假設，此假設讓我們能使用一非督導式學習的目標，也就是重建錯誤，來獲得語句中詞邊界正確與否之資訊。

5.5.3 詞切割實驗

本節實驗中我們首先觀察我們代理人所獲得獎勵之趨勢來判斷强化學習是否讓我們的模型，分段式序列對序列自動編碼器（Segmental Sequence-to-Sequence Autoencoder, SSAE），逐漸學習到系統目標。換言之，有效的强化學習能夠讓我們的模型所獲得之獎勵越來越高。我們將訓練過程中所獲得之獎勵畫成圖5.6。圖5.6中顏色較淡之曲線為模型產生之所切割出之音訊的平均長度的趨勢。然而為了能更清楚顯示趨勢，我們將原始資料使用平滑處理（Smoothing）後繪製成深色線條。平滑處理的方式為將每個時間點的原始資料與其鄰近之10個資料點的資料做平均。圖5.6之橫軸為訓練過程，也就是參數更新的次數，而縱軸則為所獲得之獎勵。從此圖可以看出，使用强化學習的學習演算法，分段式序列對序列自動編碼器能夠藉由不斷地訓練獲得越來越高之獎勵，最後收斂。因此可以判斷我們的模型確實有成功運用强化學習訓練法學習到符合系統目標的動作。

然而獎勵曲線只是針對系統目標的間接觀察，我們直接使用系統目標對我們的模型行為做觀察。由於我們的系統目標是詞切割，因此我們直接觀察分段式序列對序列自動編碼器在訓練過程中的詞切割效能曲線。圖5.7為分段式序列對序列自動編碼器在捷克文驗證集上的詞切割效能曲線。圖中之橫軸為訓練過程，也就

doi:10.6342/NTU201801733

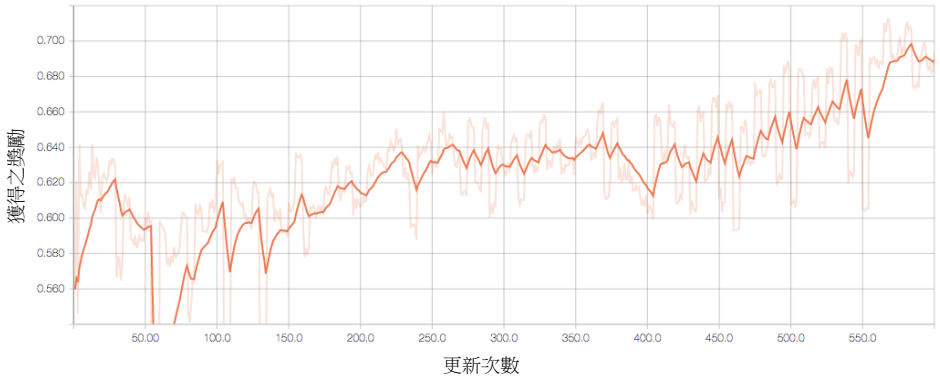


圖 5.6: 分段式序列對序列自動編碼器使用强化學習下之獎勵曲線。淡色曲線為原始資料深色曲線為將資料經過平滑處理而繪製。使用强化學習的學習演算法，分段式序列對序列自動編碼器能夠藉由不斷地訓練獲得越來越高之獎勵，最後收斂

是模型參數的更新次數。縱軸有二曲線，紅色曲線為模型所切割出之音訊的平均長度（單位為毫秒）。從圖中我們可以發現我們的模型隨著時間漸漸的學習到如何將語句做詞切割：不管是準確率（Precision）或是召回率（Recall），和訓練初期相比都有提升（圖5.7(a)(b)中的藍色曲線），而且模型所切割出之音訊的平均長度也逐漸收斂到一合理的長度（兩圖中的紅色曲線）。從此圖可以看出我們的模型在學習的過程中，所切割出之音訊片段的平均長度變動不大，然而在所切割出之音訊的平均長度約略相同的情況下，我們的模型能夠透過强化學習演算法逐漸學習出如何在正確的時間點進行詞切割，因此效能曲線才會都呈現上升的趨勢。不侷限在捷克文，在其他三個語言也有相似的學習曲線，如圖5.8至圖5.10。

doi:10.6342/NTU201801733

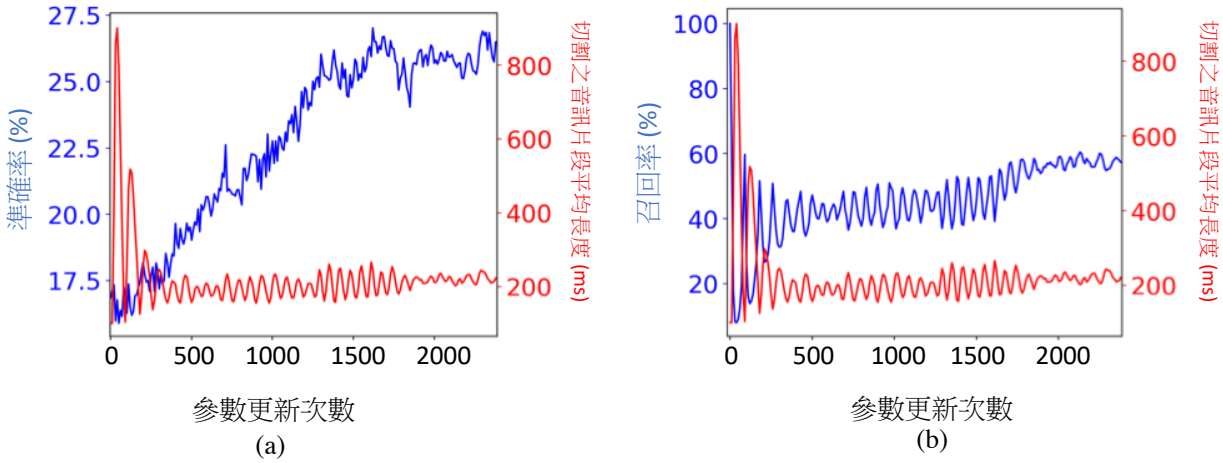


圖 5.7: 捷克語上詞切割之效能曲線

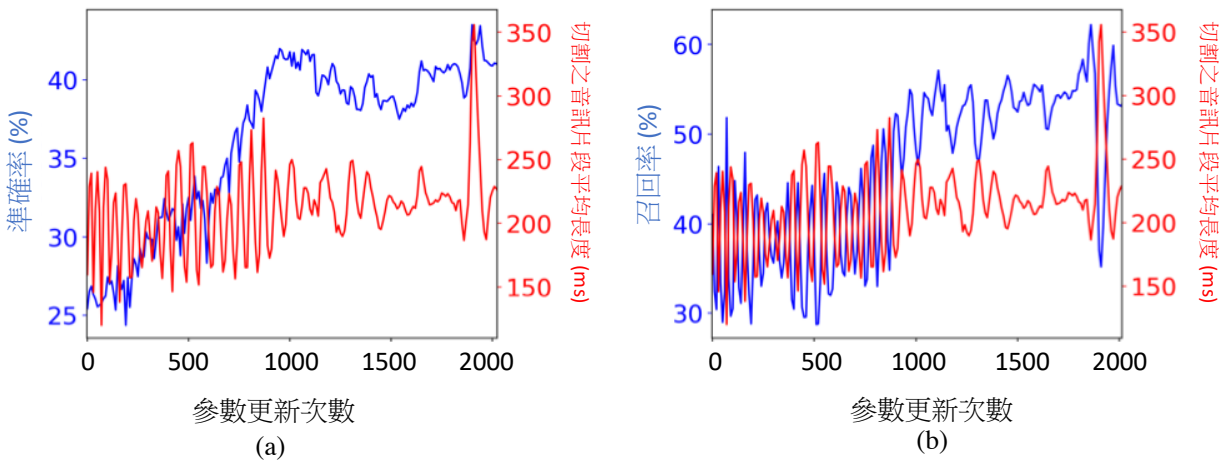


圖 5.8: 英語上詞切割之效能曲線

doi:10.6342/NTU201801733

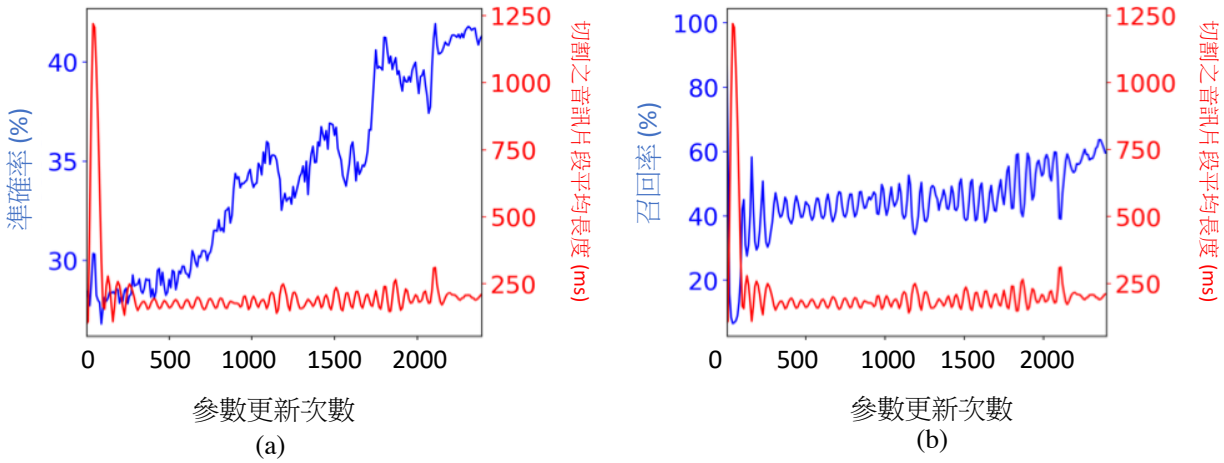


圖 5.9: 法語上詞切割之效能曲線

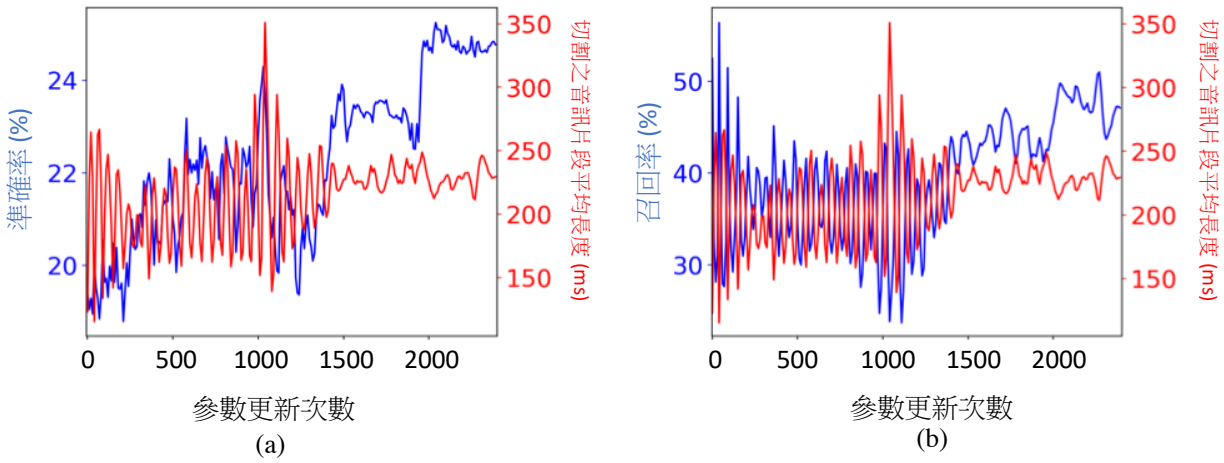


圖 5.10: 德語上詞切割之效能曲線

doi:10.6342/NTU201801733

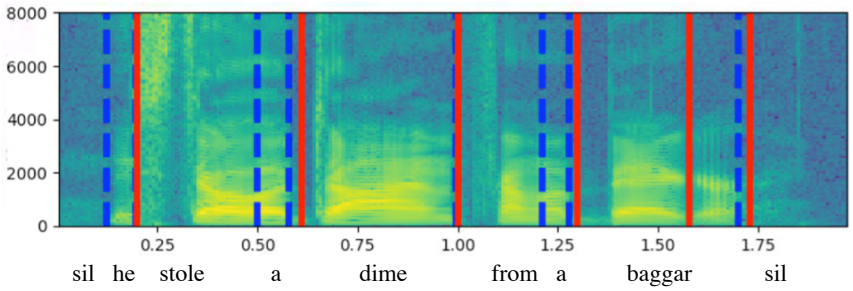


圖 5.11: 英文上詞切割之範例。圖為英文上一語句之頻譜圖。圖中之藍色虛線為真實詞邊界而紅色線條為分段式序列對序列自動編碼器所決定之詞邊界，圖片下方的文字為各詞之轉寫，”sil”為無聲音訊片段（Silence）。圖中可以發現其兩種詞邊界高度重合，顯示我們的模型確實可以藉由强化學習學習到正確切割

我們另外也定性觀察詞切割之結果。圖5.11為英文語料中一語句之頻譜圖（Spectrogram）。圖中之藍色虛線為真實詞邊界而紅色線條為分段式序列對序列自動編碼器所決定之詞邊界。圖片下方的文字為音訊之轉寫（Transcription），其中”sil”為無聲音訊片段（Silence）。由圖中可以發現其兩種詞邊界高度重合，顯示我們的模型確實可以藉由强化學習學習到正確切割。另外有趣的是冠詞”a”都被我們的模型合併至其前面的詞，推測是因為口語上冠詞幾乎都是快速帶過，在頻譜圖上並無劇烈變化，因此被我們的模型忽略。另外分段式序列對序列自動編碼器將最後一個字”baggar”分成兩段，觀察原始頻譜圖，其分段點為一頻譜不連續處，為”baggar”中的”gar”音。這發現也說明了我們的模型所發掘出的詞邊界也有可能是次詞（Subword）單位之詞邊界。

在不同模型間效能比較方面，我們使用隨機基準與另外兩種方法與分段式序列對序列自動編碼器（Segmental Sequence-to-Sequence Autoencoder, SSAE）比

doi:10.6342/NTU201801733

較它們在詞切割上的效能，實驗結果使用F1分數表示並列於表5.3。在英文上，除了F1分數外我們也列出準確率和召回率。除了隨機基準外的另外兩種方法分別是用門限激發訊號（Gate Activation Signal, GAS） [63]與階層聚合式分群法（Hierarchical Agglomerative Clustering, HAC） [48] [49]進行詞切割。 從表5.3中，我們可以發現分段式序列對序列自動編碼器在詞切割上的整體效能顯著高於另外兩種方法，尤其是在法文方面的表現，然而其在德文上略低於門限激發訊號的效能。

語言	英文			捷克文	法文	德文
方法	準確率	召回率	F1分數	F1分數		
隨機切割	0.2460	0.4108	0.3077	0.2256	0.3266	0.2541
階層聚合式分群法	0.2684	0.4621	0.3396	0.3084	0.3375	0.2709
門限激發訊號	0.3322	0.5239	0.4066	0.2953	0.3111	0.3289
SSAE	0.3706	0.5155	0.4312	0.3778	0.4814	0.3169

表 5.3: 詞切割之實驗結果。分段式序列對序列自動編碼器（Segmental Sequence-to-Sequence Autoencoder, SSAE）在詞切割上的整體效能顯著高於另外兩種方法，尤其是在法文方面的表現，然而其在德文上略低於門限激發訊號的效能

值得一提的是在使用迭代式訓練法時，我們發現在執行完第二步訓練後，再回到第一步訓練時需要將編碼器與解碼器的參數重設（Reset）至他們一開始隨機決定的參數而非接續使用上一次訓練完的參數。有經過此重設步驟可以增進穩定訓練過程之穩定度。推測是因為此步驟可以讓第二步驟中，將使用强化學習訓練切割門限時的環境（也就是提供獎勵資訊的編碼器與解碼器）維持一致，進而獲得更佳的强化學習的效果。

doi:10.6342/NTU201801733

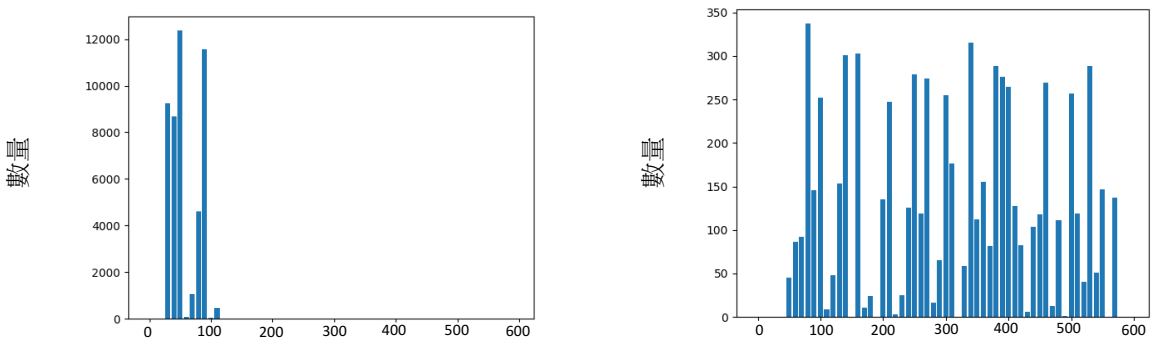


圖 5.12: 捷克文之 (a)音素音訊長度分佈 (b)詞音訊長度分佈

為了進一步了解分段式序列對序列自動編碼器所切割出來的語音片段較接近詞或是音素，我們將針對這些由分段式序列對序列自動編碼器所切出來之音訊片段的長度分佈，與詞音訊的長度分佈、音素音訊的長度分佈進行統計和比較。圖5.12所表示的是捷克文之音素音訊與詞音訊之長度分佈：橫軸為其長度，單位為毫秒（ms）；縱軸為統計數目。圖5.13則為分段式序列對序列自動編碼器所切割出之音訊片段長度分佈。首先觀察圖5.12，可以發現音素的長度分佈集中於100毫秒以下，而詞音訊的長度則介於50至550毫秒，沒有集中於哪個區段。而觀察圖5.13，可以發現分段式序列對序列自動編碼器所切出來之音訊片段長度介於100至350毫秒之間，因此可以判斷切出來的音訊片段為比音素要更高的層級，但是比詞的層級要更低一些，因此應屬於次詞（Subwords）的層級。相同的趨勢同樣發生在其他語言上，如圖5.14至圖5.19。

doi:10.6342/NTU201801733

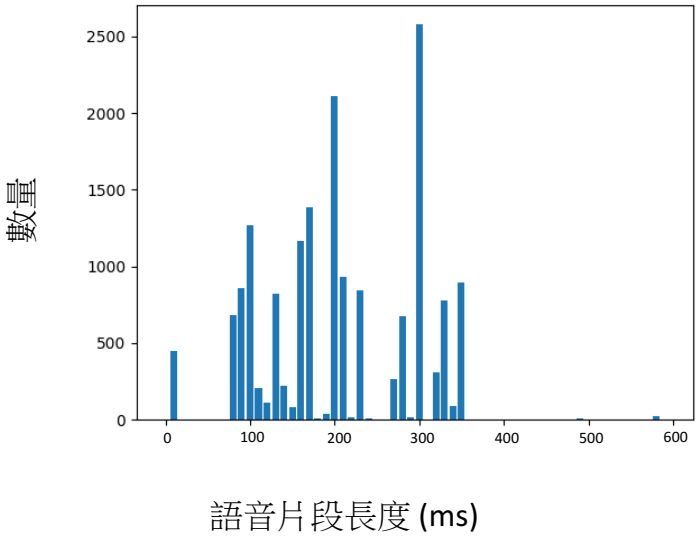


圖 5.13: 分段式序列對序列自動編碼器對捷克文切割出的音訊片段長度分佈

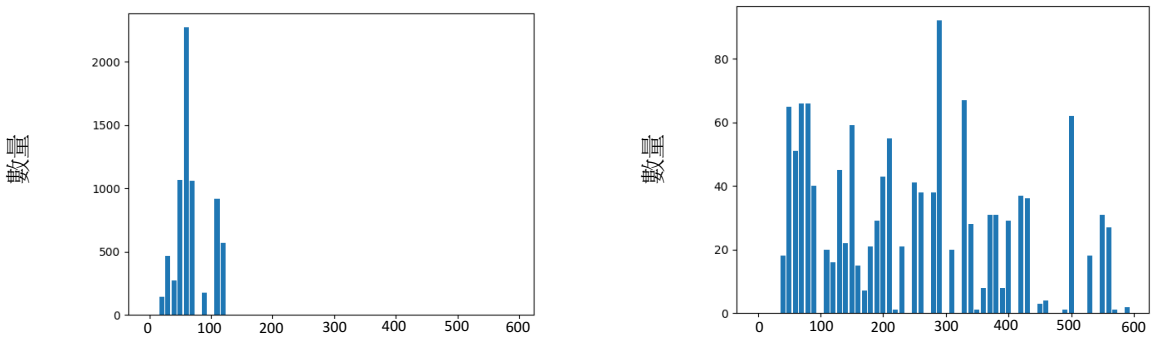


圖 5.14: 英文之 (a)音素音訊長度分佈 (b)詞音訊長度分佈

doi:10.6342/NTU201801733

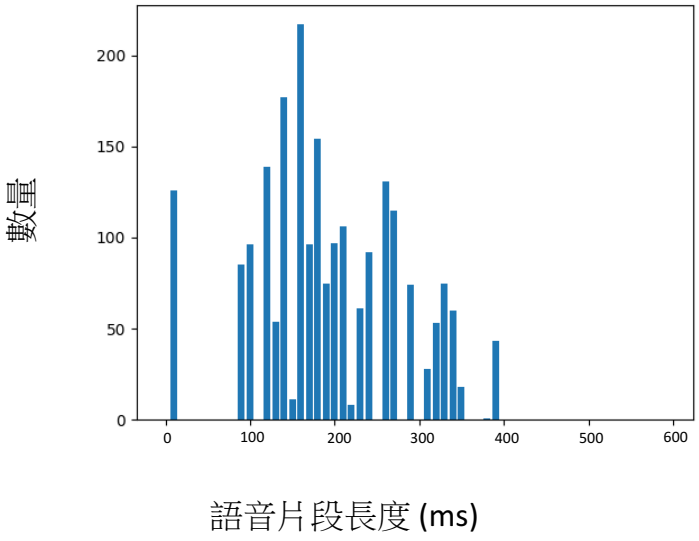


圖 5.15: 分段式序列對序列自動編碼器對英文切割出的音訊片段長度分佈

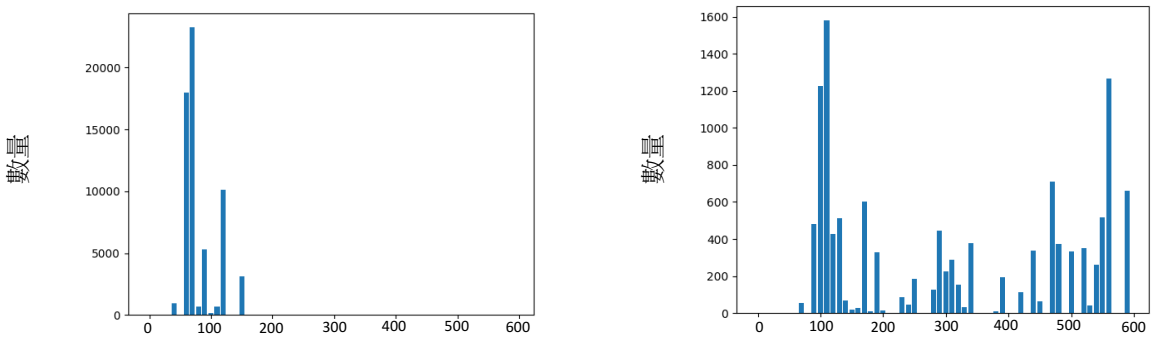


圖 5.16: 法文之 (a)音素音訊長度分佈 (b)詞音訊長度分佈

doi:10.6342/NTU201801733

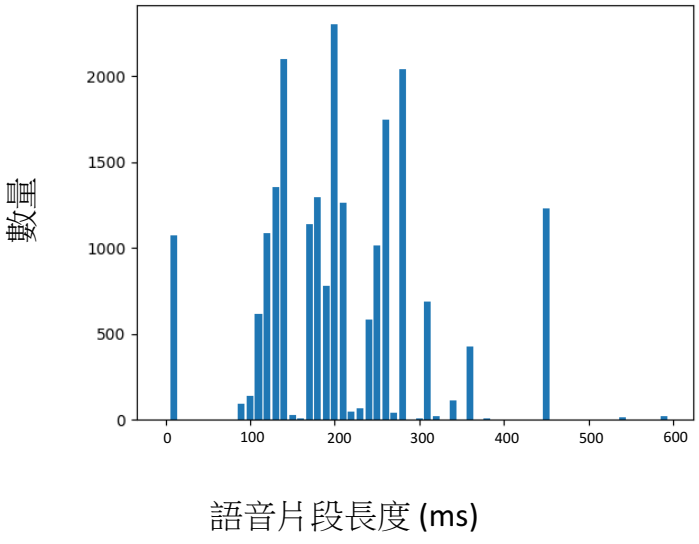


圖 5.17: 分段式序列對序列自動編碼器對法文切割出的音訊片段長度分佈

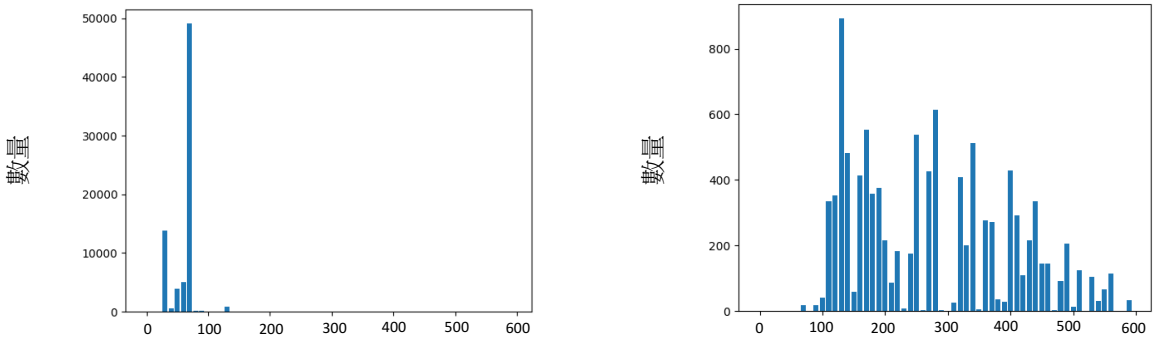


圖 5.18: 德文之 (a)音素音訊長度分佈 (b)詞音訊長度分佈

doi:10.6342/NTU201801733

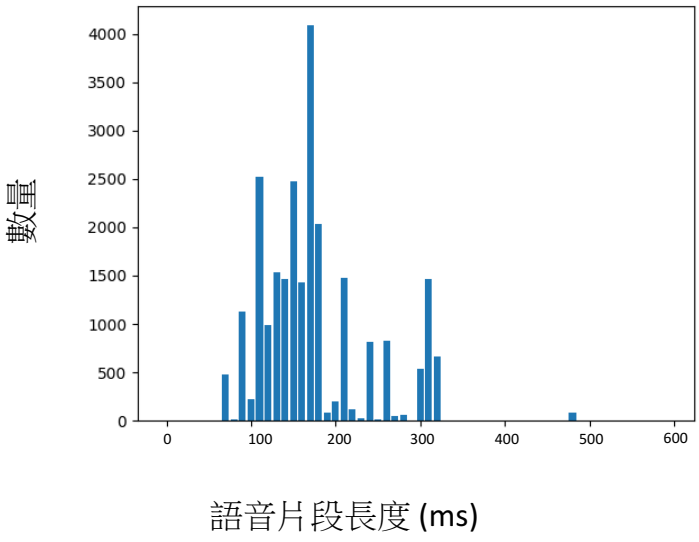


圖 5.19: 分段式序列對序列自動編碼器對德文切割出的音訊片段長度分佈

為了量化比較模型所切出之音訊片段與不同語言等級之音訊片段間的長度分佈，我們將各種音訊片段之長度分佈各近似為一高斯分佈 $Gauss(\mu, v)$（Gaussian Distribution），μ與v為此近似分佈之平均值與變異數。各語言上的詞音訊，音素音訊與使用分段式自動編碼器所切割出之音訊長度所近似之高斯分佈的數據如表5.4。表中我們可以看出德文詞音訊之標準差為為所有語言中最大，表示德文之詞音訊在長度的變化上最大。另一方面德文之音素音訊之平均值顯著低於其他語言但詞音訊之平均值卻無此情形。由於詞音訊由音素音訊所組成，因此若音素較短則表示每個詞中含有的音素數量較多，也就是每個詞音訊的音素結構較為複雜。綜合前述兩者，推測德文的語音結構應是所有語言中最複雜者，因此在詞切割實驗結果中（表5.3），分段式序列對序列自動編碼器無法獲得很好的詞切割效能。

為了衡量兩個高斯分佈之間的差異，我們使用克雷散度（Kullback–Leibler Divergence, *KLD*）來估計兩個分佈間的差異大小。我們觀察由分段式序列對序列

doi:10.6342/NTU201801733

	詞音訊		音素音訊		使用SSAE所切割出之音訊	
語言	均值 μ	標準差 $\sqrt{v}$	均值 μ	標準差 $\sqrt{v}$	均值 μ	標準差 $\sqrt{v}$
捷克文	391.44	235.62	73.22	45.26	199.65	102.78
英文	293.65	195.77	72.48	45.92	220.91	102.81
法文	296.29	226.26	72.13	45.84	204.15	100.42
德文	367.42	266.22	68.10	38.32	191.83	85.50

表 5.4: 各語言上的詞音訊，音素音訊與使用分段式序列對序列自動編碼器（Segmental Sequence-to-Sequence Autioencoder, SSAE）所切割出之音訊長度所近似之高斯分佈的數據

自動編碼器所切割出之音訊長度所近似之高斯分佈$Gauss(\mu_{seg}, v_{seg})$分別與詞音訊的分佈$Gauss(\mu_{wrd}, v_{wrd})$和音素音訊的分佈$Gauss(\mu_{phn}, v_{phn})$間的克雷散度來判斷分佈之間的差異。由於克雷散度為並非對稱，我們使用克雷散度均值（$\overline{KLD}$）來作為估計，如下式：

$$\overline{KLD}(G_1, G_2) = \overline{KLD}(G_2, G_1) = \frac{KLD(G_1, G_2) + KLD(G_2, G_1)}{2} \tag{5.12}$$

其中G_1，G_2分別代表兩個不同的高斯函數$Gauss(\mu_1, v_1)$和$Gauss(\mu_2, v_2)$。

表5.5列出不同語言上分段式自動編碼器所切割出之音訊長度所近似之高斯分佈G_{seg}分別與詞音訊的分佈G_{wrd}和G_{phn}間的平均克雷散度。表中可以看出所切割出之音訊長度在大部分的語言上明顯與詞的長度分佈較為接近，尤其是英文。顯然分段式自動編碼器在學習的過程中學習到了一些次詞單位（Subwords）的聲學模式，其長度分佈與詞相近而與音素相遠。唯一的例外是德文，可以發現由分段

doi:10.6342/NTU201801733

語言	$\overline{KLD}(G_{seg}, G_{wrd})$	$\overline{KLD}(G_{seg}, G_{phn})$
捷克文	1.8977	3.1666
英文	0.6350	3.9363
法文	1.0702	3.2585
德文	3.1128	3.9244

表 5.5: 各語言上分段式自動編碼器所切割出之音訊長度所近似之高斯分佈G_{seg}分別與詞音訊的分佈G_{wrd}和G_{phn}間的平均克雷散度

式序列對序列自動編碼器所切割出之音訊長度不論是與詞音訊或是音素音訊的分佈都相差甚遠。如前段所述，我們推測是德文所有語言中語音結構最複雜者，造成分段式序列對序列自動編碼器學習上的快男，因此所切出之音訊片段不論是與詞音訊或是音素音訊的分佈都不相近。

最後，我們好奇詞的長度對於分段式序列對序列自動編碼器之影響，因此我們的輸入音訊由語句改為詞，藉此觀察分段式序列對序列自動編碼器會如何表示不同長度的詞。我們在英文的語料上進行實驗，將不同長度之詞的音訊輸入分段式序列對序列自動編碼器，觀察其分段的數量。我們將輸入不同音訊長度的詞之分段數目分佈繪製成圖5.20。我們可以發現其幾乎為一斜直線，可以約略看出輸入長度每增加200毫秒其分段數目就會多1。分段式序列對序列自動編碼器似乎是掌握了某個長度大約200毫秒的聲學單位並用其來表示輸入之詞音訊。若確實如此，理當在進行語句切割時其切割出的音訊片段應極端集中於200毫秒。然而觀察英文的切割長度分佈圖（圖5.15）可以發現切割出之音訊片段並無此現象。推測是輸入的語音在時序上能夠提供分段式序列對序列自動編碼器更多資訊，進而讓分段式序列對序列自動編碼器能切割出長度更多樣化的音訊片段。

doi:10.6342/NTU201801733

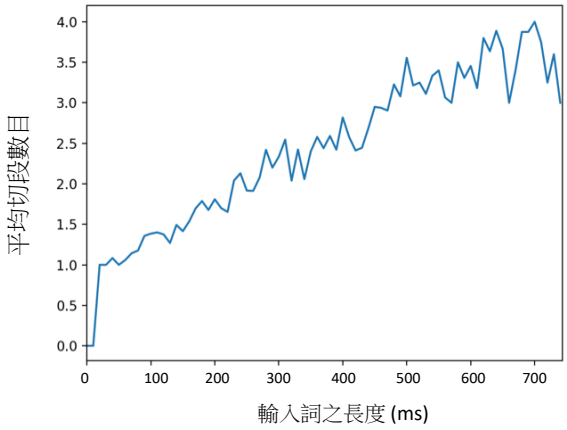


圖 5.20: 將不同音訊長度的詞輸入分段式序列對序列自動編碼器後所得之分段數目分佈

5.5.4 口述語彙偵測實驗

本節實驗中我們使用口述語彙偵測的效能來評估分段式序列對序列自動編碼器（SSAE）所產生的語音詞向量之品質，其中將語音詞向量應用於口述語彙偵測的方法如同章節5.4描述。使用情境方面，我們將語句視為語音文件並使用詞作為語音查詢指令，偵測某一語句是否含有作為語音查詢指令的詞。效能評估方面，我們使用平均準確平均值（Mean Average Precision, MAP）來衡量口述語彙偵測的效能。

我們使用張氏等人（Yaodong Zhang）所提出的方法準備語音查詢指令和語音文件 [65]。語音查詢指令為隨機選取若干聲學特徵不同的詞作為語音查詢詞（Query Words），接著將這些查詢詞在訓練集語句中的音訊片段擷取出來作為語音查詢指令。而測試集中每一個語句便作為一個語音文件。在進行口述語彙偵測實驗時，語音查詢指令之結果彼此獨立。換句話說，即使兩個語音查詢指令為相同的語音查詢詞，此關聯並不納入效能評估中。在我們的設定中我們隨機選取5個聲學特徵不同之語音查詢詞，所產生的語音查詢詞數量在英文、捷克文，法文與

doi:10.6342/NTU201801733

德文上分別為29、21，25和23，相關資訊如表5.6至表5.7。

語言	語音查詢詞	語音查詢指令數目	含有語音查詢詞之語音文件數
英文	fail	7	1
	simple	8	1
	military	2	2
	increases	7	1
	problems	5	1
捷克文	pracují	5	1
	použití	5	1
	textu	2	4
	demokracie	4	2
	abych	5	1

表 5.6: 英文與捷克文上語音查詢指令之列表

比較基準方面，我們使用標準的以音框為單位之動態時間校準（Dynamic Time Warpping, DTW）作為我們最主要的比較對象。另外我們也比較使用不同切割方法所產生出來的語音詞向量在品質上的差異，我們使用章節5.5.3中所使用的門限激發訊號（Gate Activation Signal, GAS）與階層聚合式分群法（Hierarchical Agglomerative Clustering, HAC）來作為分段式序列對序列自動編碼器以外的另外兩種非督導式學習下之詞切割方法，探討若使用現有詞切割方法所能獲得的效能。另外我們也比較使用真實詞邊界（Oracle）來進行詞切割所能獲得之口述語彙偵測效能，作為使用語音詞向量這系列方法的效能上限。

表5.8所列出實驗結果。最後一欄列出的是使用真實詞邊界（Oracle）所訓

doi:10.6342/NTU201801733

語言	語音查詢詞	語音查詢指令數目	含有語音查詢詞之語音文件數
法文	soldats	5	1
	organisme	4	2
	travaillant	6	2
	soulève	5	1
	sportifs	5	1
德文	vergeblich	5	1
	gutem	4	2
	sozial	5	1
	großes	5	1
	ernennung	4	2

表 5.7: 法文與德文上語音查詢指令之列表

練出而得的語音詞向量在口述語彙偵測上的效能。而位於第一欄的隨機基準（Ran.）則是在評估查詢指令與語音文件的相關分數時，此相關分數為一隨機亂數。我們同時也在第二欄列出以音框為單位（Frame-based）所進行的動態時間校準（Dynamic Time Warpping, DTW）的效能。

從表中可以看出使用真實詞邊界所產生之語音詞向量所得到的效能遠比其他方法的效能要高上許多，與前人論文相符 [2]。分段式序列對序列自動編碼器能夠獲得比動態時間校準更佳的效能，推測是因為動態時間校準在面對不同語者或是性別時，其沒辦法準確地對語音查詢指令和語音文件內容做出判斷。然而不同語者和性別等這些特性差異或許在語音詞向量們的訓練過程中被自動編碼器吸收，讓分段式序列對序列自動編碼器確實能夠抓住語句中的音素結構，因此分段式序

doi:10.6342/NTU201801733

			語音詞向量（使用不同切割方法）			
語言	Ran.	DTW	GAS	HAC	SSAE	Oracle
捷克文	0.38	16.59	0.68	1.13	19.41	22.56
英文	0.74	12.02	8.29	0.91	23.27	30.28
法文	0.27	11.72	0.40	0.92	21.70	29.66
德文	0.18	6.07	0.27	0.26	13.82	21.52

表 5.8: 口述語彙偵測之實驗結果，表中數據為平均準確平均值（Mean Average Precision, MAP）。第一欄為隨機基準（Ran.），最後一欄為使用真實詞邊界（Oracle）所訓練出而得的語音詞向量，為一效能上限。使用真實詞邊界所產生之語音詞向量所得到的效能遠比其他方法的效能要高上許多。分段式序列對序列自動編碼器能夠獲得比主要比較對象：動態時間校準，更佳的效能

列對序列自動編碼器能夠獲得較佳的效能。然而我們發現使用別種詞切割方法所得到語音詞向量的效能十分不理想，與隨機基準相去不遠。推測是詞切割的效能需要達到某個最低水準才能讓模型從語料中學習到有意義的資訊，進而產生具有意義的語音詞向量。

有趣的是在德文上雖然分段式序列對序列自動編碼器的詞切割效能與使用門限激發訊號相近，但是分段式序列對序列自動編碼器所產生的語音詞向量在口述語彙偵測上的效能顯著高於透過門限激發訊號所產生的語音詞向量。這個現象的原因尚未明瞭，推測是德文具有一些特別的語言結構所造成。

我們也進一步在英文上分析口述語彙偵測的結果。圖5.21所展現的是一英文上口述語彙偵測之範例。圖5.21(a)為語音查詢指令"increases"之頻譜圖，而圖5.21(b)至5.21(f)為五個具有最高之相關分數的語音文件之頻譜圖，使用相關分

doi:10.6342/NTU201801733

數由高至低進行排序。為節省版面，圖5.21(b)至5.21(f)只展現模型偵測到語音查詢指令區域，也就是子序列配對中分數最高之區域附近之頻譜圖。圖5.21中藍線表示詞邊界，而兩條紅線之間的區域為使用分段式語音詞向量偵測到語音查詢指令之區域。頻譜圖下方文字為該段語音之轉寫。

由圖5.21中我們可以看出，圖5.21(b)中含有”increases”的區段順利被分段式語音詞向量偵測到，因此具有最高之相關分數。由圖(b)中我們可以看出分段式語音詞向量偵測到的區域略大於語句中真實詞”increases”，推測其原因為緊接其後的詞”strength”的開頭也是’s’的音。圖中其他語句方面，圖5.21(c)與圖5.21(e)所偵測到的區域都是以’k’音開頭’s’音結尾，推測是因為與語音查詢指令’increases’中’creases’的部分發音相似因而具有較高順位。圖5.21(f)則推測是結尾附近’th’音與語音查詢指令結尾的’s’音相近，因而具有較高之相關分數。

5.6 本章總結

本章中介紹了如何將强化學習應用於分段式序列對序列自動編碼器以產生分段式語音詞向量。由於切割門限為一遞迴式類神經網路且其中並無使用重設機制，因此可以完整掌握語句資訊來做最有效之學校。藉由强化學習與符合系統目標之獎勵的設計，分段式序列對序列自動編碼器能夠逐漸學習到如何將輸入語句做正確詞切割並同時將切割出的音訊片段轉化為語音詞向量。實驗上除了對詞切割與口述語彙偵測的結果進行定性分析外，亦有與以往方法進行定量的效能比較。實驗結果顯示由此分段式序列對序列自動編碼器所產生之語音詞向量不論在詞切割或是口述語彙偵測的應用上都具有比以往方法更佳的效能表現。

doi:10.6342/NTU201801733

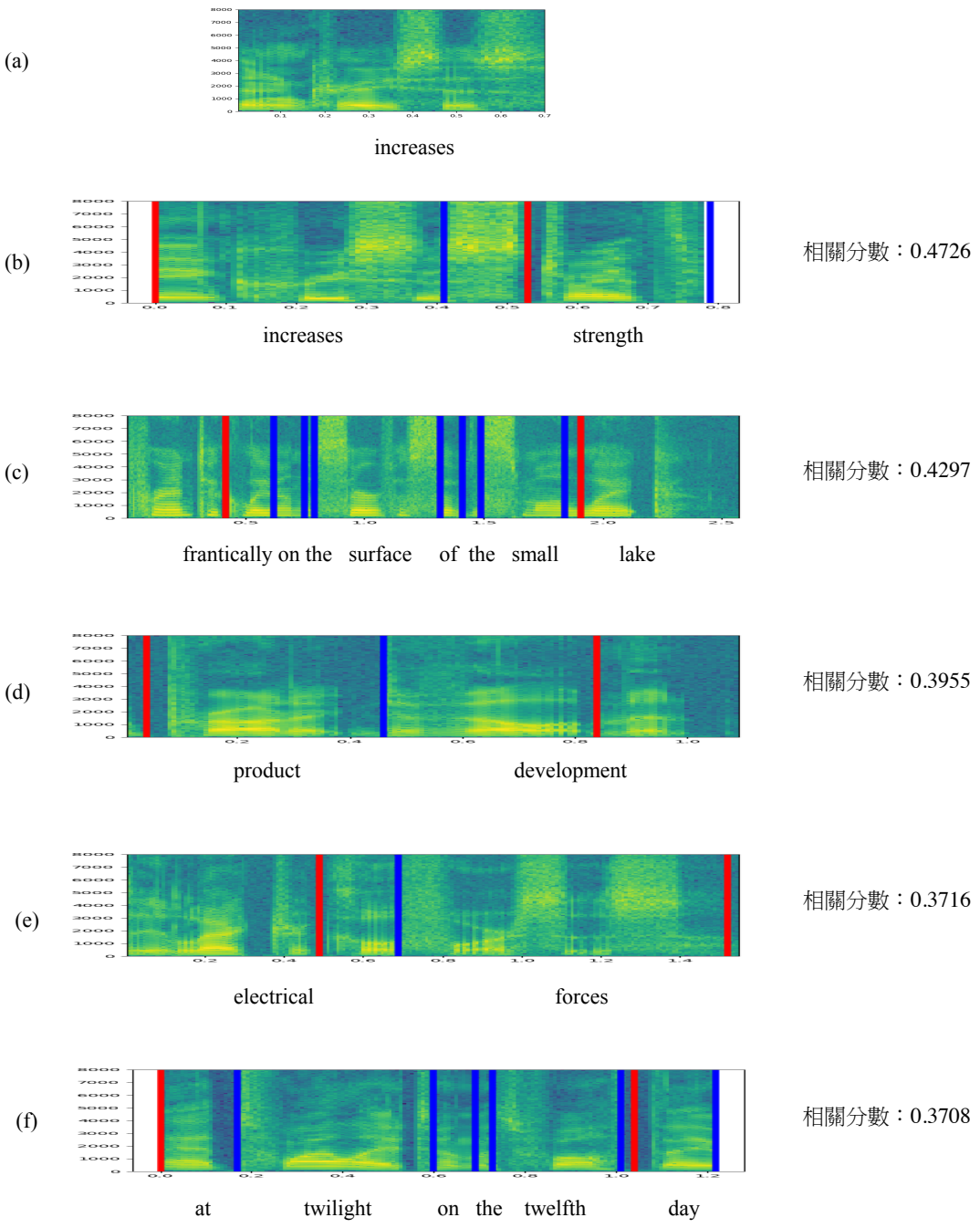


圖 5.21: 使用頻譜圖展現口述語彙偵測範例。(a)為語音查詢指令，(b)至(f)分別為Top-1至Top-5的相關語音文件之部分擷取圖。藍線表示詞邊界，兩紅線之間的區域為模型偵測到語音查詢指令的區域。每個圖下方的文字為該段語音之轉寫

doi:10.6342/NTU201801733

第六章　結論與展望

6.1　本論文主要的研究貢獻

本論文旨在探討如何使用一系列語音詞向量表示一語句。其中模型訓練過程必須在非督導式學習的框架下進行。主要的研究貢獻如下：

- 分析一種位於遞迴式類神經網路內之訊號，門限激發訊號，並發現此訊號在非督導式學習框架下的模型（如自動編碼器），與輸入音訊中語音特性之邊界（如音素邊界）具有強烈關聯，因此可以廣泛被應用於所有非督導式學習下的遞迴式類神經網路模型。
- 提出分段式語音詞向量之概念。引入切割門限之概念至原來產生語音詞向量之序列對序列自動編碼器，將其擴展成分段式序列對序列自動編碼器，用以將一輸入語句轉化為一語音詞向量序列。
- 探討使用端對端訓練的概念訓練分段式序列對序列自動編碼器，並針對其減損函數進行探討、改進。也探討使用目前端對端訓練的架構來訓練分段式序列對序列自動編碼器之潛在問題。
- 使用強化學習的概念來訓練分段式語音詞向量，改善端對端訓練模型所具有的潛在問題。將分段式語音詞向量之訓練目標設計為用以實現強化學習演算法之獎勵。
- 針對強化學習下所訓練之分段式語音詞向量，透過實驗對其詞切割與口述語彙偵測的結果進行定性分析並與以往方法進行定量的效能比較。不論在詞切割亦或是口述語彙偵測其皆具備比以往方法更佳的效能。

doi:10.6342/NTU201801733

6.2 本論文未來研究方向

在本論文第三章中所探討之門限激發訊號擷取自自動編碼器之編碼器，然而若取自其他元件（如解碼器）的門限激發訊號會具有何特性仍然需要實驗探討。另外已知在神經網路模型中層數越深的特徵為代表越抽象之概念（如語者情緒），因此由層數越深的神經網路單元所產生之門限激發訊號是否也與越抽象之特徵的轉換（語者情緒轉變）有關也是個值得探討的問題。

在第四章中使用端對端訓練雖然未獲得具體成果，但鑑於第四章中改進閾值的訓練方式以獲得合理切割音訊長度之經驗，或許可以探討對閾值使用更細緻的數學模型。另外目前也有諸多數學模型旨在處理不可微分問題，如剛貝爾軟性最大化（Gumbel Softmax），這些新的數學模型或許可以帶來不同的效果。鑑於端對端訓練為目前深層學習之主流，前述之嘗試皆是值得努力的方向。

在第五章中使用強化學習下訓練分段式語音詞向量雖然已有具體成果，但由於強化學習需要代理人和環境互動來獲得獎勵，此互動過程對於模型訓練過程而言十分花時間。另外由實驗結果可以看出詞切割效能影響口述語彙偵測之效能甚鉅，增進詞切割效能以獲得更佳的口述語彙偵測效能是個需要努力之方向。鑑於上述兩者，尋找更迅速有效的強化學習演算法是增進目前分段式語音詞向量整體效能最直接也最有價值之研究方向。

doi:10.6342/NTU201801733

參 考 文 獻

[1] 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.

[2] 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," in *Interspeech 2016*, 2016, pp. 765–769.

[3] Zhizheng Wu and Simon King, "Investigating gated recurrent networks for speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5140–5144.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[5] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur, "Recurrent neural network based language model.," in *Interspeech*, 2010, vol. 2, p. 3.

[6] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[7] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

doi:10.6342/NTU201801733

[8] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[9] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[10] Thomas Kemp and Alex Waibel, "Unsupervised training of a speech recognizer: recent experiments.," in *Eurospeech*, 1999.

[11] Lori Lamel, Jean-Luc Gauvain, and Gilles Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech & Language*, vol. 16, no. 1, pp. 115–129, 2002.

[12] Alex S Park and James R Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[13] Gautam K Vallabha, James L McClelland, Ferran Pons, Janet F Werker, and Shigeaki Amano, "Unsupervised learning of vowel categories from infant-directed speech," *Proceedings of the National Academy of Sciences*, vol. 104, no. 33, pp. 13273–13278, 2007.

[14] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, "Speech enhancement based on deep denoising autoencoder.," in *Interspeech*, 2013, pp. 436–440.

doi:10.6342/NTU201801733

[15] Andrej Karpathy, Justin Johnson, and Li Fei-Fei, "Visualizing and understanding recurrent networks," *arXiv preprint arXiv:1506.02078*, 2015.

[16] Alex Krizhevsky and Geoffrey E Hinton, "Using very deep autoencoders for content-based image retrieval.," in *ESANN*, 2011.

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

[18] Dong Yu and Michael L Seltzer, "Improved bottleneck features using pretrained deep neural networks.," in *Interspeech*, 2011, vol. 237, p. 240.

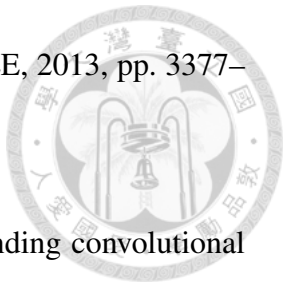
[19] Frantisek Grézl, Martin Karafiát, Stanislav Kontár, and Jan Cernocky, "Probabilistic and bottle-neck features for lvcsr of meetings," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, vol. 4, pp. IV–757.

[20] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[21] Steven Davis and Paul Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[22] Jonas Gehring, Yajie Miao, Florian Metze, and Alex Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Acoustics, Speech and Signal Pro-*

doi:10.6342/NTU201801733



cessing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013, pp. 3377–3381.

[23] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[24] Cheng-Tao Chung, Chun-an Chan, and Lin-shan Lee, "Unsupervised spoken term detection with spoken queries by multi-level acoustic patterns with varying model granularity," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7814–7818.

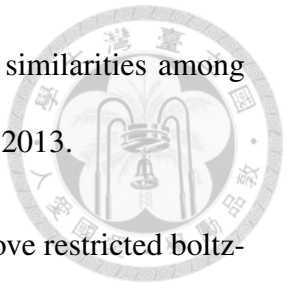
[25] Chia-ying Lee and James Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 40–49.

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

[27] Quoc Le and Tomas Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196.

[28] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2425–2433.

doi:10.6342/NTU201801733



[29] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever, "Exploiting similarities among languages for machine translation.," *CoRR*, vol. abs/1309.4168, 2013.

[30] Vinod Nair and Geoffrey 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.

[31] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[32] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

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

[35] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[36] 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.

doi:10.6342/NTU201801733

[37] Ronald J Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[38] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1889–1897.

[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

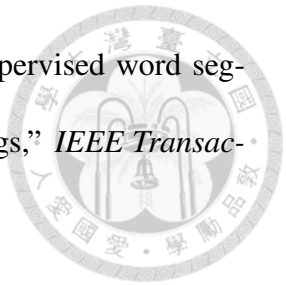
[40] Peter W Glynn, "Likelihood ratio gradient estimation for stochastic systems," *Communications of the ACM*, vol. 33, no. 10, pp. 75–84, 1990.

[41] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng, "Towards end-to-end reinforcement learning of dialogue agents for information access," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, vol. 1, pp. 484–495.

[42] Barret Zoph and Quoc V Le, "Neural architecture search with reinforcement learning," *ICLR*, 2017.

[43] Okko Räsänen, "Basic cuts revisited: Temporal segmentation of speech into phone-like units with statistical learning at a pre-linguistic level.," in *CogSci*, 2014.

doi:10.6342/NTU201801733



[44] Herman Kamper, Aren Jansen, and Sharon Goldwater, "Unsupervised word segmentation and lexicon discovery using acoustic word embeddings," *IEEE Transactions on Audio, Speech and Language Processing*, 1 2016.

[45] Paul Michel, Okko Räsänen, Roland Thiollière, and Emmanuel Dupoux, "Improving phoneme segmentation with recurrent neural networks," *CoRR*, vol. abs/1608.00508, 2016.

[46] Yossi Adi, Joseph Keshet, Emily Cibelli, and Matthew Goldrick, "Sequence segmentation using joint rnn and structured prediction models," *arXiv preprint arXiv:1610.07918*, 2016.

[47] Dac-Thang Hoang and Hsiao-Chuan Wang, "Blind phone segmentation based on spectral change detection using legendre polynomial approximation," *The Journal of the Acoustical Society of America*, vol. 137, no. 2, pp. 797–805, 2015.

[48] Yu Qiao, Naoya Shimomura, and Nobuaki Minematsu, "Unsupervised optimal phoneme segmentation: Objectives, algorithm and comparisons," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 3989–3992.

[49] Chun-an Chan, *Unsupervised Spoken Term Detection with Spoken Queries*, Ph.D. thesis, National Taiwan University, 2012.

[50] David RH Miller, Michael Kleber, Chia-Lin Kao, Owen Kimball, Thomas Colthurst, Stephen A Lowe, Richard M Schwartz, and Herbert Gish, "Rapid and accurate spoken term detection," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

doi:10.6342/NTU201801733

[51] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1," chapter Distributed Representations, pp. 77–109. MIT Press, Cambridge, MA, USA, 1986.

[52] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[53] Zih-Wei Lin, "Personalized linguistic processing: Language modeling and understanding," M.S. thesis, National Taiwan University, 2017.

[54] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig, "Linguistic regularities in continuous space word representations.," in *hlt-Naacl*, 2013, vol. 13, pp. 746–751.

[55] Chia-Hao Shen, "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," M.S. thesis, National Taiwan University, 2017.

[56] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," 2014, vol. 15, pp. 1929–1958, JMLR. org.

[57] Jennifer Fox Drexler, "Deep unsupervised learning from speech," M.S. thesis, Massachusetts Institute of Technology, 2016.

[58] Okko Johannes Räsänen, Unto Kalervo Laine, and Toomas Altosaar, "An improved speech segmentation quality measure: the r-value.," in *Interspeech*, 2009, pp. 1851–1854.

doi:10.6342/NTU201801733

[59] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 2342–2350.

[60] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

[61] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2678–2687.

[62] Yoshua Bengio, Nicholas Léonard, and Aaron Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[63] Yu-Hsuan Wang, Cheng-Tao Chung, and Hung-yi Lee, "Gate activation signal analysis for gated recurrent neural networks and its correlation with phoneme boundaries," *INTERSPEECH*, 2017.

[64] Tanja Schultz, "Globalphone: a multilingual speech and text database developed at karlsruhe university.," in *INTERSPEECH*, 2002.

[65] 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.

doi:10.6342/NTU201801733

附錄一：TIMIT 語料

TIMIT（Texas Instruments/Massachusetts Institute of Technology）是一個常被用來評估語音處理相關效能的英文語料。此語料共有6300個語句，分別由630位語者（Speakers）分別錄下十個語句。在這十個語句中，有兩個語句的內容是所有語者皆相同的，三個語句是每位語者皆不同，不會重複。而最後的五個語句則會被若干個不同的語者錄下。語句的內容涵蓋所有美式英文的音素。標準的TIMIT語料將此6300個語句分為語句內容不重複之訓練資料3696筆以及測試資料192筆。除了語音資料以外，TIMIT也有提供與時間對齊的（Time-Aligned）音素轉寫和詞轉寫（Word Transcriptions）。這些資訊能夠作為我們評估我們非督導式學習模型的效能時的依據。

doi:10.6342/NTU201801733

附錄二：GlobalPhone 語料

GlobalPhone語料是一個含有眾多不同語言的語料，內涵15種不同語言，個別具有超過80位語者的變化。其語料目的除了提供不同語言的語音辨識外，也是為了發展能夠通用於各語言的全球音素集（Global Phone Set） [64]。然而全球的語言超過4500種，收集此語料的主持人修氏（Tanja Schultz）在挑選最能代表世界語言的語言組合時，各種語言皆經過許多層面的考量，包涵語者數量、語者包含的因素範圍、地理範圍等等。本文選擇了其中的三個語言來驗證本文所提出之非督導式學習演算法在英文語言之外的概括性驗證，此三種語言分別為：捷克文，法文與德文，其基本資料如表1。與TIMIT語料不同的是，GlobalPhone語料並無提供與時間對齊的音素轉寫和詞轉寫，因此本文中將使用强迫對齊法（Forced Alignment）來獲得此資訊。

語言	捷克語		法語		德語	
資料集	訓練集	測試集	訓練集	測試集	訓練集	測試集
語者數	82	10	84	8	65	6
句子數	10367	1028	8818	839	8185	1073

表 1: GlobalPhone語料細節

doi:10.6342/NTU201801733