國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

類神經網路聲碼器在語音波形生成上的強健性分析

An Analysis on the Robustness of Neural Vocoders for Speech Waveform

王君璇

Chun-Hsuan Wang

指導教授：李琳山 教授

Advisor: Lin-shan Lee, Ph.D.

中華民國一百零九年七月

July, 2020

doi:10.6342/NTU202001501

國立臺灣大學碩士學位論文
口試委員會審定書

類神經網路聲碼器在語音波形生成上的強健性分析

An Analysis on the Robustness of Neural Vocoders for Speech Waveform

本論文係王君璇君 (R07942076) 在國立臺灣大學電信工程學研究所完成之碩士學位論文，於民國 109 年 7 月 7 日承下列考試委員審查通過及口試及格，特此證明

口試委員：李琳山 (簽名)

(指導教授)

鄭秋豫　王小川

曾仁烋　李宏毅

陳信宏

系主任、所長 蘇炎坤 (簽名)

誌謝

首先特別感謝李琳山老師的指導，感謝建立起整個實驗室的制度和氛圍。建立起每周group meeting，也邀請我擔任DSP助教和專題大助，學習到非常多。不定時的會鼓勵我，讓我更有動力做研究，並時常叮嚀我們碩論的進度，讓我可以如期完成碩論。

特別感謝宏毅老師，在忙到爆炸的情形下，仍每週撥冗討論進度，在conference死線前，實驗出問題，也臨時撥空和我們討論處理的方法。在每週group meeting時也幫忙釐清同學報告節奏太快時的一些不太懂地方，對於paper、碩論的完成都要特別感謝宏毅哥。

也要感謝實驗室的每一位以及縕儂老師實驗室的同學在每週group meeting上優秀的分享，讓我可以看到各式各樣種類、方法的論文，還有同學們的討論都讓我在思辨能力上有所提升。

感謝網管們辛苦的維護戰艦們，從硬體的購買、維護，系統上、套件上的更新，一遇到問題在Github上面開issue也都立馬修復問題，讓我們研究更加順利。

感謝國網中心提供的運算資源，雖然有發生container被重啓，導致我跑的內容不見，不過還是非常感謝可以提供相當大的運算資源，在戰艦忙碌的時候，提供可以開多台GPU平行下去做運算，讓我可以準時的完成paper和碩論。

能完成這份碩論真的非常非常感謝我的好夥伴博竣哥，從邀請我加入他的vocoder的研究，對於實驗設置、資料集的收集、測試時的問卷，都非常仰賴他的幫忙，就算發生了training dataset的出錯和我跑inference時不小心弄錯model參數等，都不慌不忙將問題各個擊破，一起完成這份研究。

也要特別感謝廷緯哥在本實驗貢獻特別多，不僅幫忙實驗的pretrain和測試時，都付出相當多的時間！

doi:10.6342/NTU202001501

感謝達融哥特別在ICASSP前，遠在美國還特別和我們通過網路電話足足花了一小時改paper架構，讓當時整個凌亂的架構重新整理，也變成現在這篇碩論的前身，為我的碩論架構訂定了一個非常好的基礎。

感謝浩然哥坐在我的隔壁時常被我問一些有關Deep Learning的一些trick之類的問題，而且是Pytorch撰寫小達人，如果是來自Pytorch程式debug，他都可以隔空抓藥找到問題。在碩一一起當DSP助教時也幫忙很多。

感謝濤哥，在Speech Synthesis上的大前輩，常常可以詢問他一些TTS上的一些trick，前處理等細節，他無一不自己親身測試過，只要和他學習那些小技巧就可以把語音生成訓練得很完美。有時候幽默的諧音梗也逗得我哈哈大笑。 感謝元瑞哥曾在線性代數課程的大力幫忙，還有多次幫忙我實驗debug的部分。 感謝NLP Master仲翊哥指點RNN的一些技術使用細節，讓我的實驗更快速完成。 感謝培傑在我嘗試做random noise實驗時，大力提供幫忙！ 感謝魁哥、昭誼、柏文一起和我在週一下午的meeting中讓我學到豐富的知識。 感謝思霖哥總共跟我換過2次report、1次intro，在我有事無法完成group meeting報告時義氣相挺，和我換過group meeting報告順序，真的非常感謝！！

另外要感謝上一屆政杰哥、儒杰哥也是語音生成的大前輩提供我很多訓練處理的細節；感謝靖平哥坐在我的隔壁也幫助我很多處理一些preprocess的細節；感謝茅大和奕禎哥也有提供我文章上的意見，讓我可以更有系統的為碩論有更好的雛形。還有感謝實驗室的每一位同學都和我多多少少討論過各式各樣的問題或細節，再次感謝同學們的幫忙，也讓我學習到相當多。

真的非常感謝所有幫我填問卷的好朋友們，很多朋友們其實也沒有到非常熟，甚至有高中時期隔壁班的同學，但在我的facebook tag之下，就算在無償的情況下，他們非常義氣幫我填寫問卷，有些人甚至填寫不只一份實驗的問卷，真的

doi:10.6342/NTU202001501

非常感謝他們的幫忙，才能促成我這次實驗順利完成。

在疫情之下，碩論的撰寫幾乎都在家中完成，要特別感謝我的媽媽在家裡每餐提供我食物，偶爾晚上一起看YouTube，是自幼以來待在家裡時間最長的一次，提供我舒適的環境和動力完成整份碩士論文。

感謝所有人的幫忙，少了任何一個人，碩論的完成都一定艱辛非常多，再次感謝大家！

doi:10.6342/NTU202001501

摘要

聲碼器為一種可以將聲學特徵值轉換至音訊波形的架構，目前以深層學習為基礎的聲碼器已被廣泛用在語音生成應用中，其中包含文句翻語音系統和語者轉換系統。 不過當遇到訓練和測試的資料分佈不一致時，以深層學習為基礎的聲碼器的表現會大幅下降。

這篇碩論主要在探討不同以深層學習基礎的聲碼器，對於訓練和測試的資料分佈不一致時的生成語音品質。 本碩論所探討的聲碼器包含WaveNet, WaveRNN, FFTNet, Parallel WaveGAN。

我們首先測試聲碼器的訓練及測試是在不同語者和不同語言時的影響。 當聲碼器分別訓練在單語者單語言、多語者單語言、多語者多語言的訓練集，並分別測試在相同語者相同語言、不同語者相同語言、不同語者不同語言時，發現影響聲碼器輸出品質最大的因素是語者的多樣性，不同語言則不會影響生成結果。 此外，我們也分析聲碼器在訓練在單語者語言的情況下，發現不同性別也會大幅影響聲碼器的輸出。

這篇碩論也將聲碼器應用在其他語音生成運用上，發現WaveNet, WaveRNN最適合使用在文句翻語音系統上；Parallel WaveGAN最適合使用在語者轉換系統上。

doi:10.6342/NTU202001501

Contents

doi:10.6342/NTU202001501

doi:10.6342/NTU202001501

圖目錄

doi:10.6342/NTU202001501

表目錄

doi:10.6342/NTU202001501

第一章　導論

1.1 研究動機

自2007年第一個手機語音助理Siri出現後，隨著行動裝置的普及和技術的進步，手機內建的語音助理Siri或是Google Assistant變得越來越強大，能完成的任務也越多。 人們也開始改變使用習慣，透過由語音助理對話來完成需用手操作事務，以提高操作效率。

語音助理除了具有處理事務的能力，更建立起一種新的機器和人溝通的介面， 不僅僅是通過文字來傳達要做的任務，更希望機器發出清晰、接近人的聲音，就好像真的是一位好朋友在對你說話，讓人更有親切感，而不再只是冰冷冷的機器。 目前各大公司也紛紛致力於做出更有親和力的語音助理，讓語音助理更具有人性。 早期Google小姐機械似語音發音，給人疏遠而刻意的感受。 隨著機器學習發展迅速，目前語音合成的技術已經可以表現出相當完美的抑揚頓挫，就算是機器生成的語音也能更加貼近人們。

數位化的音訊是一個隨時間有大小起伏的數值序列，維度相當龐大，一段CD的音訊檔中，短短一秒之中有44,100個數值點，電話中傳的音訊檔每秒也有至少16,000個數值點。 音訊檔內的數值點總個數非常多，因此在一般語音生成的任務中，不會直接產生音訊檔，會造成實行上非常困難。 為了降低生成難度，語音生成通常會先生成中繼的聲學特徵值(Acoustic feature)，再通過另一個模型把聲學特徵值轉換成人耳聽見的音訊波形。 這種將聲學特徵值轉成音訊波形的模型稱之為聲碼器(Vocoder)。

隨著語音生成的應用越來越成熟且產品化，雲端上的文字資料可以直接轉換成聲學特徵值。 新型的智慧型手機都內建著聲碼器，能快速從聲學特徵值轉換成

doi:10.6342/NTU202001501

人耳聽見的音訊波形，並且十分貼近自然人的聲音提供給使用者。

目前常見的聲碼器是以深層學習架構，生成品質相當接近人聲，且能廣泛運用在語音生成應用上。 深層學習是目前發展相當快速的領域，在各種類型任務中都有突破性的表現。 不過深層學習其中一個嚴重的缺點，是在訓練和測試資料分布不完全一致時，其結果會有相當大的落差。

聲碼器是一種將聲學特徵值轉換成語音的對應關係，理論上完美的聲碼器和訓練時的語者沒有關係，不過限於深層學習的模型限制，可能會受到訓練時語者的性別、語言、聲音特徵等而表現不如預期。 此外，在語音生成的任務中，如果前面的模型所產生的聲學特徵值並非完美，其輸出結果再通過聲碼器時，是否仍能盡量接近人聲也是一個非常重要的問題。

本篇論文想要探討的主題，是希望藉由實驗來比較出聲碼器的强健性(Robustness)，也就是比較聲碼器在訓練和測試時的資料分佈不同步的結果。 首先分析遇到訓練、測試的語者不同、語言不同、性別不同的情況，進而將聲碼器應用在不同的語音生成任務。

本論文主旨希望了解到每種聲碼器適合的運用的場景、任務。聲碼器需要什麼訓練資料才可以讓最後生成的語音更完美。

1.2 研究方向

本論文研究的主軸為聲碼器的强健性比較，從聲碼器的架構比較出發，並以實驗數據作為比較的基礎。其中强健性的比較主要分為兩大方向：

1. 人聲的聲學特徵值還原成音訊：

   在本主題中，又分成兩個實驗探討其聲碼器强健性：

doi:10.6342/NTU202001501

- 語言：聲碼器是從聲學特徵值轉換成語音的變換，理論上和語言的關係比較小，本文通過比較訓練和測試在不同語言或是不同語者，比較聲碼器會不會有過度貼合(overfitting)的現象，只在看過的語言表現較好，而在沒看過的語言表現較差，進而比較其强健性。
- 性別：聲碼器的訓練語料若為單一性別，以實驗數據觀察其在不同性別下，聲碼器會不會只在訓練的性別表現較好，用以比較强健性。

2. 聲碼器運用至語音生成運用：

- 聲碼器是一個作為語音生成應用中後處理的部分，當語音生成應用其最後輸出不是以音訊波形為輸出，而是以聲學特徵作為輸出的話，就需要聲碼器來當作最後一步轉換的步驟。
- 在語音生成應用中，會偶而發現輸出的聲學特徵不完美，若其結果通過聲碼器輸出時不會造成整個音訊聲波檔嚴重破壞仍能保持不錯，可以當成聲碼器的强健性比較。
- 本主題是用兩個已經訓練好的語音生成應用再經過不同聲碼器，比較聲碼器的結果輸出。本論文所比較聲碼器在語音生成應用有：文句翻語音系統和語者轉換系統。

1.3 主要貢獻

透過實驗比較，本論文的主要貢獻如下：

- 分析多種聲碼器在遇到沒看過的語言和性別的强健性比較
- 將聲碼器應用在語音生成上，並比較其結果表現

doi:10.6342/NTU202001501

- 分析哪一類型的訓練資料，可以有高品質的語音生成結果

1.4 章節安排

本論文之章節安排如下：

- 第二章：介紹本論文相關背景知識。
- 第三章：介紹本論文要比較的所有聲碼器
- 第四章：多種聲碼器在人聲聲學特徵的强健性比較。
- 第五章：多種聲碼器在語音生成應用上的强健性比較。
- 第六章：本論文之結論與未來研究方向。

doi:10.6342/NTU202001501

第二章　背景知識

2.1　深層類神經網路(Deep Neural Network)

2.1.1　簡介

類神經網路是啓發於生物神經系統中的一種數學模型， 可用於函數的估計和近似。 若將所需求解的問題化成給定輸入、輸出的型式，求其中間的函數模型，我們便可以使用深層類神經網路當作一個給定的黑盒子(black box)，用問題的輸入、輸出去估計該黑盒子的函數模型。

基本單元–神經元

在生物的神經系統中，神經元(Neurons)由樹突、軸突連結，分別接受其他神經元的刺激和傳遞本身的激發狀態。而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)，並對輸入值分別乘以不同的權重(weight)再加上一個偏移值(bias)，最後通過一個非線性激活函數(activation function)，當作該神經元的輸出(output)。透過數學式表示：

$$y = \sigma(\sum_{i=1}^{n} w_i x_i + b) \tag{2.1}$$

其中輸入訊號為$x_1, ..., x_n \in \mathbb{R}$，$w_i \in \mathbb{R}$是對於第$i$個輸入的權重，$b \in \mathbb{R}$代表偏移值，輸出訊號為$y \in \mathbb{R}$，非線性激活函數為$\sigma : \mathbb{R} \to \mathbb{R}$。可將輸入、權重值改寫成向量內積形式：

$$y = \sigma(\mathbf{w}^T \mathbf{x} + b) \tag{2.2}$$

神經元是深層類神經網路的函數基本單位。 設置好非線性激活函數σ，$\mathbf{w} \in$

doi:10.6342/NTU202001501

$\mathbb{R}^n, b \in \mathbb{R}$是該神經元能表現的所有函數。 深層學習的目標是藉由讓神經元在給定輸入$\mathbf{x} \in \mathbb{R}^n$、輸出$y$，學習最合適的$\mathbf{w}$和$b$當作我們想要表現的函數模型。

層狀結構

單一神經元所表現的函數空間(function space)有限，因此通常會將多個神經元並排成一層，並將每一層以上一層的訊號做為輸入，形成層狀傳遞的結構，如圖2.1所示。 透過層狀結構的神經元可以表現的函數變更加豐富。這種結構稱為全連接層(Fully Connected Layer)。

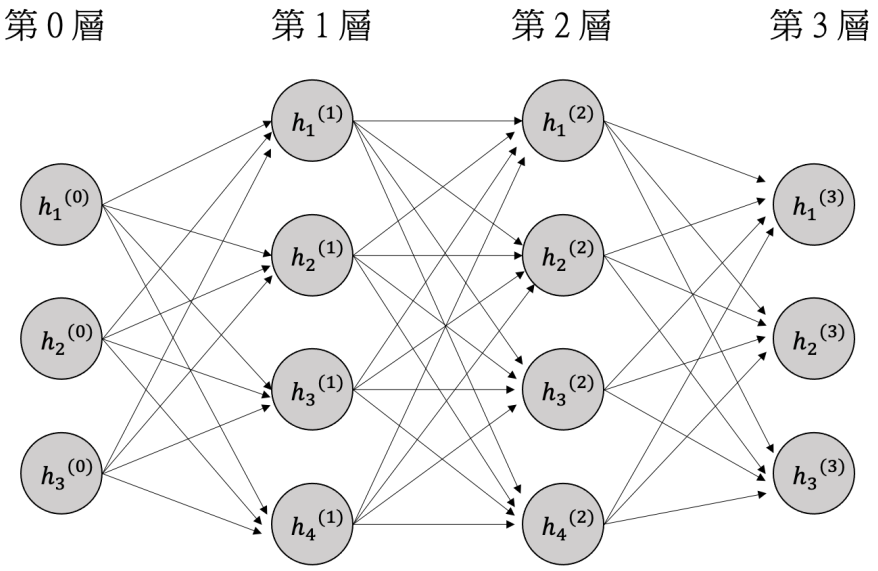


圖 2.1: 全連接層示意圖

一個深層類神經網路的最開始會直接輸入資料的數值，稱為輸入層(input layer)，最後一層則為輸出層(output layer)，剩餘的中間層稱為隱藏層(hidden layer)。 以圖2.1為例，第0層為輸入層，中間第1, 2層為隱藏層，第3層為輸出層。層狀結構還有一個非常重要的好處，是可以將層跟層之間的計算寫成矩陣乘法。

第i層和第$i-1$的關係可以寫為：

$$h^{(i)} = \sigma(Wh^{(i-1)} + b) \tag{2.3}$$

doi:10.6342/NTU202001501

其中 W, b 是第i層神經元參數並排成矩陣和向量，而σ以 $\sigma(x)_i := \sigma(x_i)$的方式擴展。

由於平行化計算的發展，圖形處理器(Graphics Processing Unit, GPU)可以非常快速地計算矩陣乘法，加速類神經網路的運算。

2.1.2 類神經網路訓練

類神經網路的訓練是通過改動神經元內的參數，也就是權重和偏差值，讓模型的輸出更靠近真實的數據結果。以θ表示模型的所有參數。輸入x經過類神經網路$F(\cdot;\theta)$後，輸出層會產生對應的輸出$F(x;\theta) = \hat{y}$。

為了衡量神經網路的表現，我們會制定損失函數(Loss Function)來定義類神經網路的表現。 損失函數$L(\hat{y}, y; \theta)$會比較真實數據y, 和模型輸出$\hat{y}$來觀察在參數θ下模型的表現。

常見的損失函數

- 數值預測：

  目標模型輸出$\hat{y}$和原始輸出y越接近表示模型表現越好，因此定義損失函數隨著y和$\hat{y}$越靠近時，損失函數$L(\hat{y}, y; \theta)$的值也越小。 常見損失函數是通過定義向量距離的方法來表示：有曼哈頓距離(Manhattan distance)和歐幾里得距離(Euclidean distance)，數學表示式分別為式2.4和式2.5，其中$\hat{y}, y \in \mathbb{R}^n$。

$$d_1(\hat{y}, y) = \|\hat{y}, y\|_1 = \sum_{i=1}^{n} \mid \hat{y}_i - y_i \mid \tag{2.4}$$

$$d_2(\hat{y}, y) = \|\hat{y}, y\|_2 = \sqrt{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \tag{2.5}$$

doi:10.6342/NTU202001501

- 分類問題：

  目標模型輸出$\hat{y}$對於種類預測機率分佈要和真實機率y越接近越好。 真實機率的表示法為將真實標注的資料以獨一餘領(One-Hot Encoding)表示，也就是C個類別表示為C維，對應類別機率給定為1，其餘維度給定為0。 此外，模型輸出結果為了讓所有類別的機率總和為1，輸出前通過軟性最大化函數(softmax function)來達成。

  而當機率分佈越接近時，我們會讓損失函數$L(\hat{y}, y; \theta)$越小。常見會使用透過交叉熵(Cross Entropy)如下。其中N是資料個數。

$$L_{CE}(\hat{y}, y; \theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} \hat{y}_j^{(i)} \ln y_j^{(i)} \tag{2.6}$$

損失函數值越小表示類神經網路的表現越好，模型訓練的目標是嘗試找到一組參數儘可能最小化損失函數，屬於一種優化問題(optimization problem)。

訓練方法

常見的方法會使用梯度下降法(Gradient Descent Algorithm)來解這種優化問題。 梯度下降法首先需要計算損失函數之於參數的梯度(gradient)，接著將模型參數往此梯度的相反方向做更新，以此來降低損失函數。 藉由不斷迭代更新參數，損失函數會逐漸接近局部最佳解(local optimal)。

反向傳播演算法(Backpropagation Algorithm)常被用來對類神經網路做更高效率的參數更新。 此演算法在計算完損失函數後，會利用連鎖率從最後一層神經元的梯度，往前一層的神經元傳遞，直到算出第一層的神經元參數為止。 整個過程就好像是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。

doi:10.6342/NTU202001501

梯度下降法有一些限制，比方說損失函數必須要可微分(differentiable)，只能收斂至局部最佳解，而非極小值，但是梯度下降法因為使用上方便，而且通常只要選擇合適的學習率(learning rate)就可以有不錯的結果，所以變成現今最常使用的深層模型的訓練方法。

2.1.3 卷積式類神經網路 (Convolutional Neural Network, CNN)

卷積式類神經網路 [1]是一種深層類神經網路的變形，是設計讓音訊、影像的資料格式保有非時變性、空間不變性(shift invariant)，也就是如果輸入在時間軸或空間軸上平移，則輸出結果也會是對應的平移。

基本上卷積式類神經網路就是全連接層的一種變形， 以一維卷積式類神經網路為例，可以看成是以數值序列取代實數做為基本操作單位的全連接層，並將乘法以一維卷積取代，一般使用通道(Channel)來稱呼卷積層中的每個神經元。以數學式表示，假定輸入訊號$x_1, x_2, ..., x_n \in \mathbb{R}^T$為$n$個有$T$個時間點的數值序列，可寫成

$$y = \sigma(\sum_{i=1}^{n} x_i \star w_i + b) \tag{2.7}$$

其中$w_i \in \mathbb{R}^k$為該通道的內核(Kernel)，其寬度為k。且$y \in \mathbb{R}^k$。

2.1.4 遞迴式類神經網路 (Recurrent Neural Network)

遞迴類神經網路的隱藏層具有「記憶」功能，除了考慮當下時間點的資訊外，也會參考過去所輸入的資訊，因此對於處理文字和聲音訊號這種時間序列經常使用此類型的類神經網路。 以下我們將分別介紹遞迴式神經網路的基本數學原理。

遞迴式類神經網路 [2]是設計處理序列式資料，且每一個時間點的資料非彼此

doi:10.6342/NTU202001501

獨立。遞迴式類神經網路在輸入第t個時間點的資訊時，前t-1個時間點重要的資訊也保留在其隱藏狀態(Hidden State)。 可透過數學表示式：

$$c_t = \sigma_h(W_c x_t + U_c c_{t-1} + b_c) \tag{2.8}$$

$$h_t = \sigma_h(W_h c_t + b_h) \tag{2.9}$$

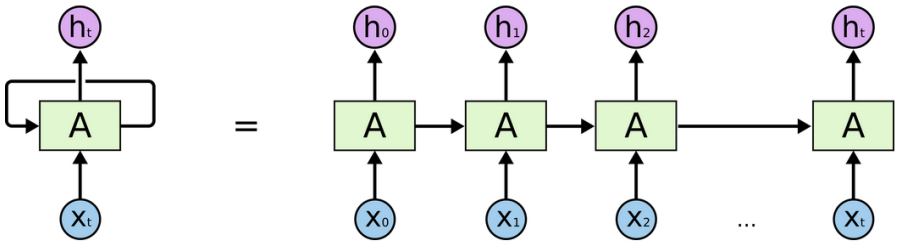


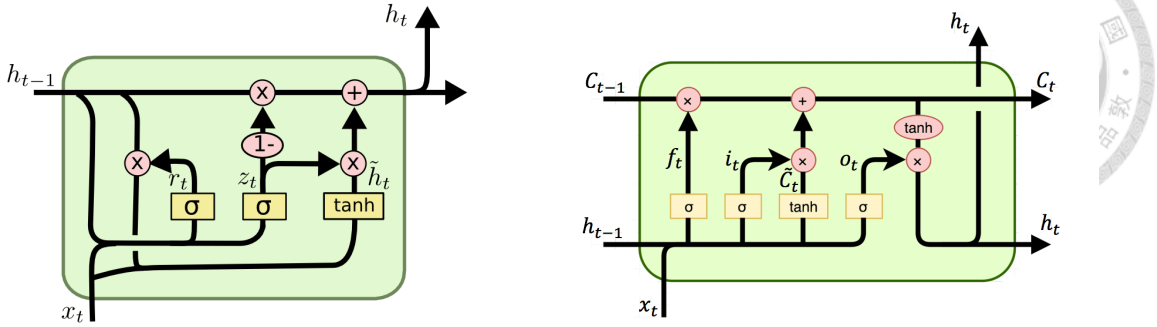
圖 2.2: 遞迴式類神經網路

圖2.2為基本的遞迴式類神經網路，圖中的A是指該網路的記憶單元，而此記憶單元在時間點t有輸入x_t，經過一連串的矩陣變換如式2.8, 2.9，爾後產生輸出h_t，如此一來，在輸出h_t時，此網路不只會考慮該時間點的輸入x_t，更會融合時間t之前所保留下來的資訊。 因此在訓練類神經網路時，此網路不只會學習到輸入和輸出之間的對應關係，更會學習到哪些資訊需要保留下來，而哪些資訊可以捨棄。

變化型

長短期記憶單元LSTM [3]是更加複雜的遞迴式類神經網路的組成單元，其透過不同的閘門(gate)來控制網路中資訊的流動，而閘門的設計是輸出介於0到1的S函數，其值代表這個閘的輸出和輸入的比例。 如圖2.3(b)，共有三種閘門，分別是輸入閘門(input gate)、輸出閘門(output gate)和遺忘閘門(forget gate)，輸入閘門負責控制輸入的資訊量有多少需要存入記憶，輸出閘門負責決定多少比例的記憶需

doi:10.6342/NTU202001501



(a) GRU (b) LSTM

圖 2.3: 遞迴式類神經網路變化型

要用來輸出，而遺忘閘門則負責決定多少比例所記憶的資訊可以被拋棄，如此一來，梯度便只會影響到閘門開啓的時候，而減少梯度消失的問題。 圖2.3(a)中的GRU [4]則是簡化讓輸入閘門和遺忘閘門連動，藉此降低參數量。

2.2 生成對抗網路(Generative Adversarial Network, GAN)

2.2.1 簡介

生成對抗網路 [5]有兩個模組：生成器(Generator)及鑑別器(Discriminator)，生成器的目的是將一個取樣出來的高斯雜訊(Gaussian Noise) z ，映射到一個真實資料的取樣點上，而鑑別器的目的則是希望能將生成器產生出來的機率分佈與真實資料的機率分佈加以區別。

2.2.2 架構

如圖2.4所示，生成器的訓練目標是根據高斯雜訊 z 生成更真實的圖片來騙過鑑別器，而鑑別器則是在判別輸入的圖片是來自於真實世界或是生成器所產生的圖片，透過這種兩兩相互對抗的訓練方式的反覆迭代(Iteration)，來希望生成器最終

doi:10.6342/NTU202001501

圖 2.4: 生成式對抗網路

可以生成與真實資料非常相近的圖片。由於兩者的訓練目標是相反的，所以通常會用一個最小值最大化(Minimax)的問題來描述生成對抗網路，而這個問題的價值函數(Value Function)可以用下列的數學式表示：

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}}[\log(D(X))] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \quad (2.10)$$

2.3 自回歸模型(Autoregressive Model)

2.3.1 簡介

在統計、計量經濟學和信號處理中，自回歸模型代表了一種隨機過程。 自回歸模型是一種處理時間序列的方法，用同一變數x在t之前的所有時間點的值來預測現在時間點的值，亦即用x_1至 x_{t-1}來預測本期x_t的表現。

2.3.2 深層學習架構下的自回歸模型

在深層學習的架構中，我們可以將前面所有時間點的資訊通過一個深層學習的模組，和這個時間所得到的資訊一起再度通過一個深層學習的模組預測出這個時間點的資訊。

遞迴式類神經網路是一種經典的自回歸模型，因為在原本隱藏層(hidden state)會存有前$t-1$個時間點的資訊，會透過這些資訊來輔助預測出當下時刻的結果。 目前也有透過卷積式類神經網路來達成自回歸模型 [6]會在章節3.2.2有更進

doi:10.6342/NTU202001501

一步的描述。

2.3.3 自回歸模型在語音生成運用

語音是一種連續變化的信號，因此可以預期若模型存有上一個時間點的訊息，對於預測這個時間點的訊息有相當程度的幫助。

本論文所要探討的聲碼器是從聲學特徵值至語音的轉換的模型，由於聲學特徵值的維度遠小於語音的維度，因此在生成過程中有相當的資訊由必須由模型自己創造(generate)。 由於語音是連續變化的信號，因此我們預期擁有自回歸能力的模型通常表現會較純粹轉換的模型表現略好，也會在後續章節實驗中被證實。

自回歸模型的一個致命性缺點是生成所需花費的時間非常高，因為在該時刻點要生成資訊時，必須要將前面所有時間點的資訊生成完畢才能開始，整個過程一定要是依序的，生成時間就會大幅提高。 因此其高品質的輸出和所需要的時間可說是一種取捨(trade-off)。

2.4 量化(Quantization)

2.4.1 簡介

量化在數位信號處理領域是指將信號的連續類比取樣值（或者大量可能的離散取樣值）近似為有限多個（或較少的）離散值的過程。量化主要應用於從連續類比信號到數位信號的轉換中。連續信號經過取樣成為離散信號，離散信號經過量化即成為數字信號。注意要取得離散信號並不需要經過量化的過程。信號的取樣和量化通常都是由類比數位轉換器(Analog-to-digital converter, ADC)實現的。

CD音頻信號就是按照44,100 Hz的頻率取樣，按16位元量化為有著$65,536(=$

doi:10.6342/NTU202001501

2^{16})個可能值的數位信號。

量化就是將聲音波形的取樣值轉換為最接近的刻度值，表示取樣值的二進位位元數決定了量化的精度。量化的過程是先將整個幅度劃分成有限個小幅度（量化階距）的集合，把落入同個階距內的樣值歸為一類，並賦予相同的量化值。

2.4.2 μ-法則量化(μ-law quantization)

透過脈衝編碼調變(Pulse-code modulation, PCM)來壓縮語音訊號的方法 [7]在傳輸上很早就有相關的研究，這種壓縮方法可以降低一個語音訊號的範圍，在類比訊號中，用這樣子的方法可以增加傳輸過程的訊噪比(signal-to-noise ratio)。

給定訊號x，經過μ-法則量化後的結果是

$$F(x) = \text{sgn}(x)\frac{\ln(1+\mu \mid x \mid)}{\ln(1+\mu)} \qquad -1 \leq x \leq 1 \tag{2.11}$$

將其μ-法則量化壓縮訊號反轉回來的反函數為

$$F^{-1}(y) = \text{sgn}(y)\frac{(1+\mu)^{|y|}-1}{\mu} \qquad -1 \leq y \leq 1 \tag{2.12}$$

除了在傳輸系統中可以使用μ-法則量化的壓縮技術，也可以在語音生成任務中將其概念引入。

將語音生成任務中，產生的時間函數$f(t) \in \mathbb{R}$，變成時間函數$g(t) \in [0, 1, ..., 2^N]$，亦可視為一種時間上的$2^N$個數的分類問題。因此對分類問題所使用的損失函數也可以使用在以μ-法則量化後的語音生成上。

doi:10.6342/NTU202001501

2.5 語音生成應用

2.5.1 簡介

語音生成已經是一個愈來愈重要的課題。最直接能想到的應用有文句翻語音系統，對於不方便閱讀文字時，可以使用語音來獲取資訊。或是對於有失語症、或是口腔有問題無法完美發出正確音調聲音，也可以幫助他們。以下提出本論文會使用到的語音生成運用。

2.5.2 文句翻語音系統(Text-to-Speech System)

語音波形包含語音內容、連續性、聲調、口吻、音色等多要素。直接給定文字輸入直接輸出聲音波形對於深層學習模型來說，通常難度比較大，不容易將所有特徵都表現得很好。

目前常見的文句翻語音系統是由一整套上下游模型(pipeline) [8]建立而成。 引入一種中介生成物為聲學特徵值(acoustic feature)，當成文句和語音中間的橋樑，藉此希望降低生成的難度並提高生成的品質。 在這種架構之中，我們通常會先建立一套模型做文句至聲學特徵值的模型，再建立出一套從聲學特徵值至音訊波形的模型，接著再將兩者相接當作最後的模型。

2.5.3 語者轉換系統

語者轉換系統 [9]是將原本語者所講的內容保留，而將原始語者轉換至目標語者上。

在語音生成的運用中，高品質的資料搜集非常困難，要對於生成文句翻語音系統的資料集，需要非常乾淨，不能有任何一點雜訊或回音，否則會影響到最終

doi:10.6342/NTU202001501

生成品質。 不過如果我們可以達到語者轉換的目標，訓練語者轉換所需要的資料相較於文句翻語音系統可以不需要那麼大量高品質資料， 因此透過一個好的語者轉換系統可以達到我們想要不同音色的語音生成。

2.6 本章總結

本章節分成三個大主軸：

1. 介紹本論文會用到的深層學習的背景知識:

   在這一章介紹本論文所會使用到的各種深層學習的模型的架構與原理。 從基本的全連接層(Fully Connected Layer)導入，介紹類神經網路的訓練方式，接著介紹其他類神經網路架構卷積式類神經網路(CNN)、遞迴式類神經網路(RNN)。 爾後介紹生成對抗網路這種特別的訓練方式。

2. 語音生成會用到的一些數學工具：

   自回歸模型、量化方法，在接下來的章節會更進一步使用這些數學工具來完成聲碼器的訓練。

3. 語音生成運用的簡介：

   在這一章介紹本論文會使用到的語音生成運用功能概述，包含了文句翻語音系統和語者轉換系統。

doi:10.6342/NTU202001501

第 三 章　聲碼器比較

3.1　簡介

語音生成過程中，為了讓語音生成難度降低、品質提高，我們通常會先生成較低維度的聲學特徵值當作中介值，最後再經由聲碼器生成我們最後語音生成的結果，此種兩步驟式的方法可以更提高生成品質。

低維度的聲學特徵值的表示方法有很多，常見的方法是將原始語音訊號通過數學轉換(Transform)，使其維度降低。 根據消息理論(Information Theory) [10]，壓縮必定會喪失一部分的資訊，而導致資訊不能完全被回復。 因此需要模型將壓縮過的資訊還原成近似原本的音訊波形，而將聲學特徵值還原成音訊波形的模型即為聲碼器(Vocoder)。

3.1.1　聲學特徵值

聲學特徵值是將原本音訊檔壓縮過的數值表現。 其中常見的做法之一是取時頻譜的大小值作為聲學特徵值。

時頻譜的取得是先將訊號在時域截短為多段分進行短時(short-time)傅立葉轉換，爾後取其轉換過後的大小(magnitude)當作成果，過程中丟棄其相位(phase)。其中傅立葉轉換(Fourier Transform) [11]是將時域的資訊轉至頻域的轉換方式，屬於無壓縮的轉換方式。 不過取其大小，而丟棄相位的步驟會將資訊量直接減半。

梅爾時頻譜(Melspectrogram)是將時頻譜通過梅爾標度濾波組(mel-scale filter banks)，梅爾濾波組是一種模仿人耳的濾波器，這會將維度從傅立葉轉換維度降至濾波組個數，會大幅降低維度。 梅爾時頻譜(Melspectrogram)是一種非常常見的

doi:10.6342/NTU202001501

聲學特徵值的方法，也是本論文中所使用的聲學特徵值。

在本論文中聲學特徵值的抽取細節均依照表3.1。

前强調(Pre-emphasis)	0.97
音框長度(Frame Length)	800 數值點
音框移動(Frame Shift)	200 數值點
視窗類型(Window Type)	漢氏窗(Hann Window)
取樣率(Sample Rate)	22,050 Hz
梅爾時頻譜維度(Melspectrogram Dimension)	80

表 3.1: 梅爾時頻譜參數

3.1.2 聲碼器

聲碼器的目的是將聲學特徵值轉換成音訊波形，並且希望音訊波形越接近人聲越好。

若聲學特徵值為時頻譜，可以透過迭代的方式估計出頻譜的對應相位，也就是章節3.2.1的葛芬-林演算法。 倘若聲學特徵值不為時頻譜的話，葛芬-林演算法則無法使用。

其他常見的方法是使用以深層學習為基礎的方法，且輸入只要能表示成任何形式的聲學特徵值都可以估計出音訊波形，在以下會分別介紹：

- 章節3.2.2介紹卷積式類神經網路的WaveNet，是達到自回歸模型的先驅者。
- 章節3.2.3介紹WaveRNN透過兩段式的方式預測，先預測比較粗略的結果，再比較精細地將正確結果預測出來。

doi:10.6342/NTU202001501

- 章節3.2.4介紹將WaveNet改良成輕量版的FFTNet，不僅需要的記憶體大小降低，也可以縮短在生成時所需花費的時間。

- 章節3.2.5介紹以對抗式生成訓練的聲碼器。

3.2 聲碼器架構比較

3.2.1 葛芬-林演算法(Griffin-Lim Algorithm)

葛芬-林演算法 [12]是一種從時頻譜(Spectrogram)重建成音訊的模型。其主要想法是透過迭代的方式，給定時頻譜去重建相位的資訊。 不過葛芬-林演算法的限制在於只能應用在聲學特徵值為時頻譜時，若聲學特徵值不為時頻譜，則無法重建原始音訊。 生成出來的結果聽起來比較乾淨但有機械音的感覺。

在本篇碩論除了章節5.3以外，我們所使用的聲學特徵值均為梅爾時頻譜，因此無法直接使用葛芬-林演算法。 不過在章節5.3會將葛芬-林演算法和其他聲碼器做比較。

葛芬-林演算法實作方式

給定一時頻譜S，欲重建一訊號，使此重建訊號的時頻譜愈接近S愈好。 設x_i是第i次迭代的訊號， F為短時距傅立葉變換，F^{-1}是反向短時距傅立葉變換。 S_i, P_i分別代表x_i的短時傅立葉轉換的大小及相位 即 $F(x_i) = S_i e^{jP_i}$, $j = \sqrt{-1}$ 重建過程如下：

doi:10.6342/NTU202001501

1. 隨機初始化P_0，則$x_0 = F^{-1}(Se^{jP_0})$

在第i次迭代：(重複以下步驟2-4直到滿足迭代停止條件)

2. 對x_i做短時距傅立葉變換取得大小和相位，即$F(x_i) = S_i e^{jP_i}$

3. 將$S_i e^{jP_i}$中的大小S_i以S取代

4. 重建訊號$x_{i+1} = F^{-1}(Se^{jP_i})$

3.2.2 波網模型(WaveNet)

WaveNet [6]是一種使用卷積式類神經網路來達到自回歸模型的創舉。它受到PixelRNN [13] 在影像應用上生成像素點(pixels)的啓發，WaveNet 將相似的概念應用於語音生成上，並且可以處理精度(resolution)至少為每秒16,000個取樣值(sample value)的訊號。它把聲學特徵值輸入當作模型考慮的條件，並在生成過程中將前一時刻的輸出當作下一刻的輸入，但在訓練過程中是將整段音訊同時在輸入、輸出對應裡一同訓練，使得自回歸的架構可以在訓練時被平行化。

不過自回歸的性質加上很深的架構，導致生成速度非常緩慢。爾後有多種方法嘗試解決生成速度緩慢的缺點，例如：Fast WaveNet [14]、Parallel WaveNet [15]等等。

WaveNet的基本架構為擴展因果卷積層(dilated causal convolution layers)，如圖3.1，藍色的輸入表示一個一個連續的音頻訊號取樣點(一般音型波形被記錄在電腦的方式)，使用這種架構的性質有：

- 因果性(Causality)：每個在時間點的訊號都是根據過去的輸入點所產生，不會看到未來的資訊。以數學表示式在$t+1$時刻點的機率為$p(x_{t+1}|x_1, ..., x_t)$，不會看到 $x_{t+1}, x_{t+2}, ...x_T$等未來的資訊。

doi:10.6342/NTU202001501

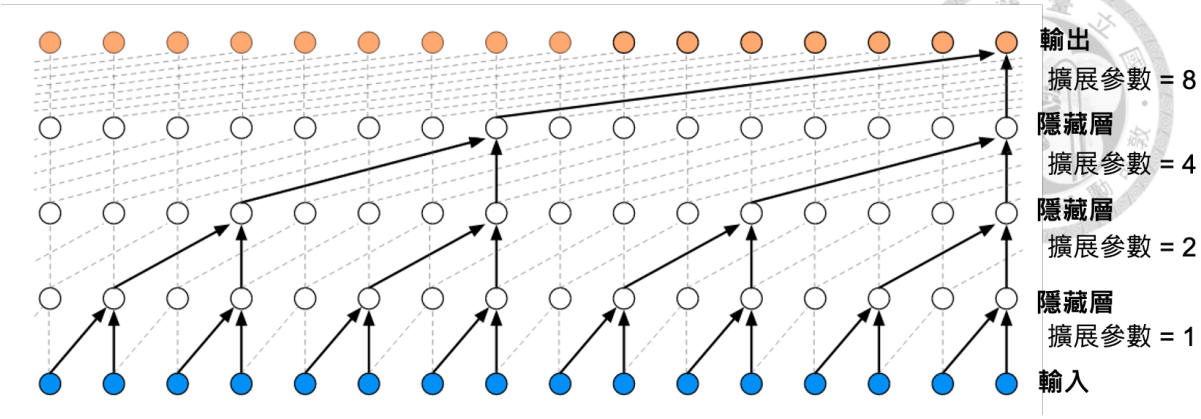


圖 3.1: 擴展因果卷積示意圖

- 擴展卷積(Dilated convolution): 相較於一般的卷積式類神經網路，擴展卷積在做卷積運算時會忽略特定時刻點的輸入，雖然會讓看到的輸入點比較稀疏，但是同樣運算時間可以看到時間軸上更廣闊的輸入點，擴大感受面(receptive field)。

- 訓練快速: WaveNet模型架構不是遞迴式的，在訓練時會針對整句的取樣點，直接做平行化的訓練。因此在計算損失函數時不需要通過遞迴式運算才能求出，故訓練時遠比RNN結構快。 不過在生成過程(Inference)，因為自回歸模型的性質，每個預測出來的輸出都會當作輸入再送回模型裡，速度會較慢。

為了讓WaveNet可以使用更多層的架構 (更深)，以及收斂速度更快， 如圖3.2，WaveNet使用殘差(Residual)的架構以及跳躍連接(Skip-connections)。 通過一個因果卷積操作之後，使用了一個閥門進行控制：

$$z = tanh(W_{f,k} * x) \odot \sigma(W_{g,k}, *x) \tag{3.1}$$

最後的結果根據每一層的輸出的中間結果進行疊加後得到。

在計算損失函數時是使用章節2.4.2的方式將音訊波形化成分類問題，並使用

doi:10.6342/NTU202001501

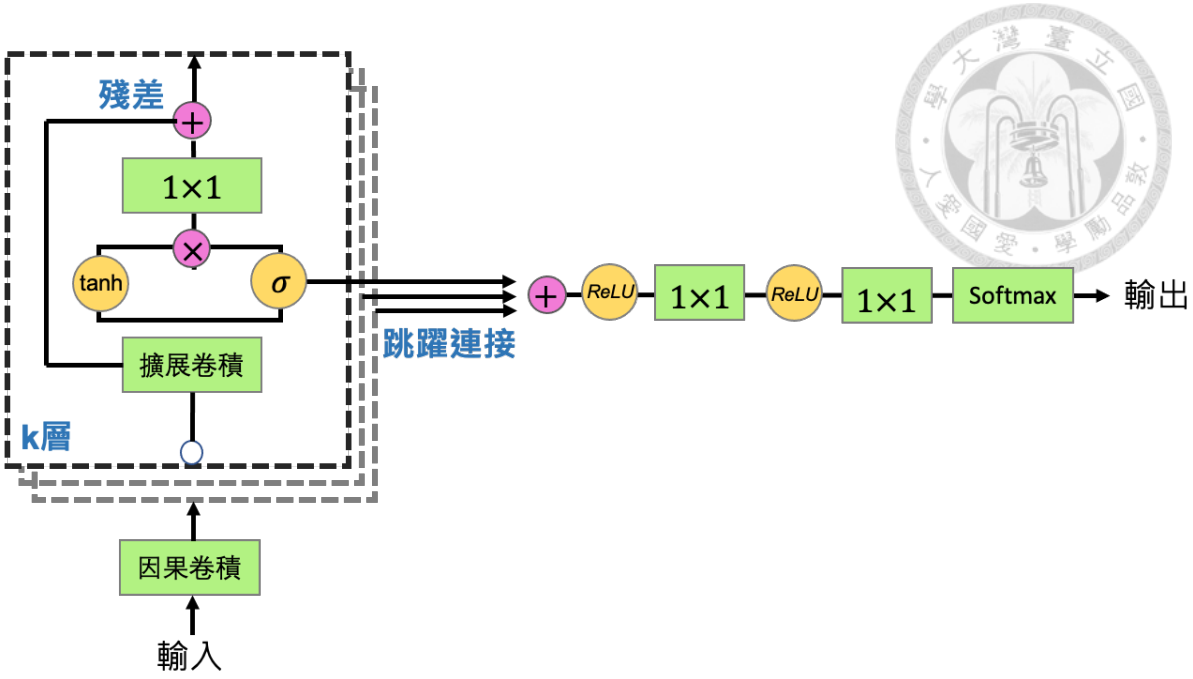


圖 3.2: WaveNet的殘差模組

交叉熵的方式來計算損失函數，再通過章節2.1.2來更新參數。

3.2.3 波遞迴類神經網路模型(WaveRNN)

WaveRNN [16]是一個輕量化的聲碼器，非常快速而且即時，在設計時主要就是設計給手機等邊緣運算(edge computing)應用所使用的。 雖然WaveRNN沒有像WaveNet一樣將前面時間點的取樣資訊再次餵進模型裡面，不過因為遞迴式神經網路的隱藏層會存有前面時刻點的資訊，我們也將其視為自回歸模型。

$$x_t = [c_{t-1}, f_{t-1}, c_t] \tag{3.2}$$

$$u_t = \sigma(R_u h_{t-1} + I_u^\star x_t) \tag{3.3}$$

$$r_t = \sigma(R_r h_{t-1} + I_r^\star x_t) \tag{3.4}$$

$$e_t = \tau(r_t \circ (R_e h_{t-1}) + I_r^\star x_t) \tag{3.5}$$

doi:10.6342/NTU202001501

$$h_t = u_t \circ h_{t-1} + (1 - u_t) \circ e_t \tag{3.6}$$

$$y_c, y_f = \text{split}(h_t) \tag{3.7}$$

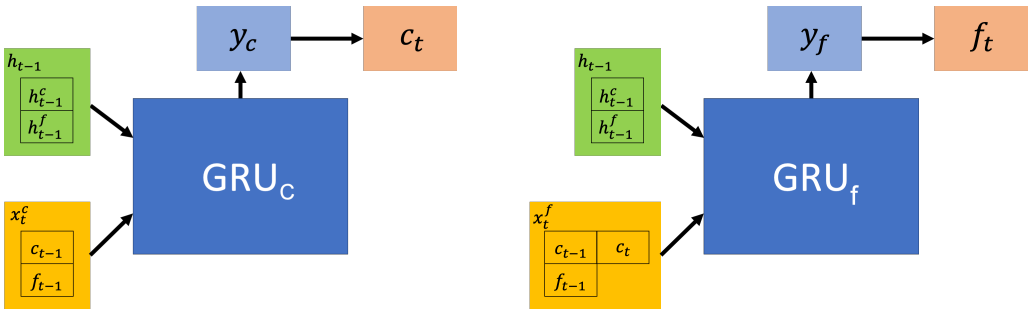
$$P(c_t) = \text{softmax}(O_2\text{ReLU}(O_1 y_c)) \tag{3.8}$$

$$P(f_t) = \text{softmax}(O_4\text{ReLU}(O_3 y_f)) \tag{3.9}$$

基本上WaveRNN的基底架構為一個變形的GRU(式3.2, 3.3,3.4, 3.5, 3.6)。 在原始論文中，在輸出最終結果時也是採用章節2.4.2中的μ-法則量化，每個時間點的輸出是16 bits，通過分類問題預測2^{16}種可能性的哪一個分類。 不過為了降低中間矩陣運算量，它將16 bits切割成前8 bits和後8bits(式3.7)，改做兩個2^8的分類問題(式3.8, 3.9)，分別輸出後再將其合併成16 bits。

從數學上等價來看，可以看成兩個步驟：

1. 概略估計：預測出前面8 bits的資訊
2. 精細估計：預測出後面8 bits的資訊



(a) 步驟一：概略估計　　(b) 步驟二：精細估計

圖 3.3: WaveRNN模型架構

我們也可以將數學式3.2至3.9等價成兩個GRU如圖3.3，其中他們的隱藏層向量(hidden vector) h_t是共用的，而因為兩者GRU的輸入不同，概略估計會輸入上個

doi:10.6342/NTU202001501

時間所預測出來的結果，而精細估計會輸入上個時間點預測出來的結果外加這個時間點概略估計的結果，所以對應GRU輸入的矩陣大小也會不同，才能共用相同的隱藏層向量。 在手機運算時，兩者可以一起通過硬體矩陣加速的方式來達到更快的生成速率。

在本篇我們所使用的WaveRNN的架構為了要和其他模型輸出比較，原始的WaveRNN為兩個步驟各吐出1個8-bit分類問題的預測值，總共16 bits，但是這樣輸出品質和其他僅輸出8 bits的做法相比並不公平，因此我們會通過兩個GRU之後，最後只會輸出1個8-bit分類問題的預測值，再算損失函數。

3.2.4 傅立葉轉換神經網路模型(FFTNet)

FFTNet [17]是WaveNet模型的改進，FFTNet把原本複雜的擴張因果卷積運算替換成圖3.4，圖的輸入$x_0, ..., x_{N-1}$，輸出x_N。

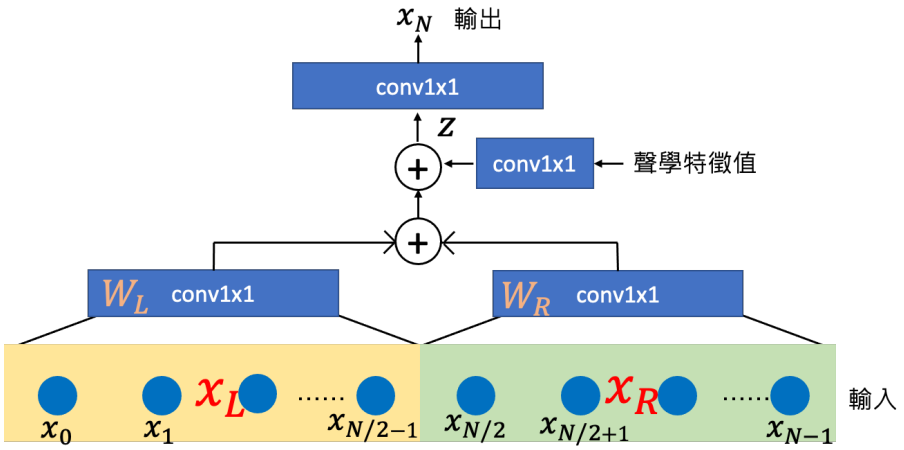


圖 3.4: FFTNet一層模組架構

首先會將輸入分成兩段x_L和x_R分別通過兩個1×1卷積層。 當我們想要把聲學特徵值當作條件餵進去的時候，模型會同時考慮之前時刻點的音訊和聲學特徵

doi:10.6342/NTU202001501

值，

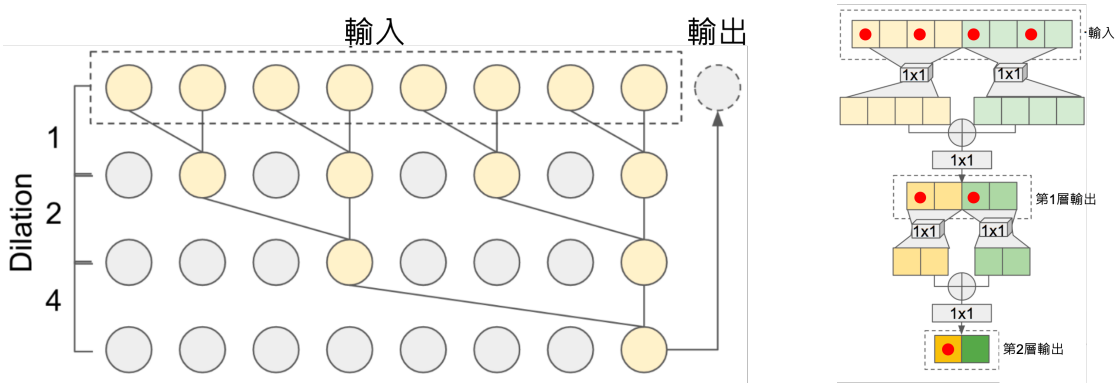
$$z = (W_L * x_L + W_R * x_R) + (V_L * h_L + V_R * h_R) \tag{3.10}$$

得到中間值z之後會通過一層1×1卷積層得到圖中的x。

$$x = \text{ReLU}(\text{conv}1\times1(\text{ReLU}(z))) \tag{3.11}$$

因為1×1卷積層不會影響到輸出長度，所以我們可知道每通過FFTNet一層模組輸出長度就會減半，這樣子的架構和WaveNet有異曲同工之妙，因此若我們將多層的FFTNet一層模組疊起來可以得到一個和WaveNet的對照圖。

透過圖3.5能看到的FFTNet對於輸出層的紅點，及它所對應得感受面(receptive field)的輸入的紅點。 可發現它所對應的感受面相當寬廣，而且所需要模型複雜度比WaveNet小很多。



(a) WaveNet架構　　(b) FFTNet架構

圖 3.5: FFTNet和WaveNet感受面對應圖

它被稱為FFTNet的原因在計算快速傅立葉轉換(Fast Fourier Transform, FFT) [18]時，也同樣可以將輸入的點分成兩半一一做運算再相加，跟這個模型的想法有異曲同工之妙。

其他FFTNet使用到的技術：

doi:10.6342/NTU202001501

1. 補0技術(zero padding): 因為FFTNet是一個自回歸模型，會需要考慮前N個點的資訊，而在音訊剛開始的時候不知道要參考哪些資訊，因此在音訊剛開始之前補上N個0。

2. 條件取樣(conditional sampling): 在做生成時，並不是對於模型輸出直接取機率最大的值當輸出結果，而是取機率取樣當作輸出結果。不過如果該時間點為濁音(voiced)，會先讓機率分佈變更加銳利再做取樣。

3. 訓練時增加雜訊：因為模型在該時刻的輸入會採計前面輸出的結果，因此若前面結果輸出不完美會導致結果自此之後都崩壞。因此在訓練時增加一些雜訊，這樣在生成時對於前面生成不完美不會那麼敏感。

3.2.5 平行生成對抗網(Parallel WaveGAN)

根據章節2.2有關GAN對抗生成的理論，可知GAN和聲碼器都是生成模型(generation model)，因此我們將對抗生成的訓練方式加在聲碼器上，如圖3.6，藉此訓練出高品質的聲碼器。

Parallel WaveGAN的生成器基底架構為WaveNet，原本輸入聲學特徵值的地方保持和WaveNet一樣，而輸入前一個時刻點的地方改成輸入和生成對抗模型一樣的高斯雜訊。 在計算損失函數時，除了生成對抗網路的計算(如式2.10)，還加上針對生成的音訊波形和原始音檔通過不同的精度的短時距傅立葉轉換(short-time Fourier transform, STFT)，並由式2.4曼哈頓距離的方法計算，希望兩者數值大小越近越好。

其他像是MelGAN [19]的聲碼器也是通過生成對抗網路的方式來訓練，通過多層級的鑑別器(Discriminator)來提升訓練出來的結果。

doi:10.6342/NTU202001501

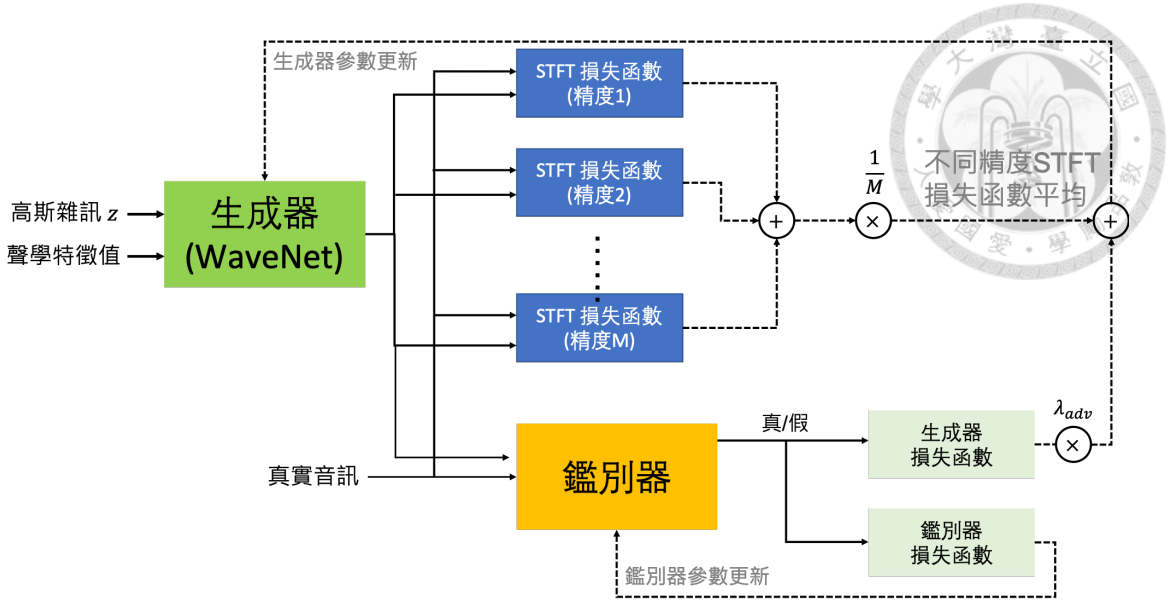


圖 3.6: Parallel WaveGAN架構

3.3 不同聲碼器優缺點比較

3.3.1 參數量比較

本次實驗中所設置的參數量如圖3.7所示，圖中單位M代表10^6。在原始的論文設置中，參數量大小的比較為 WaveNet >WaveRNN >FFTNet >Parallel WaveGAN。 在本次實驗中所使用到的參數量中，WaveNet所使用參數量設置和原始提出文章不相同，是根據我們所能使用訓練資源有關，在原始的實驗設置中WaveNet的參數總量是最大的，但在本實驗中WaveNet的總參數量小於WaveRNN。 因此本次實驗中，參數量的比較為 WaveRNN >WaveNet >FFTNet >Parallel WaveGAN

3.3.2 產生音檔速度比較

以下實驗使用的CPU為Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，GPU為NVIDIA GeForce GTX 1080 Ti 總共測試檔案25個，檔案音長分佈如圖3.8，橫軸為音檔長度(秒)，縱軸為個數。其中測試檔案的取樣率為22,050赫茲，換算成各檔案所含數

doi:10.6342/NTU202001501

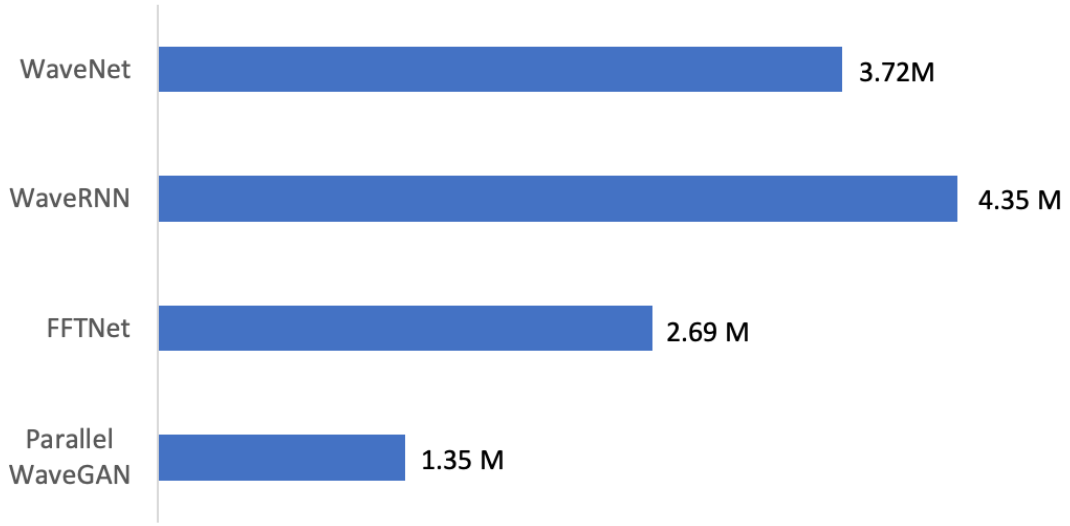


圖 3.7: 參數量比較圖

值個數的分佈圖如圖3.9，橫軸為數值點個數，縱軸為個數。

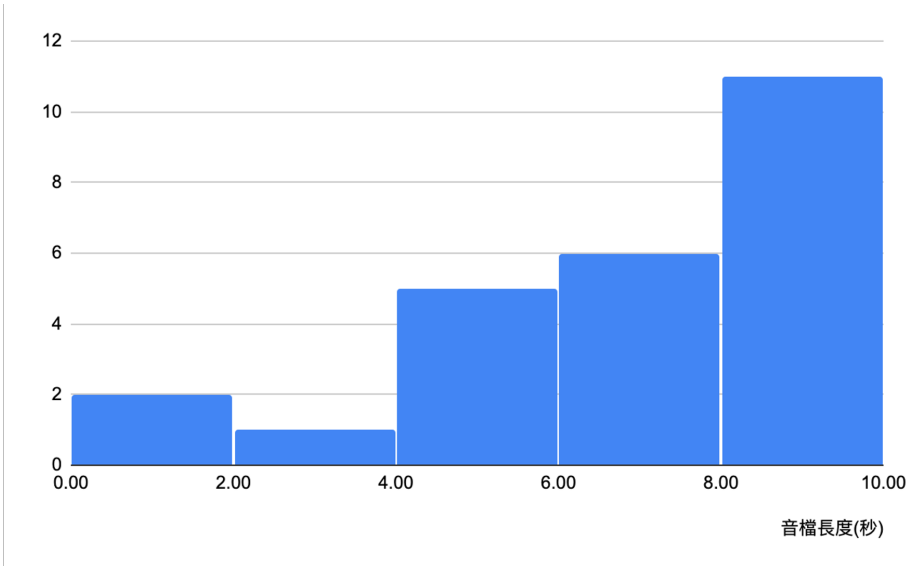


圖 3.8: 測試檔案音長分佈圖

而聲碼器最終生成的是數值點，故我們以下討論的單位為每秒可生出多少個數值點。 若模型每秒產生的數值點個數大於音頻取樣率，即可達到即時生成的效果。 在本測試實驗中25音檔加起來總共有3,812,693個數值點，而每個模型在

doi:10.6342/NTU202001501

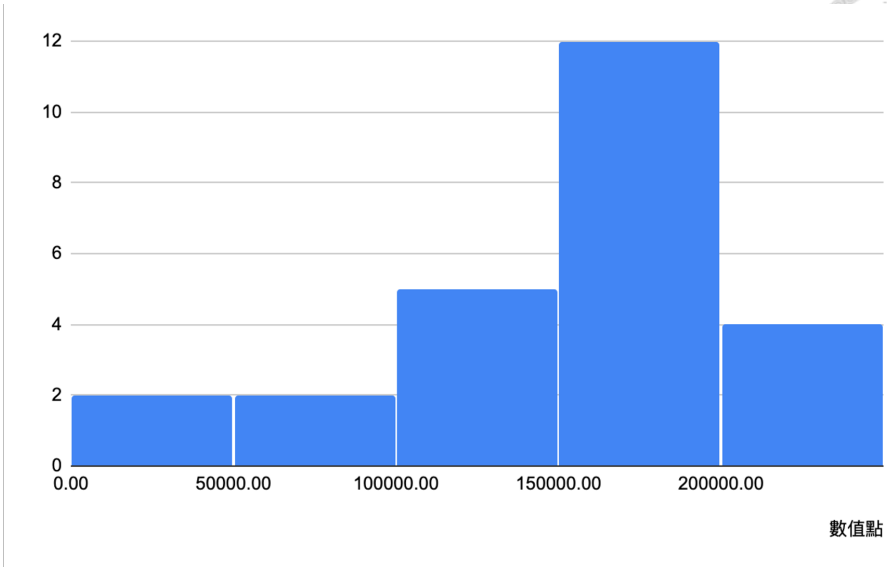


圖 3.9: 測試檔案數值個數分佈圖

僅使用CPU和使用CPU+GPU所花費的秒數如表3.2。不過這顯然受到程式碼撰寫的方式影響，且可以透過硬體加速的方式來做改善。本次測試中Griffin-Lim演算法使用的迭代次數為100次，實際經驗中100次的迭代和更多次數的迭代來說其音色人耳已分不出差別，故以此作為一個參考基準，且Griffin-Lim演算法無法使用GPU來做加速，因此CPU+GPU欄位並無記錄。

我們透過總數值點個數和時間關係可以畫出相對平均生成速度，其中僅使用CPU產生音檔平均生成速度為圖3.10，其中特別在計算Griffin-Lim演算法所需花費的時間，因為Griffin-Lim不能使用在梅爾時頻譜上面，因此我們使用相同傅立葉點數的結果做一個參考值。

在原始論文中，生成速度的比較為Parallel WaveGAN >>WaveRNN >FFT-Net >WaveNet，在本次實驗中因為參數量和實作方式的關係，因此本實驗在CPU生成平均速度為Parallel WaveGAN >>WaveRNN >WaveNet >FFTNet。但是仍可看到趨勢上WaveNet和FFTNet是在同一個數量級，而WaveRNN的生成速度是WaveNet和FFTNet的10倍以上，而兩個非自回歸(Non-autoregressive)架構的方法

doi:10.6342/NTU202001501

	CPU	CPU + GPU
WaveNet	19,596 sec	26,131 sec
WaveRNN	2,018 sec	1,737 sec
FFTNet	26,528 sec	10,047 sec
Parallel WaveGAN	162 sec	4.54 sec
Griffin-Lim Algorithm	127 sec	N/A

表 3.2: 各模型在生成25個測試音檔花費總時間

又是WaveRNN的10倍以上，並且由於Parallel WaveGAN和Griffin Lim演算法每秒生成的數值點個數較原始音頻取樣率22,050Hz還高，因此可以做到即時生成。在原始文章中，WaveRNN, FFTNet也是可以達到即時生成的效果，不過他們有特別的實作方式和特殊的硬體加速。

當使用GPU作為輔助加速，各模型產生音檔平均生成速度為圖3.11。可以發現在實用上面WaveRNN如果目標為1,600Hz可以做到即時生成，而Parallel WaveGAN在GPU的加速下其生成效率遠勝於其他自回歸架構模型。

在本論文中，所有的模型都是使用Python和PyTorch套件實作的。 而在速度測試實驗中，WaveNet CPU比GPU還快的原因猜測是因為在Python和PyTorch套件上CPU多線層優化比較好，而且不斷地將記憶體的內容和GPU記憶體內容搬運需要花費大量的時間，但這可以透過將程式碼以C++等較底層的程式語言優化，使得效率更加提升。

從以上實驗可以得知，以生成速率來說非自回歸(Non-autoregressive)的模型生

doi:10.6342/NTU202001501

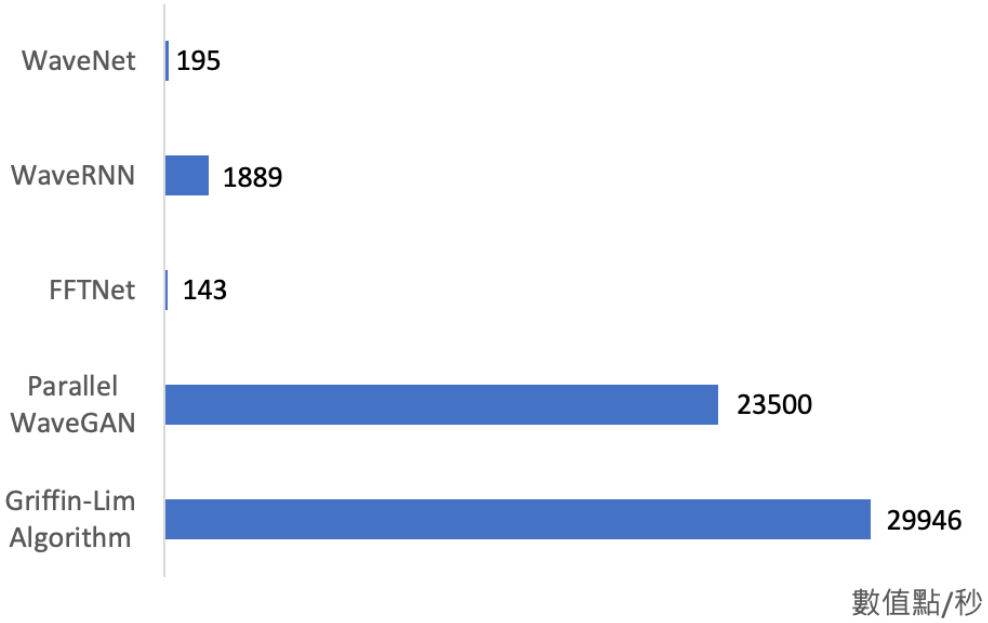


圖 3.10: CPU產生速率圖

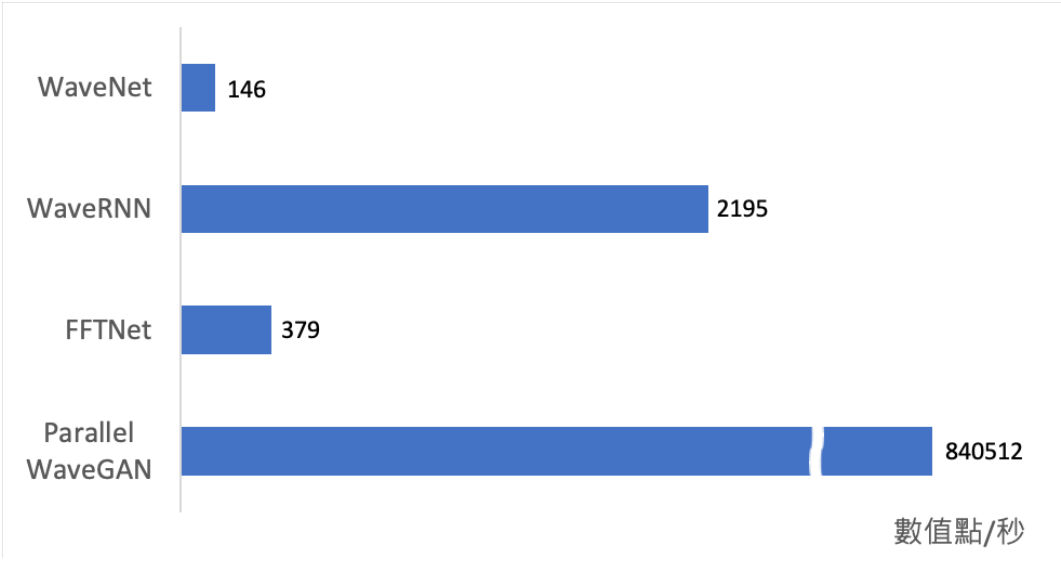


圖 3.11: GPU產生速率圖

doi:10.6342/NTU202001501

成效率遠大於自回歸(Autoregressive)模型。

3.4 聲碼器設計細節討論

在眾多聲碼器架構中，我們發現一些細節會影響最終訓練的穩定性、生成結果，以下隨著聲碼器訓練過程將發現成果一一列舉出。

3.4.1 前處理(Pre-process)

從實驗觀察中，在訓練聲碼器的時候，對於取出來的聲學特徵值來說有沒有做標準化(normalize)，對最後生成結果幾乎沒有影響，深層學習架構都會學習到合適對應的參數。 因此只要針對自己的需求建立模型加以訓練即可。

3.4.2 升取樣(Upsample)模組設計

聲學特徵值時間長度對應通常相較於原始音訊檔來說會是其降取樣(downsample)的輸出，以本次抽取聲學特徵值的方式就是音框平移，因此我們需要將聲學特徵值長度倍數增加至音框平移量，才能讓輸入聲學特徵值和輸出結果對齊(align)。

而我們通常會將這樣升取樣模組設計當作條件，希望輸出結果能依據條件輸出對應合適的音訊波形， 而實驗觀察以圖3.12的方式設計升取樣模組， 我們取升取樣倍數做為複製自己的次數後，過一層卷積層， 對於這樣的模組通常會連續過幾回，而讓每輪複製自己次數的乘積為總共升取樣倍率，再將其餵進上述分別介紹的聲碼器模組，如此聲碼器生成的結果會更穩定。

舉例來說如果需要升取樣的比率為200倍的話，通常會設計成複製4, 4, 5的架構，也就是先複製4次過一層卷積層，再複製4次過一層卷積層，最後複製5次過一

doi:10.6342/NTU202001501

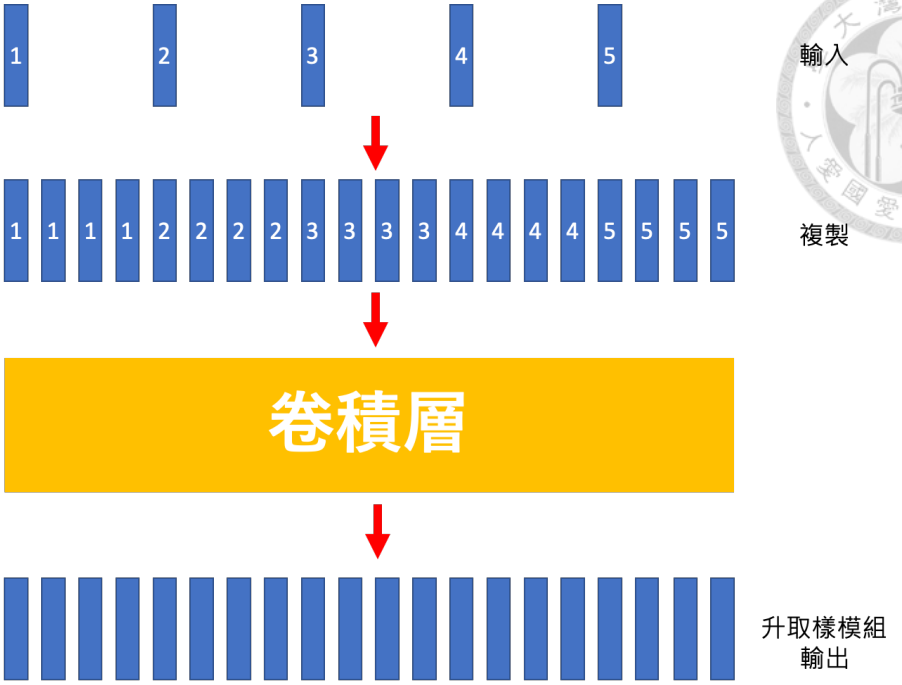


圖 3.12: 升取樣模組架構

層卷積層。 這樣子的架構會比直接複製到正確的取樣比率或是直接線性增長至正確取樣比率的生成結果穩定。

3.4.3 訓練過程

WaveNet, FFTNet這種會將前一刻時間點餵進模型當作下一刻時間輸入的模型，也就是$x_1, ..., x_{t-1}$輸出x_t當作訓練目標。 如果在計算損失函數時，我們給定$x_1, ..., x_{t-1}$，輸出$x_2, ..., x_t$做計算，收斂速度會比只算x_t的損失函數快很多。

為避免生成結果不完美，而自回歸模型連續的性質影響到生成結果，可參考FFTNet原始文章所提出加雜訊的方法。 本論文中所提到三種自回歸模型(WaveNet, WaveRNN, FFTNet)都是透過分類問題的損失函數做計算，而自回歸模型的聲碼器模型在做分類時比較容易錯分至向上或下一個分類，比方說原本想分類至第i類別，比較容易分類至i-1, i, i+1，而在測試時，如果前一個時刻錯分

doi:10.6342/NTU202001501

至i-1或i+1，可能影響自回歸模型後面的序列不知如何輸出。 因此在訓練時讓輸出加上一個平均為0，標準差為1的雜訊，因為分類目標都是整數，故將雜訊取四捨五入之後，加上原本的分類類別當作訓練目標。 如此訓練的目標可以讓生成結果更穩定，不容易因為測試時產生錯誤的分類而導致後面的序列生成結果崩潰。這種訓練方式對於所有自回歸模型的聲碼器架構都可以使用。

3.4.4 生成過程

原始FFTNet的文章中有提及條件取樣(conditional sampling)，也就是在生成時，他並不是對於模型輸出直接取機率最大的值當輸出結果，而是取機率取樣當作輸出結果，且會根據該時間點為清音或濁音，調整機率分佈變更加銳利再做取樣。 但在實驗觀察中有沒有採取機率取樣，或是調整機率分佈跟原本取最大的機率當作輸出，其實影響微乎其微，幾乎聽不出來。

3.4.5 後處理(Post-process)

在自回歸模型的架構中，有時候會聽到類似滑鼠點擊聲，聽起來有點滋滋作響，這是由於自回歸模型的架構所導致的， 當生成品質偏低的時候，有很明顯的滑鼠點擊聲時，可通過一些傳統降噪的方式，如logmmse [20]，可以稍微改善這類型的雜訊。

在非自回歸模型的結果若生成品質低落時，容易發生糊糊的聽起來有層薄膜而不清楚的感覺，此類型的輸出品質比較不能透過後處理的方式來提升生成品質。

doi:10.6342/NTU202001501

3.5 本章總結

本章節探討本篇論文所使用的所有模型架構以及在實驗中的一些觀察，並比較其參數量和產生音檔速度，在後續的章節會探討其生成品質和强健性的比較。

doi:10.6342/NTU202001501

第 四 章　多種聲碼器在聲學特徵值的強健性比較

4.1 簡介

聲碼器是一種生成模型，容易發生過度貼合(overfitting)的問題，為了瞭解不同聲碼器對於不同訓練資料是否會發生過度貼合的問題，而哪一種聲碼器具有更好的普遍化(generalization)的能力。 我們針對4種不同的聲碼器WaveNet, WaveRNN, FFTNet, Parallel WaveGAN分別訓練在多種訓練集，並分別測試在不同的測試集，並分析各聲碼器所適合的訓練集以及適合應用在哪些問題中。

4.2 資料集介紹

在本次實驗中，使用以下幾個子資料集來組成本實驗的訓練子集，其中括號內為之後表4.2內所使用的縮寫名稱。

- CMU US BDL Arctic Dataset (cmu_ma) [21]
- CMU US SLT Arctic Dataset (cmu_fe) [21]
- Internal Mandarin Dataset (man_fe)
- LibriTTS (libri) [22]
- Bible (bible)

而下列是各資料集品質定性介紹：

doi:10.6342/NTU202001501

- CMU Arctic Dataset：取BDL(男)和SLT(女)兩個單語者的資料集，語音品質口音偏重，些微背景雜訊。
- Internal Mandarin Dataset：內部搜集的資料集，女性中文語者，語音品質相當清晰，無背景雜訊。
- LibriTTS：取其中train clean的子資料集，語音品質清晰少背景雜訊。
- Bible：搜集至聖經朗讀網站www.bible.is，語音品質些微背景雜訊、些微回音。

4.3 評量方法 – 平均意見評分(Mean Opinion Score, MOS)

對於測試的音檔，會給予至少10位有效聽者戴著耳機聆聽，會詢問聽者「聲音是否自然流暢，像乾淨且無背景雜訊的人類語音?」，請聽者給予1-5分的主觀評分。 每位聽者會給予約25句做評分，因先前實驗觀察，若給予聽者過多的音檔，會因為疲乏而導致前後標準不一。 聽者都是通過FB和ptt徵求而得，皆為對語言學、語音學無特別研究的人。 此外需要觀察評分者是否有認真填寫，觀察方法為評斷他在真實音檔是否給予4或5分的分數，若有確實達到才會視為有效聽者。 在評量方法中選擇以人主觀判定是因為在音訊生成中，目前沒有一種標準可以準確地評判所生成出來的音檔像不像人類語音。 此外，對於生成出來的結果也無法和原始音檔做SNR等運算，因為像不像人聲和SNR並沒有直接的關係，而且倘若生成出來的音檔和原始音檔並沒有直接時間上的對應關係，我們也無法計算SNR。

doi:10.6342/NTU202001501

4.4 聲碼器在訓練測試不同語言及不同語者的強健性比較

觀察當訓練集和測試集是不同語言或是不同語者時，各種不同的聲碼器有什麼表現，並探討哪一種聲碼器的強健性最佳，可以訓練在少數資料卻能有普遍化應用的能力。

4.4.1 實驗設計

本實驗設計出的實驗分為兩大變因來做探討，分別是測試集語者是否在訓練集出現過以及測試集語言是否在訓練集出現過。 在訓練集我們會給定聲學特徵值和對應的原始音訊波形，而測試集會給予聲學特徵值而重建的音訊波形會交由人評分像不像人類語音。 實驗中所使用的聲學特徵值會依照表3.1的方式來抽取。

首先定義訓練集、測試集標籤的訂定原則：

- 大寫開頭代表訓練集，而小寫斜體代表測試集。
- 表示方式為：語言標籤_性別標籤
- 語言標籤: En: 英文, Ma: 中文, Lrg: 多語言
- 性別標籤: M: 女性, F: 男性, L: 男女皆有

本實驗在資料集的部分設計三種情形：

1. 單語言單語者
2. 單語言多語者
3. 多語言多語者

doi:10.6342/NTU202001501

未考慮多語言單語者這種情形是因為這類型的資料不存在，本實驗設計的訓練集和測試集對照表如表4.1。 在選擇訓練集和測試集相同語言相同語者的測試資料會特別選曾在訓練集出現過的語者且不同語句做測試。

其中依照資料集取得的方便性，我們在單語言單語者選擇中文、英文兩種語言，單語言多語者選擇英文，而多語言多語者則是將能收集到的語言(除中文以外)加進去一起訓練。 理由是我們所能方便作為測試集的語言只有中文和英文兩種語言。故當訓練集為中文，沒看過的語言就測試在英文；反之，若選擇訓練集為英文，沒看過語者的測試集就選擇為中文。 因此當設置多語言多語者的訓練集沒有包含到中文，其不同語言的測試集便可選擇中文測試集。 而測試相同語者相同語言的時候，我們在多語言多語者的訓練集都包含有單語者男性英文(cmu_ma)和單語者女性英文(cmu_fe)，這樣在選擇測試集的時候，可以選擇和單語者男性英文(cmu_ma)和單語者女性英文(cmu_fe)一樣的測試集，不用額外準備多一份測試集也可以達到目的，詳情可以同時參照表4.1, 4.2。

<table>
<tr><td rowspan="2">訓練集標籤</td><td colspan="2">訓練集特徵</td><td colspan="3">測試集和訓練集的關係</td></tr>
<tr><td>語者個數</td><td>語言個數</td><td>相同語言
相同語者</td><td>相同語言
不同語者</td><td>不同語言
不同語者</td></tr>
<tr><td>Ma_F</td><td rowspan="3">單語者</td><td rowspan="3">單語言</td><td>ma_f</td><td>ma_l</td><td>en_l</td></tr>
<tr><td>En_M</td><td>en_m</td><td rowspan="3">en_l</td><td rowspan="3">ma_l</td></tr>
<tr><td>En_F</td><td>en_f</td></tr>
<tr><td>En_L</td><td rowspan="2">多語者</td><td rowspan="2">en_m
en_f</td></tr>
<tr><td>Lrg_L</td><td>多語言</td></tr>
</table>

表 4.1: 不同語言强健性實驗訓練和測試的對應表

doi:10.6342/NTU202001501

其中測試集中的相同語言/相同語者表示訓練集曾經出過相同語言/相同語者，若測試集中的不同語言/不同語者表示訓練集不曾經出過測試集的語言/語者，而沒有出現相同語者不同語言的項目是因並沒有此類的資料集可做為測試使用。 如此一來，對語者和語言都可以做强健性的比較實驗。

對於我們構造出包含單語言單語者、單語言多語者、多語言多語者情形的訓練集後， 對於每一個聲碼器中，我們取對這5個訓練集(Ma_F, En_M, En_F, En_L, Lrg_L)做訓練，訓練完畢之後再通過做3種測試(相同語言相同語者, 相同語言不同語者, 不同語言不同語者)，分別對15個情況(5訓練集×3測試狀況)都做平均意見評分(Mean Opinon Score)， 透過設計可以了解哪種聲碼器生成時，對於訓練集不曾出現過的語者/語言仍輸出高水準的音檔，藉此評估各不同聲碼器的强健性。

為了符合我們在表4.1的訓練、測試對應關係，我們重新組合在章節4.2所介紹過的資料集，並依照我們的需求建立出單語言單語者(中文、英文)、單語言多語者(英文)、多語言多語者，訓練資料詳細數據分析都列於表4.2。 而表中Lrg_L的語者個數不為定值是因為Bible資料集有一句訓練句子包含多個語者的狀況，故無法準確地估計實際含有的語者數。

在組合出合適的訓練集之後，我們也依照表4.1的需求選擇出測試集。

而我們所選擇的測試集為以下的表4.3，也是取材自章節4.2，但文句和訓練集內的文句完全沒有重複，語者有重複的部分也有註明在表中。測試資料集的MOS的評分是和生成出來的語句一起做評分，當作一個參考值來判斷生成出來的結果。

當設計出合適的訓練集、測試集之後，我們對4種聲碼器(包含WaveNet, WaveRNN, FFTNet, Parallel WaveGAN)依照表4.1訓練和測試對應關係，每個對應關係有10句生成結果，再加上原始測試集音檔，將結果全部打散，每位聽者分配

doi:10.6342/NTU202001501

訓練集標籤	含有子資料集	語者個數	訓練集句子個數	訓練集包含語言
En_M	cmu_ma	1	1091	English
En_F	cmu_fe	1	1092	English
Ma_F	man_fe	1	8904	Mandarin
En_L	cmu_ma cmu_fe libri	560	35419	English
Lrg_L	cmu_ma cmu_fe libri bible	>600	38139	English French Japanese Korean Spanish Thai

表 4.2: 各訓練集含有的子資料集

doi:10.6342/NTU202001501

到25句，並依照章節4.3的方式給予主觀評分1-5分，並記錄平均和95%信心水準。

本實驗產生的所有音檔放置在網站: https://bogihsu.github.io/Robust-Neural-Vocoding/

測試集標籤	語者個數		測試集	測試集和哪些訓練集	MOS
	男性	女性	句子個數	有相同語者	
ma_f	1	0	10	Ma_F	4.79±0.10
en_m	0	1	10	En_M	4.67±0.14
en_f	1	0	10	En_F	4.55±0.15
en_l	5	5	10	和訓練集語者無交集	4.53±0.10
ma_l	3	3	10		4.43±0.11

表 4.3: 語言、語者强健性實驗測試集特徵分析

4.4.2 實驗結果與分析

實驗結果列表於4.4，其中分為三個小實驗來探討：

1. 訓練和測試集相同語言相同語者
2. 訓練和測試集相同語言不同語者
3. 訓練和測試集不同語言不同語者

訓練和測試集相同語言相同語者

在表4.4中的第1個區塊所有的聲碼器都表現得相當不錯。 其中又以WaveNet的自然度表現為最佳，在有些情形甚至比原本測試音檔還高分，猜測可能原測試音檔

doi:10.6342/NTU202001501

聲碼器架構	聲碼器訓練集				
	En_F	En_M	Ma_F	En_L	Lrg
訓練和測試集相同語言相同語者					
WaveNet	**4.78±0.10**	**4.71±0.11**	4.63±0.12	**4.72±0.10**	**4.70±0.13**
WaveRNN	4.48±0.13	4.61±0.13	**4.66±0.11**	4.64±0.11	4.61±0.13
FFTNet	3.87±0.17	4.29±0.15	4.45±0.10	3.28±0.19	3.58±0.17
Parallel WaveGAN	4.59±0.12	4.29±0.17	4.41±0.12	4.29±0.15	4.11±0.16
訓練和測試集相同語言不同語者					
WaveNet	2.27±0.14	2.86±0.17	3.27±0.16	**4.25±0.17**	**4.35±0.15**
WaveRNN	**2.60±0.14**	**2.89±0.15**	**3.54±0.14**	3.98±0.15	3.92±0.16
FFTNet	1.76±0.15	2.21±0.14	2.94±0.13	2.99±0.18	3.13±0.21
Parallel WaveGAN	2.35±0.15	2.85±0.16	2.88±0.14	3.80±0.21	3.85±0.17
訓練和測試集不同語言不同語者					
WaveNet	1.90±0.12	2.53±0.12	**3.85±0.15**	**4.33±0.15**	**4.33±0.17**
WaveRNN	**2.53±0.13**	**2.62±0.12**	3.30±0.15	4.30±0.16	4.16±0.17
FFTNet	1.56±0.09	1.75±0.12	2.64±0.16	2.67±0.17	3.37±0.17
Parallel WaveGAN	2.17±0.11	2.54±0.12	2.49±0.13	3.79±0.20	3.97±0.19

表 4.4: 以MOS呈現語者、語言强健性實驗訓練實驗結果

doi:10.6342/NTU202001501

有一點點麥克風雜訊，而聲碼器可些微地消除一些背景雜訊。 在訓練和測試集相同的時候，單語者的訓練集的表現會比多語者的訓練集來的略好一些，因為聲碼器只需要專注在同一語者的聲學特徵上就可以表現出有很好的結果。

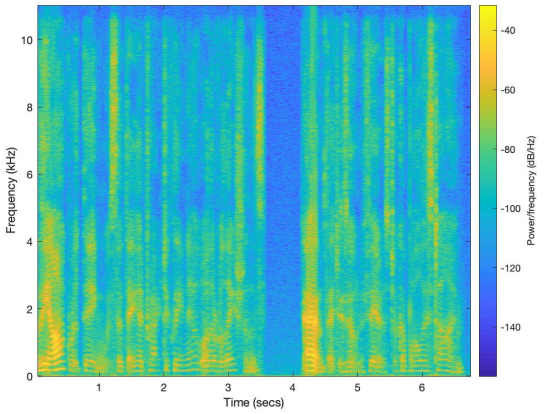
訓練和測試集相同語言不同語者

在表4.4中的第2個區塊，可發現與第1區塊有一定的差距。 在單語者的訓練集下，一旦測試在未見過的語者會有相當明顯雜訊讓聽者覺得自然度大幅下降。 其中生成結果可歸類成三種不同明顯的雜訊會影響聽者感受：

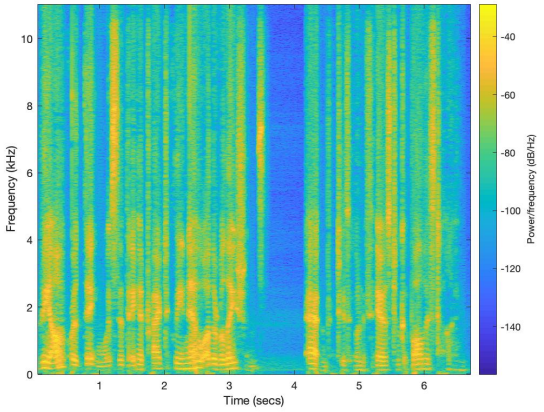
- 如圖4.1(a)，容易出現像卡通裡巫婆似沙啞撕裂的聲音，從時頻譜上可發現有些需要細緻的地方生成的比較模糊
- 如圖4.1(b)，似是喉嚨卡了一個東西講出來的話，使得講出來的話有點卡卡的，比較不順，從時頻譜上可發現能量特別集中在某些地方，聽上去就不那麼順暢
- 如圖4.1(c)，一小瞬間類似白色雜訊的噪音，從時頻譜上紅框框起來的地方為這類雜訊，在所有頻率上均具有能量

其中第一種的雜訊比較容易出現在WaveNet, WaveRNN, FFTNet中，而第二種雜訊比較容易出現在FFTNet，第三種雜訊比較容易出現在Parallel WaveGAN中。第一種雜訊是自回歸模型的特性所造成的，若某一個時刻點生成結果沒有那麼好，會導致連續幾個時刻點生成結果受到影響，便會導致這種雜訊。 第二種雜訊比較不確定造成的原因，但猜測可能是類似第一種雜訊的成因，只是影響較第一種雜訊的時間範圍廣泛。 第三種雜訊感覺是在一小段時間內的Parallel WaveGAN比較不知道怎麼生成結果，而導致那一小段的時間跟無特定機率隨意產生音訊是一樣的，對聽眾而言就像是白色雜訊。

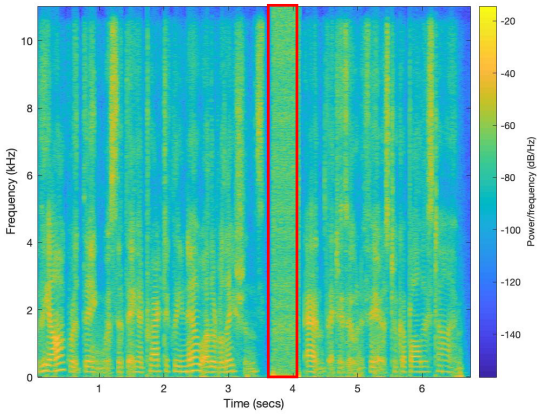
doi:10.6342/NTU202001501



(a) 卡通裡巫婆似沙啞



(b) 講話比較卡卡的



(c) 一小瞬間類似白色雜訊

圖 4.1: 未見過語者所造成雜訊的時頻譜

doi:10.6342/NTU202001501

在各聲碼器的比較中，我們可以歸納出在單語者訓練集下，WaveRNN具有比較强的强健性。 相對的，對於多語者的訓練集，WaveNet具有比較强的强健性。

訓練和測試集不同語言不同語者

在表4.4中的第3個區塊，我們可以發現跟第2區塊表現非常相近，因此我們可以從此歸納出其實訓練集語言對於聲碼器來說並不是一個影響聲碼器生成結果的關鍵因素。 這也符合我們對聲碼器的需求，因為我們所期待的聲碼器是一種從聲學特徵值轉換成音訊波形，和語言是沒有任何關係的。

4.5 聲碼器在訓練測試不同性別的强健性比較

在單語者的訓練集，測試在多語者的測試集的時候，我們無法分辨生成品質的大幅下降是因為測試集包含沒看過的語者就會大幅下降，還是主要是因為有沒看過的性別。 畢竟我們知道男女性在聲音上有相當大的差距，因此訓練和測試資料頻譜上就有相當大的差距。 因此我們在這個章節更近一步分析在單語者訓練集分別測試在兩種性別下的測試集分別會有什麼樣的結果。

4.5.1 實驗設計

其中我們所使用的單一語者資料集有男性英文、女性英文、女性中文的訓練集，而我們選擇4聲碼器(WaveNet, WaveRNN, FFTNet, Parallel WaveGAN)在表4.2中的En_M, En_F, Ma_F訓練集的模型依據表4.5的對應關係，其中測試集的選擇都為多語者的測試集，詳細數據可參照表4.6且抽取聲學特徵值的方法也是使用表3.1的設置，其中MOS值為和生成結果一起做測試當作參考值，而MOS實驗測試方法可參照章節4.3。

doi:10.6342/NTU202001501

本實驗產生的所有音檔放置在網站: https://bogihsu.github.io/Robust-Neural-Vocoding/

訓練集標籤	訓練集特徵		根據訓練集特徵對應的多語者測試集			
	性別	語言	相同語言 相同性別	相同語言 不同性別	不同語言 相同性別	不同語言 不同性別
En_M	男	英文	*en_m*	*en_f*	*ma_m*	*ma_f*
En_F	女	英文	*en_f*	*en_m*	*ma_f*	*ma_m*
Ma_F	女	中文	*ma_m*	*ma_m*	*en_f*	*en_m*

表 4.5: 不同性別強健性實驗訓練和測試的對應表

測試集標籤	語者個數		測試集句子個數	**MOS**
	女性	男性		
ma_f	3	0	10	4.54±0.15
ma_m	0	3	10	4.32±0.16
en_f	10	0	10	4.43±0.14
en_m	0	10	10	4.64±0.13

表 4.6: 性別強健性實驗測試集特徵分析

4.5.2 實驗結果分析

從章節4.4得知單語者訓練集平均表現其實和多語者訓練集再遇到沒看過的語者就已經有一個相當大的差距。 而我們從章節4.4也得知訓練集的語言對於聲碼器基本上並不影響生成品質。

doi:10.6342/NTU202001501

將實驗產生的音檔以MOS呈現在表4.7，從實驗數據得知訓練集為女性的時候，若其測試集也同樣是女性時，都可以有還不錯的表現，不過一旦測試集為男性就會有一定幅度的落差，可以發現聲碼器對於訓練集的性別是非常敏感的。

此外，雖然在測試集為同性別時，測試結果男性比不上女性。不過可以發現當訓練在女性語音而測試在男性語音時，其實表現還是遠比不上訓練在男性語音而測試在男性語音的。 因此我們可以歸納出對於聲碼器而言，訓練集的多樣性是非常重要的，也沒有任何聲碼器有足夠強的強健性，可以僅訓練在單一語者就可以有很強的普遍性可應用在各式語音上。

4.6 本章總結

本章節的目的在於觀察當訓練和測試集所遭遇的情形不同時，哪些因素是影響生成結果的。 我們設計了訓練集、測試集的對應關係，測試於人類真實語音上。

在章節4.4比較不同語言和語者的實驗，分析出語者的多樣性可以增進讓在訓練集未曾出現過的語者也表現的相當不錯。 且聲碼器的訓練集的語言並不影響生成語言的結果。

在章節4.5比較單一語者的訓練集中，發現聲碼器訓練在女性的訓練集的強健性會比男性好一些，但是若要更有普遍性的話，必定還是需要多樣的語者才能有高品質的語音生成。

總結來說，多樣性的訓練語者可以大幅提高生成的結果，讓聲碼器可以藉助類似的語者去生成出高品質的音訊。

doi:10.6342/NTU202001501

聲碼器架構	聲碼器訓練集		
	En_M	En_F	Ma_F
訓練和測試集相同語言相同性別			
WaveNet	2.41±0.23	3.47±0.24	3.57±0.20
WaveRNN	2.85±0.21	3.49±0.21	4.08±0.20
FFTNet	2.01±0.24	2.45±0.21	3.56±0.14
Parallel WaveGAN	2.68±0.22	3.47±0.20	3.34±0.17
訓練和測試集相同語言不同性別			
WaveNet	2.13±0.16	2.25±0.16	2.98±0.21
WaveRNN	2.36±0.20	2.29±0.15	3.01±0.20
FFTNet	1.52±0.15	1.97±0.20	2.34±0.15
Parallel WaveGAN	2.03±0.17	2.23±0.18	2.41±0.17
訓練和測試集不同語言相同性別			
WaveNet	1.92±0.16	3.05±0.23	4.10±0.22
WaveRNN	2.78±0.18	3.12±0.21	3.77±0.18
FFTNet	1.74±0.17	2.00±0.17	3.40±0.17
Parallel WaveGAN	2.29±0.19	2.92±0.22	2.92±0.21
訓練和測試集不同語言不同性別			
WaveNet	1.88±0.16	2.01±0.16	3.59±0.20
WaveRNN	2.29±0.17	2.12±0.19	2.84±0.21
FFTNet	1.38±0.11	1.51±0.11	1.91±0.16
Parallel WaveGAN	2.06±0.16	2.17±0.15	2.05±0.17

表 4.7: 以MOS呈現性別强健性實驗結果

doi:10.6342/NTU202001501

第 五 章　多種聲碼器在語音生成應用上的強健性比較

5.1 簡介

第4章中我們測試當聲碼器遇到訓練和測試時資料不一致的情形，但都僅測試在人聲所抽取出來的聲學特徵值上。 本章節中，我們將聲碼器使用在語音生成應用上，在語音生成應用的輸出可能不如人聲那麼完美，因此本章節會探討同樣是訓練在人聲所抽取的特徵值上，哪些聲碼器可以在表現較不完美的語音生成運用上仍有高品質的輸出。 在章節5.2會比較各聲碼器應用在文句翻語音系統上的結果；在章節5.3會各聲碼器應用在語者轉換系統上的結果。

5.2 聲碼器在文字轉語音系統下的強健性比較

5.2.1 文字轉語音系統模型介紹

Tacotron [23]為Google提出將英文字母(character)轉換至聲學特徵值–梅爾時頻譜的模型，是直接輸入輸出對應模型(end-to-end model)中容易訓練且輸出相當穩定的模型，只要擁有高品質的訓練集就可以有相當好的結果。 後來同團隊又針對Tacotron進行改良，也就是我們本次文句翻語音系統所使用的Tacotron 2 [24]模型介紹如圖5.1。

其中可以分為三大區塊：藍色框框的模組併在一起可以視為編碼器(Encoder)，灰色的是專注(attention)的機制，橘色的是解碼器(decoder)。

doi:10.6342/NTU202001501

編碼器的目的是讓英文字母轉換成向量，且希望這些向量能夠含有音素(phoneme)的資訊在其中，讓後面生成的模組可以更容易產生發音正確的語音。專注機制是因為文字的輸入長度和語音的輸出長度其實是不等長的，而我們希望模型能夠擁有文字和語音間相對應對齊(align)的關係，需要通過專注機制將兩者對齊，讓解碼器知道現在應該專注在輸入的那一個部分，才能輸出該時刻點文字所對應的語音。 解碼器則是一個遞迴性的模組會根據專注機制所得到的向量吐出對應的梅爾時頻譜。

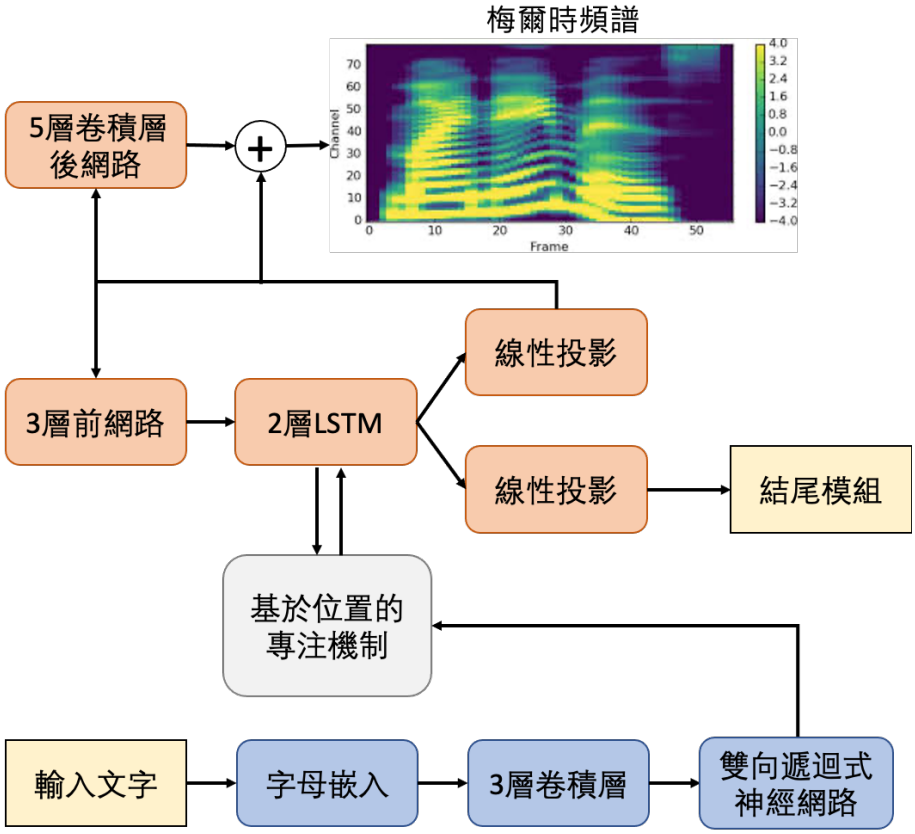


圖 5.1: Tacotron 2文句翻語音系統模型架構

除此之外，一般做文字生成時，如果文句結束時會自動吐出句終符號(End of Sentence)，但語音訊號為連續訊號故無法吐出一個特定的符號來當作句終。 因此結尾模組(stop token module)是一個辨識器(classifer)來判斷語音生成是否生成結

doi:10.6342/NTU202001501

束，一旦判斷該句結束就會停止生成。

本實驗中，文句翻語音模型所使用的訓練集為LJ Speech [25]，是來自一位女性英文專業錄音語者，非常乾淨清晰，是訓練文句翻語音模型的最受歡迎的訓練集之一。 我們使用NVIDIA已訓練好在LJ Speech的模型，並加以微調(fine tune)在我們抽的梅爾時頻譜的參數設定上。

5.2.2 實驗設計

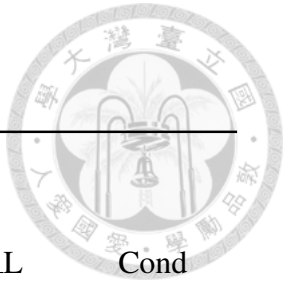
我們想要製造出聲碼器輸入為文字轉語音系統的輸出且對應輸出為原始音檔，當作聲碼器輸出的上界值。 可是文字轉語音系統是一個一對多的關係，因此生成出來的聲學特徵值和原始音檔的時間長度會不一樣，也沒有相對應的時間對應關係。 因此想讓生成出的聲學特徵值和原始音檔長度一樣需透過一些小技巧，在Tacotron 2的系統中decoder的輸入為包含有上一個時間點的輸出的聲學特徵值結果，因此若把decoder的輸入從上一個時間點輸出的聲學特徵值置換成原始音檔的聲學特徵值，便可得到和原始音檔完全一樣的聲學特徵值和音訊波型的時間對應關係。這類型的對應關係，我們在實驗中當作聲碼器的訓練集，並標籤為Cond。

此外，我們也選擇和訓練文句翻語音系統模型相同的訓練集LJ Speech來當作其中之一的訓練集，在接下來的表格中會標籤為LJ。 除了選擇LJ、Cond以外，我們選擇了單語者英文的女性訓練集(En_L)、多語者英文的訓練集(En_F)、多語者多語言(Lrg_L)的訓練集(包含英文)當作我們訓練的對象，其中訓練集的詳細資訊我們採用表4.2一樣的設置。

本實驗產生的所有音檔放置在網站: `https://bogihsu.github.io/Robust-Neural-Vocoding/`

doi:10.6342/NTU202001501

5.2.3 實驗結果和分析



聲碼器架構	聲碼器訓練集				
	LJ	En_F	En_L	Lrg_L	Cond
WaveNet	4.10±0.19	2.59±0.24	3.54±0.20	3.66±0.21	**4.21±0.16**
WaveRNN	**4.16±0.18**	3.05±0.24	3.32±0.20	3.73±0.19	3.79±0.19
FFTNet	2.75±0.27	2.16±0.29	2.50±0.27	2.28±0.28	2.86±0.30
Parallel WaveGAN	3.81±0.20	3.17±0.21	3.60±0.20	3.19±0.20	3.38±0.20
原始音檔	4.54±0.16				

表 5.1: 以MOS呈現聲碼器測試在文句翻語音模型輸出結果

通過章節4.3的方式得到MOS記錄在表5.1， 在原本預期中，Cond擁有文句翻語音系統輸出和真實語音的對應關係，應該是聲碼器的上界值，但實驗發現WaveNet以外的聲碼器在Cond的情形其實並不是最好的，猜測是因為可能對於其他聲碼器來說，真實語音的對應關係對訓練的穩定性和結果是有所提升的。對於訓練聲碼器來說，若使用的文句翻語音系統可以取得和原始音檔完全對應關係的聲學特徵值，推薦選擇WaveNet當作聲碼器的訓練。 而若是聲碼器和文句翻語音系統的訓練集是相同的，WaveRNN的MOS為4.16，是所有聲碼器表現中最好，WaveNet的MOS是4.10，也非常不錯，僅次於WaveRNN。 在單一語者的文句翻語音系統模型中，聲碼器訓練在大量資料的聲碼器來說，效果遠比不上專注在訓練在和文句翻語音系統相同的語者。

這次所使用的單語者的文句翻語音系統Tacotron 2，其生成結果非常逼近原始訓練者的特色，而且發音清晰無雜訊或回音，唯一的弱點是在語調上比真實人聲平淡，不過仍可視為和原始語者擁有同樣的音色。 因此聲碼器的訓練集

doi:10.6342/NTU202001501

為LJ Speech中，可發現在章節4.4的實驗，在相同語言相同語者的項目表現最優秀的WaveNet, WaveRNN在這樣子的情形表現仍最為出色。

而對於不同語者所訓練的聲碼器來說，其表現趨勢也和測試在一般人聲上類似。 整體來說，單語者的文句翻語音系統可以視為語調較平淡之人聲，因此若文句翻語音系統是已知的語者，可以訓練該語者在WaveRNN, WaveNet上以達到最好的效果，並且如果文句翻語音系統的架構是可以取得和原始音檔時間對應，將其對應關係訓練在WaveNet上，可以取得最佳的生成品質。

5.3 聲碼器在語者轉換系統的强健性比較

5.3.1 語者轉換模型介紹

本論文使用實驗室學長周氏所研究之語者轉換模型 [26]透過生成對抗式學習，使得就算沒有平行語料也可以達到語者轉換的目的。 所謂平行語料是指不同語者敘述相同的文句，而有著一一對應的關係。

此埋行之想法在於建立一個自動編碼器(Autoencoder)模型，就是透過編碼壓縮原始資訊，讓輸出和輸入相同的一種架構。 語者轉換模型的架構可看成兩個子

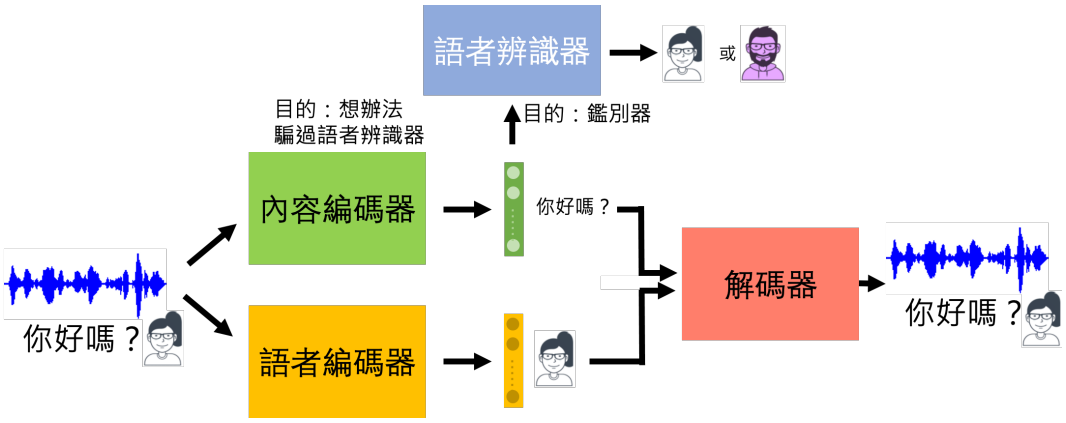


圖 5.2: 周氏語者轉換模型架構

doi:10.6342/NTU202001501

編碼器(內容編碼器、語者編碼器)、語者辨識器、解碼器。 其原理如下：

- 內容編碼器專注在把文字的內容、發音部分壓縮在輸出的向量上，而不含有任何語者的資訊。達成目的方法在於將壓縮過的資訊經過語者辨識器(鑑別器)吐出他是該語者的機率，並透過對抗式生成學習將內容編碼器和語者辨識器的參數做更新。

- 語者編碼器，將語者的資訊壓縮在輸出的向量。且當我們將內容編碼器輸出和語者編碼器輸出通過解碼器時，希望還原出來的音訊波形越接近原始音訊越好。

要生成語者轉換的文句時，只要替換語者編碼的輸出，再將綜合兩編碼器輸出一起通過解碼器就可以得到語者轉換的成果。 因此透過對抗式生成學習，可以在沒有平行語料的狀況下達成語者轉換。

我們此次選擇訓練語者轉換的訓練集為VCTK [27]，有包含多位男女英語語者，背景雜訊、口音都相當明顯。

5.3.2 實驗設計

上述周氏語者轉換模型是訓練在時頻譜上面的，故輸出也是時頻譜。 因此，我們額外加了用Griffin-Lim演算法來估計相位，當作一個比較標準值。 因訓練的聲碼器輸入是梅爾時頻譜，因此我們將周氏語者轉換模型輸出結果通過梅爾濾波器，變成梅爾時頻譜之後，再餵進聲碼器變成最終生成的結果。

我們所選擇的聲碼器訓練集有4.2中的 單語者英文的男性訓練集(En_M)、單語者英文的女性訓練集(En_F)、多語者英文的訓練集(En_L)、多語者多語言(Lrg_L)，以及訓練語者轉換模型訓練集的VCTK。

doi:10.6342/NTU202001501

本實驗產生的所有音檔放置在網站: https://bogihsu.github.io/Robust-Neural-Vocoding/

5.3.3 實驗結果和分析

從表5.2我們發現在聲碼器的生成結果比用傳統Griffin-Lim演算法來的更加自然，而且語者轉換系統生出來的結果通過梅爾濾波器做壓縮才經過聲碼器，因此若使用時頻譜當作聲碼器訓練的輸入，有機會可以更提升生成出來的音訊波形。

聲碼器架構	聲碼器訓練集				
	VCTK	En_M	En_F	En_L	Lrg_L
WaveNet	3.15±0.21	3.25±0.23	2.86±0.25	2.85±0.19	2.81±0.21
WaveRNN	3.54±0.20	3.21±0.23	2.98±0.23	2.88±0.22	2.90±0.21
FFTNet	2.71±0.22	2.19±0.21	2.30±0.23	2.28±0.23	2.51±0.21
Parallel WaveGAN	**3.83±0.20**	3.30±0.23	3.02±0.24	**3.45±0.20**	3.40±0.21
Griffin-Lim			2.72±0.21		

表 5.2: 以MOS呈現聲碼器測試在語者轉換模型輸出結果

從實驗結果可發現在我們所設定情境條件下聲碼器訓練集和語者轉換模型訓練集的語者相同時，WaveRNN和Parallel WaveGAN都相當適合，其中又以Parallel WaveGAN生成效果更加突出。 Parallel WaveGAN的表現上遠比其他聲碼器更適合當作語者轉換模型的聲碼器，不論在各訓練資料源都表現明顯比其他種類聲碼器來的更好。

此外Parallel WaveGAN只要找尋大量資料當作訓練集就可訓練出通用的語者轉換聲碼器，就算和語者轉換模型所包含的語者沒有重疊，也都可以有相當不錯

doi:10.6342/NTU202001501

的成果，因此我們會推薦可以使用Parallel WaveGAN來當作語者轉換系統最合適的聲碼器。

在這次所使用的語者轉換系統，其生成結果綜觀來看會較人類語音容易出現模糊的生成結果，而此時Parallel WaveGAN的生成方式是非自回歸模型的，生成過程中不會受到前一時刻生出的結果，最終音訊波形比較不會有殘響的效果，對於聽眾而言自然會較為接近人聲。 而相較於傳統估計相位的Griffin-Lim演算法來說，原本其估計相位的方法是根據原始人聲音訊所設計，對於生成較不完美的聲學特徵值時，估計相位的演算法的能力自然也會隨之受到影響，進而影響到生成出來的自然度。

5.4 本章總結

本章節將聲碼器訓練在人聲所抽取出的聲學特徵值上，應用在文句翻語音系統和語者轉換系統兩個語音生成的任務，並設定合適的參考值觀察出這兩類型的語者生成模型。

文字轉語音系統中，若文字轉語音系統輸出的聲學特徵值可以和原始音檔對齊當作聲碼器訓練的輸入和輸出，WaveNet是最合適的選擇。 若無法取得和原始音檔對齊的聲學特徵值，建議選擇WaveRNN或WaveNet，並且和文字轉語音系統相同語者的音檔直接做聲碼器的訓練，也可以有高品質的生成結果。 語者轉換系統中則是推薦Parallel WaveGAN為最好聲碼器模型，若使用和語者轉換系統中相同語者做訓練效果最佳，不然通過大量資料也可以訓練出通用的聲碼器。

從實驗分析來說，推測當我們所使用的語音生成系統品質非常清晰、近於人聲時，會推薦在人聲表現最好的WaveNet和WaveRNN。 若生成出來的聲學特徵值綜觀來看沒有那麼細緻，則會建議非自回歸模型的Parallel WaveGAN。

doi:10.6342/NTU202001501

第六章　結論與展望

6.1 研究貢獻與討論

本碩論透過實驗了解WaveNet, WaveRNN, FFTNet, Parallel WaveGAN四種聲碼器在訓練和測試資料分布不一致的時候，所表現的情形。

第3章中，比較各種不同聲碼器架構、參數量、生成速度外，還提供一些訓練、生成的細節可以提高最後生成穩定度和品質。

章節4.4在訓練和測試在不同語言或是不同語者的實驗中，發現聲碼器在生成語音時對於訓練時沒看過的語者品質會大幅下降， 而對於訓練時看過或沒看過的語言，生成品質都不受影響。 由於聲碼器是從聲學特徵值轉換至音訊波形的架構，語者音色如在訓練時沒看過比較難以重建某一語者原本音訊波形。而不同語言或可享成是由類似的較小的聲音單位的音訊組合而成，因此聲碼器在生成時不會因為訓練時沒看過某種語言而影響輸出結果。 章節4.5訓練在單一語者的資料集中，發現當訓練和測試語者有不同性別時輸出結果會導致生成結果大幅下降。 綜合第4章的兩個實驗，我們知道聲碼器訓練時語者的多樣性可以增進聲碼器的强健性，進而讓聲碼器在測試時有穩定且高品質的輸出。若在訓練時看過大量的不同語者的資料，生成時對沒看過的語者的敏感度會降低，而有較好的生成結果。

將聲碼器訓練在從人聲所抽取出的聲學特徵值上，在章節5.2和章節5.3分別應用在文句翻語音系統和語者轉換系統兩個語音生成的任務。 從實驗結果分析，當我們所使用的語音生成系統輸出信號品質非常清晰、近於人聲時，會推薦在人聲表現最好的WaveNet和WaveRNN。 若生成出來的聲學特徵值綜觀來看沒有那麼細緻，則會建議非自回歸模型的Parallel WaveGAN。

在本論文中比較了各種聲碼器架構以及分析了影響聲碼器生成品質的原因，

doi:10.6342/NTU202001501

了解到自回歸模型的優缺點，以及適合運用的情境場景。

6.2 未來展望

在本論文測試過程中有發現一些細節的施作方法會大幅影響最後生成品質，但這些細節測試實驗都僅是定性的而無定量的分析，是值得多做大量分析實驗去進一步觀察討論。 此外有很多細節測試實驗並沒有機會嘗試，若一一嘗試有機會可以更提高目前聲碼器的品質。

將聲碼器訓練在從人聲所抽取出的聲學特徵值上，再應用在更多語音生成任務上，去探討哪種模型更適合也是一個值得嘗試的題目。

透過本論文分析，未來可以設計出可以訓練上擁有更普遍的能力更為強健的聲碼器的訓練集，也可能搜集到更乾淨更合適的訓練集。

透過本論文分析各種聲碼器的優缺點，以及自回歸模型的優缺點，也希望未來能設計出更有普遍性、可即時生成、可運用於各式應用的聲碼器。

doi:10.6342/NTU202001501

參 考 文 獻

[1] 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.

[2] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

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

[4] 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.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[6] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[7] CCITT Recommendation, "Pulse code modulation (pcm) of voice frequencies," in *ITU*. 1988.

doi:10.6342/NTU202001501

[8] Paul Taylor, *Text-to-speech synthesis*, Cambridge university press, 2009.

[9] Yannis Stylianou, Olivier Cappé, and Eric Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[10] Claude Elwood Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[11] Ronald Newbold Bracewell and Ronald N Bracewell, *The Fourier transform and its applications*, vol. 31999, McGraw-Hill New York, 1986.

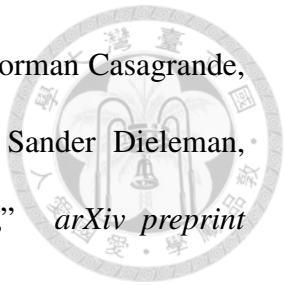
[12] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[13] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu, "Pixel recurrent neural networks," *arXiv preprint arXiv:1601.06759*, 2016.

[14] Tom Le Paine, Pooya Khorrami, Shiyu Chang, Yang Zhang, Prajit Ramachandran, Mark A Hasegawa-Johnson, and Thomas S Huang, "Fast wavenet generation algorithm," *arXiv preprint arXiv:1611.09482*, 2016.

[15] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, et al., "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

doi:10.6342/NTU202001501



[16] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu, "Efficient neural audio synthesis," *arXiv preprint arXiv:1802.08435*, 2018.

[17] Zeyu Jin, Adam Finkelstein, Gautham J Mysore, and Jingwan Lu, "Fftnet: A real-time speaker-dependent neural vocoder," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2251–2255.

[18] James W Cooley and John W Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[19] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems*, 2019, pp. 14881–14892.

[20] Yariv Ephraim and David Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE transactions on acoustics, speech, and signal processing*, vol. 33, no. 2, pp. 443–445, 1985.

[21] John Kominek and Alan W Black, "The cmu arctic speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[22] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu, "Libritts: A corpus derived from librispeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019.

doi:10.6342/NTU202001501

[23] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[24] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al., "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[25] Keith Ito, "The lj speech dataset," `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[26] Ju-chieh Chou, Cheng-chieh Yeh, Hung-yi Lee, and Lin-shan Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," *arXiv preprint arXiv:1804.02812*, 2018.

[27] Kirsten MacDonald et al. Christophe Veaux, Junichi Yamagishi, "CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit," 2017.

doi:10.6342/NTU202001501

附　　錄

以下為本次做完大量平均主觀意見分(MOS)實驗後，所觀察到的現象：

- 每人受試句子總數不宜太多，容易造成受試者前後標準不一。
- 當受試句子總數 > 30句，許多受試者會開始感到疲憊而不想繼續填寫，就算是好朋友友情幫忙填寫，在填寫完也多半會怨聲連連。
- 填寫問卷的報酬的好壞其實不太影響填答率，重點是要讓填問卷的人貼問卷時不需要花費過多的時間，或是需要使用複雜的介面完成。
- 對於不同方式取得平均主觀意見分(MOS)給予評分：
  - 好朋友: ★★★★★

    如果問卷搜集時程沒有很趕，相當推薦請好朋友填寫，大部分的好朋友會在兩三天之內或是將問卷放到週末填寫好，並且推薦在星期五或星期六的時候請好朋友填寫，免得好朋友隔了好幾天後會忘記填寫。因為好朋友不會亂填，所以整體上是非常推薦的，缺點是如果需要很多份問卷的話，僅靠好朋友是不夠的。
  - 實驗室同學: ★★★★☆

    實驗室同學對於原始訓練音檔都非常熟悉，對於真實音檔其實都有印象，先天條件下其實不太公平。 對於4-5分的音檔會相較一般人比較嚴苛，而3分以下的音檔相較一般人會容易給予較高的分數。 實驗室同學注意到的細節也比一般人更仔細，對於回聲較一般人敏感很多。綜合來說，實驗室同學們所填的結果變異會比一般人小很多，而集中在3-4分。

doi:10.6342/NTU202001501

– FB:NTU 台大學生交流版: ★★★★

台大學生交流版是由台大學生所建立的群組，可以抽獎數位同學發放酬勞或飲料，吸引同學填寫。 優點是同學們都會認真填寫，缺點是因為會受限於FB的曝光演算法，需要鼓勵同學們按讚留言延續貼文的熱度才能使貼文路人的能見度提升。 建議在設置貼文的時候，要麻煩貼寫的人標記(tag)別人，若被標記的人有來互動(按讚或留言)，路人能見率至少提升一個數量級。 問卷需要的花費的時間最好不要太多，會使同學剛打開問卷就萌生把問卷關掉的念頭。 若可以開發一個手機也可以填寫的模板，讓使用者可以使用更簡單的介面完成，通過FB:台大學生交流版的方法可以再增加一顆星。

– FB:論文問卷互助社: ★ ☆

FB:論文問卷互助社是一個公開社團，要求加入便可以參與問卷互助的部分。 有兩種方式:

1. 你填寫別人的問卷，並要求他回填你的問卷。
2. 別人填你問卷，並要求你回填他的問卷。

需要花費大量時間填寫問卷，且常常有人不回填問卷或是回填時亂填答，覺得效率蠻低落的。 建議需觀察一下他之前發文者有沒有回填的習慣，通常如果是學生論文的問卷比較容易認真回填。 如果問卷樣本相當缺少可以試試看此方法，但絕對不會是最優先推薦的方法。

– 批踢踢(發p幣): ★★★☆

批踢踢(PTT)是一個臺灣電子佈告欄(BBS)，將問卷發在上面徵求填寫，一份問卷發100 p幣(批踢踢錢的單位)，換算下來還不到新台幣2元。 因為p幣的價格非常便宜，想要鼓勵別人多填寫也可以發放更

doi:10.6342/NTU202001501

多p幣，對於發放問卷的人來說，所需花費的時間和金錢都不多，就算有一部分人亂填也不會覺得心疼。 獲得問卷速度也相當快速，是一種有效率獲得問卷的方式，不過要注意會有一部分人亂填問卷。

- Dcard: 不推薦

  Dcard平台有匿名的規定，發問卷的人無法給予填寫問卷的人任何好處，因此無法吸引人填寫問卷。故不推薦使用Dcard填寫問卷。

doi:10.6342/NTU202001501