國立臺灣大學電機資訊學院電機工程學系
碩士論文
Graduate Institute of Electrical Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

以激活函數引導與自適應實例正規化達成無監督式語音轉換

Unsupervised Voice Conversion using Activation Guidance and Adaptive Instance Normalization

陳延昊
Yen-Hao Chen

指導教授：李宏毅 教授
Advisor: Hung-Yi Lee, Ph.D.

中華民國一百一十年六月
June 2021

doi:10.6342/NTU202101337

誌謝

兩年半的日子稍縱即逝。因為高中同學 Mars 還有偉 J 室友的關係，我認識了系上的色鬼們吳宗翰還有謝濬丞。多虧他們，我才會大二就選到宏毅哥的機器學習，並在大三順勢跟了宏毅哥的專題。在專題聽到一篇學長報告的 paper，覺得很有趣，實做成語音版本，當成數位語音處理概論的期末專題，有幸受到琳山老師欣賞，吃到一次傳說中的「琳山宴」，也是一次難忘的回憶。

跟了一年多的專題，我最後也進入了語音實驗室這個大家庭。謝謝宏毅哥的指導，總是讓我們可以作自己喜歡的題目，也會和我們一起討論點子，給出很多建議，碩士期間也透過老師幫忙與介紹進入 AILabs 和 Ponddy 兩間公司實習，過程學習到很多在學校學不到的東西；感謝琳山老師在我報告 paper 的時候也會提出很多問題與評論，讓我更釐清 paper 的脈絡；在 AILabs 認識的達懿哥，帶我發了第一篇 conference paper，也在我作研究的路上給我很多靈感與建議；感謝彤恩姐，在我當網管的時候都要幫我處理一些我漏掉的事情，還請我吃早餐；感謝專題學長儒杰哥，以及網管們瑞陽哥、部長 G 良、很罩的 Yist、很色的學弟們凱為、成翰、瑋聰；也感謝實驗室的大家，每週的團體會議都會報告很先進、有趣的 paper；還有雖然不在 508，但都會一起吃吃的 G 伯翰以及簡室友和謝 Allen，有這些人一起才讓我在平常可以維持愉快的心情。

感謝我的家人，平常都在台北，一年只有寒暑假會搭飛機回金門，反而是爸媽現在比較常來台北找我。感謝平常和我一起打 LOL 的朋友們，可以講講屁話，放鬆心情。最後感謝我的女友櫻璇，平日很多時間都在實驗室，也比較少時間陪女朋友，感謝女友的包容與體諒。也謝謝這兩年多來女友的陪伴，在我實驗跑不出來、paper 寫不完，比較低潮的時候都會鼓勵我，明明自己才是最缺乏自信的，卻要在這些時候變成我的力量，未來也要一起加油！

doi:10.6342/NTU202101337

摘要

近年來，深度學習在語音轉換（Voice Conversion, VC）的應用與研究發展越來越多。從一對一語者的語音轉換（One-to-one）、多對多 (Many-to-many)、任意對任意（Any-to-any），以及一次性樣本（One-shot）語音轉換的研究逐漸成熟。許多語音轉換模型使用了表徵解纏的技術來分解一句語音中的語者特性以及文字內容，接著他們將文字內容，結合目標語者的語者特性來合成出轉換後的語音，達成語音轉換任務。在語音解纏的過程，我們會得到帶有語者特色的語者表徵（Speaker Embedding）及帶有文字內容特色的內容表徵 (Content Embedding)。一個常見的作法是，在內容表徵的抽取過程，加上資訊瓶頸讓語者資訊被過濾掉，但如果瓶頸加得太強，可能導致內容資訊的遺失，造成轉換出的語音品質不佳；如果瓶頸不夠強，又可能會讓語者資訊被過濾的不完全，導致轉換出的語音仍然帶有來源語者的特色，造成轉換失敗；這個現象即是語音解纏能力（Disentangling Ability）和語音重構能力（Reconstruction Ability）的取捨（Trade-off）。

本論文第一個部份提出了使用單一編碼器與自適應實例正規化（Adaptive Instance Normalization, AdaIN）來達成語音轉換，有效改善了前作在語音轉換的模型記憶體應用，不但大幅減少了前作模型的記憶體使用率以及運算速度，同時改善模型的輸出品質、語者相似度。在本論文的第二部分，我們嘗試探討不同的激活函數（Activation Function）對於語音表徵的解纏效果。我們使用前面提到的單一編碼器的架構，在其內容表徵上加入不同的激活函數，觀察不同激活函數在語音解纏能力和語音重構能力的取捨中，會帶來什麼不同的影響。實驗結果展示，與基礎模型（Baseline）相比，使用單一編碼器，搭配特定的 S 型函數（Sigmiode Function），能同時改善讓語音解纏能力和語音重構能力；在使用者主觀測試中，我們提出的方法也在語音品質的平均意見分數（Mean Opinion Score, MOS）和語

doi:10.6342/NTU202101337

者相似度分數取得最好成績。

doi:10.6342/NTU202101337

Abstract

Recently, the application and research development of Voice Conversion (VC) has increased. From one-to-one, many-to-many, any-to-any, all the way to one-shot VC, the research has gradually matured. Many deep-learning-based VC systems use the feature disentangling technique to separate the speaker information and the linguistic content information from a speech signal. These models convert the voice by changing the speaker information while maintaining the content information. In the process of feature disentangling, speaker embeddings and content embeddings are extracted from an audio. Applying information bottleneck on content embeddings is a general way to disentangle speaker information from content embeddings. However, the content information might be lossy if the bottleneck is too strong, which results in low-quality conversion; otherwise, the speaker information may leak into content embeddings due to a weak bottleneck. In short, there is a trade-off between disentangling ability and reconstruction ability.

In this thesis, we firstly propose to use a single encoder with Adaptive Instance Normalization (AdaIN) to achieve VC, which reduces the memory usage, meanwhile improves the voice quality and the speaker similarity of generated speech. In the second part of the thesis, we explore the effects of different activation functions on speech representation. We use the single-encoder model mentioned above as the baseline model and add different activation functions to the content embedding to see how the activation functions affect the results.

The experiment results show that using a single encoder with a proper sigmoid function applied on the speech representation improves the disentangling ability and reconstruction ability at the same time. The proposed method also obtains the best performance

doi:10.6342/NTU202101337

of the subjective evaluation, including the naturalness test and the speaker similarity test.

doi:10.6342/NTU202101337

目錄

doi:10.6342/NTU202101337

doi:10.6342/NTU202101337

圖目錄

doi:10.6342/NTU202101337

doi:10.6342/NTU202101337

表目錄

doi:10.6342/NTU202101337

第一章　導論

1.1　研究動機

近年來，深度學習（Deep Learning）[1] 技術結合其他領域的研究越來越成熟，也有許多技術已經普及到我們的生活中，例如，電腦視覺（Computer Vision, CV）的應用有：公司的上下班打卡系統、手機的解鎖系統、自動車的物體辨識系統，以及醫療影像辨識等等；語音領域也利用深度學習的技術，在語音辨識（Automatic Speech Recognition, ASR）[2] 的表現超越了傳統的統計模型，除此之外，還有語音合成（Text-to-Speech, TTS）、喚醒詞（Voice Triggering）等等，被廣泛應用在手機上；在自然語言處理（Natural Language Processing, NLP）中，問答系統（Question Answereing）、推薦系統（Recommendation System）也已經實作在許多社群平台的後端演算法。值得一提的是，深度學習在 ImageNet[3]，一個影像辨識任務的基準（Benchmark）中，辨識正確率已經超越人類；另外，在 SQuAD[4]，一個問答任務的基準中，也已經表現得比人類還要好。能有如此耀眼的成果，多虧了電腦矩陣運算能力的跳躍性提升，以及網路普及之後，大數據（Big Data）時代的來臨。

除了許多判別式（Discriminative）的任務，機器已經能夠超越人類，目前也有許多人投入深度學習在生成式（Generative）模型的研究中。電腦視覺領域中的生成模型研究成果，目前已經可以產生出非常高品質的圖像，還能應用在影片的換臉。而語音（Speech）領域的生成模型，也普及在智慧型手機的智能助手，許多大公司也推出各自的機器發音系統。雖然這些發音系統聽起來都非常近似真人，品質也非常高，但如果我們仔細觀察的話，不難發現這些發音系統，往往侷限於那些有標註的、語料豐富的語言（High Resource Languages），例如英文、日文、

doi:10.6342/NTU202101337

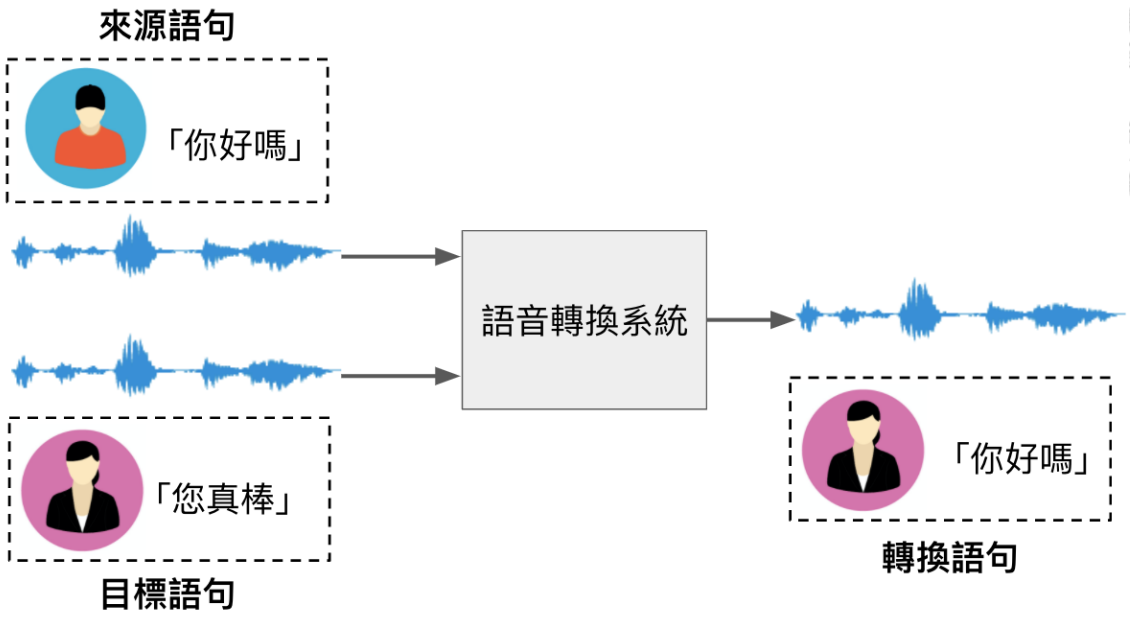


圖 1.1: 語音轉換系統

中文等。至今，語料豐富語音生成系統的研究，在聲音輸出的品質上已經非常成熟，許多研究已經開始往模型壓縮與加速的方向努力著。然而，在標註較少、語料稀少的語言（Low Resource Languages）中，我們要如何能利用這些資料有效的訓練出一個語音生成模型呢？

語音轉換（Voice Conversion, VC）也屬於一種語音生成的任務，流程如圖 1.1。一般來說，語音轉換任務需要兩個不同人說的話當作輸入，分別是來源語音（Source Speech）和目標語音（Target Speech）[1]，語音轉換模型會將來源語音的文字內容（Linguistic Content）保留，並且將來源語音的語者聲音特色，轉換成目標語音的語者聲音特色，合成出一句新的語音。語音轉換研究中，如果我們有成對標註好的平行語料（Paralleled Data），也就是說，我們有大量兩個語者說相同的句子的資料，那我們可以很直觀的訓練這個語音轉換模型：使用第一個語者和第二個語者說同句話的聲音，分別當作模型的來源語音以及模型輸出的答案，而

[1]本論文提及之「目標語音」指涉模型輸入之「目標語者的語音」，並非模型訓練目標。

doi:10.6342/NTU202101337

模型的目標語音，則使用第二個語者說的另一句話當作輸入。不過，要蒐集大量的平行語料，往往是比較困難的。實際上，我們容易取得的是大量不成對的語音資料。近年來，許多研究都著墨在這類非平行語料的語音轉換；另外，一次性樣本語音轉換，也慢慢受到矚目。大多數這類模型都利用了表徵解纏（Feature Disentangling）的技巧，設計出不需要平行語料、轉換時不限於訓練資料中的語者也能分離出語者表徵的語者編碼器，以及能夠分離出文字內容表徵的內容編碼器，再利這些用解纏之後的表徵，透過解碼器生成出最後的音訊。

以往的方法，大多針對語者表徵和內容表徵抽取，分別訓練兩種不同的編碼器。本論文將會說明如何使用單一編碼器，同時達到語者表徵和內容表徵的解纏。另外，本論文另外探討了不同激活函數（Activation Functions）應用在內容表徵上之後，對於語音轉換模型整體表現的影響。

1.2 研究方向

本論文的主要研究方向為，基於非平行語料，以自編碼器（Autoencoder）為主要架構，來達成語音表徵解纏以及語音合成，進而達到一次性樣本的語音轉換。研究方向主要包含:

- 利用自適應正規化（Adaptive Instance Normalization, AdaIN），實現單一編碼器達成語者表徵與內容表徵解纏
- 探討不同的激活函數對於隱藏語音表徵的影響，以及其間接對語音轉換品質和語者相似度的影響

doi:10.6342/NTU202101337

1.3 相關研究

一次性樣本（One-shot）語音轉換的意思是，在模型的推論（Inference）階段，來源語者和目標語者可以是任意語者；也就是，模型在訓練的過程中，可以不用看過這些語者的語料，同時，我們只需要該語者提供一句大約三秒的語料，就可以達成語音轉換。近年來，有許多研究開始探討一次性樣本的語音轉換該如何實現。

前人提出 AdaIN-VC [5]，將語音轉換視為一種風格轉換（Style Transfer）任務，借鏡了電腦視覺在風格轉換的研究 [6]，把語者特徵視為一段語音的風格，也就是這段語音的全局訊息（Global Information），並且在內容編碼器中加入了實例正規化（Instance Normalization）層，試圖過濾掉一段語音中的全局訊息，保留內容訊息；另外設計的語者編碼器，在最後一層使用平均池層，取得整段語音的全局訊息，代表語者表徵；最後在解碼器的解碼過程，利用自適應實例正規化（Adaptive Instance Normalization, AdaIN）將語者表徵融入進內容表徵，達到語者轉換。

AutoVC [7, 8] 系列的研究使用了預先訓練在語者分類任務的 D 向量（D-vector）[9] 來當作語者表徵，提出資訊瓶頸（Information Bottleneck）的想法，讓編碼之後的隱藏表徵能代表內容資訊，且不帶有語者資訊，解碼器再透過 D 向量和隱藏表徵來合成音訊。其核心概念為，解碼器在重建音訊的過程，需要該音訊的語者資訊以及內容資訊，而語者資訊可以透過 D 向量提供，那麼我們可以預期隱藏表徵就會包含需要的內容資訊，同時我們在能夠重建音訊的前提之下，將隱藏表徵的維度越調越小，形成一種資訊瓶頸，這時自編碼器為了要重建音訊，會傾向丟去不必要的資訊（已經由 D 向量提供的語者資訊），只保留內容資訊，達到語音解纏的效果。

VQVC [10, 11] 系列作品將內容資訊當作離散表徵，除了實例正規化，另外提出在內容表徵上套用向量量化（Vector Quantization, VQ）[12] 層，強制內容表徵

doi:10.6342/NTU202101337

具有離散特性。其向量量化造成的副作用，會增加音訊的重建難度，導致語音品質下降。而 VQVC+ 應用 U-net [13] 架構，利用跳躍連接（Skip-connection）改善語音重建的難度，進而提升輸出音訊的品質。

上述這些研究，都能達到一次性樣本語音轉換。然而，一次性樣本語音轉換任務至今仍然是一個艱難的任務。AutoVC 有一個明顯的缺陷，就是它要先訓練好語者分類模型，需要標註好語者的語料；再者，[14] 指出，使用分類任務預訓練好的 D 向量，雖然在表徵空間中，相似的語者的表徵會比較接近，不相似的語者表徵會比較遠，但是對於語音轉換這種生成語音的任務，D 向量卻不一定是很有效的語者表徵。AdaIN-VC 使用了語者編碼器和內容編碼器分別抽取語者表徵和內容表徵，但是使用自適應實例正規化，其實我們並不一定需要額外的語者編碼器，也就是說，我們實際上只需要單一個編碼器就可以同時得到語者表徵和內容表徵。VQVC+ 雖然透過向量量化有效的將語者資訊從隱藏表徵中分離，但受限於量量化本身離散的性質，它的語音合成品質仍然有進步空間，而這也是在 [7, 11] 指出的，語音解纏能力和語音重構能力會形成一個取捨。

1.4 主要貢獻

本論文主要貢獻如下：

- 使用單一解碼器達到語者資訊和內容資訊解纏

  基於前人的研究 AdaIN-VC，有效利用自適應實例正規化的良好性質，將額外的語者編碼器移除，大幅降低模型需要的運算資源。

- 結合 U 型網路傳遞語者資訊

  受到 VQVC+ 啟發，結合 U 型網路，將資訊瓶頸較大的語者資訊利用跳躍連

doi:10.6342/NTU202101337

接傳遞至編碼器。

- 探索不同的模型架構搭配不同激活函數的效果

  受到 AutoVC 啟發，本論文提出使用激活函數作為資訊瓶頸，探索不同激活函數對模型的影響，找到一個最適合 AdaIN-VC 的激活函數，進而改善語音轉換的整體表現。

1.5 章節安排

本論文之章節安排如下：

- 第二章：介紹本論文相關背景知識。
- 第三章：介紹如何使用單一編碼器與自適應實例正規化達成語音轉換。
- 第四章：探討不同激活函數對資訊解纏的影響。
- 第五章：本論文之結論與未來研究方向。

doi:10.6342/NTU202101337

第二章　背景知識

2.1　深層類神經網路

類神經網路（Neural Network, NN）[15] 是一種仿生物結構的數學模型，期望這個數學模型可以像生物的神經元受到訊號刺激後，有對應的不同反應。具體而言，類神經網路透過訓練資料與反向傳播法（Back Propagation），去擬合出某種輸入和輸出的對應關係。

2.1.1　全連接類神經網路

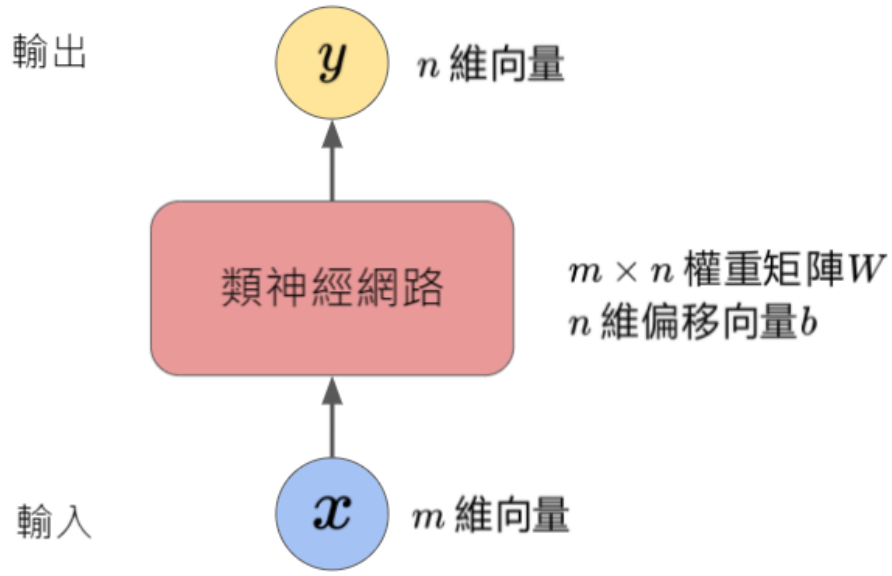


圖 2.1: 單層全連接類神經網路。輸入為 m 維向量，經過運算後輸出 n 維向量。

圖 2.1展示單一層全連接類神經網路，其數學模型由一個權重矩陣和偏移向量組成，輸出 y 與輸入 x 的對應關係為

doi:10.6342/NTU202101337

$$y = Wx + b, \tag{2.1}$$

其中 W 為類神經網路的權重矩陣，b 為偏移向量。對於輸出向量 y 的每一個元素，輸入向量 x 的每個元素都佔有一定權重，因此稱此網路為全連接類神經網路。

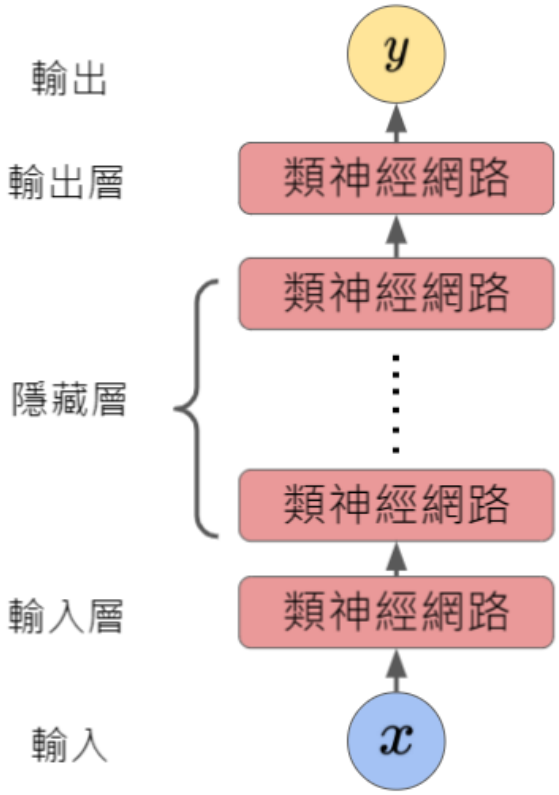


圖 2.2: 深層全連接類神經網路。由多層全連接類神經網路組成，包含輸入層、數個隱藏層，以及輸出層。輸入向量會依序和這些類神經網路運算，最後得到輸出向量。

拜現代科技所賜，矩陣運算能力大幅提升，類神經網路可以疊非常多層，圖 2.2 展示了最單純的深層類神經網路：深層全連接類神經網路。深層全連接類神經網路是單層全連接類神經網路進一步擴展，透過多層類神經網路的堆疊達到模型的深度運算。

doi:10.6342/NTU202101337

2.1.2 卷積式類神經網路

卷積式類神經網路（Convolutional Neural Network, CNN) 使用了核（Kernel）和窗（Window）的概念，將一段訊號用卷積窗切成不同的區塊，同一個區塊要對所有卷積核做內積運算，而不同的區塊對應到的卷積核是一樣的，因此具有空間相對不變性（Spatially Invariance）。而其空間相對不變性讓它被廣泛應用在電腦視覺的任務之中，例如影像辨識任務，一張全彩圖片可以用一個三通道（紅、綠、藍三個通道）乘以長乘以寬的向量來表示，而其中一個物品在圖片中的位置並不會影響它是什麼物件，因此使用二維卷積式類神經網路（2d-CNN）往往能有效抽取出影像的隱藏表徵。

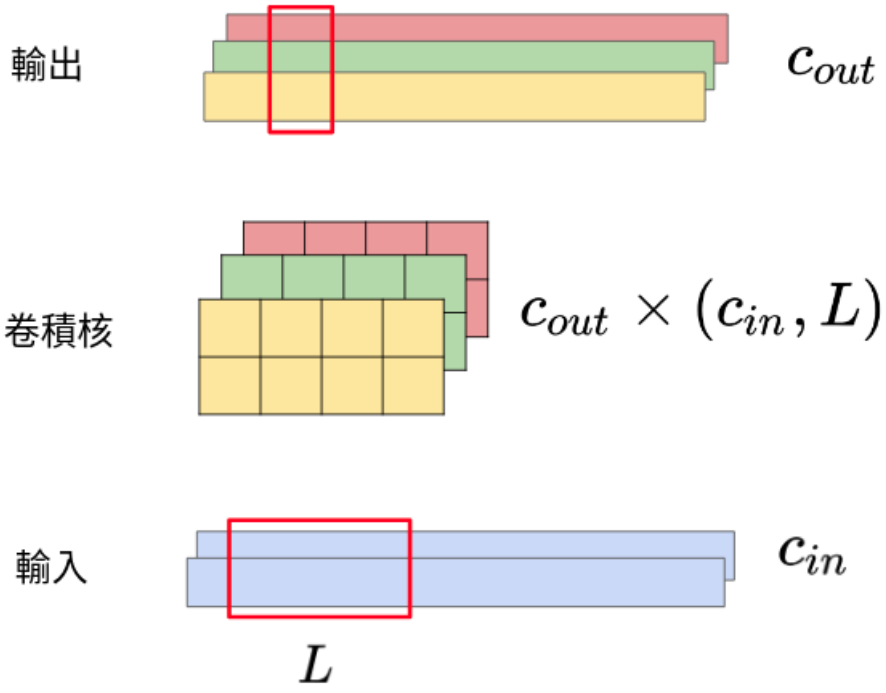


圖 2.3: 卷積式類神經網路。輸入端紅色框框為窗，輸出端紅框部份對應到輸入端紅框部份與卷積核運算後的結果。

圖 2.3展示了一個一維卷積式類神經網路。輸入為兩個通道（$c_{in} = 2$）的向量序列，輸出為三個通道（$c_{out} = 3$），窗口大小為 $L = 4$，卷積核則是由三個大小為

doi:10.6342/NTU202101337

2×4 的二維向量組成。

不只是電腦視覺，在語音訊號處理，這種空間不變性也是一個非常良好的性質。舉例來說，如圖 2.4所示，本論文使用的一個聲學特徵（Acoustic Feature）為梅爾時頻譜（Mel-spectrogram），可用一個多通道（取決於使用者定義，本論文使用 80）乘以時間長度的二維向量來表示，而一段聲音訊號在時間軸上被平移之後，也不會改變那段聲音訊號本身的音量、音素（Phoneme）、音調（Pitch）等等特徵，因此我們會使用一維卷積式類神經網路（1d-CNN）來抽取音訊的隱藏表徵。

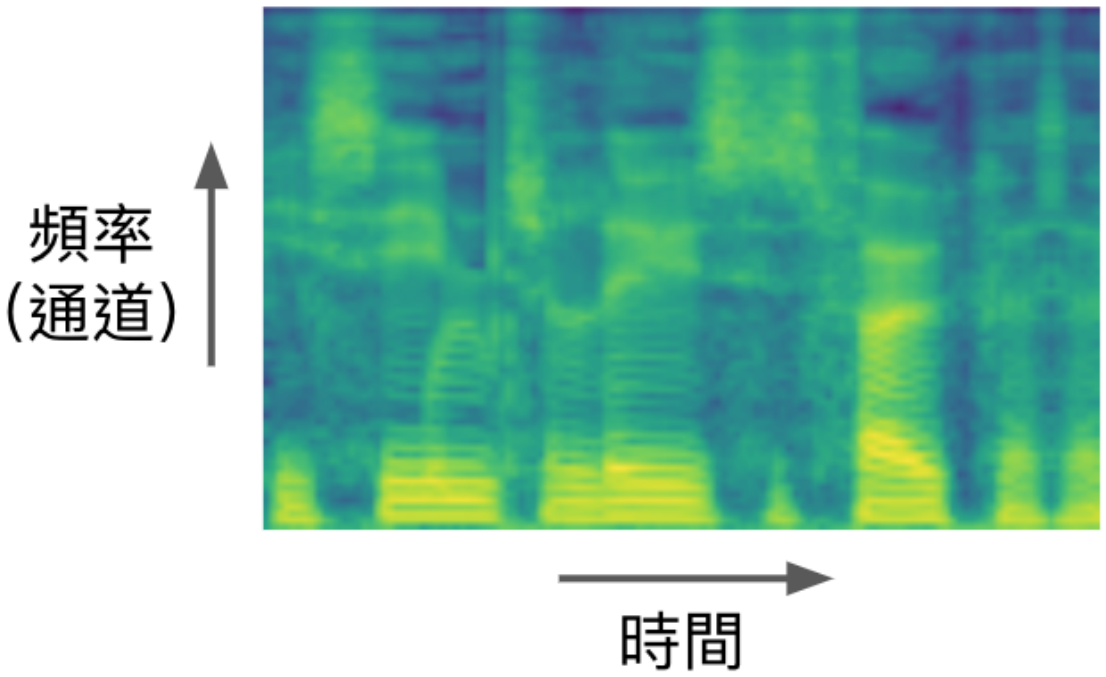


圖 2.4: 梅爾時頻譜，X 軸為時間軸，Y 軸為梅爾刻度的頻率軸，其中 Y 軸會固定一個通道數目，每一個通道代表一個頻段。

2.1.3 遞歸式類神經網路

遞歸式類神經網路（Recurrent Neural Network, RNN）也是一種類神經網路的變形，常見的遞歸式網路有長短型記憶（Long Short-Term Memory , LSTM）[16] 和門控

doi:10.6342/NTU202101337

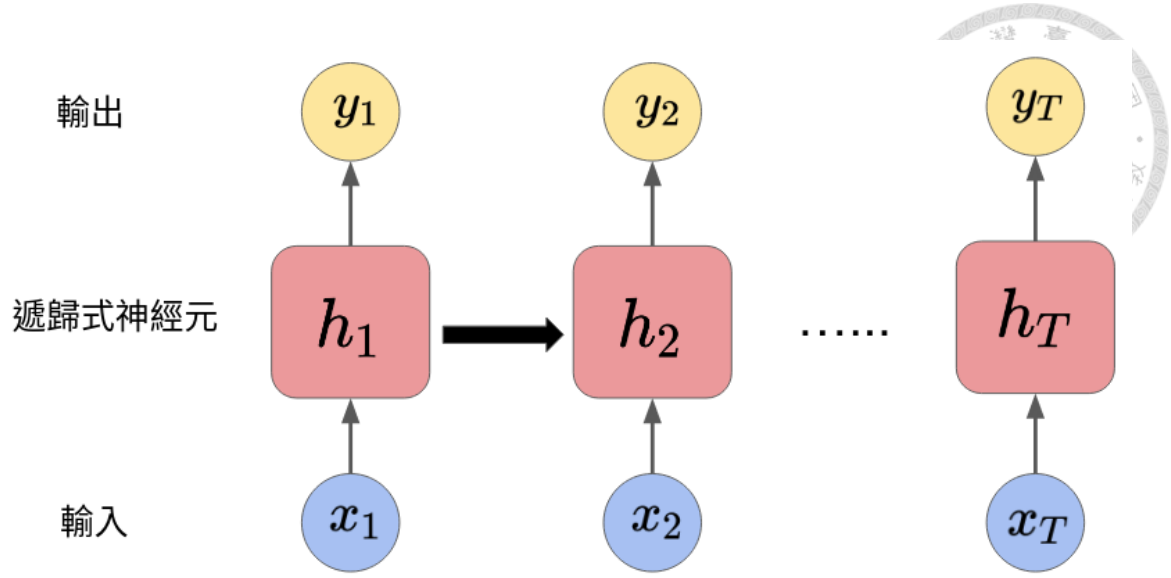


圖 2.5: 遞歸式類神經網路。輸入 x 為一個向量序列，遞歸式神經元依序與輸入向量運算，並在每一次運算之後改變神經元的隱藏狀態 h。

循環單元（Gated Recurrent Unit, GRU）[17]。它們的核心概念為記憶單元：一個可以儲存當前的狀態的神經元，並且隨時更新狀態，因此擁有記憶的特性。如圖 2.5所示，遞歸式類神經網路在實際運行時，並不會一次讀取整段輸入訊號，而是將輸入訊號在某一個維度上（通常是時間軸）一個一個讀取進來，每一次讀取都會做一次運算。這每一次運算，會根據當前輸入 x_t 以及網路當前狀態 h_t 來得到輸出 y_t，同時也更新網路當前狀態至 h_{t+1}：記住當下重要的資訊，並將一些不重要的資訊捨去。而這個記憶特性以及它依序讀取資料的特性，使得遞歸式類神經網路常被用來處理有時間序列概念的訊號，例如輸入是文字、影片等等。而聲音訊號也具有時間序列概念，因此遞歸式類神經網路也常常被應用在語音轉換的任務中。

遞歸式類神經網路可以是單向讀取，也能是雙向（Bidirectional）[18] 讀取。舉雙向的遞歸式類神經網路為例，如果我們今天的輸入長度為 T 幀的音訊，那麼這個網路能夠從第一幀至第 T 幀依序讀取資料，並且輸出一個長度為 T 幀的

doi:10.6342/NTU202101337

結果；同樣的，也要從第 T 幀至第一幀讀取資料，也輸出一個長度為 T 幀的結果；而這個雙向的網路最終輸出，就是將這兩個 T 幀的結果，每一幀的表徵串接（Concatenate）在一起。顯而易見地，雙向遞歸式類神經網路要求輸入要是有頭有尾的訊號，否則在做從尾到頭的那個方向的運算就會需要額外的運算時間。只要我們允許每次任務可以是非實時的輸入輸出，那麼我們在使用雙向遞歸式類神經網路時，就能保有幾乎不變的運算時間。

2.1.4 激活函數

激活函數（Activation Function）對於深層類神經網路來說，是至關重要的一種元素。激活函數通常是一個非線性的可微函數，其非線性的性質，讓深層類神經網路可以模擬更多複雜的非線性的函數。常見的一種激活函數是 S 型函數（Sigmoid Function），定義為

$$\text{Sigmoid}(x) := \frac{1}{1+e^{-x}}.$$

其函數圖形如圖 2.6 ，被用來模擬神經細胞的開、關兩種狀態，實際應用時，可

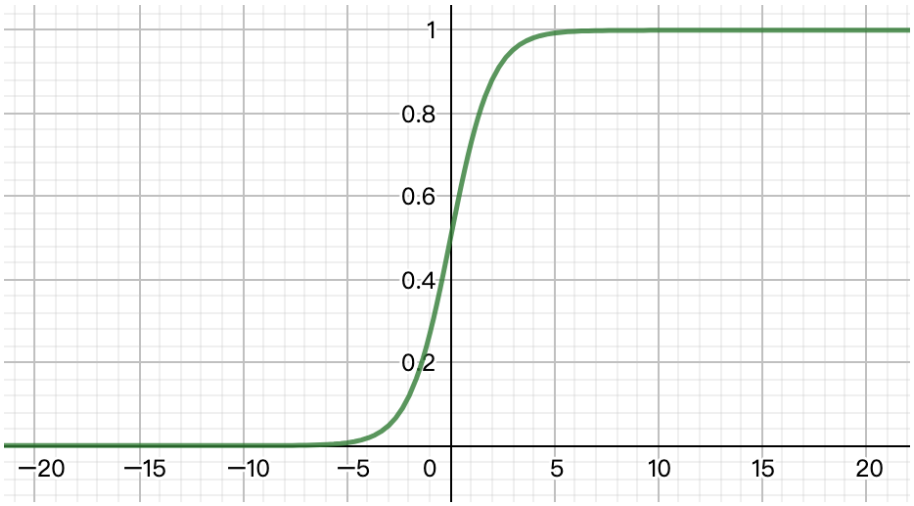


圖 2.6: S 型函數

doi:10.6342/NTU202101337

以搭配二元交叉熵（Binary Cross Entropy）損失函數（Loss Function），來達成二元分類的任務。處理多類別分類任務時，則會搭配軟性最大化函數（Softmax）

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}.$$

與交叉熵（Cross Entropy）損失函數優化模型（圖 2.7）。其他常用的激活函數還

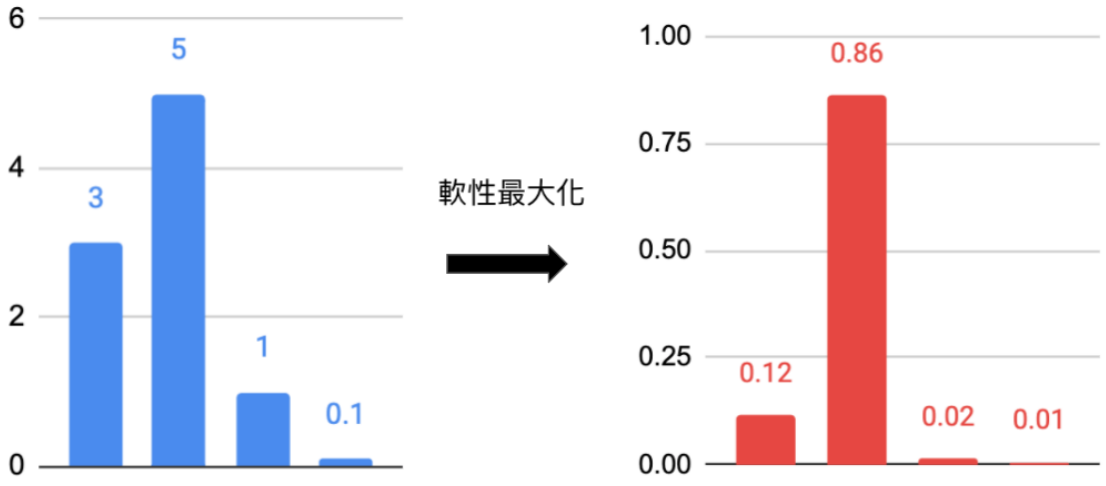


圖 2.7: 軟性最大化函數

有如圖 2.8 線性整流單元（Rectified Linear Unit, ReLU）[19]，

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

以及其諸多變形。有了這些非線性激活函數的加入，使得深層類神經網路具有能夠近似任何函數的潛能。

doi:10.6342/NTU202101337

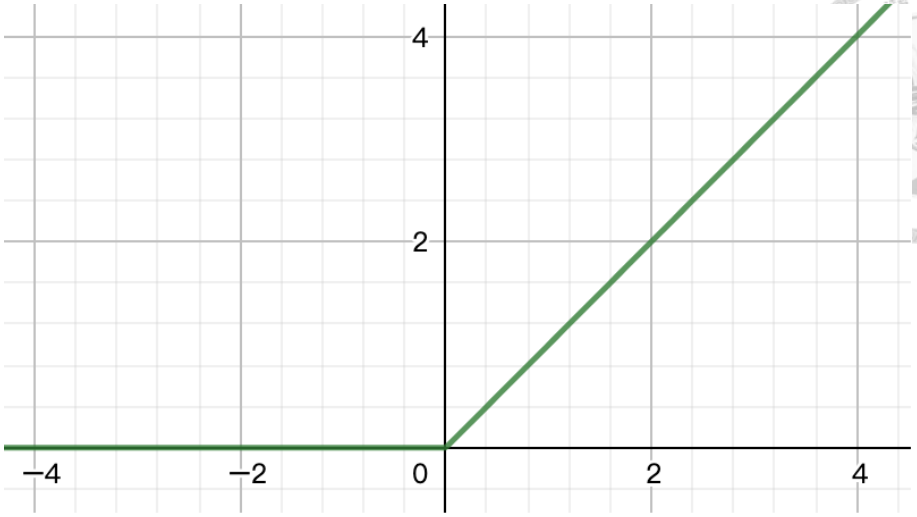


圖 2.8: 線性整流單元

2.2 資訊解纏

在深度學習領域中，我們時常希望有一個表徵抽取模型，能夠從輸入資料中萃取有效的資訊化為隱藏表徵，再交由下游任務模型訓練。然而，隱藏表徵往往帶有很多輸入訊號的不同特性，舉例來說，一段聲音訊號的隱藏表徵，可能夾雜著語者資訊、內容資訊等等。接下來，將介紹如何透過模型設計來將這些資訊從隱藏表徵裡分離出來，達到資訊的解纏效果。

2.2.1 自編碼器

在深度學習的領域中，自編碼器（Autoencoder）是一個具有自動編碼能力的模型。編碼（Encoding）本身是將訊息映射到另一種形式的過程，搭配合適的解碼（Decoding）可以將轉換後的形式再轉回原始訊息。我們需要編碼，通常是因為編碼之後的形式，會有原始訊息不帶有的良好性質，常見的應用有加密解密、資料壓縮等等。而編碼又分為無損（Lossless）和失真（Lossy）兩種形式，在深度學習

doi:10.6342/NTU202101337

領域的自編碼器，通常是失真編碼。

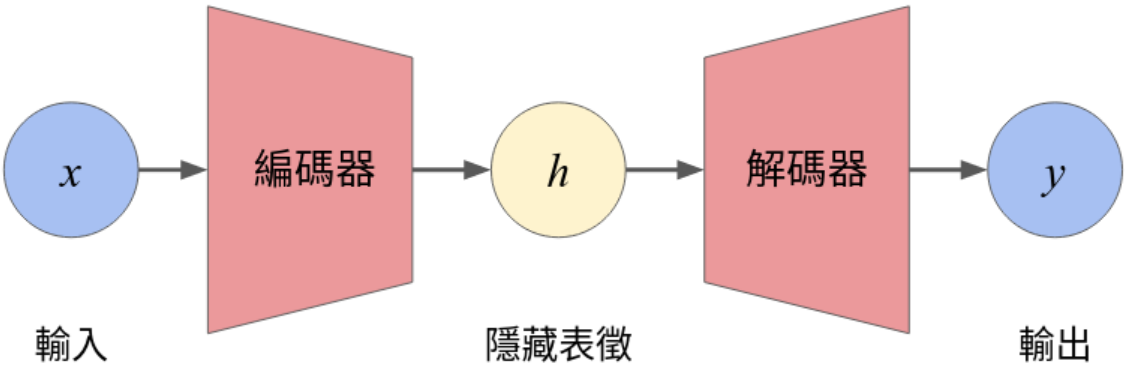


圖 2.9: 自編碼器

圖 2.9為一個通用的自編碼器架構，包含一個編碼器（Encoder）和一個解碼器（Decoder）。通常，編碼器會將一個輸入訊號映射到一個隱藏表徵空間中，解碼器則是負責將這個隱藏表徵重建原本的輸入訊號。以數學式子來表示的話，我們定義一些符號如下：

- x: 輸入訊號
- h: 隱藏表徵
- y: 輸出訊號
- E: 編碼函數
- D: 解碼函數

定義編碼過程為

$$h \coloneqq E(x), \tag{2.2}$$

解碼過程則為

$$y \coloneqq D(h). \tag{2.3}$$

doi:10.6342/NTU202101337

訓練這樣的自編碼器，我們並不需要標註過的資料，或者說，資料本身就是自己的標註。其損失函數 L 可以定義為

$$L := d(x, y), \tag{2.4}$$

其中 d 通常是 L1 距離或是 L2 距離。這樣自我重建的訓練過程，是一個自監督式學習（Self-supervised Learning）的過程。我們期待透過自監督式學習演算法訓練好的編碼器，抽取出來的隱藏表徵，能夠更好地被使用來當作下游任務（Downstream Task）的輸入特徵。

2.2.2 資訊瓶頸

透過深度學習演算法得到的自編碼器，幾乎都是失真的。更甚者，在設計隱藏表徵空間的通道數（Channel）時，只要選擇一個比輸入訊號的通道數還要小的數字，那就保證這個自編碼器是失真的（嚴格來說，要小於組成輸入訊號空間基底（Basis）的總數）。這個現象，就是資訊瓶頸（Information Bottleneck）。這個通道數目設定得越小，表示資訊瓶頸越大。除了限制通道數之外，向量量化（Vector Quantization, VQ）[12] 或使用激活函數也會形成一種資訊瓶頸。

2.2.3 自編碼器隱藏表徵解纏

一般而言，語音轉換任務會假設一段語音訊號同時帶有語者資訊（Speaker Information）和內容資訊（Content Information），且語者資訊和內容資訊帶有的資訊是互相獨立的。令 x 為聲音訊號，s 和 c 分別代表語者資訊和內容資訊。假設我們給定一段輸入訊號 x，它可以由另外兩段訊號 s 和 c 表示，且滿足 s 和 c 互相

doi:10.6342/NTU202101337

不影響，

$$H(x) = H(s; c) \tag{2.5}$$

$$= H(s|c) + H(c) \tag{2.6}$$

$$= H(s) + H(c), \tag{2.7}$$

其中 H 表示熵。如果我們有一個預訓練好的語者分類器，它的編碼函數 E_s 可以從 x 完美抽取出 s 的資訊，

$$h_s \coloneqq E_s(x), \tag{2.8}$$

$$H(h_s) = H(s), \tag{2.9}$$

那麼，[7] 指出，我們可以利用這個語者分類器和自編碼器的資訊瓶頸，來訓練另一個編碼器 E_c 來抽出 c 的資訊。簡單的數學推導如下：

$$h_c \coloneqq E_c(x), \tag{2.10}$$

$$y \coloneqq D(h_c, h_s). \tag{2.11}$$

在理想的自編碼器模型下（重建的訊號和輸入訊號總是相同），會有

$$H(x) = H(y), \tag{2.12}$$

由 式 (2.11) 和 式 (2.12) 我們可以推出

$$H(y) = H(h_c; h_s) \tag{2.13}$$

$$= H(h_c; s) \tag{2.14}$$

$$= H(h_c|s) + H(s), \tag{2.15}$$

$$= H(h_c) - I(h_c; s) + H(s), \tag{2.16}$$

doi:10.6342/NTU202101337

其中 I 表示相互資訊（Mutual Information, MI）。結合式 (2.7) 和式 (2.16)，得

$$H(c) = H(h_c) - I(h_c; s), \tag{2.17}$$

換句話說，

$$I(h_c; s) = H(h_c) - H(c), \tag{2.18}$$

由於 $H(c)$ 是定值，因此只要我們減少 h_c 的通道數，也就是降低 $H(h_c)$，那麼 $I(h_c; s)$ 也會被降低，而 $I(h_c; s)$ 被降低，代表我們的內容表徵帶有的語者資訊越少。當內容表徵的通道數越降越低，又同時保有不失真的編碼效果時，這樣的編碼器 E_c，就越接近完美的編碼器，只從 x 中分離出 c 的資訊。這整個過程，就是利用自編碼器和資訊瓶頸達到隱藏表徵的解纏。

2.3 語音生成

2.3.1 聲學特徵

一段聲音訊號，如果單純從時域（Time Domain）來看，其實很難看得出和音素、語者特徵相關的端倪。但如果我們將聲音訊號透過短時距傅立葉變換，得到的時頻譜就有很好的可解釋性。基本上，透過訓練，連人類都能從時頻譜上「看」出這句話在說什麼 [20]。如同2.1.2所說，使用梅爾時頻譜當作模型的輸入聲學特徵，可以搭配卷積類神經網路有效地抽取下游任務需要的表徵。

除了使用時頻譜，我們知道對於語音識別任務，使用共振峰基本上就可以判斷音素（Phoneme），因此對於這類任務，重要的資訊其實是它的頻譜包絡。我們將梅爾時頻譜再做一次離散餘弦變換（Discrete Cosine Transform, DCT）得到梅爾倒頻譜係數（Mel-Frequency Cepstral Coefficients, MFCC），將低維的係數留下，就能表示頻譜包絡。

doi:10.6342/NTU202101337

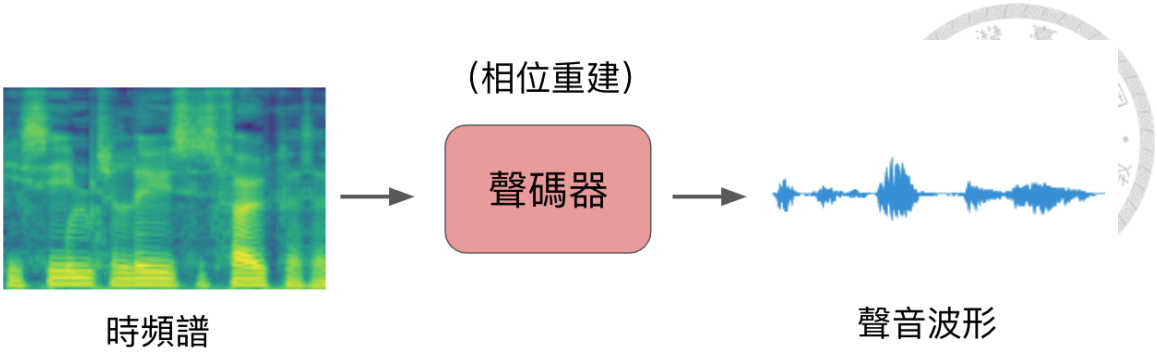


圖 2.10: 聲碼器

由於這些聲學特徵相對於時域的聲波具有許多良好的性質，常常被使用在深度學習中。然而，時頻譜如果缺少對應的相位（Phase），雖然對於語音判別性的任務往往能有很好的效果，但如果要合成回時域波形，必須要有相對的相位；如果對應的相位資訊是錯誤的，那麼合成出的聲音訊號將會帶有嚴重的機械音。

2.3.2 聲碼器

語音轉換任務通常不會直接使用聲音原始波形當作模型的輸入和輸出特徵，而是會先將原始波形轉成時頻譜或是梅爾時頻譜當作模型的聲學特徵。由於一段聲音訊號是由時頻譜和對應的相位決定的，因此，要將時頻譜重建回聲音波形，我們就需要使用聲碼器（Vocoder）。如圖 2.10 所示，聲碼器是跟據時頻譜預測對應相位，合成回聲音波形的技術。傳統基於規則的聲碼器有葛氏林氏演算法（Griffin-Lim Algorithm）[21]，使用迭代算法，目標讓重建訊號的時頻譜和給定的時頻譜越像越好。現在有許多使用深度學習技術的類神經網路聲碼器出現[22, 23, 24, 25, 26, 27]，不但合成的聲音品質遠遠超過傳統演算法，連速度都已經不再是瓶頸。基本上，目前的類神經聲碼器已經完全取代了傳統的葛氏林氏演算法。

本論文中所使用的聲碼器為 MelGAN[26]，使用對抗式生成模型（Generative

doi:10.6342/NTU202101337

表 2.1: 不同聲碼器生成的音訊品質平均意見分數比較（本表取自 [26]）

模型	平均意見分數	95% 信賴區間
葛氏林氏演算法	1.72	±0.07
MelGAN	3.49	±0.09
原始音訊	4.19	±0.08

Adversarial Networks, GAN）達成非自回歸（Non-autoregressive）語音生成，相對於其它類神經聲碼器，具有運算速度快的優勢。表 2.1為不同編碼器生成的語音品質的平均意見分數（Mean Opinion Scores, MOS）測試，其評分標準為

- 5 分：非常好
- 4 分：好
- 3 分：普通
- 2 分：差
- 1 分：非常差

2.4 本章總結

本章分成三大部分：

- 介紹深度學習框架的幾種類神經網路以及非線性激活函數

  首先我們介紹深層深度學習的基本類神經網路，接著介紹本論文模型大量使用到的卷積式類神經網路及遞歸式類神經網路以及它在語音表徵抽取的應

doi:10.6342/NTU202101337

用，最後介紹讓深層類神經網路具有非線性特質的激活函數。

- 自編碼器與資訊瓶頸的概念

  介紹語音轉換的核心架構自編碼器，以及簡單推導如何利用資訊瓶頸實踐語音表徵的解纏。

- 本論文所使用到的輸出與輸入聲學特徵，以及使用到的聲碼器

  本論文主要任務為語音轉換，不直接使用聲音波形，而是使用時頻譜作為模型的輸入與輸出聲學特徵。時頻譜具有良好的解讀性，搭配卷積神經網路，能有效地抽取出隱藏語音表徵。透過聲碼器，將模型輸出的時頻譜再轉換至聲音波形。

doi:10.6342/NTU202101337

第 三 章　使用單一編碼器與實例正規化達成語音轉換

3.1　簡介

本章節將詳細說明如何使用單一編碼器來達成語音表徵的解纏，並且達到一次性樣本的任意語者之語音轉換。基於前人研究 AdaIN-VC[5]，我們同樣假設一段由同一語者所講的內容，它的語音訊號可以拆分為全局資訊和局部資訊，如圖 3.1。其中，全局資訊表示語者資訊，局部資訊則是內容資訊，內容資訊包含了當下特定某一個時間點的音素、抑揚頓挫等其他與語者不相關的資訊。

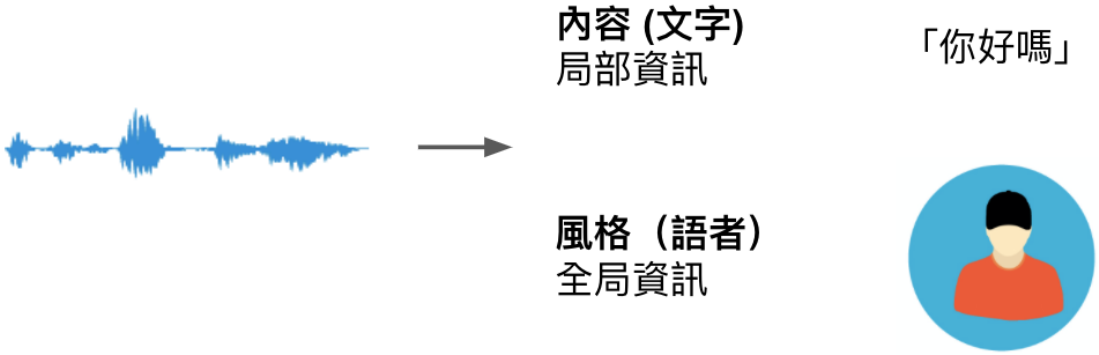


圖 3.1: 將語音轉換看成風格轉換任務，文字內容就是這段音訊的局部資訊，語者特徵就是這段音訊的全局資訊。

在訓練資料只有非平行語料的限制之下，單單利用自監督式學習的框架與自重建目標函數，我們設計的編碼器就可以將語者資訊與內容資訊區分開來，並且語者可以不存在訓練資料內，達到任意語者資訊的抽取。而在模型推論階段，只要分別使用一句來源語句和目標語句，從來源所抽出來的內容表徵，搭配從目標

doi:10.6342/NTU202101337

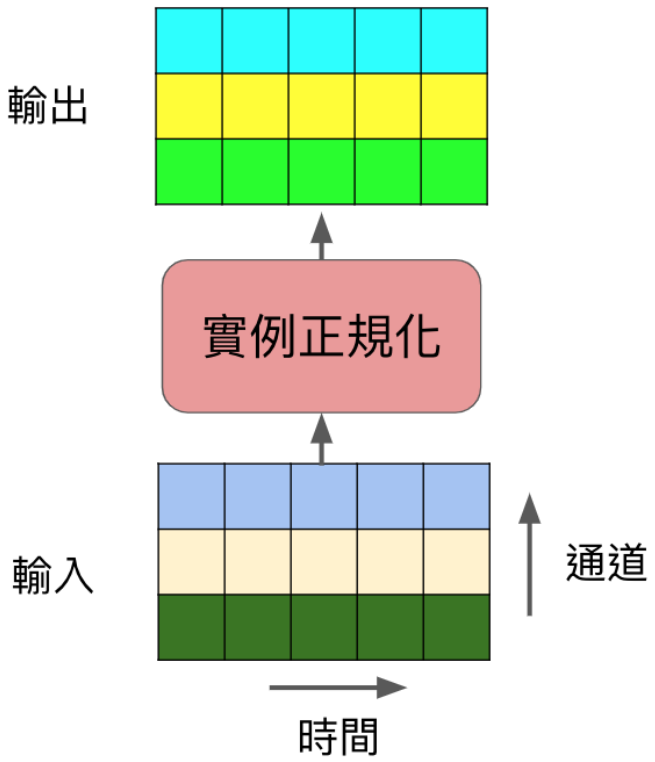


圖 3.2: 實例正規化模組。圖中展示一個三通道、序列長度為 5 的表徵輸入。

語句所抽出來的語者表徵，我們就能夠達到語音轉換的任務。

3.2 以實例正規化達成一次性樣本語音轉換

3.2.1 實例正規化與內容編碼器

實例正規化（Instance Normalization, IN）是一種類神經網路的模組，作用是將輸入表徵做逐通道（Channel-wise）正規化，示意圖如圖 3.2。令輸入序列為 $(x_{c,t})_{C\times T}$，其中 C 為通道數，T 為序列長度（時間）。實例正規化層會先將 x 逐通道算出平

doi:10.6342/NTU202101337

均數 μ_c 與標準差 σ_c，即

$$\mu_c \coloneqq \frac{1}{T}\sum_t x_{c,t},$$
$$\sigma_c^2 \coloneqq \frac{1}{T}\sum_t (x_{c,t}-\mu_c)^2,$$

取得了逐通道的平均數與標準差後，實例正規化的輸出結果 $(\hat{x}_{c,t})_{C\times T}$ 為

$$\hat{x}_{c,t} \coloneqq \frac{x_{c,t}-\mu_c}{\sigma_c}.$$

在進行訓練的時候，資料往往是一批一批（Batch）訓練，而同一批資料的語者可能都是不同的，因此在做實例正規化時需要注意，計算的平均數與標準差是逐樣本、逐通道的，也就是對於批大小（Batch Size）為 N 的一批輸入 $(x_{n,c,t})_{N\times C\times T}$ 來說，每一個樣本 x_n 會有自己的逐通道平均值 $\mu_{n,c}$ 和標準差 $\sigma_{n,c}$，分別為

$$\mu_{n,c} \coloneqq \frac{1}{T}\sum_t x_{n,c,t},$$
$$\sigma_{n,c}^2 \coloneqq \frac{1}{T}\sum_t (x_{n,c,t}-\mu_{n,c})^2,$$

此時輸出結果為

$$\hat{x}_{n,c,t} \coloneqq \frac{x_{n,c,t}-\mu_{n,c}}{\sigma_{n,c}}.$$

由於我們假設一句長度為 T 的音訊裡，語者的資訊是不會隨著時間改變的，結合實例正規化輸入與輸出的特性，將整段訊號中時間軸上共同的統計特徵（平均值與標準差）給去除掉，作用類似將這段音訊和語者有關的資訊消除。因此，AdaIN-VC[5] 以卷積類神經網路的堆疊為骨幹，利用實例正規化模組的特性，將實例正規化模組穿插在卷積類神經網路中，組成內容編碼器，將語者訊息從隱藏表徵中過濾掉。為了不與通道數 C 誤用，本章節的內容表徵以 h_e 表示，而不是以 h_c 表示。

doi:10.6342/NTU202101337

3.2.2 平均池化層與語者編碼器

有了內容編碼器，AdaIN-VC 另外為語者資訊設計了一個語者編碼器，用來抽取語者表徵 h_s。以往許多語者編碼器會使用查找表（Lookup Table）來實作。如圖 3.3 所示，訓練時，標註好每一個訓練語者的編號之後，為每一個語者訓練出他們自己的語者表徵。在推論階段，輸入對應的語者編號，模型就能根據查找表找出對應編號的語者表徵。雖然這類語者表徵抽取的方法能夠為每一個訓練資料中的語者設定他們專屬的語者表徵，讓模型在推論階段時表現較佳，但這也同時限制了模型無法泛化到非訓練語料中的語者；如果要加入新的語者，那麼可能需要數筆該語者的語料來微調（Finetune）模型，而沒辦法達到單一樣本語音轉換。AdaIN-VC 將語者編碼器也安排在自編碼器的整體架構裡，輸入為音訊的聲學特徵，而非語者編號，目的是希望能從聲學特徵自動抽取出語者資訊。為了使這個語者編碼器抽取出來的隱藏表徵不帶有音訊的內容資訊，在設計上，語者編碼器的輸出層是一個平均池化層（Mean Pooling Layer），將輸出表徵在時間維度上，取池化核大小為時間序列長度 T，效果等價於取整段平均作為最終輸出，這使得不論輸入的表徵序列長度 T 為多少，輸出的表徵都是長度為 1 的向量，圖 3.4 為以 2 為池化核大小的平均池化層的示意圖。我們知道，一段音訊的內容資訊會隨著時間變化，因此，輸出表徵的長度為 1 時，這樣的資訊瓶頸，不可能將隨時間變化的資訊，也就是內容資訊保留下來。而這個語者表徵 h_s，會再經過一些類神經網路，被拆分成通道數相同的兩個向量，分別為 $\hat{\mu}$ 和 $\hat{\sigma}$，接著才會輸入解碼器。

3.2.3 自適應實例正規化與解碼器

在解碼階段，解碼器將內容表徵 h_e 和語者表徵 h_s 利用自適應實例正規化模組（Adaptive Instance Normalization, AdaIN）來結合，最終輸出一段轉換後的時頻譜，

doi:10.6342/NTU202101337

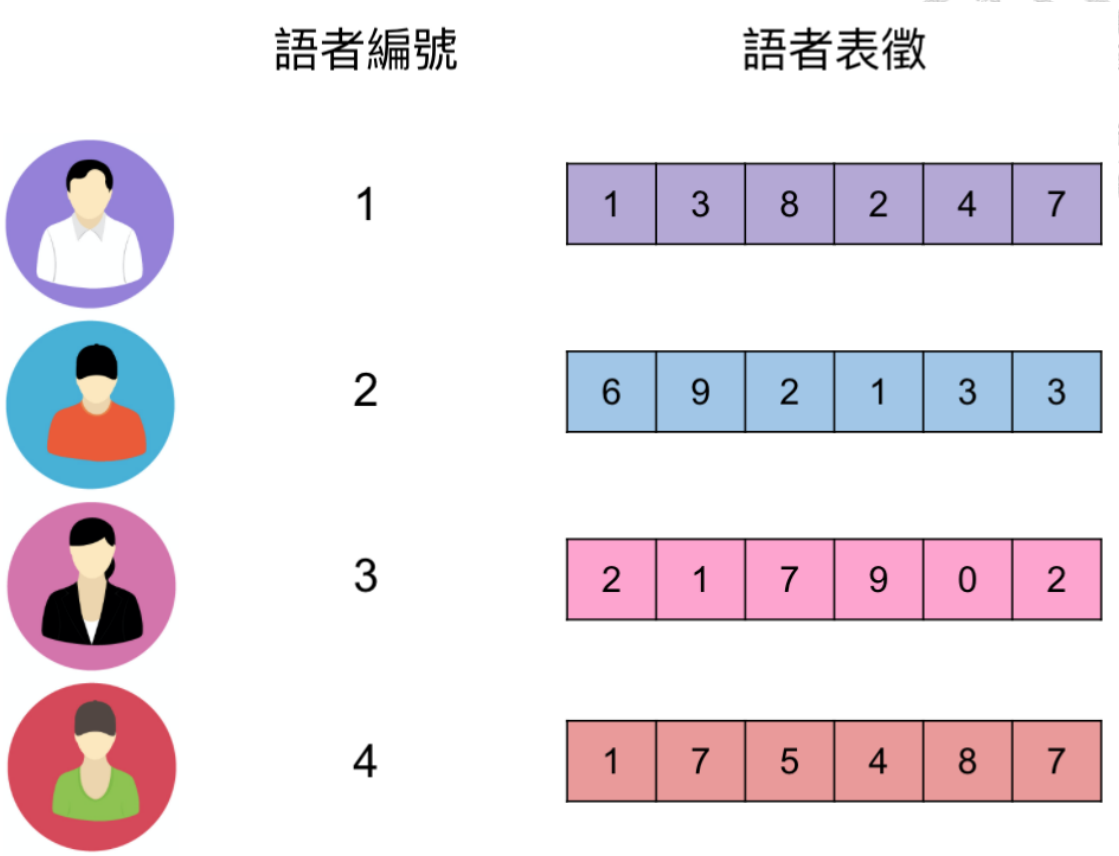


圖 3.3: 查找表。語者表徵在訓練階段是可訓練的參數。模型在推論階段會直接使用語者編號從查找表中取出訓練好的對應語者表徵。

再透過 MelGAN 聲碼器將時頻譜轉回聲音訊號。

自適應實例正規化模組可以看作實例正規化模組的逆運算，對於一批資料，它的輸入有三個部分，分別為表徵序列 $(\hat{x}_{n,c,t})_{N\times C\times T}$，平均值 $(\mu_{n,c})_{N\times C}$ 以及標準差 $(\sigma_{n,c})_{N\times C}$，輸出 $(y_{n,c,t})_{N\times C\times T}$ 定義為

$$y_{n,c,t} \coloneqq \sigma_{n,c}\hat{x}_{n,c,t} + \mu_{n,c},$$

作用類似將每一個過實例正規化之後的表徵 x_n 去正規化。顯而易見地，如果我們使用的平均值和標準差若來自 x_n 本身，則去正規化後的輸出 y_n 會符合

$$y_n = x_n.$$

doi:10.6342/NTU202101337

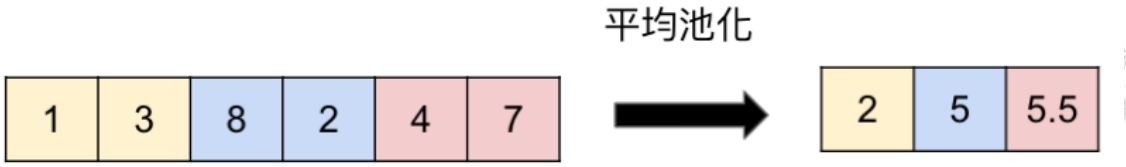


圖 3.4: 平均池化層

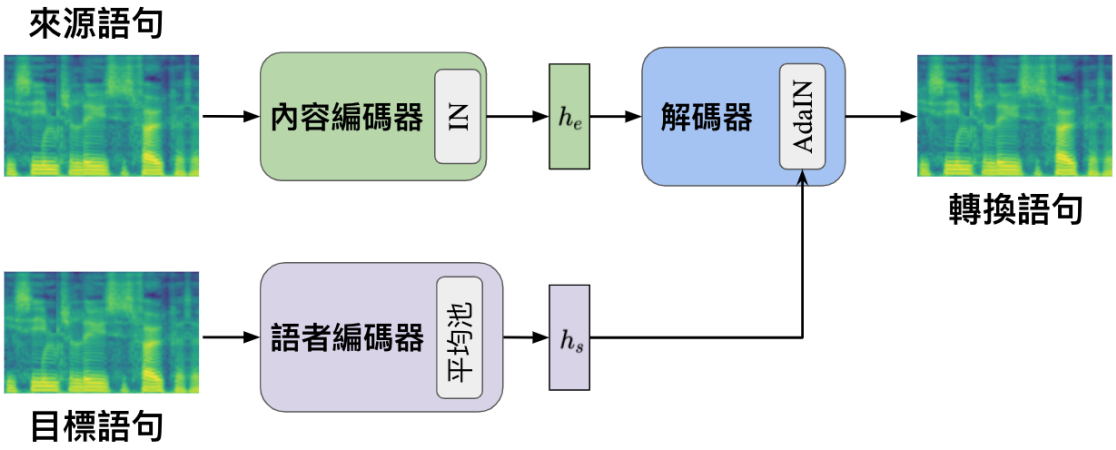


圖 3.5: AdaIN-VC

直觀上的理解，如果使用的平均值和標準差來自另一組樣本 x_m，則去正規化後的輸出 y_n，會具有和 x_m 相同的全局特性，但同時保留 x_n 的局部特性。

AdaIN-VC 的解碼器也是許多卷積類神經網路堆疊而成，並且在網路之間，穿插使用自適應實例正規化模組，將語者表徵 h_s，也就是3.2.2 節 所提及的 $\hat{\mu}$ 和 $\hat{\sigma}$，融入回內容表徵 h_e，讓原本只包含內容資訊的隱藏表徵，逐漸帶有正確的語者資訊。

doi:10.6342/NTU202101337

3.2.4 訓練與推論階段

圖 3.5為 AdaIN-VC 的流程圖。訓練時，由於沒有平行語料，因此來源語句與目標語句是相同的。令來源語句的聲學特徵為 x，模型輸出的轉換語句聲學特徵為 y，則訓練目標是要降低轉換語句和來源與句的自重建誤差，損失函數定義為：

$$L := ||x - y||_1.$$

透過上述編碼器與解碼器特別針對聲音特徵的設計，雖然我們並沒有平行語料，但我們期待這個架構可以讓內容編碼器與語者編碼器確實學會分別抽取帶有內容資訊和語者資訊的表徵，並且解碼器可以結合這兩個表徵合成出轉換後的時頻譜。

不難想像，不同於使用查找表的語者編碼器只能使用訓練過的語者表徵，這樣透過自重建目標訓練出來的語者編碼器能從輸入的聲學特徵推論出該段音訊該有的語者表徵，具有較強的泛化能力。因此，在推論階段時，這類語者編碼器可以接受任意語者的語音輸入，而且，和訓練時相同，在推論階段同樣只需要一句語音樣本，就能抽取出它的隱藏表徵，從而達成一次性樣本的語音轉換。

3.3 提出方法

3.2 節 我們介紹了前人的研究 AdaIN-VC，利用雙編碼器與解碼器的架構，達到一次性樣本語音轉換。接下來，本論文將詳細說明如何使用單一編碼器，就能同時達到抽取語者表徵與內容表徵的效果。在不增加額外參數的情況下，整個模型將少掉一半的編碼器參數量，降低模型運算量與記憶體使用量。

doi:10.6342/NTU202101337

3.3.1 單編碼器與自適應實例正規化

AdaIN-VC 使用實例正規化模組，讓內容編碼器可以過濾掉語者資訊，留下內容資訊；同時，在解碼器中，使用自適應實例正規化模組將語者資訊融入回內容資訊。接下來的部分，為了簡化符號，我們令 μ_x 和 σ_x 表示某一樣本 x 經過實例正規化模組後計算出來的逐通道平均值與逐通道標準差，$\hat{x}$ 表該樣本 x 經過實例正規化模組的輸出。根據 3.2 節 的討論，我們得出 $\hat{x}$ 可以視為 x 的局部資訊，同時，因為實例正規化模組的特性，自然而然地，μ_x 和 σ_x 可以視為 x 的全局資訊。AdaIN-VC 使用了另一個類神經網路嘗試抽取出全局表徵 h_s，再利用解碼器的自適應實例正規化和局部表徵融合在一起。然而，這整個取得全局表徵的流程，對於已經使用實例正規化的編碼器架構來說，其實是冗餘的。

本論文提出，AdaIN-VC 內容編碼器的實例正規化模組從輸入特徵 x 所抽取出來的統計資訊 μ_x 和 σ_x，就能有效代表整段音訊的全局資訊，也就是語者資訊。如果看回 AdaIN-VC，那麼這個 μ_x 和 σ_x 就是對應到3.2.2 節提到的 $\hat{\mu}$ 及 $\hat{\sigma}$。因此，我們不需要再另外設計、訓練一個語者編碼器來抽取語者表徵，只使用單一編碼器與實例正規化模組，保留下計算出來的 μ_x 和 σ_x，很自然地就能代表這段輸入訊號的語者表徵。如圖 3.6 所示，和 AdaIN-VC 一樣，訓練時使用的來源語句和目標語句是相同的；推論階段時，從來源語句抽出內容表徵 h_e，另外從目標語句抽出語者表徵 h_s，有了內容表徵與語者表徵，解碼器所做的事情，就是利用自適應實例正規化模組，將語者表徵融合進內容表徵後生成時頻譜，達到語音轉換。

3.3.2 結合 U 型網路

U 型網路（U-net）[13] 是一種自編碼器架構的延伸變型，如圖 3.7 所示，透過跳躍連接的設計，把越接近輸入端的編碼模組輸出表徵連接到越接近輸出端的解碼

doi:10.6342/NTU202101337

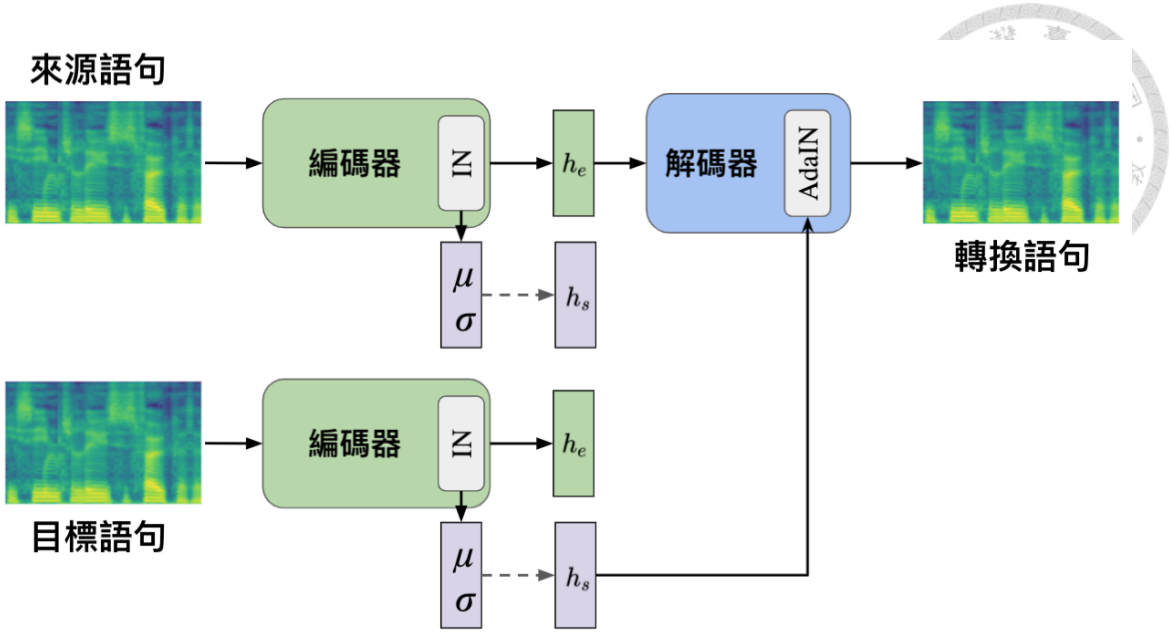


圖 3.6: 本論文提出使用單一編碼器達到語音轉換。

模組。由於越靠近輸入端的表徵，通常損失的資訊較少，這樣的架構設計讓包含較完整資訊的表徵連接到越靠近輸出的位置，以致於模型可以更容易學會自我重建，讓模型的學習難度下降。

要將 U 型網路使用在本論文提出的模型中，需要特別注意的是，我們要避免這些跳躍連接架構將我們不想要的資訊傳遞到解碼器端。前面提到，跳躍連接的優勢在於，越靠近輸入端的表徵會被傳遞到越接近輸出端的對應解碼位置，讓自我重建任務更容易達成。然而，由於非平行語料語音轉換任務的限制，訓練階段模型只能以自我重建為目標，如果我們讓跳躍連接能傳遞內容表徵，同時對內容表徵不加限制，這會讓語者資訊也容易被滲透到這個表徵中，導致推論階段的轉換失敗。

VQVC+[11] 以其前作 VQVC [10] 為基礎，結合 U 型網路在自編碼器的架構，試圖透過跳躍連接層來加強模型的重建能力，有效改善了 VQVC 重建能力較差，導致語音生成品質不佳的問題。其模型如圖 3.8 所示，它在內容表徵的跳躍連接之前，使用了向量量化（Vector Quantization, VQ）的技術，強制內容表徵轉換到

doi:10.6342/NTU202101337

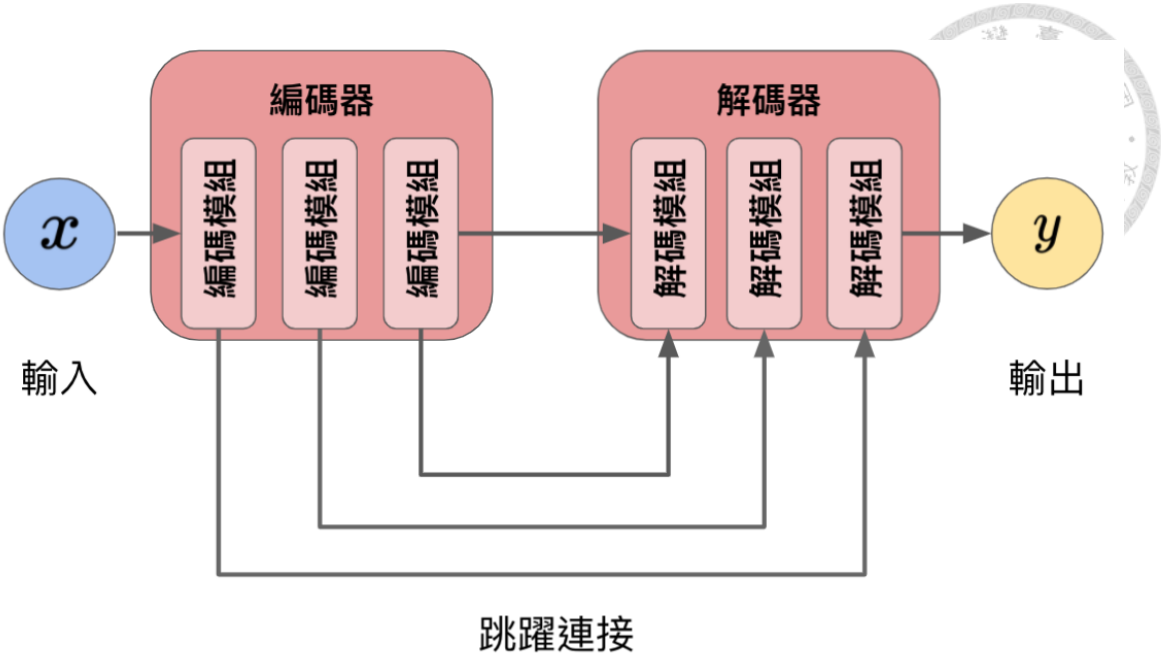


圖 3.7: U 型網路

模型	重建誤差
VQVC	0.262
VQVC+	0.188

表 3.1: VQVC 與 VQVC+ 的重建誤差。VQVC+ 使用跳躍連接的 U 型網路，大幅降低重建誤差。(本數據取自 [11])

離散的空間中，讓內容表徵產生強大的資訊瓶頸，留下局部真正重要的資訊，其它全局的資訊則是由語者表徵所提供。表 3.1 展示了跳躍連接層對自編碼器訊號重建能力的改善。

觀察 VQVC+ 的架構設計，本論文認為，內容表徵只要利用編碼器與解碼器中間貫穿的隱藏表徵來傳遞就夠了；而語者表徵由於本身只帶有 1 個幀的資訊，自然產生極強的資訊瓶頸，就可以利用跳躍連接層來傳遞，而不會有推論階段資訊滲透的問題。

doi:10.6342/NTU202101337

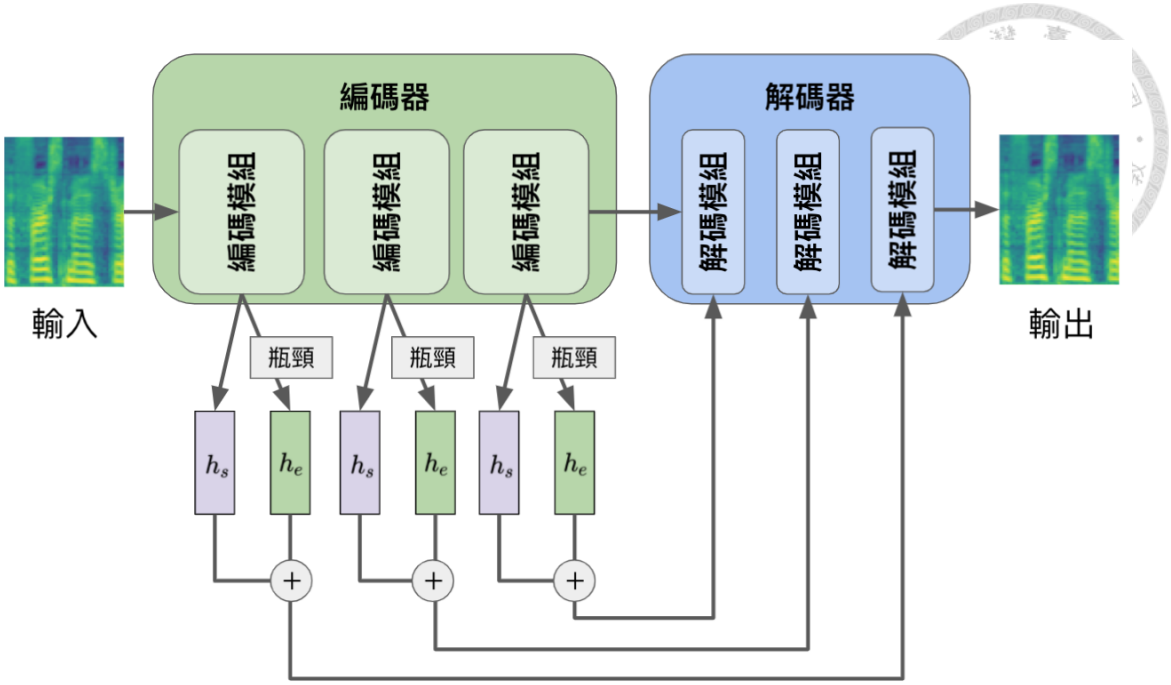


圖 3.8: VQVC+

3.4 網路架構與實施

3.4.1 模型使用元件

批次正規化

對於深層類神經網路，在訓練時如果輸入訊號的分佈變化太大，可能會造成訓練不穩定，結果難以收斂。[28] 提出使用批次正規化（Batch Normalization），將每一層類神經網路的輸入先正規化之後再餵進下一層，有助於改善訓練的穩定性。對於一批批大小 N，通道數 C，序列長度 T 的資料 $(x_{n,c,t})_{N\times C\times T}$ 批次正規化模組首先計算出整批的逐通道平均值和整批的逐通道標準差

$$\mu_c \coloneqq \frac{1}{NT}\sum_n\sum_t x_{n,c,t},$$

$$\sigma_c^2 \coloneqq \frac{1}{NT}\sum_n\sum_t (x_{n,c,t}-\mu_c)^2,$$

doi:10.6342/NTU202101337

利用 μ_c 和 σ_c 將輸入正規化，此時輸出結果為

$$\hat{x}_{n,c,t} \coloneqq \frac{x_{n,c,t} - \mu_c}{\sigma_c}.$$

不難發現，當批大小為 1 的狀況，批次正規化的效果和實例正規化做的是相同的運算。然而，批次正規化希望拿到的 μ_c 和 σ_c，是越具有整個資料集的代表性越好，因此在訓練時若使用到批次正規化模組，那麼批大小就不能設定太小。

帶泄露線性整流單元

帶泄露線性整流單元（Leaky Rectified Linear Unit, Leaky ReLU）[29] 是線性整流單元的一種變形，

$$\text{LeakyReLU}(x) \coloneqq \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases}.$$

其中 α 預設為定值 0.1，也可以隨模型訓練。理論上，帶泄露線性整流單元具有線性整流單元的優點，同時不會落入線性整流單元壞死現象（Dead ReLU Problem）。線性整流單元壞死現象是指，在輸入小於 0 的區域，會讓神經元不被激活，導致在訓練的反向傳遞階段，對應到的類神經網路參數梯度也不會被激活，造成無法更新的窘境。

編碼模組

編碼器由許多編碼模組堆疊而成，每個編碼模組架構如圖 3.9 所示。編碼模組中使用兩層一維卷積層堆疊，中間通過批次正規化以及帶泄露線性整流單元作為激活函數，在模組最後使用實例正規化模組，模組輸出會繼續向前傳播，而模組計算出來的逐通道平均值和逐通道標準差也會保留下來，透過跳躍連接傳遞到對應的解碼模組。

doi:10.6342/NTU202101337

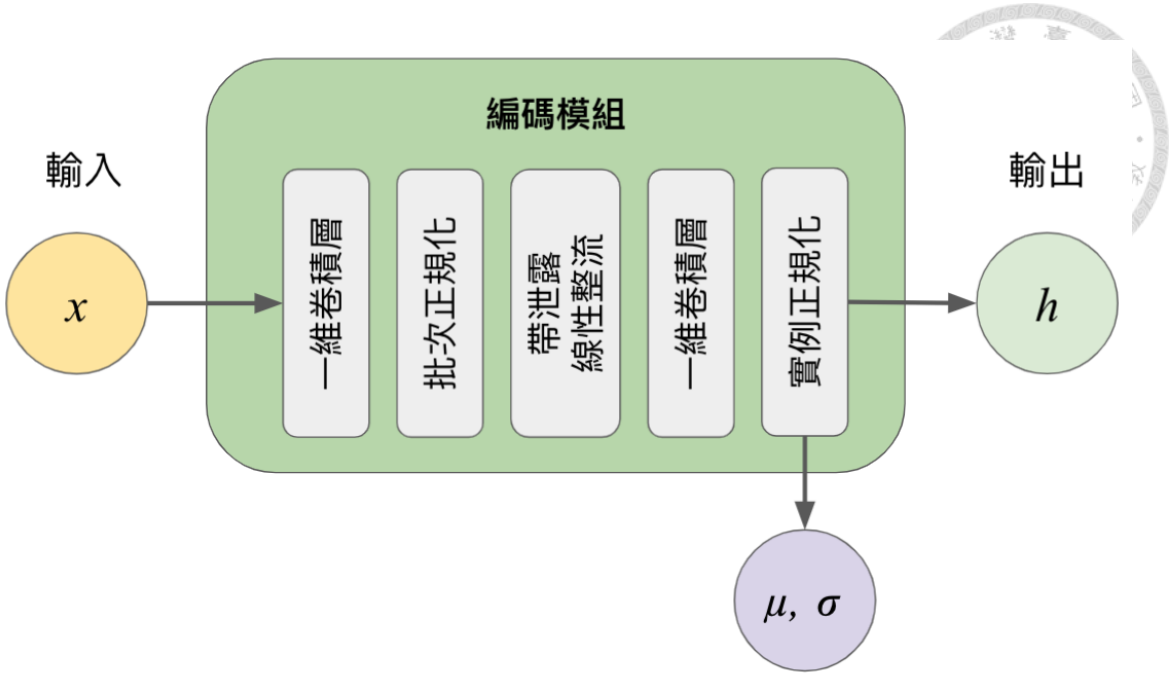


圖 3.9: 編碼模組

解碼模組

解碼器由許多解碼模組堆疊而成，每個解碼模組架構如圖 3.10 所示。解碼模組和編碼模組十分相像，使用兩層一維卷積層堆疊，兩卷積層之間使用批次正規化和帶泄露線性整流單元串接。不同於編碼模組，解碼模組的最後一層使用自適應實例正規化模組，接收從相應的編碼模組跳躍連接過來的逐通道平均值和逐通道標準差，將輸入去正規化。

3.4.2 完整模型架構

圖 3.11 展示了本論文提出的模型，使用自編碼器為主體，結合單一解碼器以及 U 型網路架構達到語音轉換。輸入與輸出的聲學特徵皆為梅爾時頻譜（Mel-spectrogram）。

doi:10.6342/NTU202101337

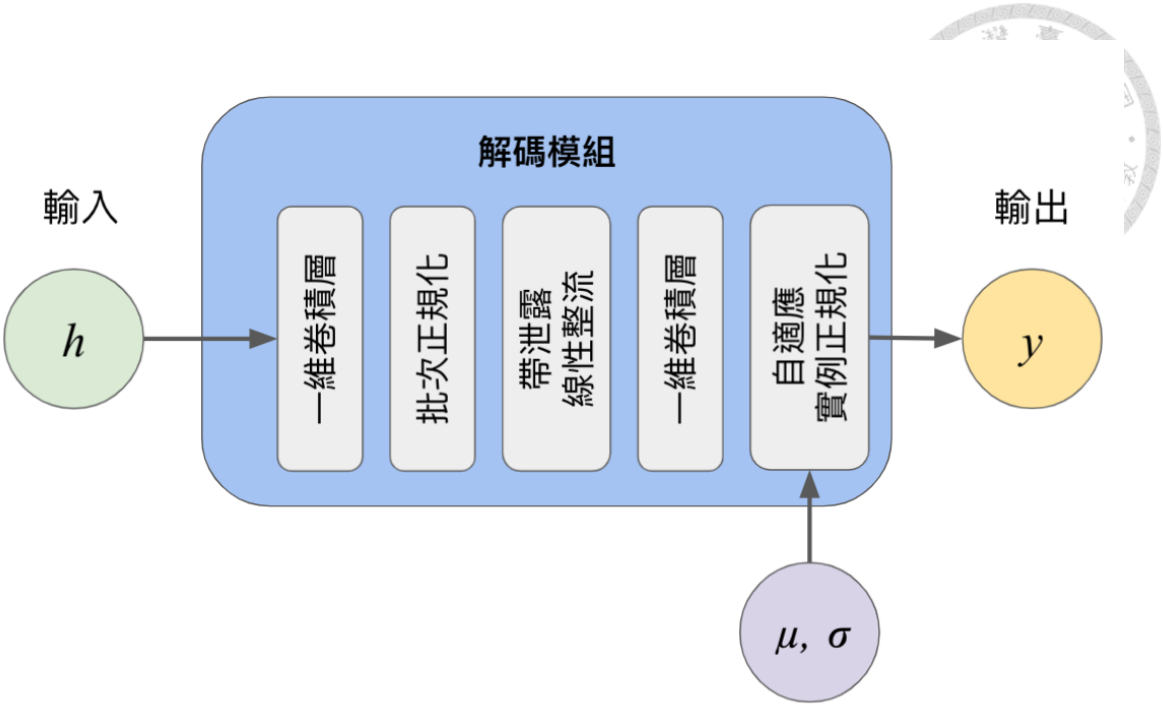


圖 3.10: 解碼模組

3.4.3 訓練細節

聲學特徵

本模型使用之聲學特徵為梅爾時頻譜，詳細特徵抽取參數如表 3.2。針對每一句輸入音訊，我們會將開頭和結尾的無聲片段截斷，訓練批大小為 32，每個訓練片段會隨機取 128 幀，大約為 1.48 秒。由於模型輸入與輸出皆為梅爾時頻譜，本論文另外使用 MelGAN [26] 聲碼器將時頻譜重建回聲音波形。

訓練目標

由於訓練語料非平行，我們採用自監督學習方式，訓練目標為自我重建（Self Reconstruction）。具體而言，就是降低自編碼器輸入訊號 x 與輸出訊號 y 的 L1 距

doi:10.6342/NTU202101337

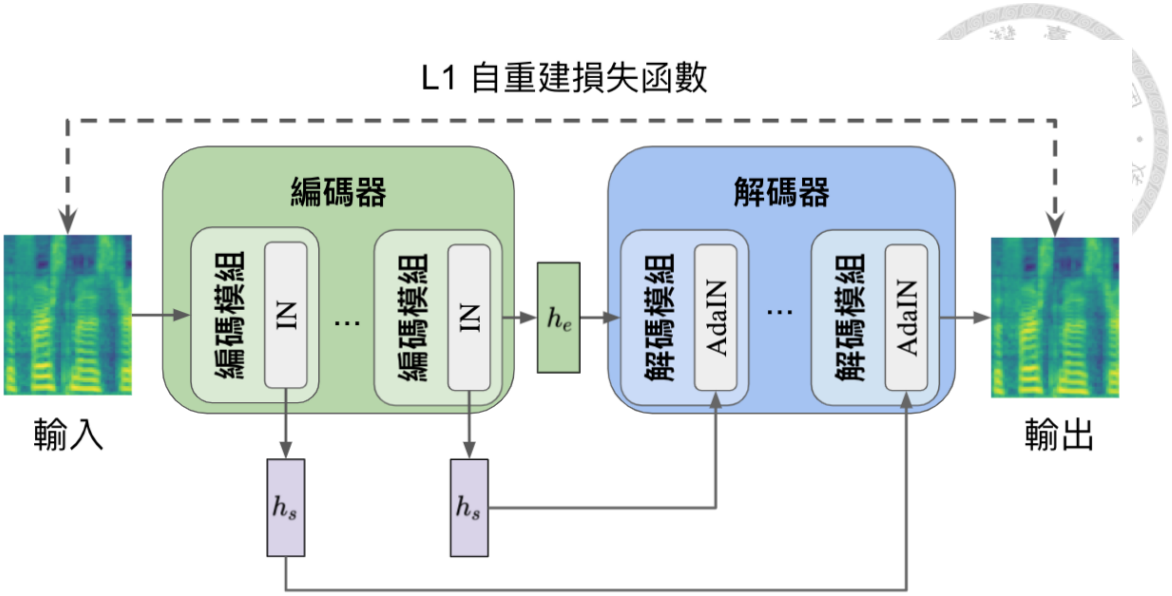


圖 3.11: 本論文提出之模型

離

$$L := ||x - y||_1.$$

超參數

本論文使用之最佳化器（Optimizer）為 Adam 最佳化器 [30]，其參數設置如表 3.3。Adam 最佳化器受到動力學中，「動量」的啟發，讓模型在更新參數時，給學習率一個類似「動量」的項，在梯度下降時，有機會從局部最佳解（Local Minimum）或鞍點（Saddle Point）等不理想卻平緩的死胡同中跳脫出來。另外，為了訓練穩定，我們還使用了梯度剪裁（Gradient Clipping）[31] 的技術。

資料集

使用訓練資料集為 VCTK Corpus [32]，該資料集統計數據如表 3.4。我們隨機選擇其中 80 人的每人各 200 句話作為訓練資料集，剩餘的 29 人以及沒有使用到的語句則當作測試資料集。

doi:10.6342/NTU202101337

頻格數量（FFT Size）	1024
音框跳距（Hop Length）	256
音框點數（Window Length）	1024
窗口類型（Window Type）	漢氏窗（Hann Window）
採樣率（Sampling Rate）	22050 赫茲
梅爾通道數（n_{mel}）	80
梅爾最低頻（f_{min}）	0 赫茲
梅爾最高頻（f_{max}）	11025 赫茲

表 3.2: 聲學特徵參數

3.5 實驗

3.5.1 實驗設定

本節實驗主要想比較雙編碼器與單編碼器模型的語音生成表現，因此以雙編碼器為基準模型，與本論文提出之單編碼器模型比較。

- 雙編碼器

  參考 AdaIN-VC [5] 所提出的語者編碼器架構當作語者編碼器實作雙編碼器架構，並且訓練一個在客觀評估上表現和 AdaIN-VC 差不多的模型作為基準模型。除了語者編碼器是額外加入之外，本模型之架構都和單編碼器所使用的架構相同。

- 單編碼器

  即本章節所提出之架構，使用單一編碼器與解碼器達到語音轉換。

doi:10.6342/NTU202101337

最佳化器	Adam
學習率（Learning Rate）	0.0005
β_1	0.9
β_2	0.999
梯度剪裁值 (Gradient Clipping Value)	5

表 3.3: 最佳化器參數

人數	109
語句數	約每人 400 句
時長	約每句 3 秒
總時長	約 40 小時

表 3.4: VCTK Corpus 統計數據

3.5.2 視覺化實驗結果

時頻譜

如圖 3.12 所示，觀察模型產生的時頻譜，可以發現不論使用單編碼器抑或雙編碼器架構，都能將語者資訊從來源語音分離，留下內容資訊，達到語音轉換效果。

隱藏表徵視覺化

隱藏表徵散佈在高維空間中，本論文使用 t-隨機鄰近嵌入法（t-distributed stochastic neighbor embedding, t-SNE）[33] 將隱藏表徵降維至二維空間以視覺化。結果如圖 3.13 和 圖 3.14 ，分別展示了語者表徵和內容表徵降維後的分佈。如圖 3.13所示，使用額外語者編碼器來編碼的語者表徵在表徵空間中，有被分得較開的趨勢，

doi:10.6342/NTU202101337

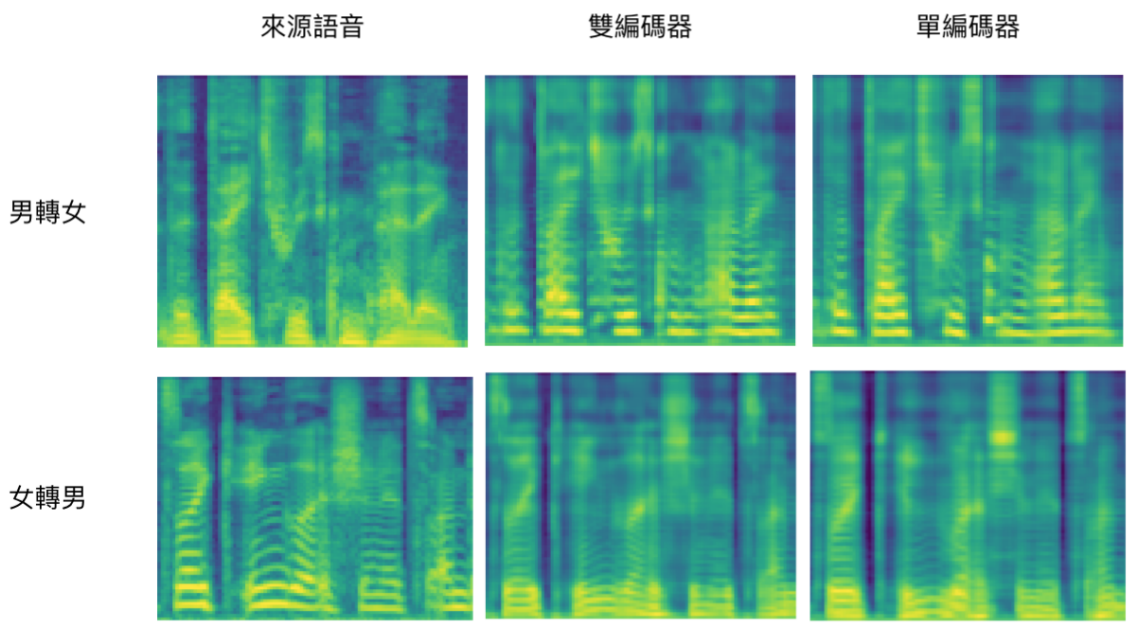


圖 3.12: 時頻譜比較，上排為女轉男，下排為男轉女。最左邊為原始音訊，中間為雙編碼器模型轉換後結果，右邊為單編碼器模型轉換後結果。

但只使用單編碼器，藉著自適應實例正規化的特性，也同樣讓語者表徵具有語者資訊。如圖 3.14所示，我們可以明顯看出來，使用雙編碼器所編碼出的內容表徵，來自同樣語者的表徵仍然有群聚在一起的現象，而單編碼器則沒有這個現象；這表示雙編碼器模型所編碼出的內容表徵仍然帶有非常多語者資訊。

3.5.3 客觀評估

接著一一介紹本論文使用之客觀評估項目，並將結果統整於表 3.6。

運算資源

我們衡量了模型大小以及單一筆語音轉換的模型運算速度。

doi:10.6342/NTU202101337

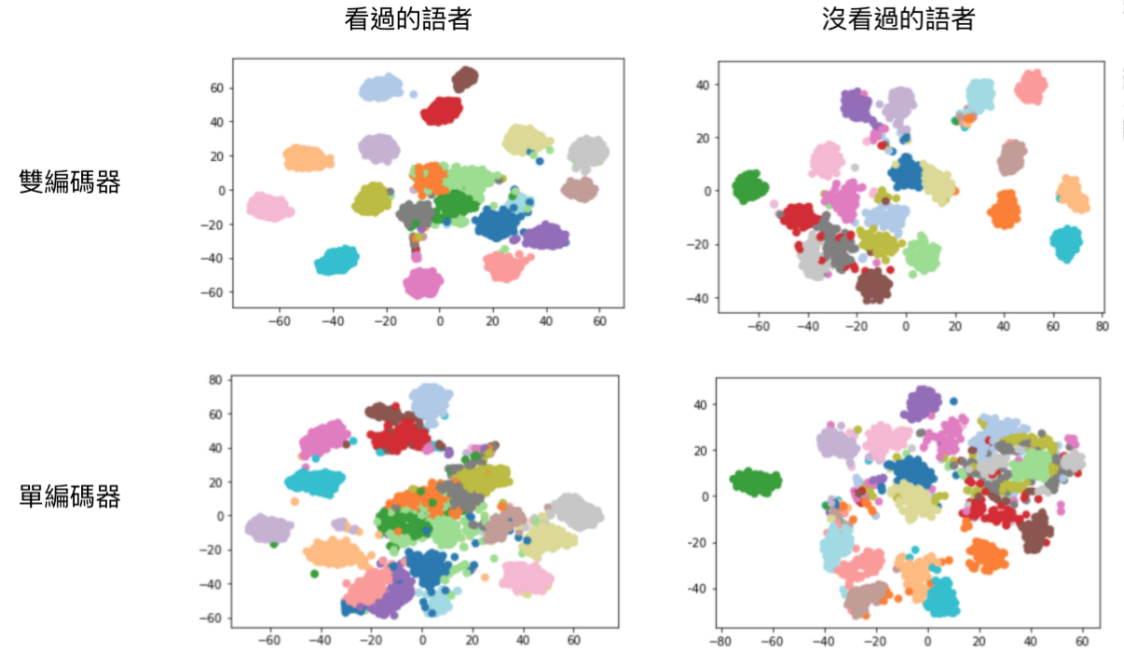


圖 3.13: 語者表徵分佈。不同顏色的點代表不同語者。

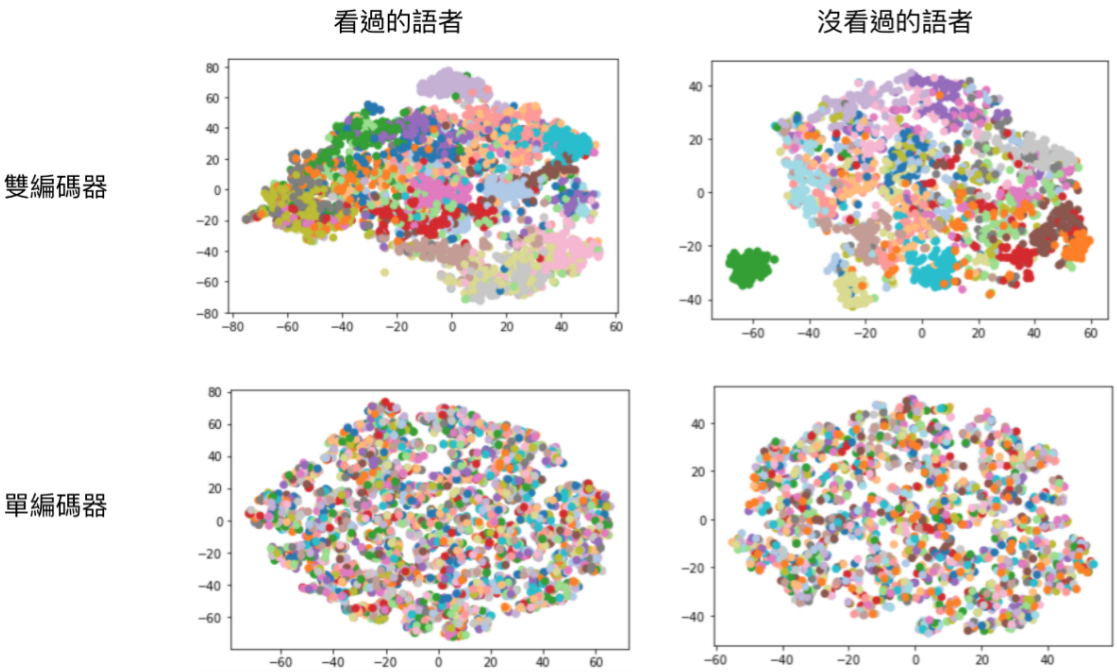
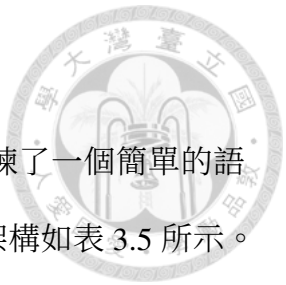


圖 3.14: 內容表徵分佈。不同顏色的點代表不同語者。

doi:10.6342/NTU202101337



語者分類正確率

為了要衡量模型的解纏能力，我們為模型抽取出的內容表徵訓練了一個簡單的語者分類器，看看是否能從隱藏表徵中提取出語者資訊。分類器架構如表 3.5 所示。我們希望內容表徵中帶有的語者資訊越少越好，因此內容表徵上的語者分類正確率越低越好。本實驗使用之訓練資料和3.4.3 節模型使用之訓練資料相同，測試資料為該 80 名訓語者的語句中，沒有被使用到的那些語句。

模組	參數
全連接層	(c_{in}, c_h)
一維卷積層 + 線性整流單元	$(c_h, c_h, 3)$
一維卷積層 + 線性整流單元	$(c_h, c_h, 3)$
一維卷積層 + 線性整流單元	$(c_h, c_h, 3)$
全連接層	(c_h, c_{out})

表 3.5: 語者分類器架構，由上至下為類神經網路第一層至最後一層。一維卷基層之參數依序為輸入通道數、輸出通道數、核大小。

模型重建能力

我們使用模型在測試資料集的 L1 重建誤差來衡量模型重建能力。重建誤差越低，表示模型抽取表徵並重建回時頻譜的能力越高，通常與輸出的聲音品質有高度相關。

doi:10.6342/NTU202101337

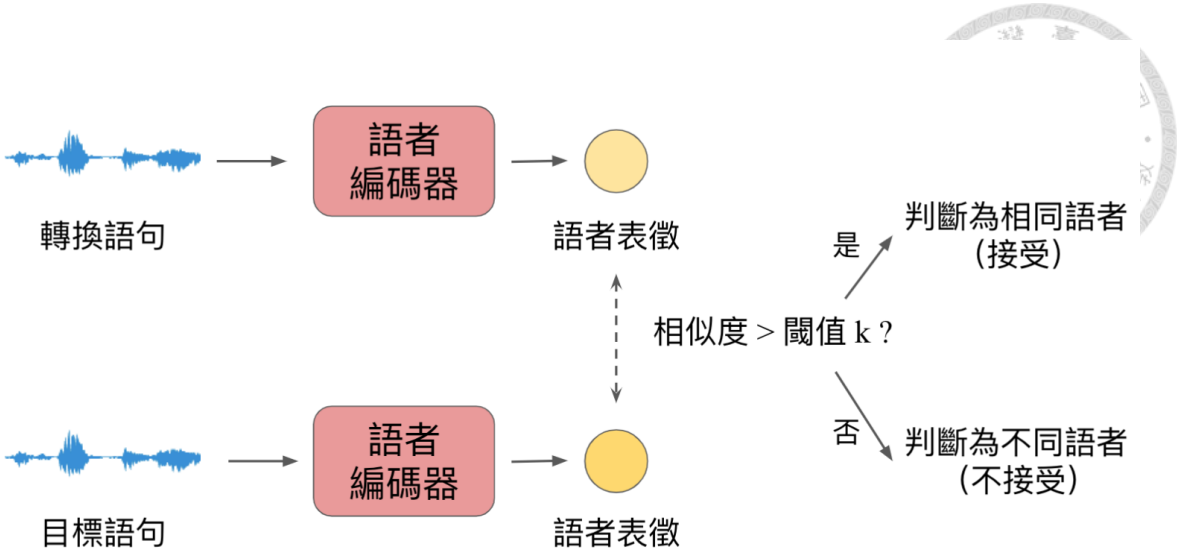


圖 3.15: 語者驗證接受度

語音辨識錯誤率

我們使用語音辨識（Automatic Speach Recognition, ASR）模型來辨識模型合成的聲音，計算字符錯誤率（Character Error Rate, CER）來衡量文字內容資訊是否被保留。[1]

語者驗證接受度

我們使用語者驗證接受度（Speaker Verification Accept Rate, SVAR）來驗證模型合成的聲音與目標語者是否為相同語者。給定一定數量的語句，語者驗證接受度是用來衡量其中有多少句子會被語者驗證系統判定成目標語者的比例。此衡量方式流程圖如圖 3.15，使用了第三方預訓練好的語者編碼器來抽取語者表徵（Speaker Embedding），計算轉換語音的語者表徵和目標與者的語者表徵的相似度，若此相似度超過某一個閾值 k，則判定此語音被語者驗證模型接受。實際上，我們使用的閾值 $k = 0.6597$，此時資料集的語者驗證等錯誤率（Equal Error Rate, EER）為 5.65%。[1]

doi:10.6342/NTU202101337

語音品質

使用 MBNet [34] 來預測模型合成語音的平均意見分數（Mean Opinion Score, MOS）。[1]

評估結果

客觀評估之評估結果統整於表 3.6。可以看到我們使用的雙編碼器架構模型，已經大幅減少運算資源，並且在其他衡量標準都達到和 AdaIN-VC 差不多的表現。另外，使用單一編碼器，運算資源又比雙編碼器來得少，而其他衡量標準也都維持模型表現。

	AdaIN-VC	雙編碼器	單編碼器
模型大小 (M)	4.9	1.8	**1.2**
運算速度（it/s）	0.36	23.72	**34.71**
語者分類正確率（%）	**76.7**	81.4	87.9
重建誤差	0.156	0.142	**0.133**
語音辨識字符錯誤率（%）	39.4	**33.0**	36.5
語者驗證接受度（%）	86.4	87.1	**89.9**
預測平均意見分數	**3.35**	3.26	3.29

表 3.6: 客觀評估

[1]感謝黃子賢同學提供評估程式。https://github.com/tzuhsien/Voice-conversion-evaluation

doi:10.6342/NTU202101337

	雙編碼器	單編碼器
額外語者編碼器	有	
層數	6	6
編碼模組隱藏通道數	128	128
內容表徵通道數	3	3

表 3.7: 模型架構細節

3.5.4 主觀評估

除了客觀評估之外，本實驗結果也使用主觀評估來衡量模型。本評估所使用之模型實施細節如表 3.7。

語者相似度

我們請受試者聽三段語音

- A：目標語者原始語音
- B：單編碼器合成之語音
- C：雙編碼器合成之語音

其中 A 語音固定為目標與者原始語音，B 和 C 段隨機調換順序，受試者並不會事先知道哪段語音是由哪個模型所生成。聽完三段語音之後，我們請受試者填選下列選項

- A 語者和 B 語者較相似
- A 語者和 C 語者較相似

doi:10.6342/NTU202101337

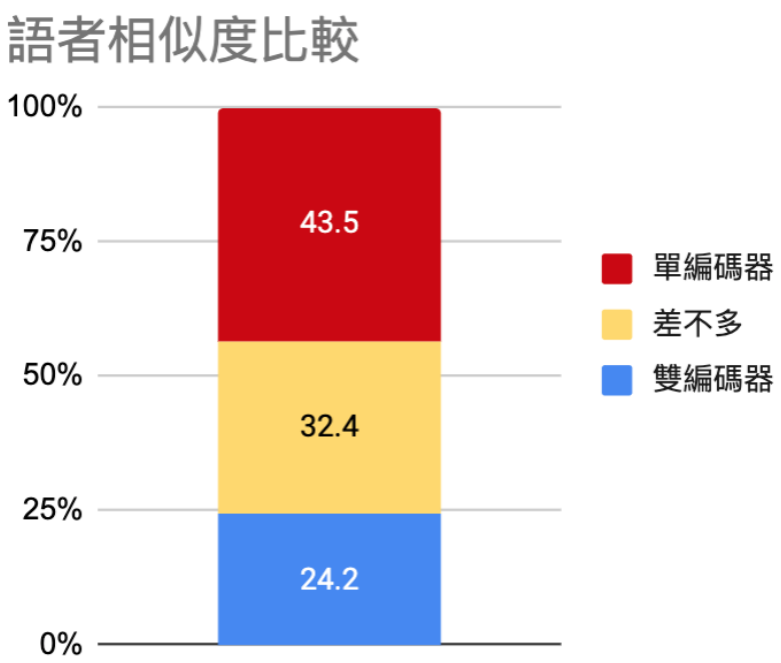


圖 3.16: 語者相似度比較

- 差不多

得到結果如圖 3.16 ，顯示使用單編碼器模型不但降低運算資源，在實際生成之語音也得到較佳的語者相似度。

語音品質

我們使用平均意見分數（Mean Opinion Score, MOS）來衡量模型合成的語音品質。評分標準如下

- 5 分：非常好
- 4 分：好
- 3 分：普通
- 2 分：差

doi:10.6342/NTU202101337

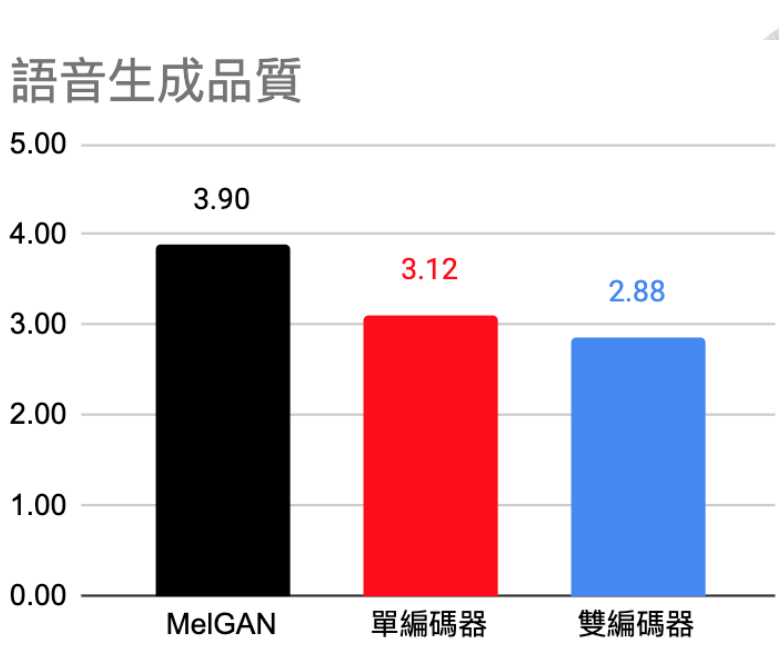


圖 3.17: 語音生成品質主觀分數

- 1 分：非常差

除了單編碼器與雙編碼器模型之外，我們也加入 MelGAN 生成的原始音訊一起比較，作為參考的分數上限。施測結果如圖 3.17 所示，單編碼器的語音生成品質也好過雙編碼器語音生成品質，但距離參考分數上限仍然已有一段進步空間。

3.6 本章總結

本章節介紹基於 AdaIN-VC 的概念，並結合 U 型網路，利用實例正規化本身的特性，達到語者資訊與文字內容資訊的表徵解纏。實驗結果顯示，使用單一編碼器除了大幅降低模型運算資源，也保持和雙編碼器模型 AdaIN-VC 一樣的表徵解纏能力與音訊重建能力；在主觀評估上，單編碼器模型比起雙編碼器模型也具有較佳的語者相似度和語音品質分數。

doi:10.6342/NTU202101337

第 四 章　以激活函數形成資訊瓶頸對表徵解纏的影響

4.1　簡介

對於任意語者的一次性樣本語音轉換任務來說，模型解纏能力（Disentangling Ability）是舉足輕重的。目前使用非平行語料、以自編碼器為模型主體的大多數研究都使用資訊瓶頸（Information Bottleneck）來達到表徵解纏。然而，AutoVC [7] 和 VQVC+ [11] 的作者都明確指出，對於以自重建為目標的語音轉換模型來說，模型的解纏能力和模型重建能力（Reconstruction Ability）會行成一個取捨（Trade-off）。當解纏能力越高，則模型自重建能力就越差，反之，自重建能力越好，解纏效果就越差。從表 3.6，我們其實可以發現，雖然模型轉換後的語音有接近 90% 的語者驗證接受度，但利用內容表徵，我們還是能訓練出超過 70% 正確率的語者分類器。因此，如何改善內容表徵的解纏效果，仍是值得研究的議題。本章將使用第三章提出之架構作為基準模型，嘗試使用不同激活函數作為資訊瓶頸，對內容表徵作進一步的限制，並觀察不同激活函數對模型的解纏能力和模型自重建能力的影響。

4.2　透過資訊瓶頸達成表徵解纏

4.2.1　AutoVC：減少表徵通道

如同本論文在2.2.2 節提到的方法，AutoVC 使用一個以廣義端到端損失（Generalized End-to-end Loss, GE2E）預訓練好的語者編碼器，或稱為 D 向量（D-vector）

doi:10.6342/NTU202101337

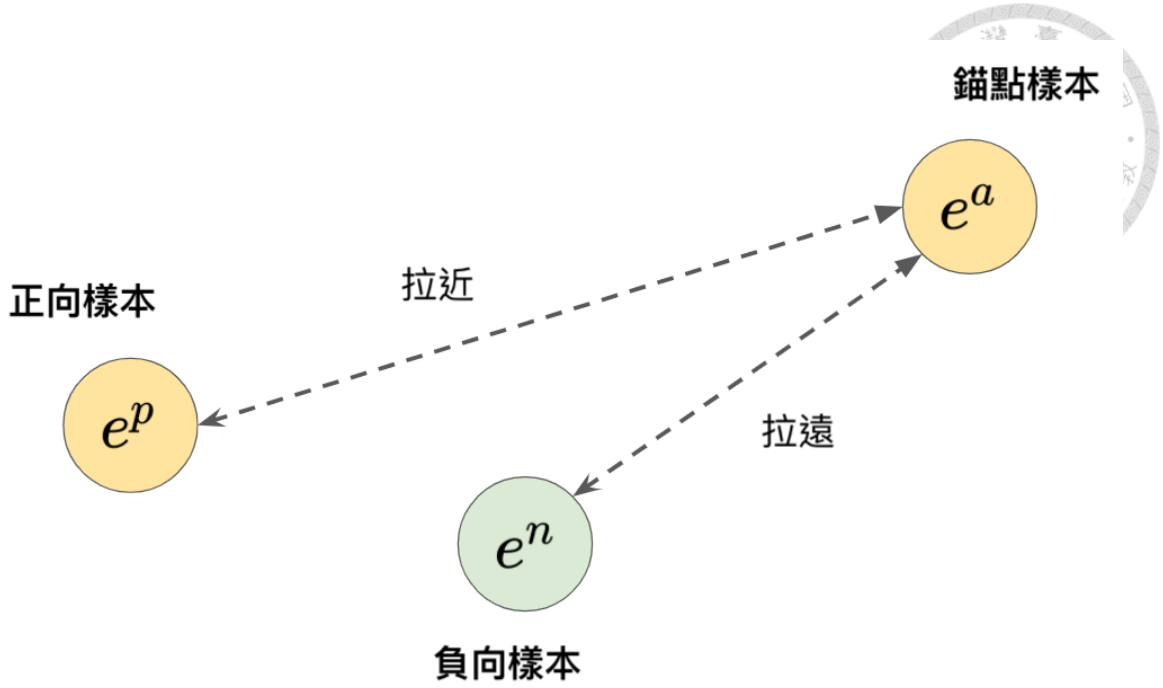


圖 4.1: 三元組損失

[9]，以及減少自編碼器表徵通道，達到任意對任意語者的一次性樣本語音轉換。

三元組損失

廣義端到端損失前身為三元組損失（Triplet Loss）[35]，如圖 4.1 所示，訓練時，會一次取三個樣本：錨點樣本（Anchor Example）、正向樣本（Positive Example）與負向樣本（Negative Example），其中正向樣本 x^p 與錨點樣本 x^a 來自相同類別，負向樣本 x^n 與錨點樣本 x^a 來自不同類別，訓練目標是拉近這正向樣本與錨點樣本在表徵空間中的距離，同時拉遠負向樣本與錨點樣本在表徵空間中的距離。

$$L \coloneqq d(e^a, e^p) - d(e^a, e^n),$$

其中 d 為距離函數，e^a, e^p, e^n 分別表示表徵空間中錨點樣本、正向樣本、與負向樣本點。

doi:10.6342/NTU202101337

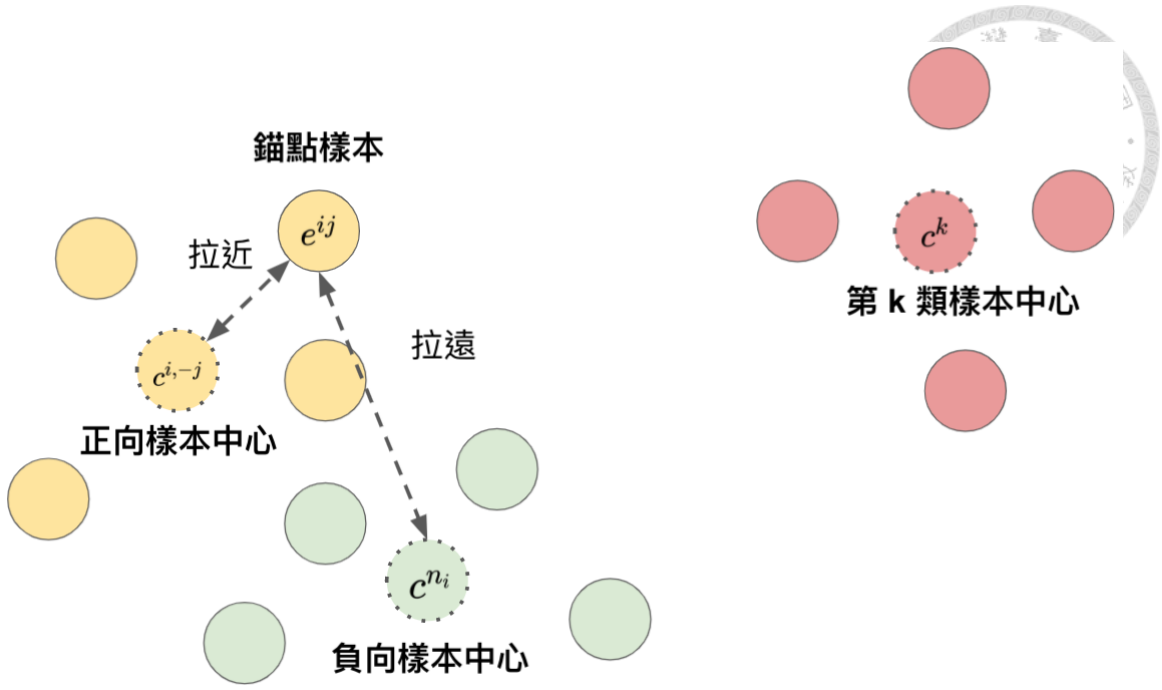


圖 4.2: 廣義端到端損失。錨點樣本會選擇距離最近的非同類樣本中心當作負向樣本中心，圖中綠色樣本中心因為離錨點樣本較近，因此被選為負向樣本中心。

廣義端到端損失

廣義端到端損失示意圖如圖 4.2，不同於三元組損失，廣義端到端損失在訓練的批運算過程中，會至少選擇三種類別當作這次的運算資料，每種類別取出一批樣本。在計算損失函數時，選擇一個點作為錨點樣本，其將與該批中，排除掉自己以外的所有正向樣本點的中心點作拉近；另外，從其他批負向樣本中，選擇一批中心點和正向樣本中心點最近的那批負向樣本當作這次運算對象，作拉遠的運算。舉例來說，在一批運算中，會有 I 個類別，每個類別分別都有 J 個樣本，並定義

- x^{ij}：第 i 類中的第 j 個樣本。
- e^{ij}：x^{ij} 在表徵空間中對應到的點。
- d：距離函數，常用 L1 或 L2 距離，或是負餘弦相似度。

doi:10.6342/NTU202101337

對於兩個點 e^m 和 e^n 來說，他們的餘弦相似度定義如下

$$S(e^m, e^n) \coloneqq \frac{e^m \cdot e^n}{||e^m|| \cdot ||e^n||},$$

若使用餘弦相似度來定義距離函數則可使用

$$d(e^m, e^n) \coloneqq -S(e^m, e^n).$$

這個定義。此時損失函數 L 為

$$L \coloneqq \sum_i \sum_j \left[d(e^{ij}, c^{i,-j}) - d(e^{ij}, c^{n_i}) \right],$$

其中

$$c^i \coloneqq \frac{1}{J} \sum_j e^{ij},$$

$$c^{i,-j} \coloneqq \frac{1}{J-1} \sum_{j' \neq j} e^{ij'},$$

$$n_i \coloneqq \arg\min_{i' \neq i} d(c^i, c^{i'}).$$

AutoVC 使用這樣預訓練的語者編碼器（D 向量），在訓練語音轉換任務時，此語者編碼器參數是固定的，並且假設此語者編碼器能夠抽出我們需要的語者資訊。

減少表徵通道

圖 4.3 展示 AutoVC 的模型，其解碼器由卷積類神經網路以及遞歸式類神經網路所組成。如同2.2.3 節 所述，固定預訓練好的 D 向量語者編碼器，當作理想的語者表徵抽取模型，藉由調整內容表徵的通道大小，使通道大小剛好能讓內容表徵通過，屏蔽掉語者資訊，達成資訊解纏。值得一提的是，AutoVC 架構本身被設計成不需要平行語料，但 D 向量語者編碼器的預訓練，是需要大量有標註語者的資料的。另外，[14] 也指出，使用這種語者驗證（Speaker Verification）任務預訓練的語者編碼器，不一定適合當作語音生成任務的語者編碼器。

doi:10.6342/NTU202101337

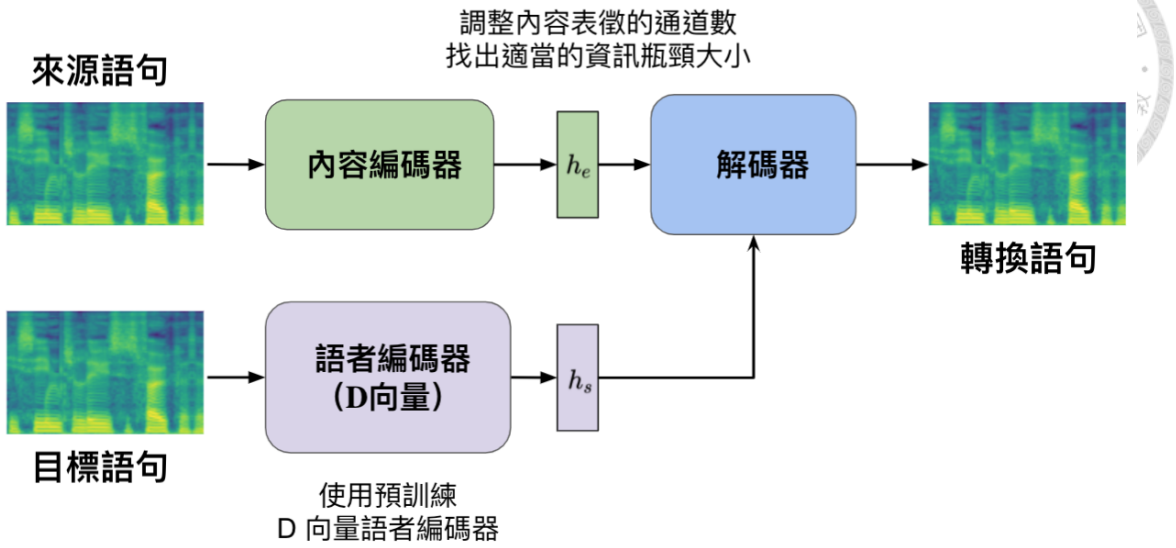


圖 4.3: AutoVC

4.2.2 VQVC+：向量量化

向量量化

向量量化（Vector Quantization, VQ）模組是一種離散編碼的技術，需要我們一開始就設定碼本大小（Codebook Size）：在原本的表徵空間中，取一定數目 N 個點當作編碼點（Code），這個 N 即是碼本大小。圖 4.4 展示向量量化模組，輸入一個表徵後，向量量化模組輸出即為該輸入表徵最靠近的編碼點，由於輸出只能從有限的碼本取出，因此形成離散的表徵。而這個向量量化模組的編碼點是可以被訓練的參數。當我們把碼本大小 N 設定得越大，向量量化模組帶來的資訊瓶頸就越小，反之則資訊瓶頸越大。如表 4.1，VQVC+ 的實驗結果顯示，如果內容表徵的跳躍連接之前只使用實例正規化，或者是設定的碼本大小太大，這樣雖然讓重建任務的重建誤差（Reconstruction Error）變小，但因為資訊瓶頸不夠強，語者資訊容易透過越靠近輸入端的表徵滲透到輸出端，造成模型在推論階段，內容表徵會帶有來源語者的資訊，這種不如預期的資訊滲透現象（Information Leak），而導致

doi:10.6342/NTU202101337

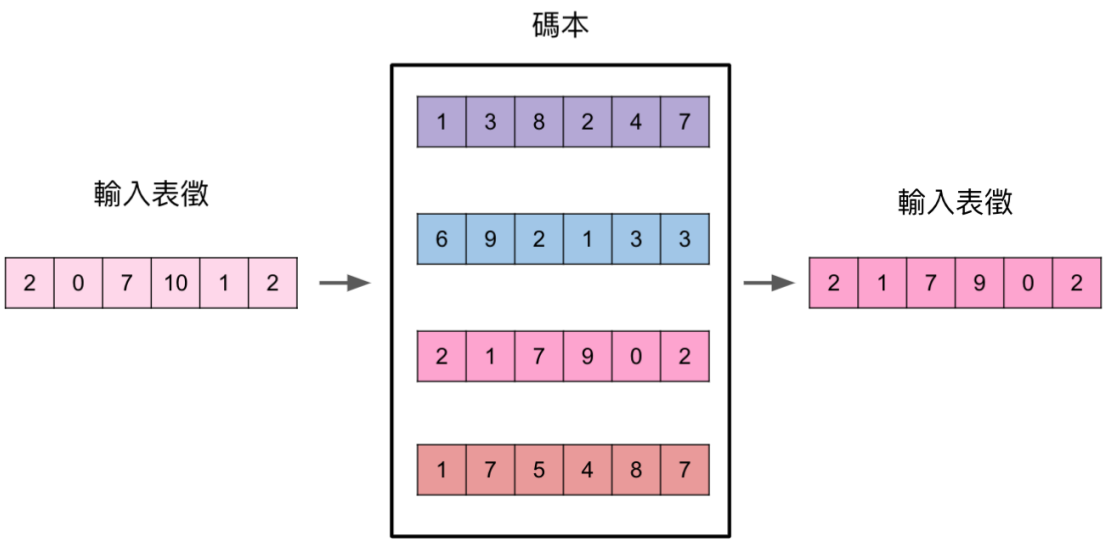


圖 4.4: 向量量化。本示意圖中之碼本大小為 4。

轉換失敗。實際上，使用向量量化雖然在語者相似度上取得不錯的效果，跳躍連接確實也改善模型重建能力，但轉換後的語音內容仍然還有一段進步空間。

4.3 提出方法

4.3.1 激活函數引導

不論是減少表徵通道數或者是使用向量量化，都可以視為在隱藏表徵抽取過程中的資訊瓶頸。而 AutoVC[7] 和 VQVC+[11] 確實也使用這些資訊瓶頸，在非平行語料的語音轉換任務中獲得一些成果。然而，這兩篇論文的作者都指出，模型的重建能力和資訊解纏能力勢必要有取捨。本論文提出了使用激活函數（Activation Functions）作為另一種資訊瓶頸，探討不同激活函數對第三章 所提出模型之資訊重建與解纏能力的影響。在表徵輸出層加上激活函數，對模型運算資源的影響是微乎其微，但到目前為止，並沒有研究探討激活函數作為語音解纏的資訊瓶頸的效果。本論文探討了不同的激活函數作為資訊瓶頸，對模型解纏能力與重建能力

doi:10.6342/NTU202101337

碼本大小	語者分類準確率 (%)	重建誤差
32	19.5 / 11.8 / 6.8	0.210
64	23.2 / 16.6 / 7.0	0.188
128	33.3 / 17.0 / 10.3	0.180
256	35.8 / 18.1 / 12.5	0.165
IN	71.2 / 36.8 / 5.0	0.145

表 4.1: VQVC+ 不同資訊瓶頸對應的資訊解纏能力與重建誤差。第二欄是 VQVC+ 的三個編碼模組輸出的內容表徵的語者辨識準確率，準確率越高代表內容表徵帶有語者資訊越多，也就是資訊瓶頸越弱。第三欄是重建誤差，越小代表模型重建能力越強。IN 則表示沒有使用向量量化，只使用實例正規化作為資訊瓶頸。（本表格取自 [11]。）

的影響。

4.4 網路架構與實施

4.4.1 網路架構

本章節所提出之模型架構如 圖 4.5 ，與第三章提出之模型幾乎相同，唯一不同之處，在於內容編碼器最後一層輸出加上了不同激活函數。所選的激活函數有指數線性單元（Exponential Linear Unit, ELU)、線性整流函數、S 型函數及其變種。

4.4.2 訓練細節

本章節使用的資料集、聲學特徵、超參數等訓練細節都和第三章相同。

doi:10.6342/NTU202101337

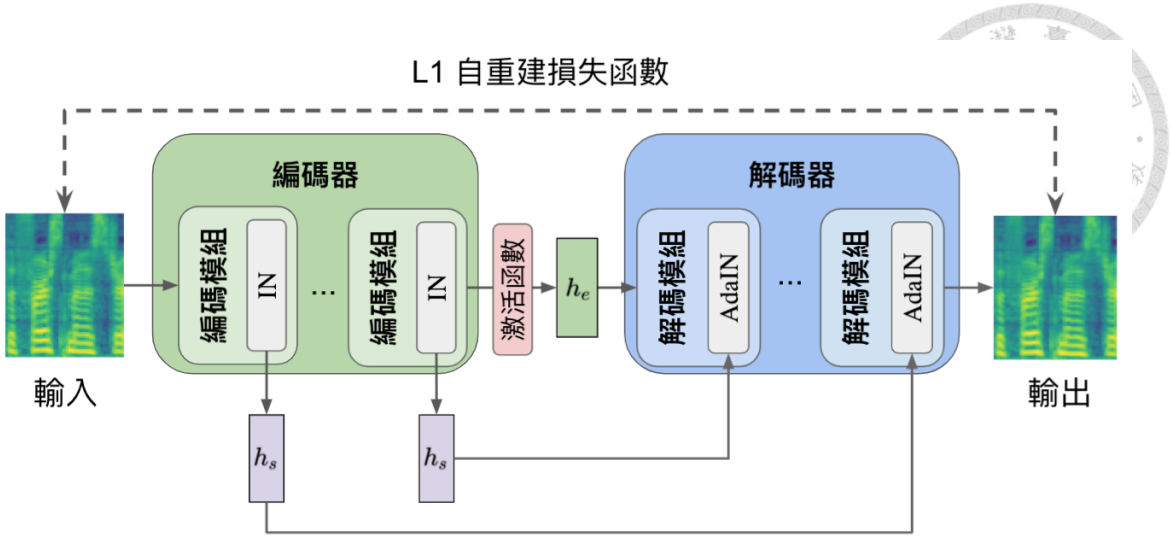


圖 4.5: 本論文提出之模型

4.5 實驗

4.5.1 視覺化實驗結果

時頻譜

圖 4.6 展示使用單編碼器模型產生的時頻譜，以及加上 S 型函數作為資訊瓶頸所生成之時頻譜的結果。結果顯示，不論有沒有加上 S 型函數，模型都有轉換語者特徵的能力。

隱藏表徵視覺化

本節和第三章相同，使用 t-隨機鄰近嵌入法（t-distributed Stochastic Neighbor Embedding, t-SNE）[33] 將表徵降至二維之後視覺化。圖 4.7 和 圖 4.8 為降維後結果。從降維後的表徵分佈來看，可以發現加上 S 型函數的資訊瓶頸後，語者表徵降維結果和沒有加上之前差不多；然而，加上 S 型函數之後，反而使得相同語者的內容表徵產生了些微群聚的現象。

doi:10.6342/NTU202101337

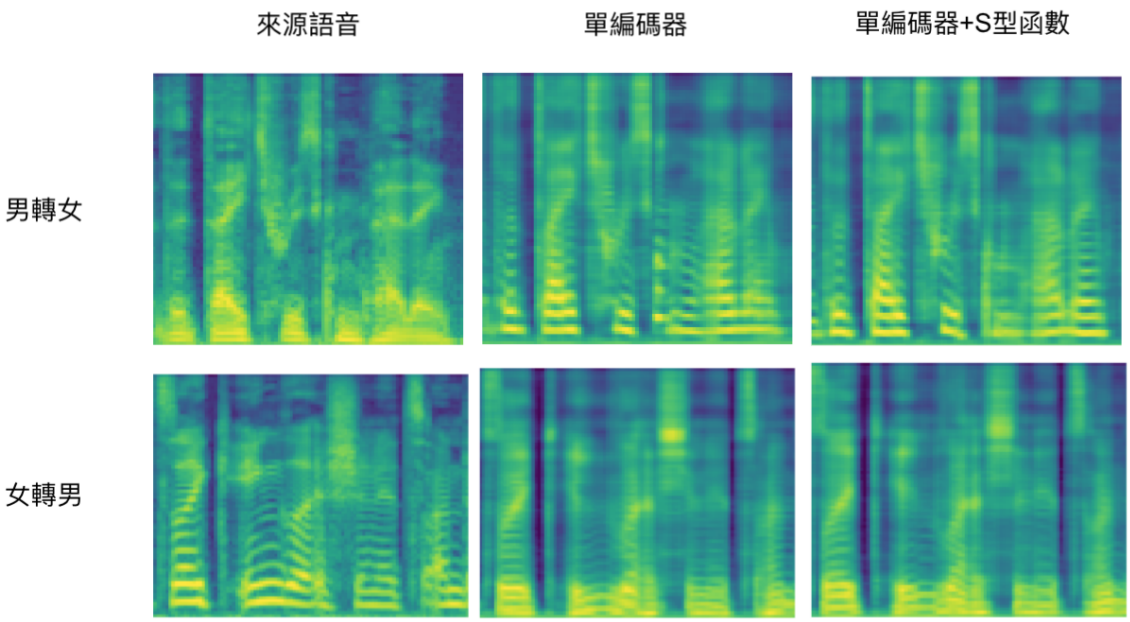


圖 4.6: 時頻譜比較，上排為男轉女，下排為女轉男。最左邊為原始音訊，中間為單編碼器模型，右邊為本章提出之模型轉換後結果。

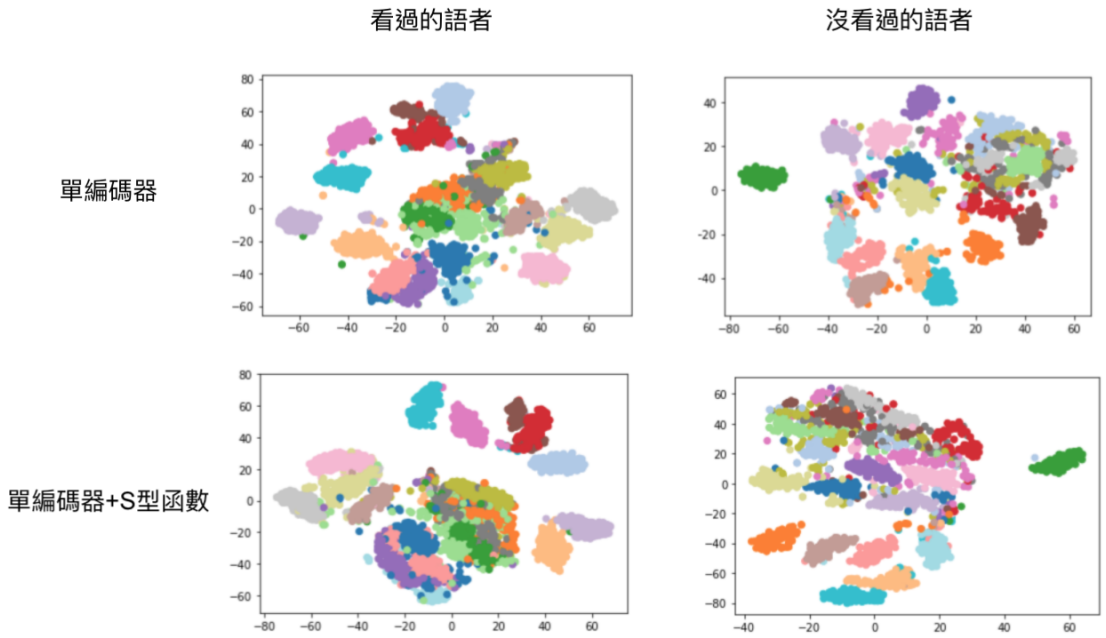


圖 4.7: 語者表徵視覺化

doi:10.6342/NTU202101337

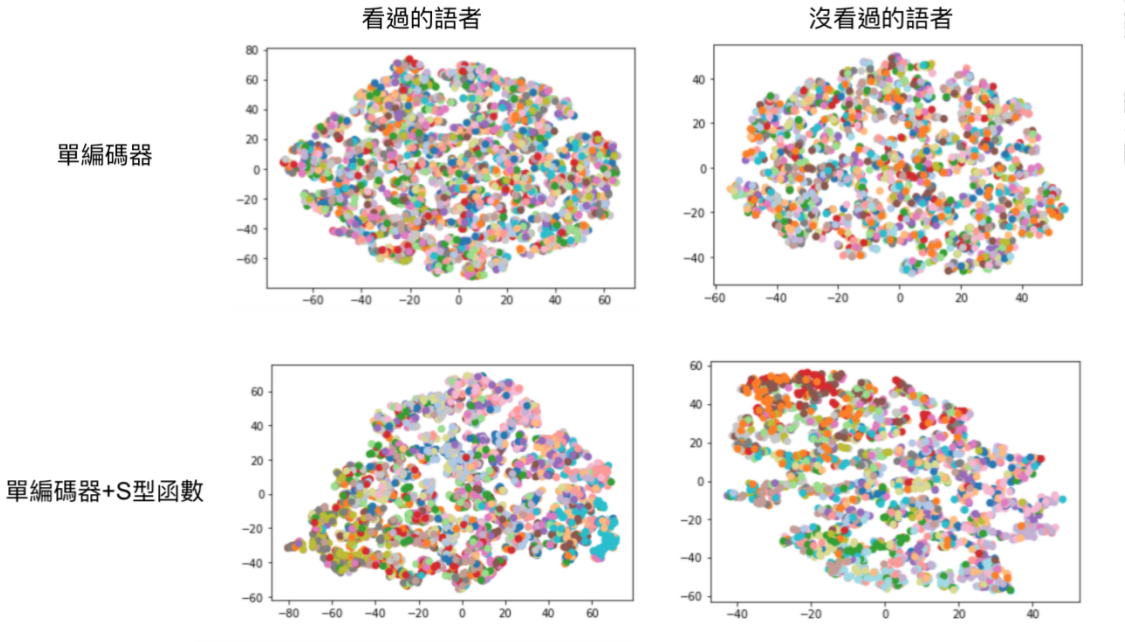


圖 4.8: 內容表徵視覺化

4.5.2 激活函數的影響

首先，我們針對第三章提出模型的內容表徵加上不同激活函數，在語音轉換模型訓練完之後，固定模型參數，另外訓練一個語者分類器，測試內容表徵帶有多少語者資訊，此語者分類器架構如表 3.5 所示，正確率越低越好。實驗結果如表 4.2，我們發現使用 S 型函數當作資訊瓶頸，可以讓內容表徵的語者正確率大幅下降，同時保持模型的重建能力；除此之外，藉由調整變種 S 型函數的變化率 α 到某一個範圍能得到最佳的結果，其中變種 S 型函數定義為

$$\text{Sigmoid}_{\alpha}(x) = \frac{1}{1+\exp(-\alpha x)}.$$

doi:10.6342/NTU202101337

	語者分類正確率（%）	重建誤差
無激活函數	80.5	0.150
ELU	79.8	0.150
ReLU	78.6	0.150
Sigmoid, $\alpha = 1$	58.8	0.149
Sigmoid, $\alpha = 0.5$	0.9	0.151
Sigmoid, $\alpha = 1 \times 10^{-2}$	1.3	0.149
Sigmoid, $\alpha = 1 \times 10^{-6}$	0.9	0.178

表 4.2: 不同激活函數對單編碼器模型影響

4.5.3 S 型函數分析

4.5.2 節實驗結果顯示，使用合適的 S 型函數所構成的資訊瓶頸，能在不傷害模型重建能力的前提下，使內容表徵中的語者資訊進一步被抹除。我們接著固定 S 型函數的 $\alpha = 0.5$，比較有加上 S 型函數和沒有使用 S 型函數的兩種模型，在不同的表徵通道數的資訊解纏、重建能力取捨表現。實驗結果如圖 4.9。圖表橫軸為模型重建誤差，縱軸為內容表徵上的語者分類正確率。這兩個數值都越低越好，因此這條曲線要越往左下方靠越好。先看藍色的曲線，改變單編碼器模型的內容表徵通道數，不難理解，通道數越大，則資訊瓶頸越小，語者資訊越有可能會滲透到內容表徵，因此有很高的語者分類正確率；同時，帶有越多資訊，也越有助於模型重建，因此重建誤差越小。再來看紅色的曲線，加上了 S 型函數作為資訊瓶頸，不但沒有傷害模型重建能力，同時降低了內容表徵中語者的資訊，讓取捨曲線進一步往左下移動，達到雙贏的局面。另外，我們也比較不同模型深度，加上 S 型函數引導的效果。本實驗使用 $\alpha = 0.5$ 的 S 型函數，結果如表 4.3。表中的層

doi:10.6342/NTU202101337

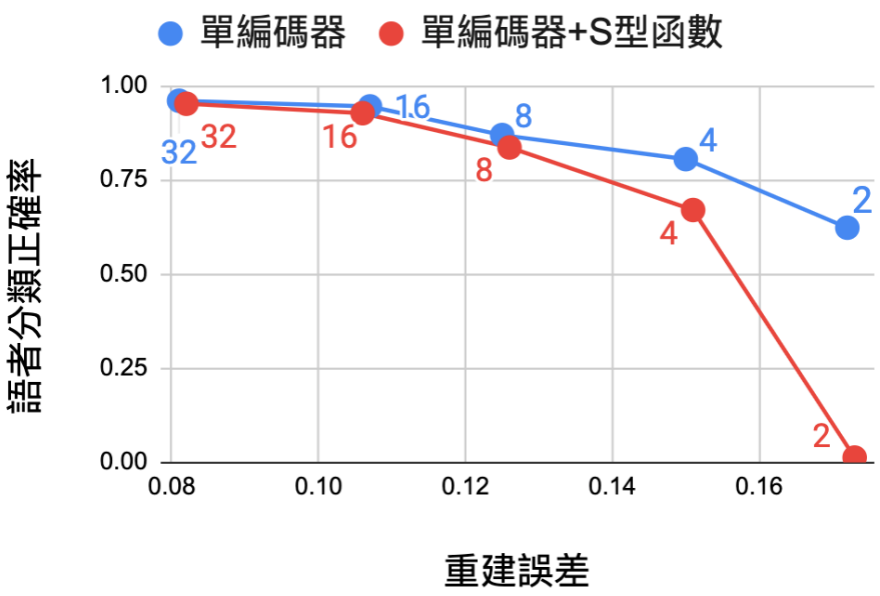


圖 4.9: 模型解纏能力與重建能力的取捨曲線。圖中點旁的數字代表模型內容表徵所選用的通道數。

數為編碼器中編碼模組的數量以及解碼器中解碼模組的數量。我們發現單編碼器模型在不同的深度，只要加上 S 型函數，都會顯著降低內容表徵的語者分類正確率，同時又不會傷害模型的重建能力。

4.5.4 客觀評估

本節主要比較之模型為第三章提出之單編碼器模型。另外我們也列出 AdaIN-VC、AutoVC 和 VQVC+ 的實驗跑分。這裡我們使用他們官方開源的程式碼來訓練模型，但為求公平，使用的資料集、聲學特徵等等都和第三章所使用的相同。如同3.5.3 節 ，我們也比較了以下幾個客觀分數

- 語者分類正確率
- 重建誤差
- 語音辨識字符錯誤率

doi:10.6342/NTU202101337

層數	S 型函數	語者分類正確率（%）	重建誤差
2		87.9	0.139
4		91.2	0.128
6		93.2	0.125
2	有	79.3	0.138
4	有	74.9	0.131
6	有	76.8	0.125

表 4.3: S 型函數對不同深度模型的影響，上半部是沒有加上任何激活函數的模型，下半部是加上 S 型函數（$\alpha = 0.05$）引導的模型。

- 與者驗證接受度
- 預測平均意見分數

實驗結果見表 4.4。單編碼器搭配 S 型函數的模型幾乎在所有客觀評估分數上的到最佳表現，雖然字符錯誤率和預測主觀分數不如 AutoVC，但 AutoVC 其實他的語者驗證接受度是比較低的，在沒看過的語者轉換任務中，其所生成之語音事實上常常不像目標語者。

4.5.5 主觀評估

本實驗結果也使用主觀評估來衡量模型生成語音的表現。本評估中，我們使用的模型架構如表 4.5。

doi:10.6342/NTU202101337

	AdaIN-VC	AutoVC	VQVC+	單編碼器	單編碼器*
語者分類正確率（%）	76.7	28.4	40.1	87.9	**1.2**
重建誤差	0.156	0.156	0.190	**0.133**	**0.133**
語音辨識字符錯誤率（%）	39.4	**33.0**	52.4	36.5	35.9
語者驗證接受度（%）	86.4	81.2	79.9	**89.9**	88.7
預測主觀分數	3.35	**3.42**	3.25	3.29	3.33

* 本章提出之模型，使用 S 型函數作為資訊瓶頸

表 4.4: 客觀評估

語者相似度

我們請受試者聽三段語音

- A：目標語者原始語音
- B：單編碼器合成之語音
- C：雙編碼器合成之語音

其中 A 語音固定為目標與者原始語音，B 和 C 段隨機調換順序，受試者並不會事先知道哪段語音是由哪個模型所生成。聽完三段語音之後，我們請受試者填選下列選項

- A 語者和 B 語者較相似
- A 語者和 C 語者較相似
- 差不多

doi:10.6342/NTU202101337

	單編碼器 +S 型函數	單編碼器
層數	6	6
編碼模組隱藏通道數	128	128
激活函數	Sigmoid, $\alpha = 0.05$	
內容表徵通道數	3	3

表 4.5: 模型架構細節

得到結果如圖 4.10 ，顯示使用合適的 S 型函數可以在語者相似度評分上得到較佳的成績。

語音品質

使用平均意見分數來衡量模型合成的語音品質。我們加入 MelGAN 生成的原始音訊一起比較，作為參考的分數上限。結果如圖 4.11 所示，單編碼器加上 S 型函數的引導，讓語音生成品質得到提升。

4.6 本章總結

本章探討不同激活函數對模型表徵與表現的影響，並且透過實驗發現使用合適的變種 S 型函數能改善單邊碼器模型的解纏能力與重建能力取捨。另外實驗也顯示，對於不同深度的模型，使用 S 型函數引導，都能達到進一步的資訊瓶頸，同時不傷害模型重建能力。主觀實驗也顯示，加上 S 型函數引導能略為帶來語音品質的進步。

doi:10.6342/NTU202101337

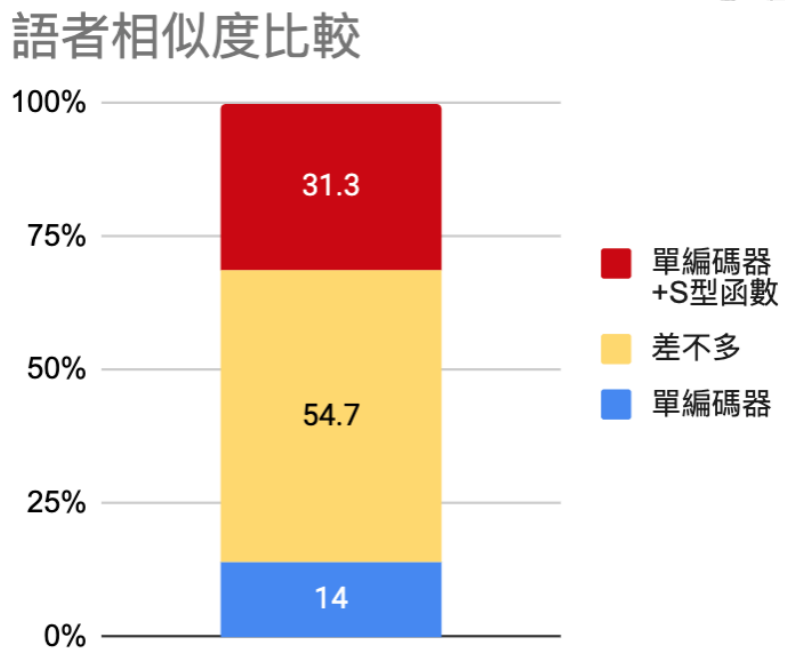


圖 4.10: 語者相似度比較

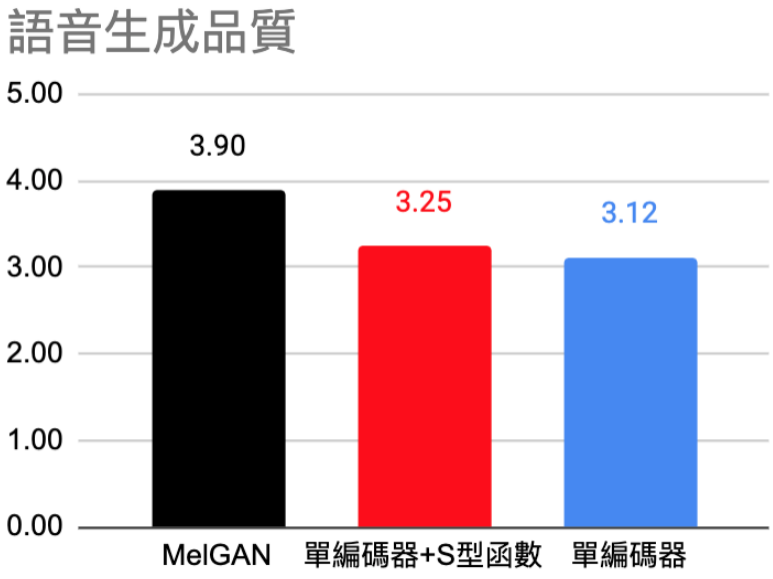


圖 4.11: 語音生成品質主觀分數

doi:10.6342/NTU202101337

第 五 章　結論與展望

5.1　研究貢獻與討論

本研究主要方向為一次性樣本任意對任意語者轉換任務模型的改善。主要的改善有

- 我們主要基於 AdaIN-VC 之想法，使用自適應實例正規化的特性，將雙編碼器的模型架構改為單編碼器。
- 將上述模型結合 U 型網路，使用跳躍連接網路將資訊瓶頸較強的語者資訊傳遞到解碼器，自然避免了資訊滲透。
- 在參數量大幅下修的情況下，提升運算速度、降低運算資源，且達到與 AdaIN-VC 同等表現。
- 探討激活函數作為資訊瓶頸對表徵解纏能力的影響。
- 進一步找到合適的激活函數，讓單編碼器語音轉換模型表現提升。
- 提供完整、可以重現（Reproducible）實驗結果的開源代碼，並且整合 AdaIN-VC、AutoVC 與 VQVC+ 等模型，包裝成統一的語音轉換模型研究框架。

5.2　未來展望

語音轉換任務至今也越來越多研究投入。一次性樣本的語音轉換的品質，仍然還有很大的進步空間。以文字生成語音（Text-to-Speech, TTS）為例，目前生成品質

doi:10.6342/NTU202101337

已經很不錯，甚至早已普及到我們生活周遭。然而，一次性樣本語音轉換的品質，距離商業化還是需要後續更多研究來突破。

我認為合理使用豐富語料語言的預訓練語音表徵，或者是其他任何方式所學到的語音表徵，也許是未來十分重要的研究課題。畢竟在真實世界，即使某種語言的語料十分稀少，但語言和語言之間，可以預期有許多共通特性。透過預訓練的表徵，我相信能夠有效調適到新的語言或是新的語音任務。

另外，本論文透過實驗方法，發現使用合適激活函數能夠引導模型達到更好的表現。選擇合適激活函數造成的資訊瓶頸，雖然能普遍讓表徵解纏能力改善，但當中的細節、根本原因，目前尚不明朗，這也是未來研究方向。而透過自適應實例正規化，雖然能達到語者轉換的效果，但單純使用平均值與標準差這兩種簡單的數值來代表實際上十分複雜的語者資訊，恐怕還是不夠的；可能的解法有加入更高階的動差（Moment）的概念進模型之中，或是使用直方圖等化（Histogram Equalization, HEQ）[36] 的技巧來將語者表徵的分佈相互對應。

除此之外，本論文只討論英文口語語音的研究。然而在聲調語言（Tonal Language），例如中文，直接套用本論文之方法會有轉換前後的聲調不正確的議題。這個問題在男女互轉的情況下尤其嚴重。可能的解法是加入基頻（F0）作為模型其中一個輸入，讓模型能多出一個資訊來預測輸出音訊的音調。

未來，也會有更多自監督式語音表徵出現，以及更多有趣、厲害的想法被拋出並實踐，期待任意語者的一次性樣本語音轉換，有一天能夠被普及化，並改善我們生活品質。

doi:10.6342/NTU202101337

參 考 文 獻

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, 2012.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

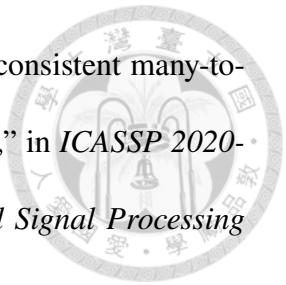
[4] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[5] J. chieh Chou and H.-Y. Lee, "One-Shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization," in *Proc. Interspeech 2019*, 2019, pp. 664–668. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2663

[6] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[7] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," *arXiv preprint arXiv:1905.05879*, 2019.

doi:10.6342/NTU202101337



[8] K. Qian, Z. Jin, M. Hasegawa-Johnson, and G. J. Mysore, "F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6284–6288.

[9] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[10] D.-Y. Wu and H.-y. Lee, "One-shot voice conversion by vector quantization," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7734–7738.

[11] D.-Y. Wu, Y.-H. Chen, and H.-Y. Lee, "Vqvc+: One-shot voice conversion by vector quantization and u-net architecture," *arXiv preprint arXiv:2006.04154*, 2020.

[12] A. v. d. Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," *arXiv preprint arXiv:1711.00937*, 2017.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[14] T.-h. Huang, J.-h. Lin, and H.-y. Lee, "How far are we from robust voice conversion: A survey," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 514–521.

doi:10.6342/NTU202101337

[15] W. S. MCCULLOCH and W. H. PTTs, "A logical, calculus of the ideas immanent in nervous activity."

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

[17] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[18] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

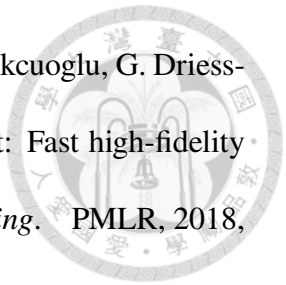
[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[20] V. Zue and R. Cole, "Experiments on spectrogram reading," in *ICASSP'79. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4. IEEE, 1979, pp. 116–119.

[21] D. Griffin and J. 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.

[22] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

doi:10.6342/NTU202101337



[23] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," in *International conference on machine learning*. PMLR, 2018, pp. 3918–3926.

[24] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[25] J. Lorenzo-Trueba, T. Drugman, J. Latorre, T. Merritt, B. Putrycz, R. Barra-Chicote, A. Moinet, and V. Aggarwal, "Towards achieving robust universal neural vocoding," *arXiv preprint arXiv:1811.06292*, 2018.

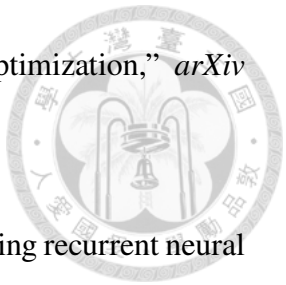
[26] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *arXiv preprint arXiv:1910.06711*, 2019.

[27] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *arXiv preprint arXiv:2010.05646*, 2020.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[29] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1. Citeseer, 2013, p. 3.

doi:10.6342/NTU202101337



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

[31] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*. PMLR, 2013, pp. 1310–1318.

[32] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.

[33] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[34] Y. Leng, X. Tan, S. Zhao, F. Soong, X.-Y. Li, and T. Qin, "Mbnet: Mos prediction for synthesized speech with mean-bias network," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 391–395.

[35] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[36] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.

doi:10.6342/NTU202101337