臺北市立建國高級中學 108 學年度科學展覽作品說明書

科別：資訊科

組別：高級中等學校組

作品名稱：「聲不由己」- 聲音風格轉換

關鍵詞：自動編碼器、機器學習、Deepfake

編號：

摘要

我們利用了一篇 IBM 所發表的論文（Qian et al., 2019），實作以自動編碼器為基礎的人聲轉化器。其功能為輸入兩個人的錄音片段，能輸出另外一段音檔，為第一個人所講的話與第二個人聲音風格的結合。除了高效能之外，不需要很多的輸入和不限定人的轉換是這個模型最大的特點，不需要耗費巨大的資源、可以利用單個 GPU 做訓練的特性，對於做實驗，微調也有很大的幫助，相較於其他（例：以 GAN 為基礎）的模型，多有過之而無不及。其訓練方法也有其特殊之處，由於兩個人講相同的話的難取得，利用自我轉換以計算損失函數，以利訓練。

壹、 研究動機

高一上時我們班前往新加坡進行海外參訪的活動，而在這 7 天的行程當中，除了在日常的交流與參訪行程收穫滿滿外，讓人同樣回味無窮的就是休閒娛樂的時間，而這其中一項便是惡作劇電話了，這也帶給了我們對於本次研究的靈感－要如何打出一個好的惡作劇電話呢？最重要的當然就是要模仿另一個人的聲音了，模仿得越像就越容易成功，這便催生出了我們的研究主題－聲音風格轉換。

貳、 研究目的

一、利用機器學習技術協助完成聲音風格轉換。

二、藉由調整不同的訓練參數或資料集探討不同的訓練結果。

參、 研究設備及器材

一、設備：筆電、具有 GPU（NVIDIA GeForce 1070）的桌機

二、編輯器&平台：MacVim、Sublime Text 3、GitHub

三、語言：Python

四、使用套件：pytorch、tensorflow、resemblyzer、numpy

肆、 研究過程與方法

一、資料尋找與蒐集

（一）、尋找可用模型架構

我們找到了許多的聲音轉換的模型。不過，有一些模型沒有辦法達到任意兩個人之間的轉換，而只能將任何人轉成特定一個人（Sun et al., 2016），或者是以 GAN 為底的模型（Donahue et al., 2018），不過有文章提到其所需要的資料量非常龐大，收斂與否也不易調整，就先不列入考慮了。也有看到 Google 所提出的 Tacotron 模型（Yuxuan Wang et al., 2016），然而其主要是文字轉換聲音的模型。

（二）、決定參考 IBM 論文模型進行訓練

決定使用 IBM 所發表的模型（Qian et al., 2019）。 其可說是集結上述其他的模型於一身，模型不大，不會很難訓練；所需資料量並不是太多（實際上，約 10GB 的音檔），而也可以順利達成多人對多人的聲音風格轉換。我們找到其在 GitHub 上面的部分專案程式，有範例音檔以供轉換，證明其能夠運行。然而，作者僅僅附上模型架構與轉換程式，並未公開訓練程式與資料格式等訊息，我們須自行實作。

二、理論基礎

（一）、何謂自動編碼器

最基本的自動編碼器由兩個部分所組成：一個「編碼器」和一個「解碼器」，編碼器負責讀取輸入，並將其轉換為一個編碼。解碼器讀入方才的編碼後，產生輸出。舉例來說，若有一個能除去一個圖片的雜訊的自動編碼器程式，那當使用者輸入一張有雜訊的圖片的時候，編碼器會看到，並將之轉換為一個編碼，可能其中蘊含著各種資訊、例如筆畫的位子、順序等，給那個解碼器。解碼器得到這個編碼之後，會根據它產生一個最終輸出，也就是我們想要的，沒有雜訊的圖片 。編碼器與解碼器的內部可以自行設計，以這個圖片輸入輸出的模型為例子的話，或許有一些捲基層、全連接層，而產生編碼序列的也可能有一些 LSTM 的架構含在裡面。值得一提的是，輸入之後，編碼器所產生出來的編碼不是人看得懂的，像是編碼器和解碼器之間的秘密語言般，非此兩者，無法解讀。

（二）、AutoVC 自動編碼器

1. 模型簡介

AutoVC 自動編碼器即是將將自動編碼器的概念，應用於聲音轉換之上：若要達成人聲間的聲音風格轉換，所需要的是將一個人的「說話風格」與「說話內容」分離。具體來說，給定一個人所說的話，可以透過一個「風格編碼器」$E_S(\cdot)$ 得到風格編碼；一個「內容編碼器」$E_C(\cdot,\ \cdot)$ 將剛才所產生出的 $E_S(\mathrm{X})$ 和 X 所說的一併輸入，得到內容編碼 $E_C(\mathrm{X})$。相反地，對於某一個未知的一句話，若能得到其內容編碼 $E_C(\mathrm{Y})$ 與風格編碼 $E_S(\mathrm{Y})$，也可以透過「解碼器」來還原原本的那一句話 $\mathrm{D}(\cdot,\ \cdot)$。重點是，解碼器並沒有限制內容與編碼的來源相同！只要提供內容編碼和風格編碼，縱使是不同句話，甚至（特別是）不同人所出來的內容、風格，也能夠完成轉換與生成。舉例來說，假設有兩個人 A、B，分別講了一句話（稱為a和b），然後我們想要得到以 B 的聲音說出 A 本來想表達的那一句話（意即將b說話的內容

以a說話的風格表達），那只需要得到 B 的風格（$E_S(\mathrm{b})$）和 A 的內容（$E_C(\mathrm{a})$），並合併之送入解碼器，即可得所求。

我們稱 $\tilde{X}_{a\to b} = D(E_C(a),\ E_S(b))$ 為「初步近似」（initial estimate），因為難免會有聲音粗糙不足之處，為了讓所產生的聲音更加精緻，近似人聲，還加了一層「後層」（Postnet）$R_{a\to b}$，目的在於微調人聲，是鑲嵌於D內的一小部分。故最後的轉換結果為 $\hat{X}_{a\to b} = \tilde{X}_{a\to b} + R_{a\to b}$

2. 訓練過程

訓練時，因為鮮少有「兩個人講一模一樣的話」的例子供訓練，所以轉用另外一個方式：自我轉換。若同一個人講了兩句話，稱為 X_1 與 X_2，那當我們嘗試將前者的聲音轉換為後者的聲音時，因為聲音係出同一個人，理論上應該是一模一樣的聲音，藉此可以計算損失函數。論文中，提出並證明

$$\mathrm{L} = \min_{D(\cdot,\ \cdot),\ E_C(\cdot)} (L_{recon} + \lambda L_{content})$$

是一個可以達到轉換目標的損失函數。此處，

$$L_{recon} = \mathbb{E}\left[\| X_1 - \hat{X}_{1\to 2} \|_2^2\right], L_{content} = \mathbb{E}\left[\| E_C(X_1) - E_C(\hat{X}_{1\to 2}) \|_1\right]$$

，λ 為比例常數。然後因為想要排除上述所提到的後層，單純訓練主要解碼器的部分，所以通常也會加入一個初步近似損失值

$$L_{recon0} = \mathbb{E}[\| X_1 - \tilde{X}_{1\to 2} \|_2^2]$$

而新的損失函數就是：

$$\mathrm{L} = \min_{D(\cdot,\ \cdot),\ E_C(\cdot)} (L_{recon} + \lambda L_{content} + \mu L_{content0})$$

，同上 μ 為比例函數。雖然此新版的損失函數沒有經過以上嚴謹證明可行，但是實際上在實作的時候，有發現其確實會增加訓練的效能。

三、實作程式

（一）、資料處理

我們所用的資料集主要來自愛丁堡大學的 VCTK Corpus，其中含了約 109 人所說的話，每一個人約說 400 句話。他們所說的語言為英文，來自各地的腔調（例：美國、倫敦、曼徹斯特等腔調皆不同，需要列入訓練）、總共約 10GB 的音檔。這些音檔都不長（約 5～10 秒的片段），且是在錄影室內錄製，音質佳，可供訓練。我們也有找到一個中文的語音資料庫，然而它都是同一個人所講的，較沒有辦法訓練出其他特徵，暫時不考慮。

有了聲音檔案，接下來就是要如何處理了：首先，必須將聲音全數轉換為陣列形式的頻譜圖，並存下來；一組測試資料也有一些限制：

1. 兩句話必須出自同一個人
2. 兩句話不可以相同

我們選擇生成所有的 (i, j, k)，代表第 i 個人的第 j 句話和第 k 句話。這樣，只需要事先保證 i, j, k 在合理範圍內，且 $j \neq k$，先生成出這個所有可能的訓練句子的序列，再將其隨機打亂，就可以開始訓練了。

（二）、Style encoder（聲音風格編碼器）

我們的風格轉換器是用預先訓練好的，以 GE2E Loss（Li Wan et al., 2017） 為理論基礎，輸入一個人所說的話，輸出一個長 256 的向量，為那一句話的風格編碼。我們找到的風格編碼器稱為 Resemblyzer，由 Resemble AI 團隊開發。利用此團隊所公開的程式套件，我們可以訓練內容編碼器與解碼器，而不需要全部都自己重新訓練，徒增變數量，減少模型的穩定度。

（三）、訓練程式

首先，先確認作者的模型的輸入、出格式，且能夠正常運作；然後再開始撰寫

訓練程式。我們將程式分成三大部分：

1. 主要驅動程式

   作為使用者與後面程式的介面，能在轉換與訓練模式間切換。

2. 訓練程式

   行主要的訓練作業，將所有參數、數值輸入之後，開始訓練並紀錄訓練結果於圖上，且定時儲存訓練結果，以防止突發狀況發生，損失訓練成果。

3. 資料相關程式

   即上面所提到的作業，當訓練程式向資料驅動程式（dataloader）請求資料的時候，其必須將資料整理成訓練程式看得懂、能用的格式並送回。

當然，還有一些其他部分，例如在轉換時，會需要將所生成的頻譜圖轉換成音檔的程式（vocoder），與一些輔助函數的程式，於此不做贅述。

四、調整參數

藉由調整程式的參數，如學習率（learning rate）、批大小（batch size）等參數，或者調整資料集的人數、句數多寡、比例，或甚至初始條件的不同，都有可能會影響成果，所以就要進行這方面的微調與實驗，才能做出好的結果。

伍、 研究結果

完成訓練程式的實作後開始訓練聲音轉換模型，以下包含我們用不同參數、資料集訓練出的結果，其對應的損失函數曲線及聲音轉換成效：

一、使用不佳的 style encoder

Loss 雖有下降，但聲音風格轉換的效果不夠好，有時甚至出現偏向雜訊的結果，難以從產生出的音檔推斷原始文句。

二、使用人數較少的資料

已能成功完成聲音風格轉換，但聽起來與原本的預期仍有一段差距，內容變得稍微模糊不清，但已能辨識文句的內容。

三、使用人數較多的資料

（一）、正常聲音轉換

能成功完成任意兩個人之間的聲音風格轉換，雖然有時仍會出現一些模糊的內容，但大致上與一般人說話無異。

（二）、轉換中文文句

將鍾文音檔丟入訓練好的模型中轉換，轉換出的音檔結果會有外國腔調的效果。

陸、 討論

一、使用不同的 style encoder 會對訓練結果造成不同影響

（一）、利用 One Hot 的 style encoding 進行訓練

使用一一對應的向量來代表每一個人的聲音風格，在訓練時可以排除 style encoder 所帶來的誤差，也較容易實作，能更早看到訓練成果。缺點則是在訓練完成後，只有在訓練資料集裡的人能有機會完成聲音風格轉換，無法達成我們的研究目的。

（二）、利用風格重疊率高的 style encoding 進行訓練

利用 style encoder 對於每個音檔產生出對應的 style encoding，理論上訓練完成後便可以成功進行任何人對任何人的聲音風格轉換，但如圖（6-1）所示，由於產生出的 style encoding 在不同人身上的聲音風格重疊率高，對於同樣的 style

encoding，有可能其實為兩個不同人的說話音檔，這會造成訓練上的困難，很難成功將兩個人的聲音區隔開來，也就難以完成聲音風格轉換，因此聲音轉換的實際成果不彰。

（三）、利用風格重疊率低的 style encoding 進行訓練

修正（二）的問題，我們改變了 style encoder 的實作方式，如圖（6-2）所示，style encoder 已能較好的分隔出不同的人說話的風格，將這樣的 style encoding 加入訓練，便得到了較好的訓練成果。

二、資料集對訓練結果造成不同影響

（一）、中文資料庫與英文資料庫的取捨

前面有提到，中文和英文資料集的問題，因為中文資料集只有用到一個人說的話，雖然音質乾淨但是我們還是沒有辦法用；英文的 VCTK Corpus 則有不同人所說的大量資料。

（二）、VCTK 資料集中資料多寡的影響

如果要使用所有可能的錄音對的話，會有 $109 \times 400^2 \approx 1.6 \times 10^7$ 個錄音對，而在一個錄音對平均上要約一秒的時間計算的情況下，如此龐大的資料量很快就變得不可行（一千六百萬秒約為 185 天）。

因此我們勢必須要對這些資料的人數與音檔對數進行一些取捨。一開始我們使用人數較少、音檔對數較多的資料來進行訓練；後來則決定將人數調高，為了不讓訓練時間變得過長，也將音檔對數相對降低。

1. 訓練過程的差異

將人數調高的結果造成損失函數的下降也跟著變得緩慢，訓練時間也相對拉長，如圖（6-3）所示，橘色的圖曲線為一開始資料人數比較少的訓練曲線，可以看到它收斂的很快，一下子就達到了漸近線，而藍色的曲線則是用比較多人數的資

料集去訓練，其收斂效率緩慢許多，到了一萬步時，損失函數仍然持續下降。

2. 訓練成效的差異

我們發現，經過同樣的時間後，後面的訓練結果比較好。我們推測是因為音檔對數本來就是足夠的，再增加也徒勞無功；然而人數增加了五倍，增加了訓練即得多樣型，讓模型更能辨別不同的說話特徵，於是得到了更好的訓練結果。

（三）、以英文訓練的模型套上中文錄音片段

我們推測因為模型的訓練都是以外國的來源為主，所以比較容易抓到說英文的特徵，而說中文的特徵，比如捲舌或平仄等外國語言比較少的部分，則比較少抓住，進而形成很像外國人說話的結果。

三、未來展望

（一）、利用中文資料集訓練，優化模型對中文音檔間的轉換

（二）、升級硬體設備與增加平行處理功能，加速訓練與實驗的過程

（三）、繼續調整參數，拉長訓練時間，以求模型的效果達到最佳化

（四）、訓練另外一個模型將聲音對上目標的人的五官，製造成影片，增加逼真度

柒、結論

一、我們發現，基於自動編碼器的模型能達到：

（一）、不需要太多的資源，不論是訓練時間（用一個 GPU 訓練，約 1～2 日）、轉換時間（約 10～20 分鐘）或訓練資料量，都不至於無法達成，能在自己的家中進行實驗。

（二）、能實現多對多（many-to-many）的轉換，不限輸入、輸出的人選。

（三）、有效（此指內容清晰能辨為原本的內容，且風格有明顯轉變之跡象）的轉換不

同風格的聲音。

二、發現批大小、風格編碼器的選擇，與資料量選擇的比例多寡與訓練成果的關係，從一堆雜訊到無法控制轉換風格，都是有可能發生的風險，須謹慎防範。

捌、參考資料

Sun, Lifa & Li, Kun & Wang, Hao & Kang, Shiyin & Meng, Helen. (2016). Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. 1-6. 10.1109/ICME.2016.7552917.

Donahue, C., McAuley, J., and Puckette, M. Adversarial audio synthesis. arXiv preprint arXiv:1802.04208, 2018.

Yuxuan Wang and RJ Skerry-Ryan and Daisy Stanton and Yonghui Wu and Ron J. Weiss and Navdeep Jaitly and Zongheng Yang and Ying Xiao and Zhifeng Chen and Samy Bengio and Quoc Le and Yannis Agiomyrgiannakis and Rob Clark and Rif A. Saurous, arXiv:1703.10135

Kaizhi Qian and Yang Zhang and Shiyu Chang and Xuesong Yang and Mark Hasegawa-Johnson, arXiv:1905.05879v2