Study of transformer-based TTS and its embedded implementation
基於變換器之文字語音轉換研究及嵌入式實現

Sian-Yi Chen

Advisors : Tay-Jyi Lin and Chingwei Yeh

Outline

- **三點貢獻**
  1. 生成指定對象聲音，並且音質、音色良好
  2. C版本音質更好
  3. Transformer C-based implementation
- **論文章節規劃**
- **python 與 C 版本結果差異**

論文章節規劃

1. Introduction
   - 研究背景
   - 貢獻、動機
2. TTS
   - 傳統TTS
   - 近期趨勢的TTS方法
3. Multi-speaker, x-vector Transformer-TTS model
   - x-vector介紹
   - Transformer介紹
4. Experiments
   - 環境設置
   - 實驗結果
5. Implementation
   - 軟體實現過程
6. Conclusion

論文結構規劃

論文章節主要分為6大章：

0. Abstract
1. Introduction
   - 背景介紹：描述TTS的應用，並初步介紹TTS
   - 研究動機：transformer為神經網路中的經典
   - 研究貢獻
   - 論文架構：說明論文各章節安排

2. Text-to-speech (TTS)
   - 典型的TTS系統：介紹各類傳統TTS以及最具代表的參數語音合成系統
   - 近期趨勢的TTS方法：介紹VCC2020，以及我所參考的ASR+TTS

3. VCC2020 baseline Transformer (multi-speaker, x-vector Transformer-TTS model)
   - 網路架構
     - 網路特色
     - 網路各層功能介紹
       - word embedding、self-attention、Masked Multi-Head Attention
     - 架構分析:網路分成encoder與decoder

4. Experiments
   - 環境設置：說明使用的電腦硬體規格、軟體框架版本
   - Dataset：使用陽明大學320句文字，並自行錄製語料，300句訓練，
   - 訓練方法：使用微調(fine-tune)的方式取得神經網路的bias和weight
   - 實驗
     - 調整網路中的超參數(更改特徵提取的頻率、FFT point)
     - 調整訓練語句數量
     - 使用不同人選、性別進行微調
   - Experimental Results

5. Implementation
   - C code實現過程

6. Conclusions
   - 將實驗結果做結論
   - Future work
     - 可優化轉換時間(過多開檔動作)
     - 輸入輸出因python版本高度模組化而難以在C重現，若將前後步驟補齊可省略許多功能
     - 在網路結構最後是由conv1D收尾，感覺可以更改成其他神經網路或是修改conv1D參數查看實驗結果

References(提供20篇文獻)

Python 與 C 版本結果差異

Transformer 學習音色對象：震威
輸出語句：小朋友們在看螞 (蟻搬家) ，320句語料中的第301句

震威原音色

python 版本輸出結果

C 版本輸出結果

Outline

1. Introduction
   - 研究背景
   - 貢獻、動機
2. TTS
   - 傳統TTS
   - 近期趨勢的TTS
3. Multi-speaker, x-vector Transformer-TTS model
   1. x-vector介紹
   2. Transformer介紹
4. Experiments
   - 環境設置
   - 實驗結果
5. Implementation
   - 軟體實現過程
6. Conclusion

研究背景

- 身處科技日新月異的時代，與機器的互動成為生活中不可或缺的一部分。有別於過往使用按鍵互動，如今使用語音進行互動已是常見的互動型態，其中Google助理、Siri、電腦的朗讀功能皆是經典的應用。

- 上述應用皆使用將文字轉換成聲音的功能，這樣的技術就稱為語音合成（Text To Speech, TTS），而在語音合成中，有著許多不同追尋的面向，例如生成速度、自然度、清晰度、相似度…等。

研究動機

- 現在語音轉換隨著科技越發進步，想到如果媽媽在小孩睡前講床邊故事的時候，透過錄製短短幾句語料，就可以利用媽媽的聲音來撥放有聲書，不僅可以讓小孩安心睡，家長更可以空出時間來做其他事情。

- 而有聲書追求的是自然度與相似度，因此嘗試找尋並使用能夠有效達成自然度與相似度的TTS。

Outline

1. Introduction
   - 研究背景
   - 貢獻、動機
2. TTS
   - 傳統TTS
   - 近期趨勢的TTS
3. Multi-speaker, x-vector Transformer-TTS model
   - x-vector介紹
   - Transformer介紹
4. Experiments
   - 環境設置
   - 實驗結果
5. Implementation
   - 軟體實現過程
6. Conclusion

典型的TTS系統

1. 發音合成 (Articulatory Synthesis)：為模擬人類發聲器官的行為來產生語音
2. 共振峰合成 (Formant Synthesis)：根據 source-filter model 的規則生成語音
3. 拼接合成 (Concatenative Synthesis)
    - 單元選擇合成 (Unit Selection Synthesis)：從資料庫中選擇適合的單元進行拼接
    - 統計參數合成 (Statistical Parametric Synthesis)

近期趨勢的TTS方法

- 在 speech synthesis 分類中，傳統與現今技術我認為可以用 E2E model 作為分界點，而目前的聲學 (acoustic) 演算法主要分為三類，RNN based、Transformer based、CNN based。

- 我所使用的TTS是Transformer based，其中參考了VCC2020 (Voice Conversion Challenge 2020) 作為我的 baseline

- 在大會中有出現幾種神經網路Tacotron、LSTM、Transformer等，其中Transformer在MOS評分上分數又居於最高[1]，且大會提供了baseline[2]，因此選用它作為我的baseline

[1] T. Hayashi, et al., ”ESPNET-TTS: unified, reproducible, and integratable open source end-to-end text-to-speech toolkit,” *arXiv preprint arXiv:1910.10909* , 2019
[2] W.-C. Huang, T. Hayashi, S. Watanabe, T. Toda, “The sequence-to-sequence baseline for the voice conversion challenge 2020: cascading ASR and TTS,” *arXiv preprint arXiv:2010.02434*, 2020

Outline

1. Introduction
   - 研究背景
   - 貢獻、動機
2. TTS
   - 傳統TTS
   - 近期趨勢的TTS
3. Multi-speaker, x-vector Transformer-TTS model
   - x-vector介紹
   - Transformer介紹
4. Experiments
   - 環境設置
   - 實驗結果
5. Implementation
   - 軟體實現過程
6. Conclusion

x-vector

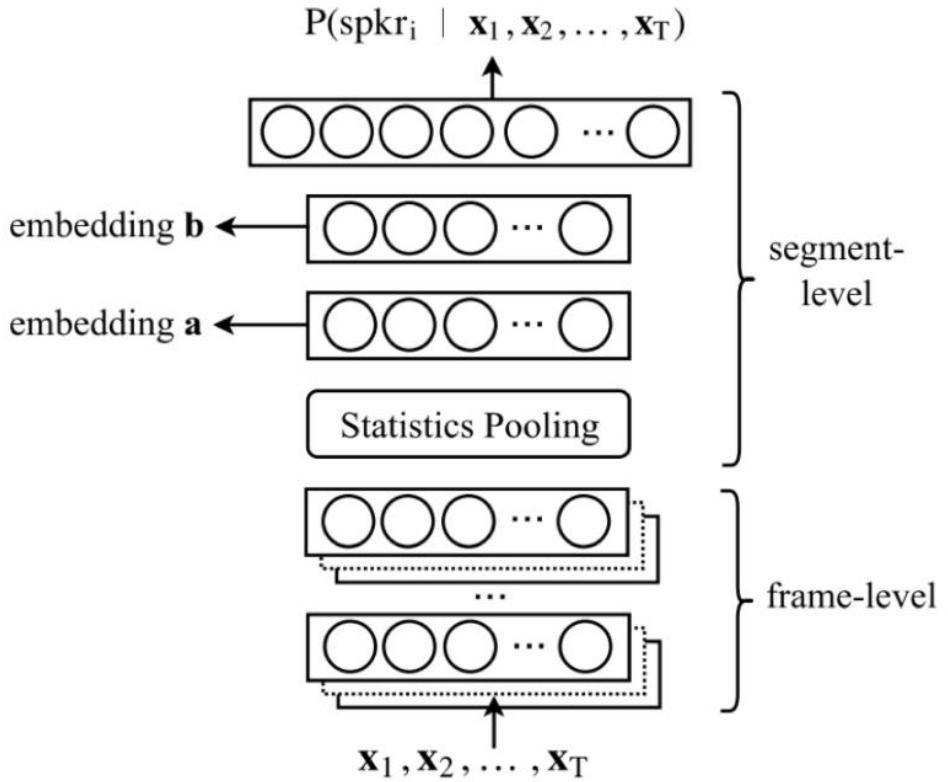
- X-vector 考慮了整段聲音訊號，他會計算每一小段的聲音輸出的特徵，得到 mean 和 variance 然後 concat 起來
- 最後再放進 DNN 裡面進行訓練，判斷這個聲音是誰的

- 在實際預測的時候，輸入的語音長度不同，因此會把語音結成多段，然後取這幾段的特徵平均值作為最後的 speaker embedding

- **X-vector 可以接受任意長度的輸入，然後轉換成固定長度的特徵表示**

- X-vector 包含多層 TDNN，一個 pooling 層，和兩層 embedding 層，一層 softmax

- x-vector layer
  - TDNN 層：相對於 LSTM 可以並行訓練，相對於 DNN 可以增加時序上下文關係
  - pooling 層：取出 TDNN 輸出的平均值還有標準差，再將兩者皆起來，得到句子級別的特徵表達
  - embedding 層：具有較好的特徵表達能力
  - softmax 層：節點數為訓練集說話人的個數



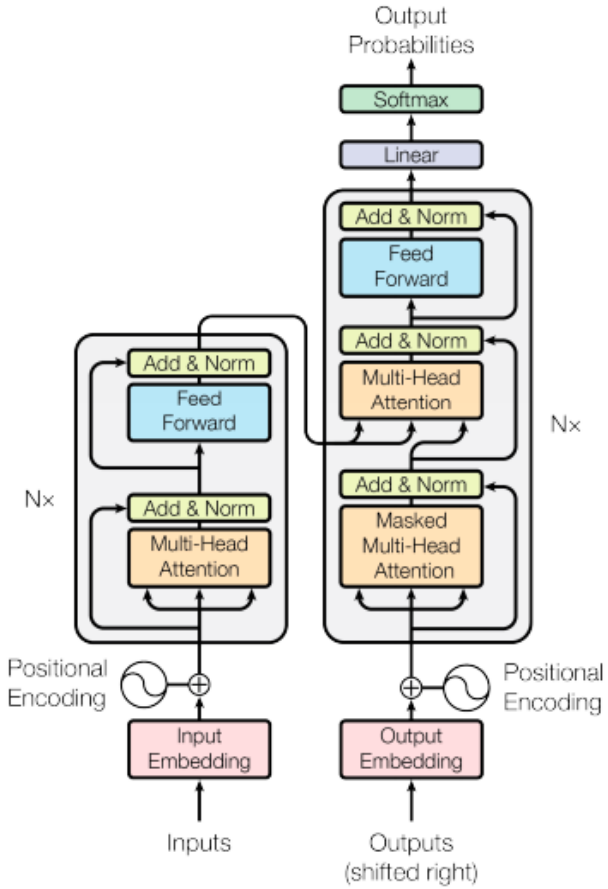
Transformer

Transformer 是近代語音處理的一個熱門神經網路，在 2017 年由 Google 提出 “**Attention Is All You Need**” 這篇論文 [1] 中出現，它由一組 encoder 與 decoder 組成，架構是基於 Seq2Seq + self-attention。

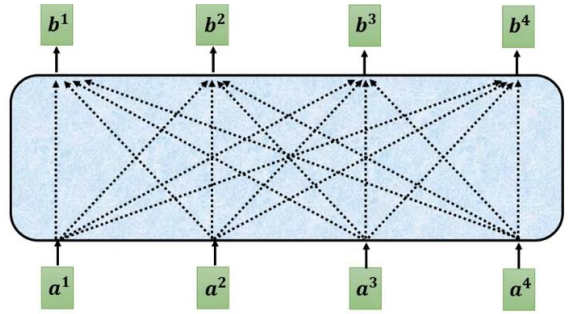
Sequence to sequence model 輸入一個sequence 就會輸出一個 sequence，而它的輸入、輸出的長度均由神經網路自行學習，兩者長度沒有對應關係。

Self-attention (自注意力機制) 則是 Transformer 主要取代 RNN (LSTM、GRU) 的主要原因，這個機制有兩大優點，一是他可以平行化運算，去掉 RNN 遞迴的結構，二是每一個輸出的向量，都會看過整個輸入的序列，因此當輸入的 sequence 太長的時候， RNN 容易忘記一開始的訊息，但使用 self-attention 的 Transformer 就可以知道要把注意力放在哪些資訊上，而不會因為輸入的資訊太長而忘記一開始的資訊。

所以 Transformer 的優點是除了可以處理更長的序列之外，還提升了運算效率。



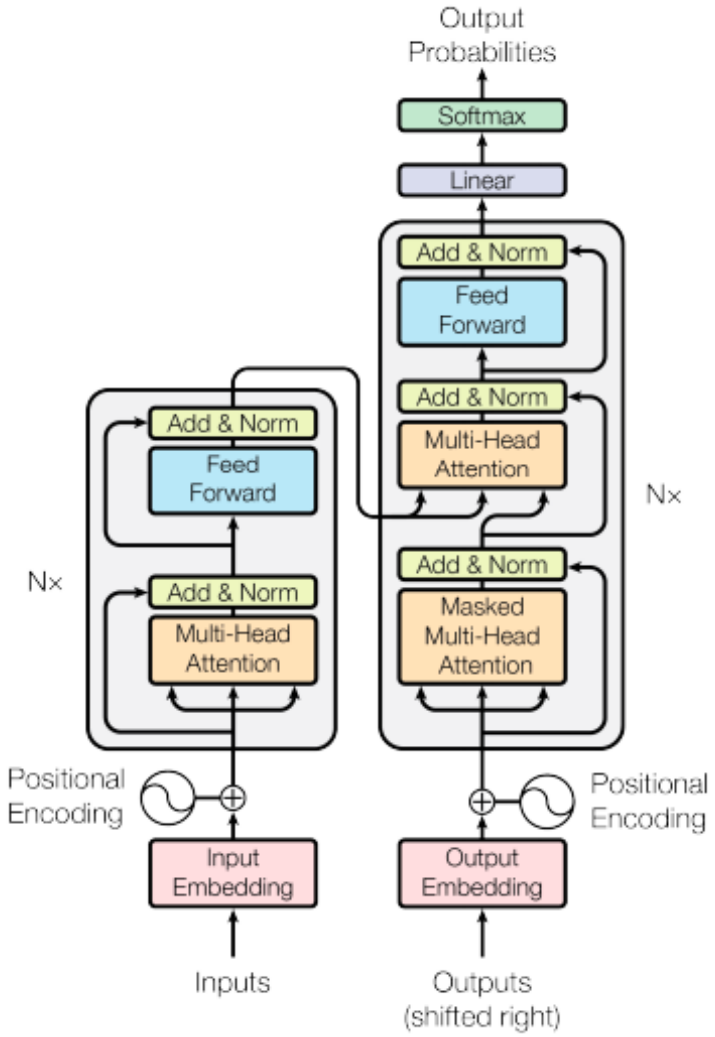
The Transformer - model architecture.



Self-attention (自注意力機制)

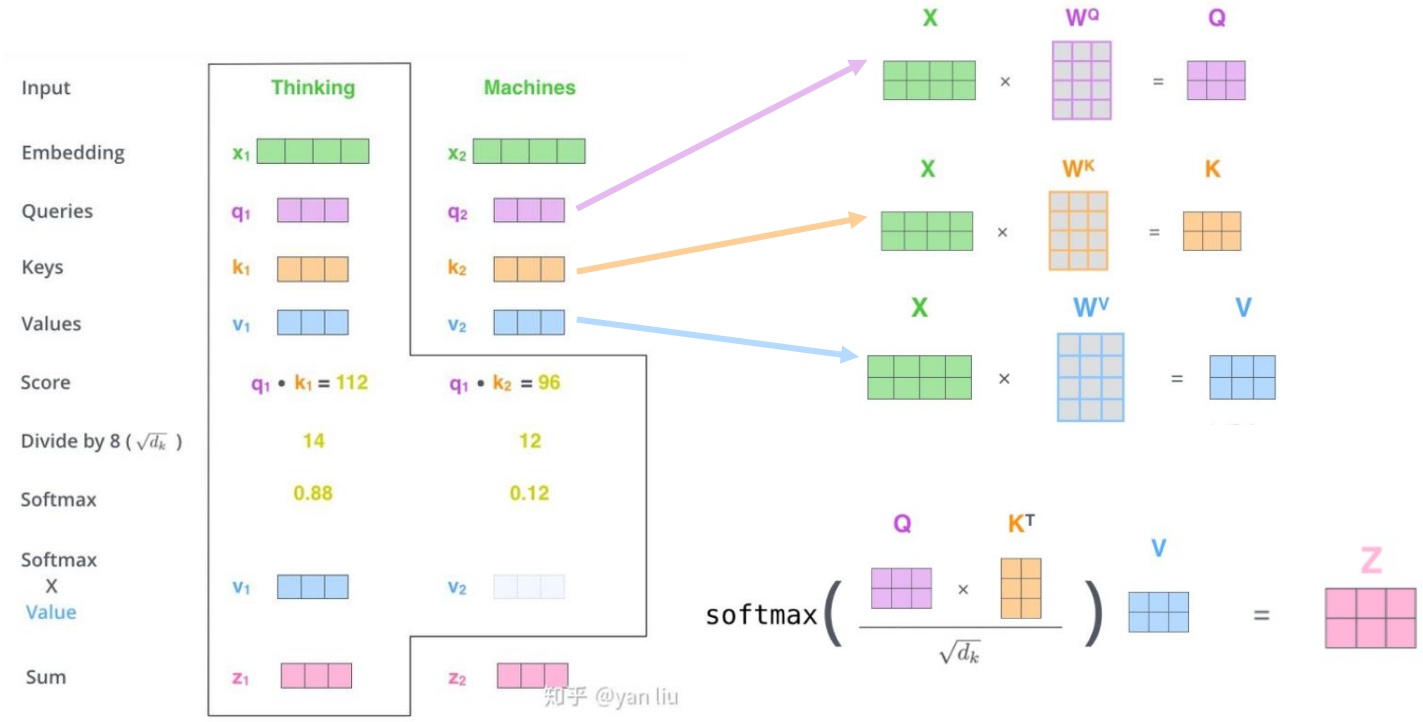
[1] A. Vaswani, et al., “Attention Is All You Need”, *arXiv preprint arXiv:1706.03762*, 2017.

Transformer



Transformer定義為：第一個完全依賴 self-attention 去計算 input、output 之間關係並不使用序列對齊的 RNN 或 CNN 的 transduction model。

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Query，Key，Value的概念取自於資訊檢索系統，舉個簡單的搜索的例子來說。
當你在網購平臺搜索某件商品（年輕女士冬季穿的紅色薄款羽絨服）時
你在**搜尋引擎**上輸入的內容便是 **Query**
然後搜尋引擎根據Query為你匹配 **Key**（例如商品的種類，顏色，描述等）
然後根據Query和Key的相似度得到匹配的內容（**Value**）。

Outline

1. Introduction
   - 研究背景
   - 貢獻、動機
2. TTS
   - 傳統TTS
   - 近期趨勢的TTS
3. Multi-speaker, x-vector Transformer-TTS model
   - x-vector介紹
   - Transformer介紹
4. Experiments
   - 環境設置
   - 實驗結果
5. Implementation
   - 軟體實現過程
6. Conclusion

Training processes

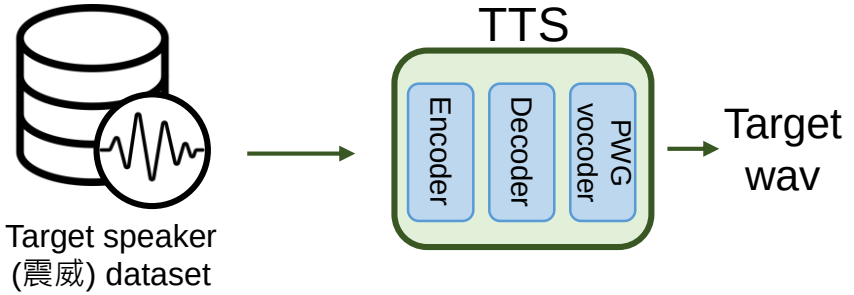
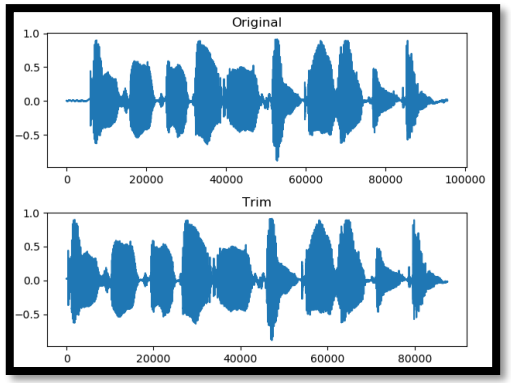


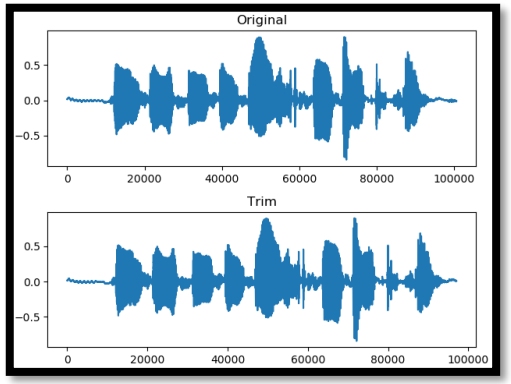
Figure 3: model fine-tuning

TTS 訓練流程

1. Data preparation
   - 選擇語料
   - 建立語料與文本的關聯檔
   - 降頻至 16kHz
   - 檢查準備的資料目錄、格式是否正確
2. Feature Generation (使用 TTS 預訓練中計算的統計數據對特徵進行標準化)
   - 切除空白音檔
   - 生成 fbank
   - 生成指定的 train、test 語句列表
   - 使用 cmvn dump pretrained model feature
3. Dictionary and Json Data Preparation
   - 使用 TTS 預訓練中內置的字典對標記進行索引
4. x-vector extraction
   - 生成 MFCC 並計算 energy-based VAD
   - 對於 Kaldi-based X-vector pretrained model 提取 X-vector
5. fine-tuning
   - Train E2E-TTS model (encoder)
6. Decoding
   - 對於 test set 做 TTS 解碼 (decoder)
7. Synthesis
   - 使用訓練過的 Parallel WaveGAN 將生成的 mel filterbank 轉回波形



大部分音檔可以正常切除



部分音檔前無聲區沒有被切除

Outline

1. Introduction
   - 研究背景
   - 貢獻、動機
2. TTS
   - 傳統TTS
   - 近期趨勢的TTS
3. Multi-speaker, x-vector Transformer-TTS model
   - x-vector介紹
   - Transformer介紹
4. Experiments
   - 環境設置
   - 實驗結果
5. Implementation
   - 軟體實現過程
6. Conclusion

軟體實現過程

Outline

1. Introduction
   - 研究背景
   - 貢獻、動機
2. TTS
   - 傳統TTS
   - 近期趨勢的TTS
3. Multi-speaker, x-vector Transformer-TTS model
   - x-vector介紹
   - Transformer介紹
4. Experiments
   - 環境設置
   - 實驗結果
5. Implementation
   - 軟體實現過程
6. Conclusion

Conclusion

- 在進行fine-tune時，女生的聲音訓練的較男生的好