Innovations of Wang TTS (based on VCC 2020 ref. design)

Sian-Yi Chen

Advisors : Tay-Jyi Lin and Chingwei Yeh

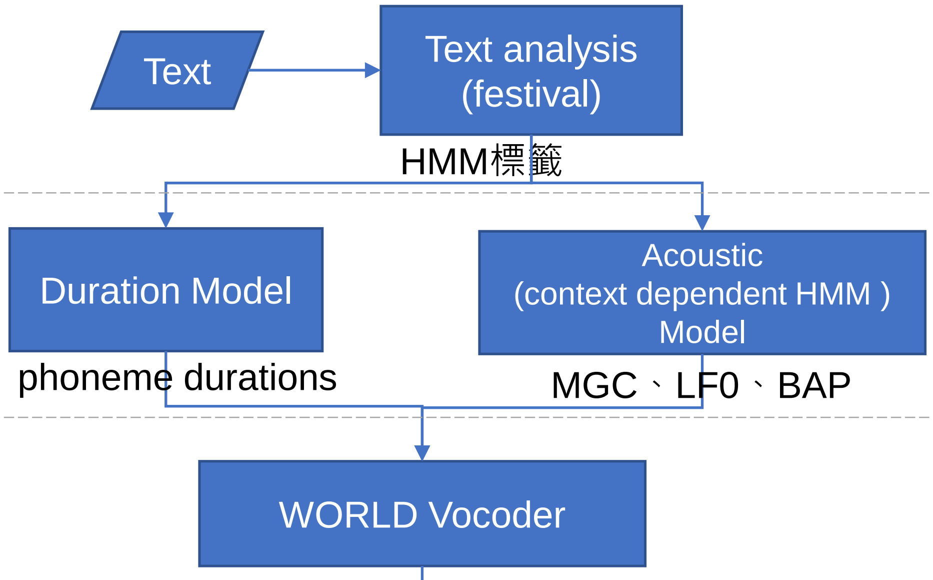
Outline

- ◆ 目標：描述清楚傳統TTS的I/O，並且有上下關係的說明如何改善baseline的音質

1. 傳統TTS
   - I/O與模塊處理
   - 傳統TTS架構與優缺
2. 現今TTS
   - 傳統與現今TTS的區別
   - 現今架構的分類
   - 現今TTS的優缺點
3. 基於VCC 2020 baseline TTS
   - 架構介紹
4. VCC 2020 baseline TTS問題與改善
   - 問題說明
   - 改善方案
5. 三點貢獻發想

Traditional TTS design

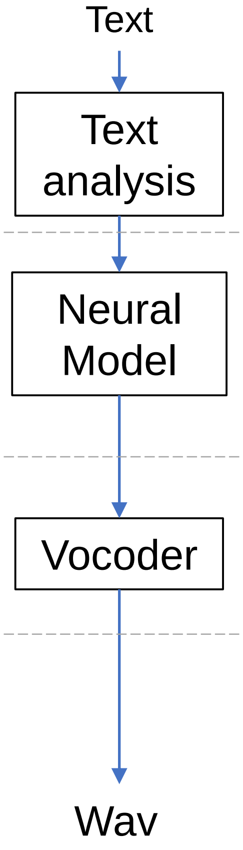
- block diagram with clear I/O & processing of each block



Output:

1. I came for information more out of curiosity than anything else.

2. Anyway, no one saw her like that.



1. **txt to label**
2. **label to training-data**

1. **Neural Network Architecture**
2. **I/O**

- **use 16kHz sampled audio signal**

- Advantages and disadvantages

統計參數合成 (Statistical Parametric Synthesis)

- 優：與更早以前的方法拼接合成(Unit Selection Synthesis) 比較，生成音檔更自然、更靈活方便修改參數、比串接式合成成本更低，不須要大量資料庫

- **缺：**
  1. **生成的語音還是具有較低的理解性**
  2. **很容易與人聲作區別**
  3. **像機器人的聲音**
  4. **在文本處理階段需要具備語言學、聲學的先備知識**

Modern TTS design

- Traditional TTS vs. Modern TTS
  - □ 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，傳統技術中在文本分析階段需要透過人工的方式進行向量化，才能讓神經網路進行訓練，為了解決傳統種種的缺點，發展出E2E model希望這些繁瑣的過程都能透過神經網路自行學習。

- 現今TTS架構的分類
  - □ RNN based
  - □ Transformer based
  - □ CNN based

- 現今TTS的優缺點 (尚未完成)

  - □ 因此選擇了Transformer based [1]作為我的baseline。

[1] 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.

Based on VCC 2020 ref. design

使用 VCC 2020 reference 的 VC baseline，架構由 2 個 model 組成，分別是 ASR + TTS，而兩個 model 互相獨立，ASR 將文字辨識出來後就可以與整個系統切開來，因此我們著重在 TTS model 的部分 (如圖一)。

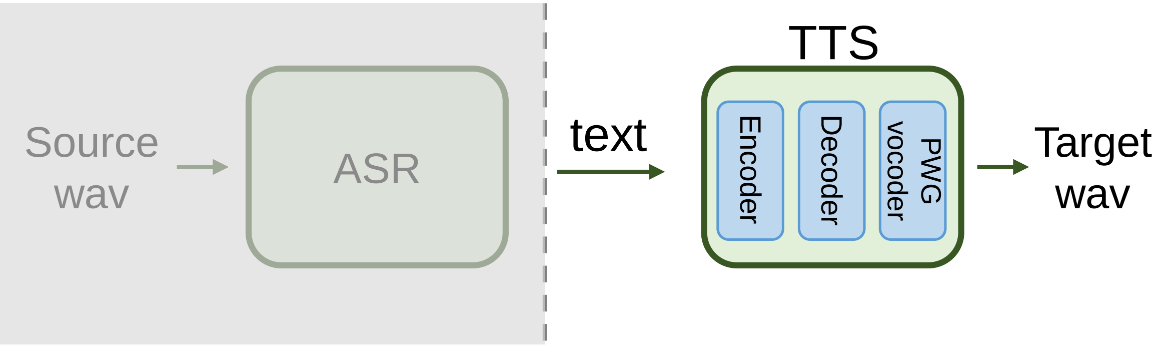
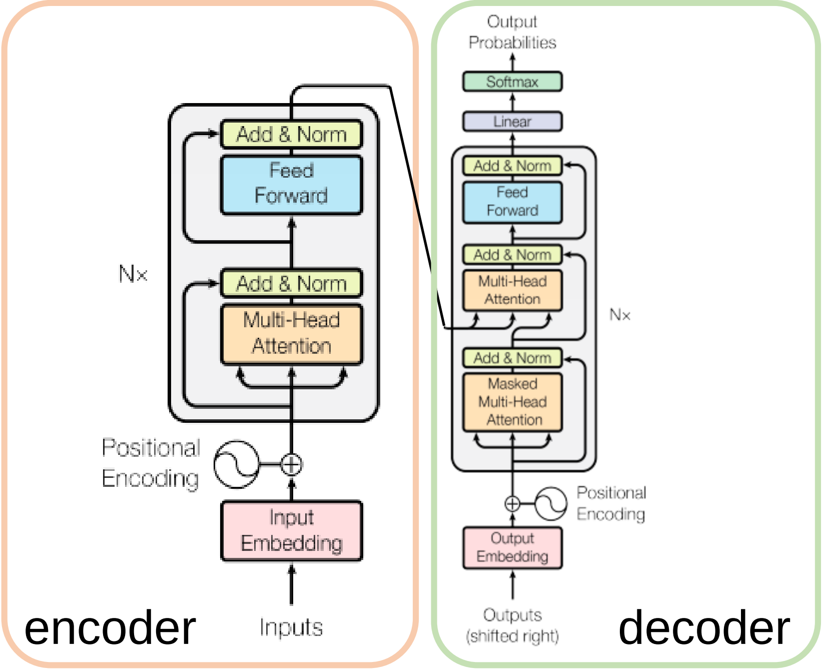


Figure 1: System structure

VCC2020 Baseline pre-training model: multi-speaker, x-vector Transformer-TTS model



架構可以拆解成以下四個區塊

1. Feature representation (MFCC)
   - 使用MFCC特徵作為輸入。
2. Embedding (x-vector)
   - 利用x-vector將輸入轉換成固定長度的特徵表示，也就是embedding的部分。
3. TTS model (Transformer)
4. Vocoder (Parallel WaveGAN)

VCC 2020 baseline TTS問題與改善

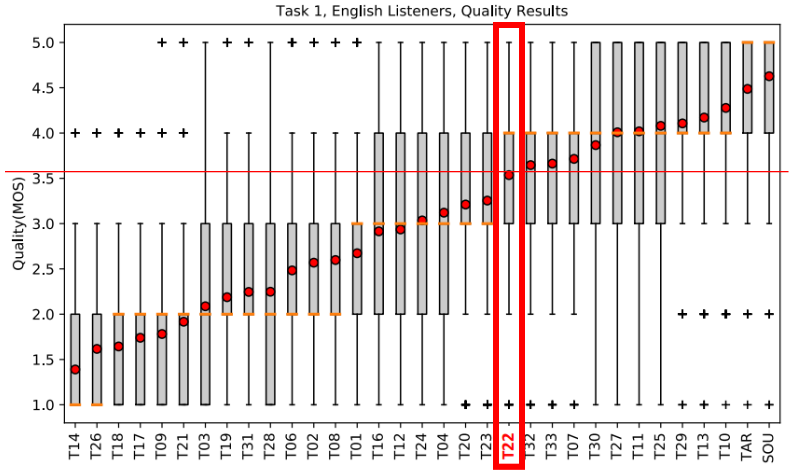
- 在論文中表示「ASR and TTS models still much room for improvement」，系統在VCC2020競賽中獲得3.5分的MOS分數，並獲得約90%的相似度

- 架構拆解成四個區塊
  1. Feature representation (MFCC)
  2. Embedding (x-vector)
  3. TTS model (Transformer)
  4. Vocoder (Parallel WaveGAN)

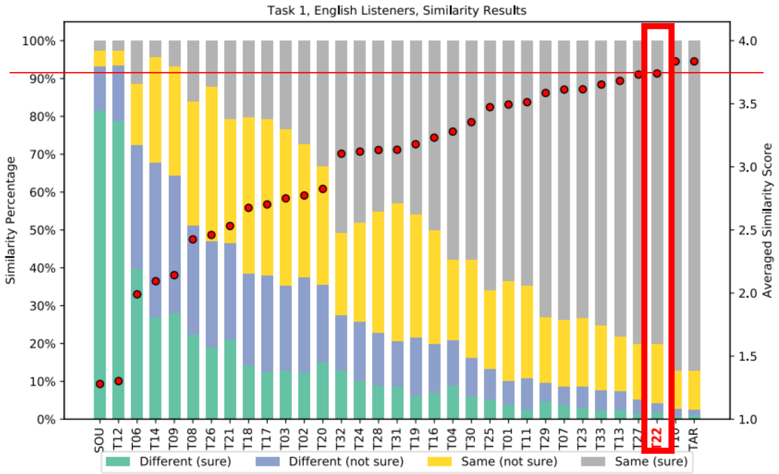
- 方案 (可以有效達到音效的改善)
  1. 認為將3或4使用的模型更改
  2. 提升目前使用的取樣率(16kHz)

- 原因
  1. Transformer算是較單純的模型
  2. 目前使用的Vocoder為non-AR，換成autoregressive model是普遍認知較好的方法

- 三點貢獻發想
  1. 合成出王老師的聲音
  2. 與baseline相比，音質提升
  3. 再次證明ASR+TTS這種cascade的方式是有競爭性的



(a) *Naturalness results for task 1.*



(b) *Similarity results for task 1.*

Text analysis (txt to label)

在Merlin提供的標籤中有兩種類別，分別是state align用狀態對齊與phoneme align用音素對齊，預設使用state align方式對齊。

State align使用HTK(Speech Recognition Toolkit)生成，以發音狀態為單位的標籤文件，而每個音素都由多個狀態組成，這邊則是指定生成5個HMM狀態的標籤。

txt to label

生成txt檔(一)，並使用festvox中一個名為EHMM的工具生成全文標籤，

EHMM(ergodic HMM)是一種對齊方法，它解釋了音素標籤之間可能存在停頓的可能性。

將txt檔轉換成全文標籤後，並依照HMM-base標籤格式(二)生成具有5個狀態得HMM標籤(三)

(一)：Text 檔

- (arctic_a0001 "Author of the danger trail, Philip Steels, etc.")
- (arctic_a0002 "Not at this particular case, Tom, apologized Whittemore.")

(二)：Context-dependent label format for HMM-based speech synthesis in English

p1ˆp2-p3+p4=p5@p6 p7

/A:a1 a2 a3

/B:b1-b2-b3@b4-b5&b6-b7#b8-b9$b10-b11!b12-b13;b14-b15|b16

/C:c1+c2+c3

/D:d1 d2 /E:e1+e2@e3+e4&e5+e6#e7+e8 /F: f1 f2

/G:g1 g2 /H:h1=h2ˆh3=h4|h5 /I:i1=i2

/J: j1+ j2- j3

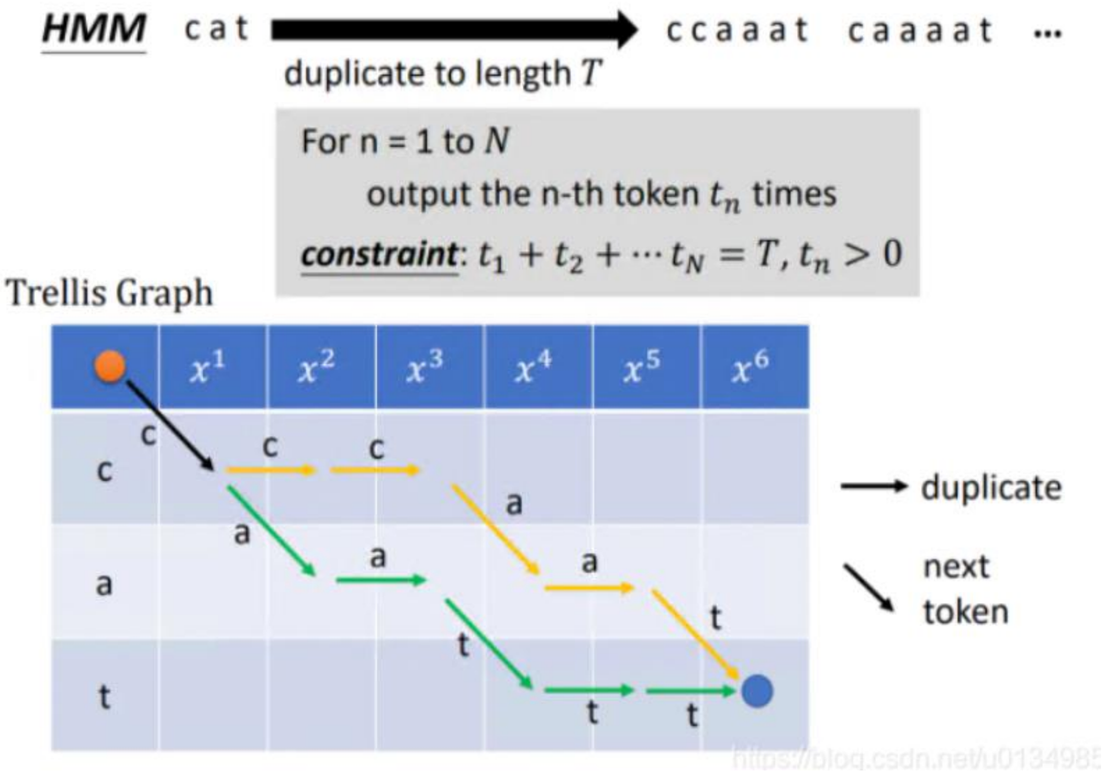
lab_format.pdf

(三)：具 5 個狀態的 HMM 標籤

- 0 50000 x^x-sil+sil=ao@x_x/A:0_0_0/B:x-x-x@x-x&x-x#x-x$x-x!x-x;x-x|x/C:0+0+0/D:0_0/E:x+x@x+x&x+x#x+x/F:0_0/G:0_0/H:x=x@1=2|0/I:0=0/J:14+8-2[2]
- 50000 100000 x^x-sil+sil=ao@x_x/A:0_0_0/B:x-x-x@x-x&x-x#x-x$x-x!x-x;x-x|x/C:0+0+0/D:0_0/E:x+x@x+x&x+x#x+x/F:0_0/G:0_0/H:x=x@1=2|0/I:0=0/J:14+8-2[3]

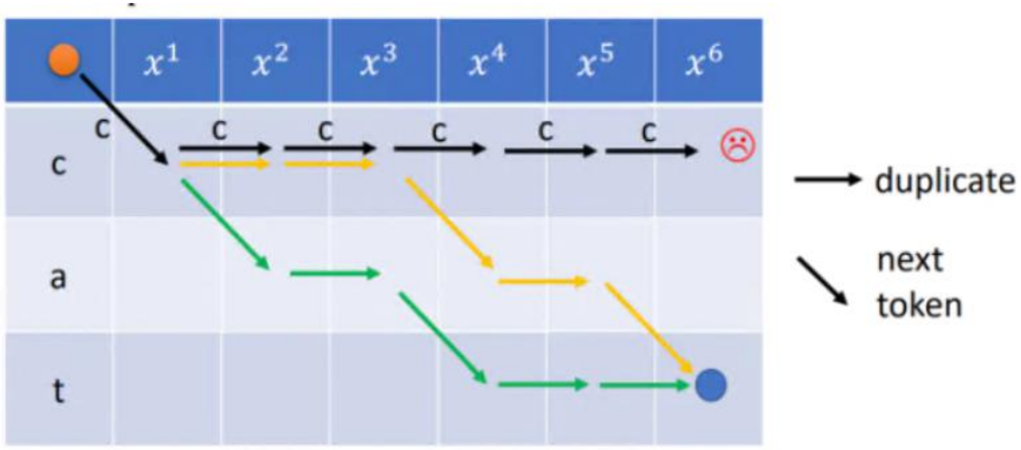
HMM alignment

HMM (Hidden Markov Model)



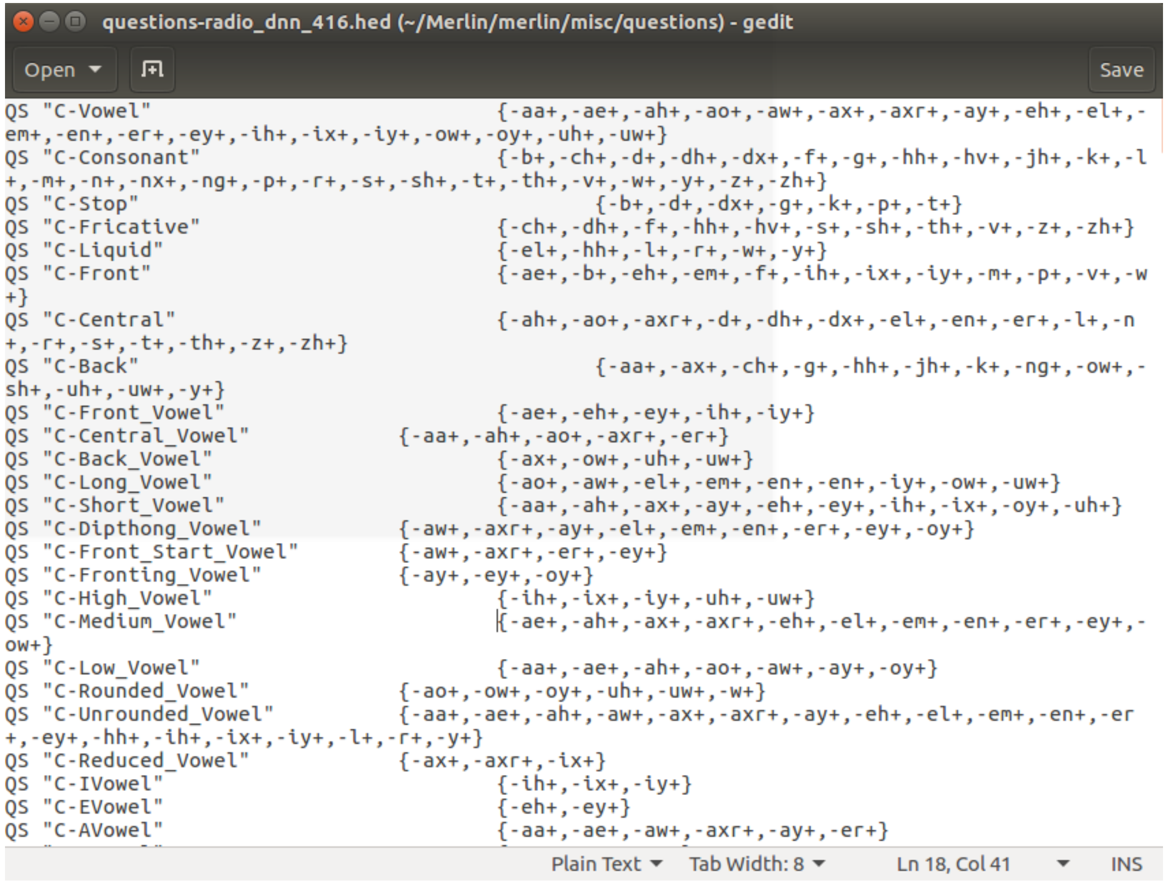
```
For n = 1 to $N$
    output the n-th token $t_n$ times
constraint: $t_1 + t_2 + \cdots t_N = T, t_n > 0$
```

一個token可以重複N次，但是所有token重複的次數和耀等於acoustic features的長度T，也就是灰色部分所描述的公式。
表中橫軸代表acoustic features，縱軸表示token。
從左上角開始走到右下角，每步只能有兩個方向：向右走或是向下走，但終點一定要走到右下角才算是合法的路徑。
從起點一直走到終點所有合法的路徑就是所有可能的alignment。



Text analysis (label to training-data)

進神經網路訓練之前，需要將標籤檔轉換成二進位檔或是向量化，也就是現在神經網路做的Embedding，在Merlin中有兩種轉換的文件，差別為生成檔案的維度不同，分別為416與600維，此文件稱為**問題集(Question file)**。
問題集針對不同的語言需要自行設計，這邊使用的是416維的問題集，也就是由416道題目所組成，內容包含判斷前後文的聲韻母為何？聲母、韻母、韻律、位置特徵劃分等等。



questions-radio_dnn_416.hed (~/Merlin/merlin/misc/questions) - gedit
Open
Save

```
QS "C-Vowel"                            {-aa+,-ae+,-ah+,-ao+,-aw+,-ax+,-axr+,-ay+,-eh+,-el+,-
em+,-en+,-er+,-ey+,-ih+,-ix+,-iy+,-ow+,-oy+,-uh+,-uw+}
QS "C-Consonant"                        {-b+,-ch+,-d+,-dh+,-dx+,-f+,-g+,-hh+,-hv+,-jh+,-k+,-l
+,-m+,-n+,-nx+,-ng+,-p+,-r+,-s+,-sh+,-t+,-th+,-v+,-w+,-y+,-z+,-zh+}
QS "C-Stop"                                     {-b+,-d+,-dx+,-g+,-k+,-p+,-t+}
QS "C-Fricative"                        {-ch+,-dh+,-f+,-hh+,-hv+,-s+,-sh+,-th+,-v+,-z+,-zh+}
QS "C-Liquid"                           {-el+,-hh+,-l+,-r+,-w+,-y+}
QS "C-Front"                            {-ae+,-b+,-eh+,-em+,-f+,-ih+,-ix+,-iy+,-m+,-p+,-v+,-w
+}
QS "C-Central"                          {-ah+,-ao+,-axr+,-d+,-dh+,-dx+,-el+,-en+,-er+,-l+,-n
+,-r+,-s+,-t+,-th+,-z+,-zh+}
QS "C-Back"                                     {-aa+,-ax+,-ch+,-g+,-hh+,-jh+,-k+,-ng+,-ow+,-
sh+,-uh+,-uw+,-y+}
QS "C-Front_Vowel"                      {-ae+,-eh+,-ey+,-ih+,-iy+}
QS "C-Central_Vowel"            {-aa+,-ah+,-ao+,-axr+,-er+}
QS "C-Back_Vowel"                       {-ax+,-ow+,-uh+,-uw+}
QS "C-Long_Vowel"                       {-ao+,-aw+,-el+,-em+,-en+,-en+,-iy+,-ow+,-uw+}
QS "C-Short_Vowel"                      {-aa+,-ah+,-ax+,-ay+,-eh+,-ey+,-ih+,-ix+,-oy+,-uh+}
QS "C-Dipthong_Vowel"           {-aw+,-axr+,-ay+,-el+,-em+,-en+,-er+,-ey+,-oy+}
QS "C-Front_Start_Vowel"        {-aw+,-axr+,-er+,-ey+}
QS "C-Fronting_Vowel"           {-ay+,-ey+,-oy+}
QS "C-High_Vowel"                       {-ih+,-ix+,-iy+,-uh+,-uw+}
QS "C-Medium_Vowel"                     {-ae+,-ah+,-ax+,-axr+,-eh+,-el+,-em+,-en+,-er+,-ey+,-
ow+}
QS "C-Low_Vowel"                        {-aa+,-ae+,-ah+,-ao+,-aw+,-ay+,-oy+}
QS "C-Rounded_Vowel"            {-ao+,-ow+,-oy+,-uh+,-uw+,-w+}
QS "C-Unrounded_Vowel"          {-aa+,-ae+,-ah+,-aw+,-ax+,-axr+,-ay+,-eh+,-el+,-em+,-en+,-er
+,-ey+,-hh+,-ih+,-ix+,-iy+,-l+,-r+,-y+}
QS "C-Reduced_Vowel"            {-ax+,-axr+,-ix+}
QS "C-IVowel"                           {-ih+,-ix+,-iy+}
QS "C-EVowel"                           {-eh+,-ey+}
QS "C-AVowel"                           {-aa+,-ae+,-aw+,-axr+,-ay+,-er+}
```

Neural Network Architecture

Input: 416 dimensions label binary file

Duration model (DNN): 4*512 (tanh)

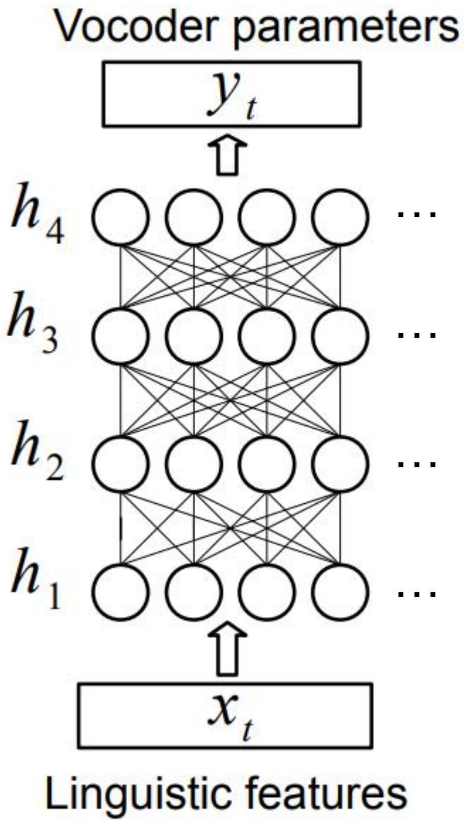
- **Output: 預測出每個音素 5 個狀態的持續時間**

- Batch size: 256
- Learning rate: 0.002
- Train file number: 50
- Valid file number: 5
- Test file number: 5

Acoustic model (DNN): 4*512 (tanh)

- **Output: mgc: 60維; bap: 1維; lf0: 1維;**

- Batch size: 64
- Learning rate : 0.002
- Train file number: 50
- Valid file number: 5
- Test file number: 5



圖一：前饋神經網路(DNN)