

模型描述

- RNN
 - 輸入參數維度：(800, 108)
 - 方便起見，先將所有序列重複直到長度為 800，不足部份以末尾的 silence 的參數補齊
 - 108 維是 fbank 的 69 維 + mfcc 的 39 維
 - 模型層
 - 雙向的 LSTM 層，輸出維度 128 維，權重以常態分佈初始化
 - Dropout 層，比例=0.4
 - 雙向的 LSTM 層，輸出維度 128 維，權重以常態分佈初始化
 - Dropout 層，比例=0.2
 - 雙向的 LSTM 層，輸出維度 128 維，權重以常態分佈初始化
 - Dropout 層，比例=0.2
 - 雙向的 LSTM 層，輸出維度 128 維，權重以常態分佈初始化
 - Dropout 層，比例=0.2
 - 全連接 (Dense) 層，輸出維度 64 維，activation 函數=relu
 - Dropout 層，比例=0.2
 - 全連接 (Dense) 層，輸出維度 39 維，activation 函數=softmax
 - 損失函數：交叉熵損失函數，優化器=adam
- CNN + RNN
 - 輸入參數維度：(800, 108)
 - 同 RNN
 - 模型層
 - 一維卷積層，卷積核 64 個，時域窗長度=11，權重以 He 常態分佈初始化
 - Dropout 層，比例=0.2
 - 一維卷積層，卷積核 32 個，時域窗長度=11，權重以 He 常態分佈初始化
 - 雙向的 LSTM 層，輸出維度 128 維，權重以常態分佈初始化
 - Dropout 層，比例=0.4
 - 雙向的 LSTM 層，輸出維度 128 維，權重以常態分佈初始化
 - Dropout 層，比例=0.2
 - 雙向的 LSTM 層，輸出維度 128 維，權重以常態分佈初始化
 - Dropout 層，比例=0.2
 - 雙向的 LSTM 層，輸出維度 128 維，權重以常態分佈初始化
 - Dropout 層，比例=0.2
 - 全連接 (Dense) 層，輸出維度 64 維，activation 函數=relu
 - Dropout 層，比例=0.2
 - 全連接 (Dense) 層，輸出維度 39 維，activation 函數=softmax
 - 損失函數：交叉熵損失函數，優化器=adam

如何提升性能

- 描述模型與技巧

- 在 trimming 的時候，一次考慮一個寬度為 5 的時域窗，若該時域窗中有 3 個以上的 frame 被預測為同個 phone 才輸出該 phone，最後再將連續的 phone 刪除。
- 在得出最佳模型時，使用 3 個不同的模型（分別為 RNN, CNN+RNN, CNN+RNN+CNN），取這 3 個模型中每個 phone 被預測為不同分類的機率的乘積來做判斷。
- 使用的原因
 - 一開始在預測的時候，因為模型預測的準確率不佳，因此使用這方法過濾一些預測錯誤的雜訊。隨著模型的準確率愈來愈高，此方法依舊有效，只是時域窗的寬度和門檻也要隨之調整。

(寬度, 門檻) \ 準確率	60%	85%	95%
(2, 2) (naive)	-	-	7.33734
(5, 3)	21.99759	9.60240	7.18072
(7, 3)	22.31084	9.67710	7.28433
(7, 4)	15.60000	11.06265	8.71566

- 這是參考 Ensemble 的作法。因為使用這方法已經是 deadline 當天，所以沒有真的去做 Ensemble (Boosting / Bagging)，而是直接拿 3 個不同模型的預測機率共同預測結果，但是也有小幅提升。

Model	RNN	CNN+RNN	CNN+RNN+CNN	合併前兩者	合併三者
Edit Distance	7.90602	8.82409	10.63132	7.26746	7.18072

實驗結果與設定

- 比較並分析 RNN 和 CNN 的結果
-
- 比較並分析與其他模型的結果