MLDS HW1 Report

學號：R05921012　姓名：吳宗澤

最後我選擇使用 RNN 模型是由 Keras 所架成的，Masking (mask_value = 0.) -> Bidirectional (GRU (128)) -> Bidirectional (GRU (128)) -> TimeDistributed (Dense(512)) -> TimeDistributed (Dense (256)) -> TimeDistributed (Dense(48))，並利用 Adam 去做更新。CNN 模型則是在相同的 RNN 模型前加入 Conv1D (64, kernel_size=3, stride=1, padding='same') -> Conv1D (32 , kernel_size=3, stride=1, padding = 'same')，也是同樣利用 Adam 去做更新的。

一開始我在使用 RNN 是使用 LSTM 去實作並利用 mfcc 當作 feature input 去做輸入，輸入後也要將每個 sequence 自己做 padding 到一樣長，但我發現使用 LSTM 接 Dense 的話，validation accuracy 上升的速度很慢，大概 60~70 個 epoches 才會到準確率 70，而在考量資料 frame-wise prediction 這種特性後，應該把 Bidirectional 也放進去因為 phone 從前往後推和從後往前推應該是有關係的，而果然放進去以後大概 30 個 epoches 就可以到 70%的準確率，而在改成 GRU 以後效果也較原本的 LSTM 好，當然為了避免 overfitting，我在各個 Layer 層也有加入 dropout 和 batchnormalization 去讓我的 model 更好。

除此之外，我發現我在把 fbank 的 feature 丟進 RNN 的時候效果是不太好的，比用 mfcc 的 feature 還差些。而在考慮加入 CNN 時，我覺得每個時刻的資訊都很重要所以並沒有用 maxpooling 將資訊量縮小，這個動作適用於影像但不適用於這裡。而在加入 CNN 之後 mfcc 訓練出來的 model 幾乎沒有什麼長進，但在 fbank 與 CNN 結合之後訓練出來的 model 竟然 edit distance 足足少了 1 點多，這樣的效果與我們之前在處理頻率域的頻譜的時候常常用到影像處理的方法一樣，因為 CNN 也算是一種影像 filter 的方法並取出高維的重要的特徵出來。

我曾經有嘗試過幾種方法去做幾種將 mfcc 和 fbank 做結果 fusing 的方法，但因為如果我將兩者同時讀進去 model 並作處理時，電腦的 ram 有點不太夠去做處理，因此，我選擇將兩個 models(一個是 mfcc 在單獨用 RNN 時的 model，另一個則是 fbank 用在 CNN+RNN 時的 model)的 output predictions 放進 Random forest 去做最後的 ensemble，但因為資料量太大又樹的數量又會讓 sklearn 的 random

forest 消耗的 ram 線性成長，後來我是改用把他們的 predictions 當成新的資料丟進另一個只有 fully connected layers 的 model 做訓練，並 output 出新的 48 維結果出來，這樣 fusion 的結果其實也讓我的 edit distance 有所進步。而若將兩個 model 的 output 直接取平均效果沒有上一種的結果好。

最後我用了許多之前訓練出來比較好的 mode 們去 predict 我們的 testing 資料，然後將將所有的結果取平均做 ensemble 才能得到我在 kaggle 目前最佳的成績。

而在截止時間之後，我稍微觀察了一下 output 的結果和想到這是屬於 frame-wise 的 prediction，因此如果在我們 prediction 出來的 frame 裡面若是他沒有在鄰近重複的出現(例如：aaabaaccc)，b 在這邊就屬於一種 prediction 的 noise，因此我試著 filter 掉這樣的預測，若是重複的長度沒有大於 1、2、3、4 等等，發現刪掉小於三的長度的預測，可以讓 edit distance 變成只有 7.5 多。

總結：Performance : multi-model ensemble prediction with filtering output (length : 3) > multi-model ensemble prediction with filtering output (length : 2) > multi-model ensemble prediction with filtering output (length : 1) > multi-model ensemble prediction with filtering output (length : 4) > multi-model ensemble prediction > mfcc-rnn-fbank-cnnrnn-fusing prediction with fully connected layers> fbank-cnnrnn-fusing prediction > mfcc-rnn prediction =(almost the same) mfcc-rnncnn prediction > fbank-rnn prediction