1. Model description (2%) :
    a. RNN (1%) :

主結構：(Sequential)
Masking(mask_value=0)
Bidirectional LSTM(216, dropout=0.2, activation='tanh')
Bidirectional LSTM(216, dropout=0.2, activation='tanh')
LSTM(48, dropout=0.2, activation='softmax')))
input data：
concat(mfcc, fbank, axis=1), dim = 39 + 69 = 108
output class:
48 phone label

    b. RNN+CNN (1%) :

主結構：
cnn_1:Conv2D(32, (1, 54), padding='valid', activation='tanh')
MaxPooling2D((1,2))
TimeDistributed(Flatten())
cnn_2:Conv2D(16, (1, 8), padding='valid', activation='tanh')
MaxPooling2D((1,2))
Conv2D(16, (1, 8), padding='valid', activation='tanh')
MaxPooling2D((1,2))
Conv2D(16, (1, 8), padding='valid', activation='tanh')
MaxPooling2D((1,2))
TimeDistributed(Flatten())
rnn: Merge([cnn_1, cnn_2], mode='concat'))
Bidirectional LSTM(432, dropout=0.2)
LSTM(49, dropout=0.2, activation='softmax')
input data：
concat(mfcc, fbank, axis=1), dim = 39 + 69 = 108
output class:
48 phone label + 1 zero padding label

2. How to improve your performance (1%)：
   a. RNN：
      i. 增加RNN深度。
      WHY：有助於model習得更高層次的時序變化，在連音時減少輸出sequence的零碎轉換。
      ii. 使用Bidirectional LSTM。
      WHY：有助於model使用前後資料決定phone label，類似target delay。
   b. CNN + RNN：
      i. 結合不同kernal size的cnn作為input feature detector。
      WHY：將time與feature視為2D圖面，利用小而深及大而淺的kernal對feature維度抓取不同pattern。
      ii. 選用tanh作為cnn activation function。
      WHY：有避免輸出數值大於1造成LSTM飽和，tanh由-1至+1可幫助model收斂。
   c. General:

preprocessing：

1. 針對每一個feature做normalization。
2. 將所有資料（每個檔案都頭尾串再一起成一個大的連續資料）以timestep=200的window滑行切成sample，overlap約20%~33%。
3. 擴充trainning set，參考Vocal Tract Length Perturbation: A frequency warping approach to speaker normalization演算法。
   WHY：利用演算法模擬不同說話者的口腔共振結構，隨機將原始數據頻率及強度做轉換，以產生類似不同口腔結構的摹擬數據。
   Improvement: Kaggle Public Score 9.23163 -> 8.47457

$$G(f) = \begin{cases} \alpha f, & 0 \le f \le f_0 \\ \frac{f_{\max} - \alpha f_0}{f_{\max} - f_0}(f - f_0) + \alpha f_0, & f_0 \le f \le f_{\max} \end{cases}$$

ref: http://ieeexplore.ieee.org/document/650310/?tp=&arnumber=650310&url=http:%2F%2Fieeexplore.ieee.org%2Fiel4%2F89%2F14168%2F00650310

train：

1. timestep = 200, batch size = 64 (for cnn), 160 (for rnn)。
2. optimizer = Adam, clipvalue一開始設定0.5，當val_acc停滯 (~10 epoch) 時調至0.01，再停滯時調整至0.005，完成training。

predict：

1. 將test data依Instance ID分割為timestep上限780的sample，不足者補上silence data (phone ‘sil’ 之data)。

3. Experimental results and settings (1%)：

RNN v.s. CNN：

理論上CNN可以分析出一段時間中頻域上的pattern，接著搭配RNN等結構儲存時域上的特徵，但實作中發現單用一層淺的CNN或是多層CNN都無法幫助後續的RNN增進判斷準確率。唯有在同時使用大而淺及小而深的CNN於RNN之前才能接近與直接使用多一層RNN（BiLSTM）的準確率。實驗中尚未做出優於純RNN的model，或許是數據的處理不夠完善讓CNN辨識出可靠的feature。

Model \ Score	val_acc (%)	Kaggle Public Score
CNNs + BiLSTM	76.42	10.96045
BiLSTM + BiLSTM	77.7	9.23163

此表中的兩個model皆未使用Vocal Tract Length Perturbation演算法擴充原始數據。

Other Model：

利用Keras Merge Layer的擴充性，嘗試整合寬淺CNN、窄深CNN及Bi-LSTM三種前端的Hybrid Model，後端則是Bi-LSTM及LSTM。

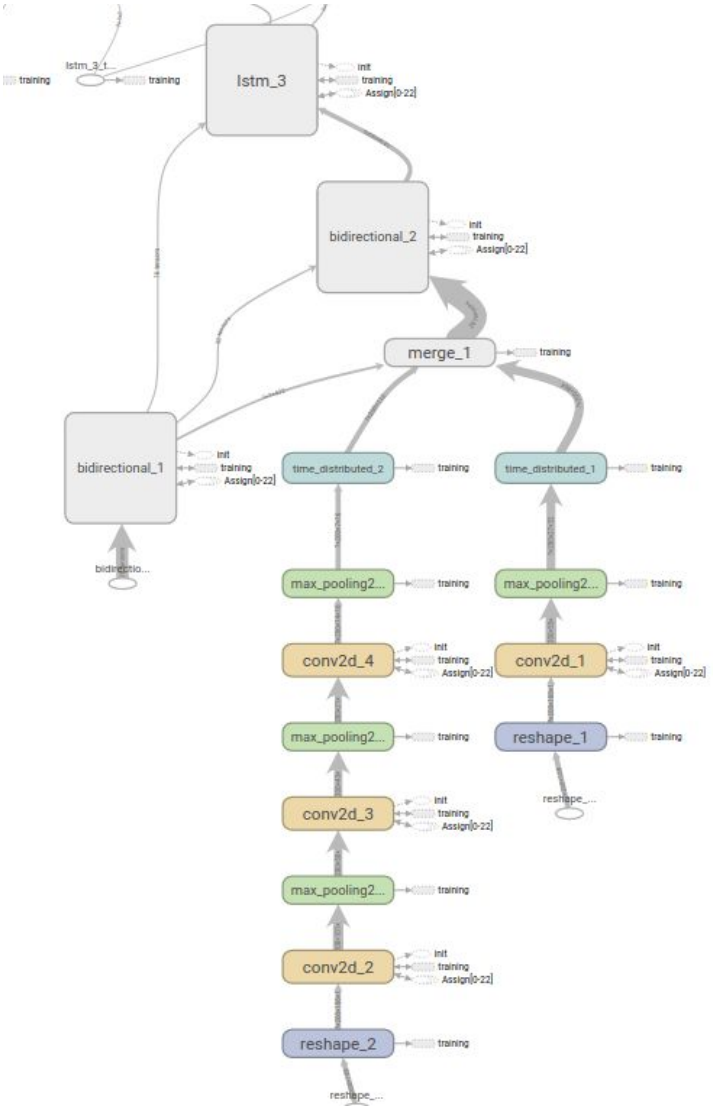
設計目標：結合CNN及RNN前端以不同方式處理資料，接著輸出復合的feature vector由後端的Bi-LSTM記憶並整合。

實驗結果：

val_acc： 77%

Kaggle Public Score：9.79096

#未使用Vocal Tract Length Perturbation演算法，可與上表一同比較。



Best Model:

決定選用deep RNN + Vocal Tract Length Perturbation演算法擴充原始數據（擴充4倍）。

ADLxMLDS HW1 Report
R05942027 陳重源 2017/10/28