MLDS HW1: TIMIT

學號：B04901066 系級：電機三 姓名：洪國嘵

1.Model description (2%)

皆使用keras實作，input=(3696, 777, 108)，
output=(3696, 777, 48)(activation='softmax')，每層間都加 Dropout(0.2)
# model Score = (Private, Public)(皆使用output filter，請參考"如何改善表現")
---RNN (1%)

使用兩層hidden_layer為1024的LSTM (14.62409, 14.50282)

best同樣是RNN，使用三層hidden_layer=2048之LSTM (9.83132, 10.06214)
---RNN+CNN (1%)

在兩層LSTM前再加一hidden_layer為1024的Conv層 (12.98072, 13.18644)

2.How to improve your performance (1%)
---Write down the method that makes you outstanding
---Describe the model or technique (0.5%)
---Why do you use it (0.5%)
觀察輸出結果，發現有些相似的語音可能會判斷錯誤，舉例來說：
'aaaaabbbbLLLvvvv' 變成 'aaaeadbbbLLLgvvv'
此時若直接依照規定priming，error就會高達4，因此我決定在最後model輸出的字串再加上一個filter，若是連續次數太少(e.g. <3)便直接刪除，因為根據train.lab，通常一個音都會連續出現5次(最少的為三次而且很少見，兩次或一次屈指可數)，並且刪除。雖然可能增加deletion的loss，卻可大幅改善insertion和substitution。使用某個model直接輸出在public上成績可能是25，加了filter便能來到10幾分，影響程度十分劇烈。

3.Experimental results and settings (1%)
---Compare and analyze the results between RNN and CNN (0.5%)
---Compare and analyze the results with other models (0.5%)
---other models can be variant of basic RNN, like LSTM, or some novel ideas you use
CNN相當於在時間維度或feature維度加上filter，因為一聲音段所代表的音理論上是連續的，因此CNN對於處理掉雜訊應是有相當程度的幫助，然而因為我採用的model參數較多，所以變成RNN和CNN之間預測準度差不多(CNN稍微較好)。
因此我就對第二題 improve performance的參數來做實驗，設定repeat filter，即連續出現n次才將那個phone輸出，使用完全相同之model、參數、設定，僅n不同，得到如下結果：

repeat >= n可通過	1	2	3
Private Score	26.56144	16.40240	13.48674

顯見多花幾分鐘對output再做一些手腳，就能對結果有劇烈變化，我發現這項特色後便和同學討論，大家一致發現不論何種model皆是repeat取3有最好結果(取4準度又會變差)，因此這幾乎可以當作最後一個必加的filter。