報告製作：鄭皓謙
學號:r06922115

TIMIT dataset 分析過程：

作業可以選擇mfcc或是fbank資料集，我選擇了mfcc，配合找到的mfcc wav2features library方便我以後再深入了解speech recognition。

這次作業要的是phonemes的輸出，而不是frame wise的一對一輸出，原本我想嘗試接兩層RNN做auto-decoder-encoder，不過可能是模型太長而且中間我並沒有另外做masking，效果不好，所以最後採用的模型是 one to one, frame wise output，然後再另外做triming。

考慮給的資料輸入每個 time step frame有重疊的資訊，以及在features dim男女音可能在features dim上有shift offset的對應關係，先用CNN在兩個軸上做特徵擷取，經過flatten後接上RNN做一對一的輸出，最後做48-39 mapping跟triming。

資料被我padding到最長777長度，所以每個sample後面都會接一長串的padding 0，考慮到資料的平均長度，每筆資料通常有一半是padding的值，後面會提到如何處理的。

由於padding的關係，所以用keras內建的accuracy metric不能夠顯示有效資料長度內的準確率，所以我在loss.py裡面另外寫了loss_with_mask以及acc_with_mask。

RNN模型：

我第一個過baseline的模型就是用兩層LSTM直接做一對一輸出，前面幾次嘗試時，想省時間，以為句子會有間隔sil的段落，所以用SimpleRNN直接做應該也可以得到相近的表現，實則不然，浪費了很多時間。

這個應用非常適合使用bidirectional RNN的架構，句子中phoenme與phoneme的連續關係，甚至是句子與句子間，在人類發聲系統有前後關聯，那麼RNN架構就很適合再在這個場合應用。mfcc的理想是取出人類可以感知的幾項特徵，假設這些特徵是人類辨識語音的關鍵要素，那麼以上假設就應該成立。

如果沒有另外切句子的話，每個setence間的距離其實足夠長到讓LSTM的優勢展現出來，使用bidirectional RNN的影響更深，所以不用想用純RNN做了。

CNN模型

前面提到RNN可以處理phonme與phoneme之間的順序關係，CNN可以更專注地處理mfcc提取特徵時，時間維度上的相近特性，以及不同人發聲在特徵維度上的對應關係，例如男女發音頻域差別。

我選用的kernel size分別是(1,5)及(3,5)，考慮到希望加強特徵維度上的範圍，而時間軸上的關係則希望在RNN部份能夠做好，最後輸出的時候會做triming，所以strides直接設為1，並沒有特別強調時間軸或是特徵軸上的跨度特性。

雖然最後輸出會做triming，也許意味著只取一小段聲音訊號突出的部分，更能找出明顯的特徵，但是模型還是沒有加上Pooling，我認為為了讓之後的RNN能夠做到更好的一對一對應，選擇不做pooling，希望在時間軸上一對一的關係更明確。而在mfcc中的特徵描述，我沒有觀察出明顯subsampling的特性，所以在特徵軸上也沒有加上pooling。

RNN與CNN表現差別

當RNN前面接上CNN以後，確實表現上有提升，我理解成CNN處理了更近的關係，當模型只看最近5個frame或是10個frame時，在直覺上是能夠判斷的更準確，配合上RNN，也許就能夠根據句子或是單字，而不僅僅是phonme的前後幾個phoneme，去做更精準的預測。

模型實作過程：

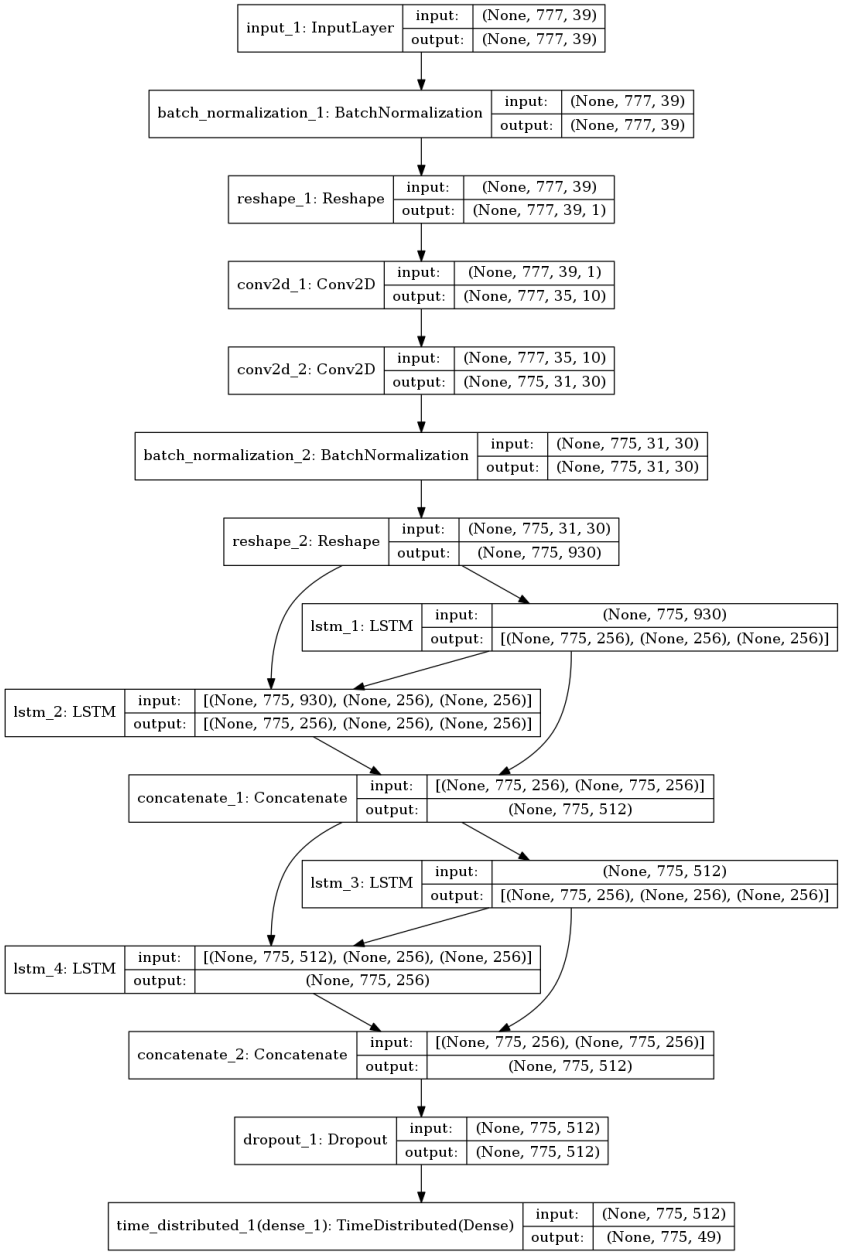
之前嘗試非固定長度輸出時，用categorical cross entropy，並沒有特別處理padding的部分，結果輸出都是sil以及padding symbol，儘管在第一層加上Masking，直接用RNN效果也沒有比另外做triming好，所以改用一對一模型。

在一對一模型的途中，我發現因為padding資料的關係，大幅影響初期模型gradient decent的方向，這個可以由定義有做masking的loss function去避免，以及配合sample weight去調整。我將sil的權重調到0.2而padding value的權重則調整為0.01。

這次dataset並沒有觀察到明顯subsampling的性質，所以CNN後面並不會接pooling layer，而為了加強在time step上的關係，使用了bidirectional 的LSTM。

兩層CNN —> 兩層RNN —>softmax dense

CNN kernel選用的size是(1,5)跟(3,5)，希望由LSTM去處理time step上關係。



試過的技巧

除了bidirectional，我嘗試過讓RNN模型加入Mask層，不傳入padding的部分，RNN matrix僅僅有效長度內的資料做改進，在純RNN的環境有一些可以觀察到的改進，但是當接上CNN時，並沒有辦法排除掉padding的部分，由於CNN層裡面加了bias。

當我不希望RNN被那些padding的值影響時，我嘗試過不在CNN加bias，但是表現沒有變好，我猜測是mfcc進來的資料分佈，無法被經過原點的線逼近，也就是CNN學不好，那麼就沒有必要加入CNN；另一個想法是，我先不對padding的部分算loss，那麼這樣模型開始學不好padding的部分，不過因為是一對一模型，所以padding的部分放棄沒關係，所以我改用了有做masking的categorical cross entropy。

最後我希望的目標是能夠做到seq to seq的模型，所以儘管是padding的部分也應該要學到，雖然這次沒有做出來，但為了改進的空間，我嘗試了另一個作法：先排除padding的loss做訓練，第二次一樣的模型，載入先前訓練的各個weights,bias，再加入padding的loss做訓練。如此作法是希望能夠避免過早掉入padding所造成的local minimum，確實有改進overfitting的部分，最後模型用的方法是改變每個sample weight。

我另外在mfcc資料上以及做完CNN後，加上了normalization。

另外嘗試過將RNN的最後一個輸出當作一個總結，再用最後這個總結做decoder，也就是auto-encoder-decoder，不過架得不好，所以沒有採用。

還有不做RNN，直接兩層的Dense做sigmoid，最後softmax，這個模型很快，在考慮padding的準確率只有50–60%，我有看到許多應用其實是用這種快速的模型去做前面的處理，我猜想是在應用上，後面的處理能夠彌補如此低的一對一模型準確率。

可以改進的地方

這個模型在跑training時容易跑到overfitting，由於我不夠清楚dropout該加在哪些地方，又或是架構過於複雜，也是可以改進的部分。

loss function我是寫了一個有masking的版本，可以只對有效的長度做training，不過我來不及train出足夠好的模型以及後面做triming的判斷改寫。只透過調整sample weight的方式讓模型盡量在初期不要踩到padding所造成的淺洞。如果是輸出不定長度的模型，我想配合CTC loss可以做得更好。

目前輸出做triming的部分，是將重複phoneme沒超過長度3的當作錯誤判斷去掉，我覺得可以另外練一個模型去學習更廣泛的分佈關係。3這個閾值是根據直覺設定的，並沒有研究資料集裡面的統計特性。

RNN另外把hidden state接出來，原本打算做encoder decoder，不過失敗了，考慮之後加上masking的可能，所以沒用keras的bidirectional layer warper包。

成績最好的那次輸入來不及放上kaggle　9.959

Submission and Description	Private Score	Public Score	U
final.csv 12 minutes ago by Payo add submission details	11.30361	11.80225	
b5.csv 4 hours ago by Payo cnn\gpu	9.95903	10.30508	