r04325008

資料處理好之後，有 3696 個句子，最長的 trainning 句子有 777 個 frame，所以我先把所有的句子 padding 到 777，之後再使用 mask 讓 cost function 不計算 padding 的 cost。Target 先處理成 39 個類別之後才放進模型裡。

Model Description

1. RNN
   我使用 keras 的 sequential model。第一層先使用 masking，讓之後的 cost function 可以忽略輸入全是 0 的 frame。接著使一層的 Bidirectional LSTM。Bidirectional LSTM 後面接一層的 LSTM。在這兩個 RNN 後都有接 batch normalization，目的是為了要讓最佳化的更好。最後在 RNN 後面接上一些 fully connected layers，並使用 relu 作為 activation。最後一層的 activation 使用 softmax，讓模型可以預測機率。模型建後好以後，使用 categorical cross entropy 計算 cost，並使用 adam 解最佳化問題。
2. CNN
   CNN model 的前處理以 RNN 資料的前處理為基礎，再將每一個 frame 和其前後兩個 frame concatenate 在一起，形成一個維度為（3, 39）的 frame。第一個 frame 和最後一個 frame 則在左右個補上 39 維的 0。
   RNN + CNN 的 model 與 RNN 相似。但在 masking 之前加入了 ConvLSTM2D。這個方法可能沒有辦法很正確的做到 masking，因為 padding 的 input 在經過 convLSTM2D 之後不會全為 0。但 keras 的 ConvLSTM2D 不支援 masking，所以沒有辦法將資料傳進 model 後馬上做 masking。經過 Convolution 之後，將結果攤平，接著放入兩層 LSTM，其後都接著 batch normalization，最後接上 fully connected 做更多非線性轉換。一樣使用 softmax 轉換成機率，並用 categorical cross entropy 計算 cost，使用 adam 解最佳化問題。

How to Imporve Performance

1. Bidirectional LSTM 和 batch normalization
2. 使用 Bidirectional LSTM 可以讓模型同時考量聲音資料兩個方向的資訊，讓模型可以從更多的資料中抽取更多的資訊。使用 batch normalization 是因為一般的 update 在計算時都是 assuming 其他 variable constant 的情況下計算出來的，但實際上在 update 時所有的 variable 都會變動，為了解決這個問題所以使用 batch normalization，讓 update 時更可能讓 cost 下降。

Experimental results and settings

1. 我做出來的模型 rnn 是比較好的。這一方面可能是因為 cnn 的部分 masking 做的不是很正確（但應該還是可以學習到的，因為 target 是 0 的話，cost 還

是會是 0，所以 cost function 的錯誤應該只是一個常數項的錯誤，update 的方向應該還是正確的），另一方面可能是因為我只是用了前後兩個 frame，或許 frame 數再多一點效果會更好。最後因為覺得 cnn 已經很複雜了，所以只加上了上兩個普通的 LSTM，沒有使用 bidirectional LSTM，這也可能造成 cnn 的結果較差。

2.

model	accuracy
2 lstm(200 200)+desne+softmax	62.24%
2 lstm(100 100)	68%
2 lstm(100 100)+relu+dropout 0.9	64.13
2 lstm(100 100)+relu	66.16%
3 lstm(100 100 100) + relu dropout 0.9	63.22%
1 lstm(600)	51%