

1. Model description

(1) RNN:

➤ Input

- 將同個 sentence id 的 frame 組成序列, 產生 3 維的陣列(序列長度、batch 資料數、特徵維度)丟進 RNN
- 特徵維度: 70(fbank)

➤ RNN

- 3 layers
- Hidden dimension = 256
- 層與層之間加上 dropout, rate = 0.4
- Activation function(最後一層): ReLu

➤ Fully connected layer

- 經過 RNN 後, 序列中的每一個 frame 都有通過 RNN 之後的 outputs, 對於這些 outputs 一一餵進此全連接層
- input dimension = 512 (256*2)
- output dimension = 256
- activation function: ReLu

➤ Dropout layer

- rate= 0.4

➤ Fully connected layer

- 最後將每一個 frame 的 outputs 丟進此層, 得到每一個 frame 對所有可能的 label 的分布
- input dimension = 256
- output dimension = 49

➤ Log softmax

- 透過 softmax 得到每個 frame 對各個的 label 的機率值

(2)CNN + RNN:

➤ Input

- 將同個 sentence id 的 frame 組成序列, 產生 3 維的陣列(序列長度、batch 資料數、特徵維度)丟進 RNN

➤ CNN (1 Dimensional)

- Input channels: 3 個, 對於每一個時間點的 frame, 分別取其前後一個 frame, 疊成 3 個 channel. (第一個以及最後一個 frame 則以 1 個 zero vector 做 padding).
- Output channels: 6 個
- Filter size= 5, 每次沿著 feature 維度的方向平移 1
- activation function: ReLu

➤ Max Pool

- Pool size = 2, 每次平移 1
- 最後將 features 扳平, 得到每個 frame 經過 convolution 後的新特徵值

➤ Dropout layer

- Rate = 0.4

➤ RNN

- 3 layers
- Hidden dimension = 256
- 層與層之間加上 dropout, rate = 0.4
- Activation function on Last layer: ReLu

➤ Fully connected layer

- 經過 RNN 後, 序列中的每一個 frame 都有通過 RNN 之後的 outputs, 對於這些 outputs 一一餵進此全連接層
- input dimension=512 (256*2)
- output dimension=256
- activation function: ReLu

➤ Dropout layer

- rate= 0.4

➤ Fully connected layer

- 最後將每一個 frame 的 outputs 丟進此層, 得到每一個 frame 對所有可能的 label 的分布
- input dimension = 256
- output dimension =49

➤ Log softmax

- 透過 softmax 得到每個 frame 對各個的 label 的機率值

2. Methods to improve performance

- I. 正規化特徵值：將數值減去平均再除以標準差，將數值控制在一定的範圍內，原本沒有正規化之前，不同維度的特徵值分布的範圍不太一樣，導致學習的效果不太好，經過正規化之後可有效改善表現。
- II. 將 RNN 改為 LSTM：資料集中有些句子的 frame 數還滿多的，因此覺得 frame 間的 long-term dependency 應該會影響預測，而 LSTM 比 RNN 更能捕捉到這種關係，推測對於預測的準確應有所幫助。
- III. 使用雙向的 LSTM：原先只有使用單向的 LSTM，表現不是很好，因為感覺一個音的辨識應該跟前後的音都有關係，故改用雙向的 LSTM，結果也有所改善。
- IV. 加上 dropout layer 避免 overfitting：訓練的時候發現訓練資料集的正確率很高，然而驗證資料集的正確率卻不太好，因此推測有可能是 overfitting，故加上 dropout layer，經過幾次的實驗後覺得 dropout rate 設成 0.4 的效果較好。

3. Experimental results and settings

- I. 比較 RNN 與 CNN：
 - i. 若只有 RNN(如第 1 題(1)中描述的)，在 kaggle 上表現大概是 12.29，
 - ii. 加了一層 CNN 之後，若 channel 數設成 1，表現大概是 12.27，進步不太多
 - iii. 加了一層 CNN 並且將 channel 數設成 3，也就是考慮前後各一個的 frame 之後，進步較多，在 kaggle 上的表現大概是 10.2，推測是因為可以捕捉到上下文的 local 的特徵，因此可以改善表現。
- II. 比較 RNN 與 LSTM：
 - i. 在一樣的參數之下，RNN 收斂的速度比 LSTM 慢，當 LSTM 在訓練資料集的正確率已到達 85%上下時，RNN 的正確率還在 73%上下，且最後在測試資料集的表現也比較不好。