ADLxMLDS HW1 Report

B03902086 資工四 李鈺昇

Model description

1. RNN

設計大致上如投影片，用的是 LSTM cell，其他參數如下：
4 層 LSTM（dropout = 0.2），
每層 500 neurons，
最後再過一層 fully-connected 並用 softmax 輸出 48 個 phoneme 的機率。

2. CNN+RNN

設計大致上如投影片，用的是 LSTM cell，其他參數如下：
先過一層 Conv1D（kernel_size = 5，32 filters，strides = 1，activation = ReLU），
接著 2 層 LSTM（dropout = 0.2），
每層 300 neurons，
最後再過一層 fully-connected 並用 softmax 輸出 48 個 phoneme 的機率。
（kernel_size = 5，32 filters 是調出來最好的設定）

3. 共同點

有了機率就可以用 argmax 找到最大值作為答案。
使用的 optimizer 兩個都一樣是 Adam，learning rate = 0.0005

Improving the performance

- 首先，因為一開始我都只嘗試一兩層 LSTM，然後努力調參數；後來經高人指點後發現四層、加上加大每層的 neurons 數量就可以做到非常高分，因此我也這麼嘗試，果真 edit distance 低了許多。缺點就是 training time 變得非常久，原本一個 epoch 大概五分鐘以內，變成需要一兩個小時，等得比較煎熬。但這樣就可以直接過 baseline！

- 另一個改善的地方是，原本我就直接天真地把預測出來相鄰且重複的 phoneme 移除，後來發現這樣子可能會發生這種情況：如果原本是 aaabaaacccc，那麼原先的做法會變成 abac，但是仔細想想之後發現 b 可能只是 noise，所以如果能夠變成 ac 的話應該會比較棒。

  所以後來的做法是：對於每個位置，也看看他左右的各 3 個，把這樣 7 個取眾數之後輸出（如果找眾數的時候遇到一樣多的話就拿到目前為止出現最多次的）。這個作法可以把原本的提高大概 2~3%。
- 最後是要不要先 map 到 39 的問題。經過實驗，發現 predict 完之後再 map 會比較好一點點。

Experimental results and settings

Comparison of RNN with CNN+RNN
我做的 RNN 大概最好可以到 7、8 左右，但是 CNN+RNN 只能做到 9、10左右，有可能是把 CNN 配上 RNN 之後 會改變 RNN 原有的特性，也可能是因為 CNN+RNN 我的 RNN 層數只有試到兩層而非四層。

Packages used

arrow==0.10.0
h5py==2.7.1
keras==2.0.8
numpy==1.13.3
tensorflow-gpu==1.3.0