

Machine Learning HW5 Report

學號：R07946013 系級：資料科學碩二 姓名：吳泓毅

1. (1%) 請說明你實作之 RNN 模型架構及使用的 word embedding 方法，回報模型的正確率並繪出訓練曲線*

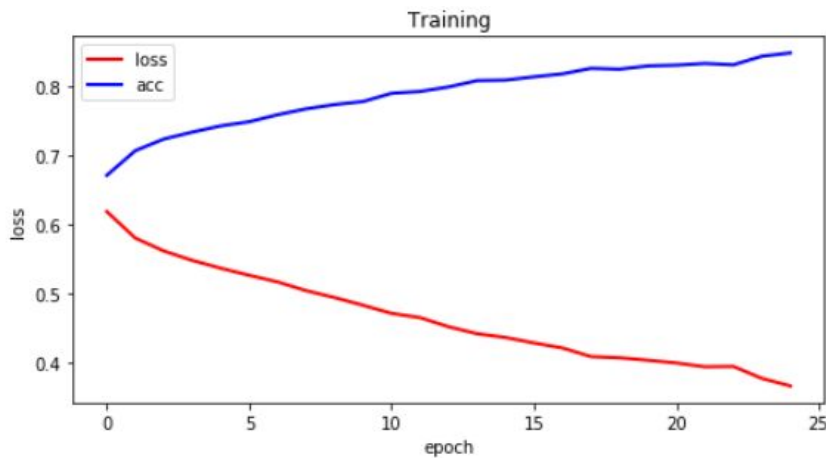
RNN 模型: GRU

與LSTM不同的是，GRU用了兩個 gate: reset gate 和 update gate, reset gate 用來控制現在的 input 和之前的 hidden state進而更新現在的 hidden state, 而 reset gate 則是用來在剛剛得到的 hidden state 和之前的 hidden state 之間取加權平均。

Word embedding 方法: word to vector

Word to vector 簡單來說就是用現在看到的字去預測旁邊的字，所以會把一個文本變成(input_word, output_word)的形式，然後把one-hot encoding 的字 mapping 到一個 low dimensional space，而在low dimensional space我們可以計算字跟字之間的相似度當作預測值，每個字跟字的相似度在經由softmax變成 distribution，而loss function 就是資料中的 empirical distribution 跟 predicted distribution 的 cross entropy。

Kaggle Accuracy: 0.79302



2. (1%) 請實作 BOW+DNN 模型，敘述你的模型架構，回報模型的正確率並繪出訓練曲線*。

3. (1%) 請敘述你如何 improve performance (preprocess, embedding, 架構等), 並解釋為何這些做法可以使模型進步。

Embedding:

Iter \rightarrow 50

Negative sampling: 5 \rightarrow 13

Skip-gram rather than BOW

提高Negative sampling 也讓結果變得更好, 猜測是因為文本比較小, 有些字比較不容易被更新到, 所以提高 negative sampling 可以使這些字更容易被更新到, 而原本高頻的字也不會被影響太多。

In nn structure:

LSTM \rightarrow GRU

(Hidden_layers, hidden_size): (2,15) \rightarrow (4,30)

增加model 的複雜度。

4. (1%) 請比較不做斷詞 (e.g.,用空白分開) 與有做斷詞, 兩種方法實作出來的效果差異, 並解釋為何有此差別。

原本沒做segmentation 最好的結果到0.75813, 做了segmentation後就到了0.77209。

當文字中出現縮寫或表點符號時, segmentation會把標點符號前的字分開, 例如 "I've" 會變成 "I" 跟 "'ve"比較合乎我們平常的用法, 而且也不會再產生原本不存在的字。

5. (1%) 請比較 RNN 與 BOW 兩種不同 model 對於 "Today is hot, but I am happy." 與 "I am happy, but today is hot." 這兩句話的分數 (model output), 並討論造成差異的原因。

```
1. X=array([[ 0,  1,  0,  3],
            [ 1,  0,  1, -2],
            [ 1,  1,  1,  4],
            [ 0,  1,  1,  0],
            [ 0,  1,  0,  2],
            [ 0,  0,  1, -4],
            [ 1,  1,  1,  1],
            [ 1,  0,  1,  2]])
```

Compute (z, zi, zf,zo):

```
array([[ 3, -2,  4,  0,  2, -4,  1,  2],
       [ 90,  90, 190,  90,  90, -10, 190,  90],
       [  0,  0, -100,  0,  0, 100, -100,  0],
       [ 590,  590,  590,  590,  590,  590,  590,  590]])
```

Iteratively compute c and y:

```
[(0, 2.8577223804673),
 (2.8577223804673, 1.190455346189415),
 (1.190455346189415, 3.928055160151634),
 (3.928055160151634, 1.964027580075817),
 (1.964027580075817, 2.743607945993673),
 (2.743607945993673, 2.671663106145307),
 (2.671663106145307, 0.7310585786300049),
 (0.7310585786300049, 2.127123445270767)]
```

2.

$$\begin{aligned}
L &= \sum_{c \in C} -\log \frac{\exp(u_c)}{\sum_k \exp(u_k)} \\
&= \sum_{c \in C} -e_c^T W'^T e_i e_j^T \omega_{input} + |c| \log \sum_k \exp(e_v^T W'^T W^T \omega_{input}) \\
\frac{\partial L}{\partial W_{ij}^T} &= \sum_{c \in C} -e_c^T W'^T e_i e_j^T \omega_{input} + |c| \frac{\sum_v \exp(e_v^T W'^T W^T \omega_{input}) e_v^T W'^T e_i e_j^T \omega_{input}}{\sum_v \exp(e_v^T W'^T W^T \omega_{input})} \\
&= \left(\sum_{c \in C} -w'_{ic} \right) e_j^T \omega_{input} + |c| \frac{\sum_v \exp(e_v^T W'^T W^T \omega_{input}) w_{iv}}{\sum_v \exp(e_v^T W'^T W^T \omega_{input})} e_j \omega_{input} \\
\frac{\partial L}{\partial W_{ij}^T} &= -e_j^T W^T \omega_{input} + |c| \frac{\sum_v \exp(e_v^T W'^T W^T \omega_{input})}{\sum_v \exp(e_v^T W'^T W^T \omega_{input})} e_j^T W^T \omega_{input}
\end{aligned}$$