Machine Learning HW6 Report

學號：b06901007　系級：電機二　姓名：戴子宜

2019.5.8

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。

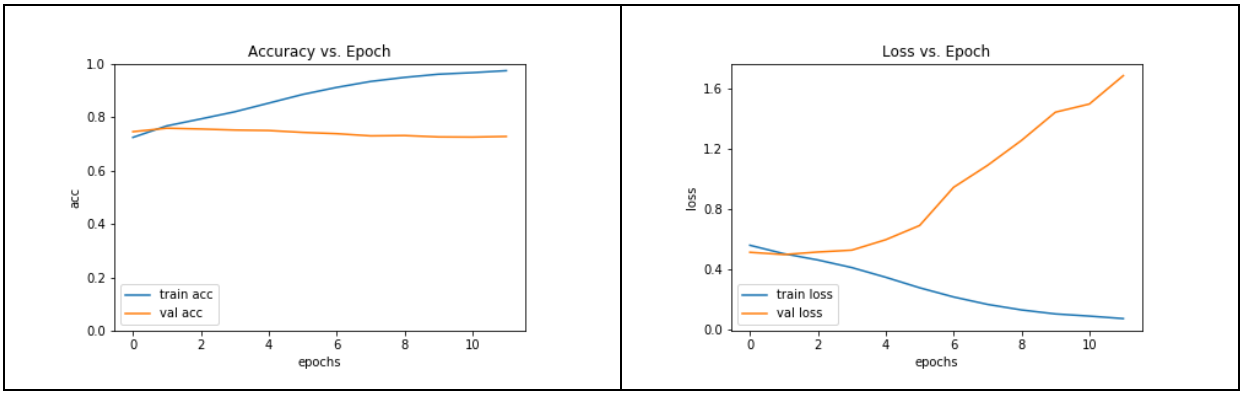
   (a) RNN 模型架構：

   Word Embedding layer（原本 Word2Vec 的 model 的 pretrained weights）
   → Bidirectional LSTM layer（共 128 units，回傳的是 sequence）
   → Bidirectional LSTM layer（共 128 units）
   → Linear Layer（ReLU 為 activation，共 128 neurons）→ Dropout（機率 0.7）
   → Linear Layer（ReLU 為 activation，共 64 neurons）→ Dropout（機率 0.7）
   → Linear Layer（sigmoid 為 activation function，共 1 neuron）
   如果最後輸出大於 0.5，label 即為 1，不然就是 0。
   訓練方式是利用 Binary Cross Entropy 為 Loss，利用 Adam 更新參數。

   (b) Word Embedding 方法：

   利用 jieba 將原本的 x_train.csv 和 x_test.csv 的句子斷句為詞
   → 利用 emoji 把句子中的 emoji 轉為文字
   → 把 train 和 test 的 data 都去訓練 Word2Vec 的 embedding model，有加入'pad'這個詞去訓練，每個詞會被對應到一個 128 維度的 vector
   → 把句子的長度 padding 至 50 個詞

   (c) 模型在 validation set 正確率為 0.7579。



2. 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。

   (a) DNN 模型架構：

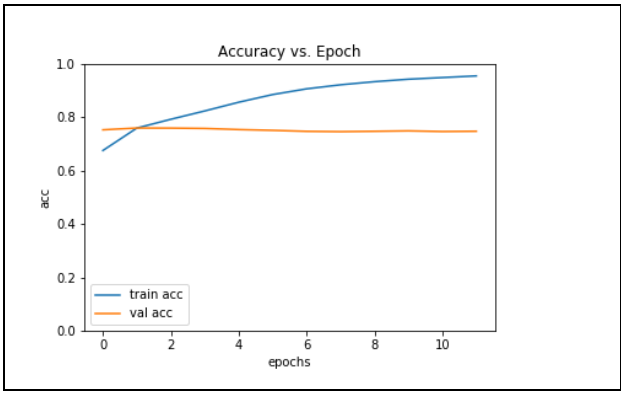
   → Linear Layer（ReLU 為 activation，共 1024 neurons）→ Dropout（機率 0.7）
   → Linear Layer（ReLU 為 activation，共 256 neurons）→ Dropout（機率 0.7）
   → Linear Layer（ReLU 為 activation，共 128 neurons）→ Dropout（機率 0.7）
   → Linear Layer（sigmoid 為 activation function，共 1 neuron）
   如果最後輸出大於 0.5，label 即為 1，不然就是 0。
   訓練方式是利用 Binary Cross Entropy 為 Loss，利用 Adam 更新參數。

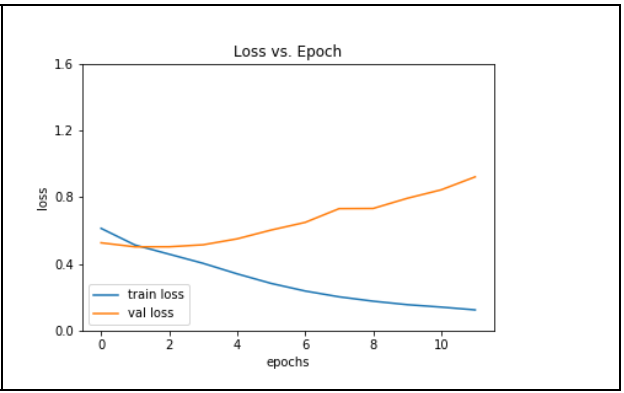
   (b) BOW 方法：

利用 train 和 test 的 data 利用和 1.相同的方法斷句為詞，去訓練 Word2Vec 的 embedding model，把 min_count 設為 20 減少整個 model 個 vocab size

→ 將每個句子利用 BOW 的方法轉為長度為 vocab size (11777) 的 vector

(c) 模型在 validation set 正確率為 0.7585。





3. 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。

(a) Embedding：

在訓練 Word2Vec 的 model 時，將 iter 由 5 調高至 100，可以讓 model 訓練得較好。

(b) 架構：

把原本的 RNN 的 LSTM layer 修改成 Bidirectional LSTM，而且多加一層回傳 sequence 的 Bidirectional LSTM layer，可以讓 val acc 增加約 0.01，因為 Bidirectional 可以讓模型除了考慮到已經出現的詞來判斷留言，還可以用另一個方向判斷。

把 embedding layer 在 Word2Vec 訓練好的 weight 放入 model 訓練也可以提升正確率，因為這樣就可以在訓練過程中也修改到 embedding layer 的參數。

4. 請比較不做斷詞 (e.g., 以字為單位) 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。

不做斷詞的模型在 kaggle public score 可以得到 0.74670，有做斷詞的則可以得到 0.7620，有做斷詞的效果較好，因為同樣的字經過組合變成的詞可能有不同的意思，但如果以自為單位的話模型會比較難判斷，在訓練 word2vec 的時候也比較困難。

5. 請比較 RNN 與 BOW 兩種不同 model 對於 "在說別人白痴之前，先想想自己"與"在說別人之前先想想自己，白痴" 這兩句話的分數（model output），並討論造成差異的原因。

	在說別人白痴之前，先想想自己	在說別人之前先想想自己，白痴
RNN	0.59436840	0.60376656
BOW	0.72377460	0.72377460

BOW 的 model 對於兩句話的判斷是相同的，因為轉換成 BOW 時，並不考慮詞的順序，所以得到的 vector 是一樣的。

RNN 的 model，第二句得到的分數比較高一點，因為有考慮到順序的關係，所以判斷第二句較第一句惡意，但兩者最後都會被 label 為惡意留言。