# Report HW4

學號：b07901039　系級：電機二　姓名：劉知穎

1. (1%) 請說明你實作的 RNN 的模型架構、word embedding 方法、訓練過程 (learning curve)和準確率為何？(盡量是過 public strong baseline 的 model)

(1) RNN 架構

使用 bidirectional 的 LSTM model。

fix_embedding = true、embedding_dim=250、 hidden_dim=150、num_layers=1

classifier 使用一層 linear(input=300、output=1)、dropout=0.5、activation function 使用 sigmoid。learning rate=0.001、epoch=10、batch_size=128。

參數量：14427451 (trainable: 482701)

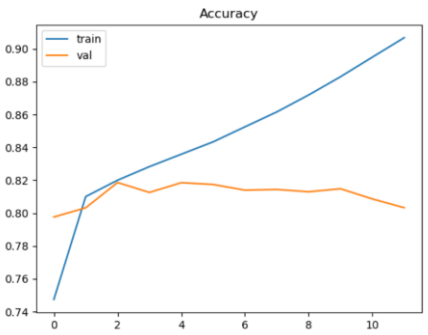
(2) word embedding

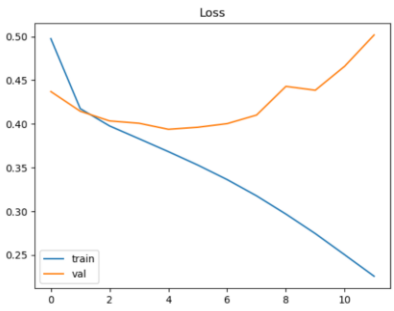
dimension of word vector=250、window=5、min_count=5、使用 skip-gram algorithm。

(Window- Maximum distance between the current and predicted word within a sentence.)

(min_count- Ignores all words with total frequency lower than this.)

(3) 訓練過程




validation accuracy 在 epoch = 2 時有最大值，validation loss 在 epoch = 2~4 之間有最小值，在 epoch > 4 之後 over-fitting。

(4) 準確率

validation accuracy: 0.823

kaggle private set accuracy: 0.82621

2. (2%) 請比較 BOW+DNN 與 RNN 兩種不同 model 對於"today is a good day, but it is hot"與"today is hot, but it is a good day"這兩句的分數(過 softmax 後的數值)，並討論造成差異的原因。

(改用 sigmoid 討論)

DNN 架構：使用四層 linear。輸入輸出維度分別是(250*35, 512)、(512, 512)、(512, 256)、(256, 1)。activation function 在前三層使用 ReLU，最後一層使用 sigmoid。dropout=0.5。

參數量：18819503 (trainable:4874753)

validation accuracy: 0.765

RNN 架構：與第一題相同。

| "today is a good day, but it is hot" | |
|---|---|
| RNN | DNN |
| 0.5609408020973206 | 0.5760492086410522 |

| "today is hot, but it is a good day" | |
|---|---|
| RNN | DNN |
| 0.9718842506408691 | 0.6314681172370911 |

兩者在第一句上都誤判成正面情緒，第二句則都判斷正確。差別在於 RNN 對於兩句的分數有明顯的差異，DNN 則相當接近。原因在於 DNN 不能判斷同樣文字出現順序不同對語意的影響，RNN 則能根據前後的文字(bidirectional model)學習。例如本題將"a good day"和"hot"的順序對換，RNN 可能可以學到在 but 之後的句子對於判斷較重要，因此第二句的分數相當接近 1。

3. (1%) 請敘述你如何 improve performance (preprocess、embedding、架構等等)，並解釋為何這些做法可以使模型進步，並列出準確率與 improve 前的差異。(semi supervised 的部分請在下題回答)

(1) preprocess

嘗試過 remove stopping words、remove 除了 letter、numbers 和一些常見標點符號的文字(發現 training data 裡有許多非英文的字母)、將常見的 abbreviation 轉換成正式的文字。

但準確率沒有上升、反而還有些微下降。

(2) embedding

調整 window 和 dimension of word。

嘗試過將 window 從 3 到 7，dimension of word 200, 250, 300 的更種組合，但結果沒有很大的差別。

(3) 模型架構

a. 改使用 bidirectional lstm。因為 bidirectional 的 lstm 可以從一個字之前與之後的字判斷整句的文意，因此較單向的 lstm 學習力更強。

| | validation accuracy |
|---|---|
| 單向 lstm | 0.819 |
| 雙向 lstm | 0.823 |

b. 調整 sen_len。sen_len 太小時，會有太多句子被截斷，造成模型學習到的資料太少，或沒有學習到一個句子中主要判斷正反文意的部分，因此學習力不佳。當 sen_len 太大時，會有太多句子補上空字符的向量(<PAD>)，造成模型學習到<PAD>對文意的影響，因此學習效果也不佳。經過實驗，sen_len 大約為 35 時有最好的學習效果。

| sen len | validation accuracy |
|---|---|
| 20 | 0.808 |
| 30 | 0.820 |
| 35 | 0.823 |
| 40 | 0.820 |
| 45 | 0.816 |
| 50 | 0.502 |

4. (2%) 請描述你的 semi-supervised 方法是如何標記 label，並比較有無 semi-supervised training 對準確率的影響並試著探討原因 (因為 semi-supervise learning 在 labeled training data 數量較少時，比較能夠發揮作用，所以在實作本題時，建議把有 label 的 training data 從 20 萬筆減少到 2 萬筆以下，在這樣的實驗設定下，比較容易觀察到 semi-supervise learning 所帶來的幫助)。
(使用 20000 筆資料當 training data、20000 筆資料當 valiation data。)

semi-supervised 方法：
使用 self-training。用由 labeled 的 training data(2 萬筆)訓練出的模型對 unlabeled data(20 萬筆)進行預測，將預測值大於 0.95 的 data 標記為 1，小於 0.05 的標記為 0。加入新 label 的 data 後一起在訓練一次。
新增 5690 筆正向資料 、31442 筆負向資料。
original validation accuracy: 0.762
semi-supervised validation accuracy: 0.795
當用原資料 20 萬筆時，semi-supervised 的效果不顯著，兩者結果差不多。

| | valdation accuracy | kaggle public accuracy |
|---|---|---|
| original | 0.823 | 0.82621 |
| semi-supervised | 0.821 | 0.82278 |

將資料減少到 2 萬筆時，較能看出 semi-supervised 產生的影響。因為 semi-supervised training 能從 unlabeled data 產生 labeled data，如此訓練的資料變多，模型就變更穩固。