Homework 5 Report

學號：r06521605
系級：土木所電輔組碩一
姓名：許舜翔

1. (1%) 請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何？(Collaborators:)

答：

- 架構（含資料處理）

  模型是利用 keras 套件建立（如圖一），我參考 keras sample 中 imdb-lstm 的模型架構，並利用 Glove 先 pre-train 出 embedding 層。而資料處理部分，設定詞向量維度為 300，字典大小取 60000，input 的 max_len 取 100，以上皆有考慮標點符號，LSTM 層中設定 dropout=0.2, recurrent dropout=0.2。

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 100, 300)          24000000
_________________________________________________________________
lstm_1 (LSTM)                (None, 128)               219648
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 24,219,777
Trainable params: 219,777
Non-trainable params: 24,000,000
_________________________________________________________________
```

圖一

- 訓練過程

  Optimizer = adam, learning rate = 0.01, batch size = 256, epoch 大約在 6~8 次後會收斂到最好的模型。

- 準確率

  最佳結果為 0.82088/0.81992 (Public/Private)，有實作 semi-supervised 來提升精度，而在實作前為 0.81535/0.81362。

2. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何？(Collaborators:)

答：

- 架構

  利用的字典大小為 20000，text_to_matrix 採用的 mode 為 count，由於資料量龐大，DNN 模型隱藏層設置僅一層，該層的 units 為 512，完整架構如圖二所示。

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 512)               10240512
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 513
=================================================================
Total params: 10,241,025
Trainable params: 10,241,025
Non-trainable params: 0
_________________________________________________________________
```

圖二

- 訓練過程

  Optimizer = adam, learning rate = 0.01, batch size = 256, epoch 大約在 3~4 次後會收斂到最好的模型，且都會有 overfitting 的現象。

- 準確率

  為 0.79601/0.79654 (Public/Private)，因記憶體需求過大，故未能實作 semi-supervised。

3. (1%) 請比較 bag of word 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"這兩句的情緒分數，並討論造成差異的原因。

(Collaborators:)

答：

使用第一題及第二題的架構訓練出來的模型，評價上述兩句的情緒分數，結果如下表。

	today is a good day, but it is hot	today is hot, but it is a good day
RNN	0.07397	0.99972
BOW	0.57235	0.57235

可以看出使用 RNN 模型評估的結果，較接近我們直覺認為的情緒反應，且能區別兩句話的差別，而對於 BOW 模型來說，因為不考慮語序，所以兩者分數相同，除此之外分數也接近 0.5，未能明確判斷究竟是屬於哪一類，推測應為在不考慮語序的情況下，當正向的字出現越多，則歸類為 1，反之則為 0，所以當一句話在沒有特別多正向或負向的字詞時，就無法做出正確的判斷。若題目的句子改為"today is a good day "與"today is hot"的話，評分結果如下表，此時 BOW 模型方能做出區別，並做出較正確的判斷。

	today is a good day	today is hot
RNN	0.99993	0.24155
BOW	0.82327	0.40573

4. (1%) 請比較"有無"包含標點符號兩種不同 tokenize 的方式，並討論兩者對準確率的影響。

(Collaborators:)

答：

使用第一題 RNN 架構，做不同方式的比較，結果如下表。

	Public	Private
包含標點符號	0.82088	0.81992
不包含標點符號	0.81767	0.81744

在維持其他條件不變下，雖然差異不大，但、未包含標點符號的精度稍低，推測應為某些特定句子，會受到標點符號的影響而產生不同的語意，因此拿出兩種模型針對原訓練集評估的結果中，差距超過 0.8 的句子來觀察，例如原句為"gettin to bed soon .. up pretty early tomorrow morning !"，前者訓練資料包含標點符號的模型，會認為是偏負面的句子，評估分數為 0.056；而在後者未包含標點符號的模型，則是認為為偏向正面的句子，評估分數為 0.883，而實際的 label 為 0。

5. （1%）請描述在你的 semi-supervised 方法是如何標記 label，並比較有無 semi-surpervised training 對準確率的影響。

（Collaborators: ）

答：

設定不同大小的 threshold，並對大於該值或小於 1-threshold 的數據分別標上 1 跟 0，並將標籤後的資料加上原本資料重新進行 training，利用的資料包含 test 及 no-label。以第一題中 RNN 的架構訓練出來的模型為基準，測試 semi-supervised 在不同閥值下產生的影響，模型未實作前的準確率為 0.81535/ 0.81362，改變的準確率如下表，結果顯示大多都會改善原先的精度，但以這次作業來說，當閥值取到 0.85，精度的提升明顯下降，且會造成反效果。

Threshold 設定值	新增資料量	實作 Semi-supervised training 之後	
		Public	Private
0.7	935,478 (from no-label data) 159,254 (from test data)	0.82023	0.81958
0.8	788,077 (from no-label data) 134,334 (from test data)	0.81995	0.81851
0.85	695,634 (from no-label data) 118,504 (from test data)	0.81423	0.81426