HW5

學號：B03901109 系級： 電機四 姓名：陳緯哲

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: 賴又誠，https://github.com/mike87179/project/tree/master/hw4)

答：

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 40, 120)           0
_________________________________________________________________
masking_1 (Masking)          (None, 40, 120)           0
_________________________________________________________________
bidirectional_1 (Bidirection (None, 200)               132600
_________________________________________________________________
batch_normalization_1 (Batch (None, 200)               800
_________________________________________________________________
dense_1 (Dense)              (None, 80)                16080
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 80)                0
_________________________________________________________________
batch_normalization_2 (Batch (None, 80)                320
_________________________________________________________________
dropout_1 (Dropout)          (None, 80)                0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 81
=================================================================
Total params: 149,881
Trainable params: 149,321
Non-trainable params: 560
_________________________________________________________________
None
```

Word Embedding 的部分參考了 Collaborator 的方法，使用了 gensim 的 word2vec，將每個字轉為 120 維度的向量，並使用 Skip-gram 的方式來取 feature；RNN 訓練的過程中使用了 adam、Earlystop 等技巧，並將 batch size 設為 512，雖然 epoch 設定為 30，但很少會訓練到 30 個 epoch；在 public 上的準確率為 0.82459，private 則有 0.82332

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？ 答：

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 40)                0
_________________________________________________________________
masking_1 (Masking)          (None, 40)                0
_________________________________________________________________
embedding_1 (Embedding)      (None, 40, 120)           2400000
_________________________________________________________________
dense_1 (Dense)              (None, 40, 80)            9680
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 40, 80)            0
_________________________________________________________________
batch_normalization_1 (Batch (None, 40, 80)            320
_________________________________________________________________
dropout_1 (Dropout)          (None, 40, 80)            0
_________________________________________________________________
flatten_1 (Flatten)          (None, 3200)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 3201
=================================================================
Total params: 2,413,201
Trainable params: 2,413,041
Non-trainable params: 160
_________________________________________________________________
```

BOW 使用了助教提供的模型，字彙量因為 memory 的關係，無法設定太大，最後調整成 1000，轉換方式設定為 count，並使用 Adam、Earlystop，準確率有 0.75929(public)、0.75897(private)。

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。

答：

原本使用 CBOW 對這兩句話分別測得的情緒分數為 0.74 與 0.98，在 label 上都被標為 positive，雖然我個人認為前者應該是 negative，而他較後者的情緒分數也確實較低。

後來我改為使用 Skip-gram，對於題目的那兩句話分別測得 0.22 與 0.98，我認為會改善如此多的原因是在於 CBOW 沒辦法準確利用" BUT" 對語意進行分析，而 skip_gram 的方式則可以準確分析 But 前後語意的差異；另外 BOW 的部分完全沒辦法分出差異，主要原因就是他無法利用詞彙的順序來判別語意，只能從單詞的意義中去判別。

	RNN(CBOW)	RNN(Skip-gram)	BOW
Today is a good day…	0.74	0.38	0.62
today is hot…..	0.98	0.98	0.62

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。

答：

	有標點符號	只留下問號與驚嘆號	無標點符號
Public	0.82286	0.82459	0.82297
private	0.82142	0.82332	0.82110

有無標點符號的準確率如上表，兩者之間的差異並不到，但無標點符號的效果稍微好一點，我在想是否是因為有太多標點符號不存在正面或負面的意義，因此將標點符號的部分只留下問號與驚嘆號，兩個較能表達情緒的符號，果然有改善準確率。

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。

答：

在 semi-supervised 的部分，因為記憶體的關係，我只取 semi_data 的前 10000 筆資料來進行訓練，先讓 model 對這些 data 做預測，再將預測的結果與有 label 的 data 串聯再一起，並進行 training，這個過程重複 3 次左右，門檻設定為 0.1，也就是預測結果高於 0.9 或低於 0.1 的時候才會進行訓練，最後結果如下，可以看出加上 semi-supervised 的效果並沒有改善準確率。

	supervised	Semi-supervised
Public	0.82459	0.82365
private	0.82332	0.82321