Machine Learning (2017, Fall)

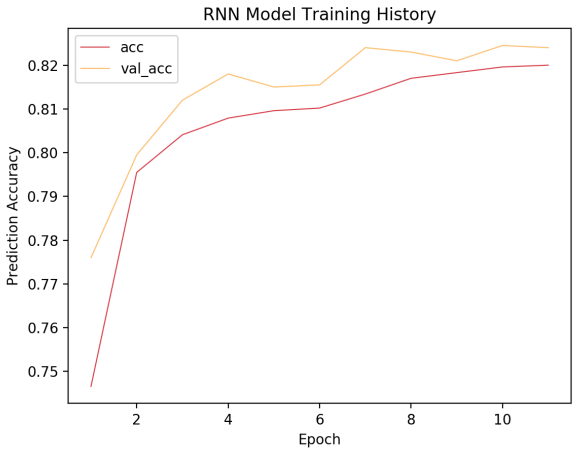
Assignment 4 – Text Sentiment Classification

學號：b03901086　系級：電機四　姓名：楊正彥

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？(1%)

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 30, 200)           25868200
_________________________________________________________________
gru_1 (GRU)                  (None, 30, 512)           1095168
_________________________________________________________________
gru_2 (GRU)                  (None, 256)               590592
_________________________________________________________________
dense_1 (Dense)              (None, 256)               65792
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 257
=================================================================
Total params: 27,620,009
Trainable params: 1,751,809
Non-trainable params: 25,868,200
_________________________________________________________________
```

這次作業的 RNN model 架構如上圖，一共疊了兩層 GRU、兩層 dense 和一層 dropout，而所選用的 batch size 為 512、optimizer 為 Adam、loss function 為 binary cross-entropy、除了最後一層 dense 的 activation 選用 sigmoid 以外，其他層皆使用 relu。使用 gensim 所 pretrained 的 Word2Vec 為 200-dim 的 word embedding model，所使用之 training data 為這次作業所提供的所有 txt 檔。



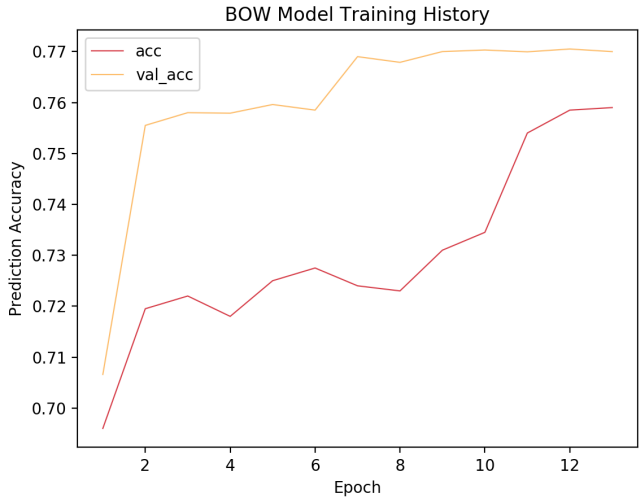
最後模型在 epoch=10 時即得到 val_acc 最高的 model，其準確度如下表：

Validation Accuracy	Testing Accuracy	Public Score	Private Score
0.82450	0.81850	0.82478	0.82334

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

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 1024)              10241024
_________________________________________________________________
dense_2 (Dense)              (None, 512)               524800
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 513
=================================================================
Total params: 10,766,337
Trainable params: 10,766,337
Non-trainable params: 0
_________________________________________________________________
```

這次作業的 BOW model 架構如上圖，一共疊了三層 dense 以及一層 dropout，其餘的訓練相關設定如 batch size、loss function、activation、optimizer 設定皆與 RNN model 相同。而此 BOW model 的 input 為一個 20000-dim 的 BOW vector，僅記錄前 20000 個出現頻率最高的單字出現次數。



最後模型在 epoch=12 時即得到 val_acc 最高的 model，其準確度如下表：

Validation Accuracy	Testing Accuracy	Public Score	Private Score
0.77050	0.76788	0.76737	0.76602

3 請比較 BOW 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。(1%)

Testing Sentences	BOW Model	RNN Model
today is a good day, but it is hot	0.81553212	0.63922944
today is hot, but it is a good day	0.81553212	0.94836336

這兩句話在 BOW model 中所得到的 prediction score 會是一樣的因為其各個單字出現的次數完全一樣，所以轉換成 BOW vector 會得到一樣的結果。而如果透過 RNN model 則能夠透過語句前後不同順序來判斷，得到較精準的情緒分數（這裡以 prediction score 作為情緒分數，為最後一層 sigmoid 的 output）。

4 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。(1%)

為了比較有無標點符號兩種 tokenize 方式，在本題將 Tokenize 中的 filters 做修改，因為其 default 為拿掉大部分的標點符號，得出的結果如下表，雖然說差距仍在 1%但仍可以看出來保留標點符號能夠得到較為精準的 classification 結果。

Model	Validation Accuracy	Testing Accuracy	Public Score	Private Score
w/ Punctuation	0.82960	0.83000	0.82874	0.82741
w/o Punctuation	0.82105	0.82900	0.82147	0.82192

5 請描述在你的 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。(1%)

在 semi-supervised learning 的部份我們分別嘗試了三種 threshold value，用 pretrained 好的 RNN model（所使用的為 w/o 標點符號的 model）對所有的 unlabeled data 做預測，當 prediction score 大於 0.5+V_{th}或者小於 0.5-V_{th}才會將這些 data 加進 labeled data 中 shuffle 後重新 train 一個新的 RNN model。

V_{th}	Public Score	Private Score	Newly Labeled Data Numbers (positive/negative)
0.00	0.82147	0.82192	-/-
0.05	0.81927	0.81919	20899/17032
0.10	0.81836	0.81693	43415/36761

可以看出來在本次作業中進行 semi-supervised training 所達到的效果不是非常好，可能是因為 twitter 的 corpus 太大太雜而且 oov 太多所以以致於準確率沒辦法進步，也有可能因為V_{th}取得太小導致 semi-supervised training 時 RNN 沒有太多新的 info 可以更新自己的 model。