

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？

Layer (type)	Output Shape	Param #
bidirectional_4 (Bidirection	(None, 41, 1024)	2510848
dropout_6 (Dropout)	(None, 41, 1024)	0
bidirectional_5 (Bidirection	(None, 41, 1024)	6295552
dropout_7 (Dropout)	(None, 41, 1024)	0
bidirectional_6 (Bidirection	(None, 1024)	6295552
dropout_8 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 1024)	1049600
activation_4 (Activation)	(None, 1024)	0
dropout_9 (Dropout)	(None, 1024)	0
dense_5 (Dense)	(None, 1024)	1049600
activation_5 (Activation)	(None, 1024)	0
dropout_10 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 2)	2050
activation_6 (Activation)	(None, 2)	0
Total params: 17,203,202		
Trainable params: 17,203,202		
Non-trainable params: 0		

preprocess 部分我是使用 gensim pretrain 作業資料的 word2vec(vector size = 100)，因此在 model 中沒有 embedding 層。在每一個句子我有加上開始和結尾(<s>, </s>)，句子的 padding 用 </s>，以及把總出現次數 < 5 的字視為 <unk>

我用了三層的 Bidirectional LSTM (512)，後面再接兩層 1024 的 Dense layer，中間都有使用 Dropout(0.5)，Activation 部分用的則是 relu。我使用的 batch size = 32，過程大概是在第 3 個 epoch 的時候 validation acc 達到 0.82，之後從 validation 上來看大致上是收斂的穩定狀態。在 kaggle 上的準確率為 0.82578，而我的 best model 是和其他不同架構的 model ensemble 的結果。

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

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 128)	12928
activation_1 (Activation)	(None, 128)	0
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 256)	33024
activation_2 (Activation)	(None, 256)	0
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 512)	131584
activation_3 (Activation)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 2)	1026
activation_4 (Activation)	(None, 2)	0
Total params: 178,562		
Trainable params: 178,562		
Non-trainable params: 0		

preprocess 仍然是用 word2vec 的結果，把一整個句子的 vector 都加起來當作 dnn 的 input。用了 3 分別為 128, 256, 512 的 dense layer，activation = relu，另外在最後 output activation = softmax

訓練過程 validation acc 其實並沒有達到穩定，而 training loss 在第 12 個 epoch 開始不太會下降，此時的 validation acc 是 0.749

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。

rnn: [2.68 1.31] / [0.50 3.49]

bow: [0.47 0.52] / [0.42 0.57]

這兩種方法是用相同的 word2vec model，其中 bow 在兩句上的差異不太明顯，我想其中原因是使用 bag of word 的方法，這兩句話基本上是相同的（一點點的差異可能是因為浮點數誤差等原因），沒辦法充分表現出字詞的順序問題。而在 rnn 上就有顯著的差異，第一句是負面，第二句是正面，成功的捕捉到這兩句話的不同。因為 rnn 有考慮到字詞的順序，並且我使用 bidirectional，因此兩個方向的 feature 都有抓到。

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。

有: validation accuracy = 0.821

無: validation accuracy = 0.813

以上的準確率都是在除了 tokenize 不同之外其他的方法都相同的情況下，epoch = 4 時的 validation 結果，有包含標點符號的結果稍微好一點點，有可能是因為在一些情況下，符號也隱含著一些情緒方面的內容。也就是說如果 input 有一些更多的資訊，rnn 有可能可以稍微學的更好。

5. (1%) 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。

無: validation accuracy = 0.821

有: validation accuracy = 0.824

我標記 label 的方法是取 predict nolabeled data 結果中，機率最高的前 100000 個結果（因為 labeled data 有 200000 筆），無論是正面或負面，標記的方式是直接視為 0 或 1。最後再把標記的 data 送回 model 繼續 train，最後得到的 validation accuracy 比起原本只稍微好一些。推測可能的原因為，在原本的 rnn 中，就已經有非常多的參數可以 train，而 semi supervised 能夠增加的資訊量有限，因此並不能大幅度的提升結果。