學號：b04611041 系級： 工海三 姓名：簡暐晉

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: No)

答：

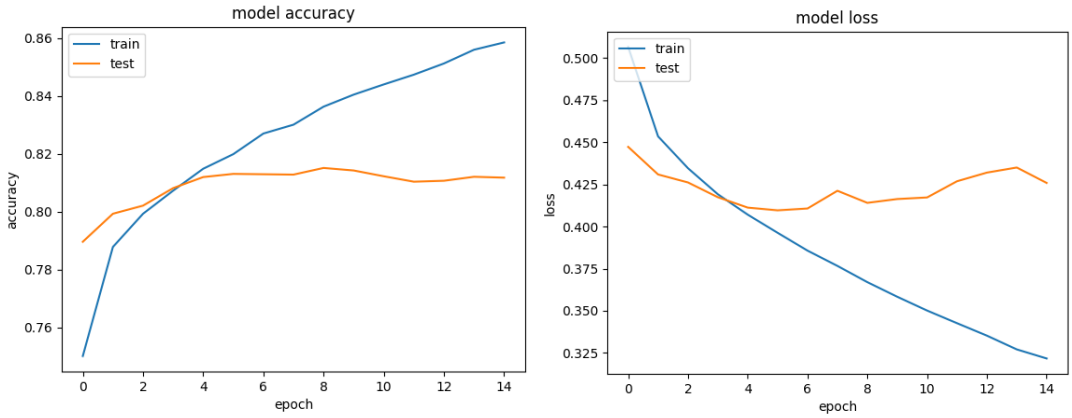
模型架構:

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 36, 200)	7157600
bidirectional_1 (Bidirection	(None, 128)	135680
dense_1 (Dense)	(None, 256)	33024
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 1)	65

Total params: 7,367,521
Trainable params: 209,921
Non-trainable params: 7,157,600

我先用 GloVe pre-train 出 word vector 當作我的 embedding layer，使用的 data 為 training, testing 和 unlabel data 的句子，並選擇出現頻率大於 3 次的 word。vector dimension 為 200 維。並且讓他在 training 時不再更動 (trainable=False)，接著使用一層 128 unit 的 LSTM，activation function 為 tanh。之後接到 3 層 dense layer，units 分別為 256, 128, 64, activation 皆為 relu，後面各加了 3 層 dropout，dropout rate 皆為 20%，最後 output 1 個 unit 使用 binary_crossentropy 來 predict 答案，optimizer 為 adam。

訓練過程:



我選擇 data 的 20%作為 validation data，可以看到大約在第 6 個 epoch 後，validation accuracy 便上升的很緩慢，最佳結果約可接近 82%。最後我用全部的 data train 10 個 epoch 後當作最後結果，單一 model 在 kaggle 上的正確率為 0.82039，ensemble 我則是另外加了 2 個 model(lstm 改成 gru 以及不用 bidirectional)，最後的 public 正確率為 0.82827。

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:No)

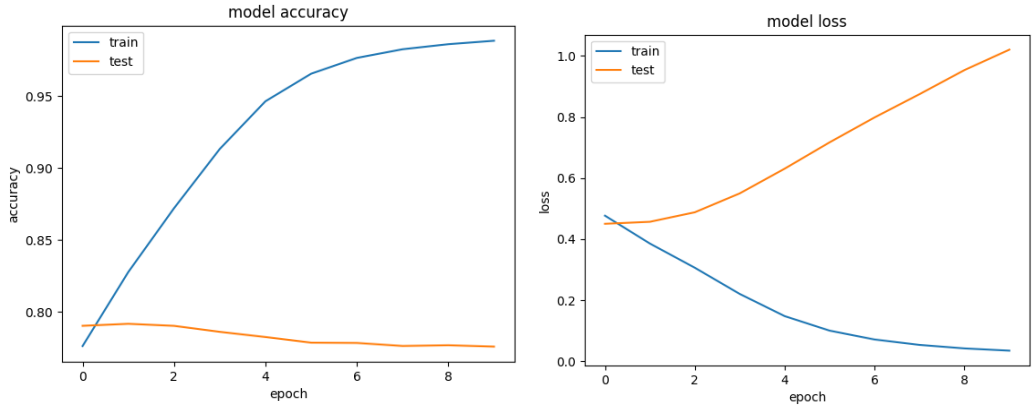
答：

模型架構：

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 256)               6267904
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 257
=================================================================
Total params: 6,268,161
Trainable params: 6,268,161
Non-trainable params: 0
```

我選擇出現次數 5 上的字做成詞袋向量(約 15000 維)，並 count 文字在句子的出現次數當作 word vector，最後接到 1 層 256 units 的 dense layer，activation function 為 relu，另外中間加 1 層 droput，dropout rate 為 20%，最後 output 1 個 unit 來 predict 答案。

訓練過程：



我同樣選擇 20%的 data 作為 validation，可以看到大概在第二個 epoch 之後 validation accuracy 就不再上升，並且與 RNN 的 model 比起來較容易 overfit。而 bag of word 的 model 在 kaggle 上的正確率只有 0.79695，相較於 RNN 的 model 低了不少，由此可知字的出現順序對於情緒的判斷有一定程度的影響。

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: No)

	today is a good day, but it is hot	today is hot, but it is a good day
Bag of word	0.66114289	0.66114289

RNN	0.23048772	0.9712913

我的 bag of word 使用文字的出現次數作為 vector，因此對於這兩句話來說，因為出現的字都一樣，因此產生的 vector 也一樣，所以 output 的結果都一樣。而 RNN 因為會考慮字的出現順序，因此對於這兩句話可以對於這兩句話則可以產生出不同的結果，並且 predict 出正確的答案。

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: No)
   答：

我使用沒有包含標點符號的方法，在 kaggle 上的正確率為 0.81122，和有用標點符號的方法相比起來，正確率約少了 1%，由此可知標點符號對於情緒判斷可能有一定程度的幫助。例如有驚嘆號可能表示當下心情激動等等。

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。(Collaborators:No)
   答:

iteration	Training Data size	Training accuracy	Validation accuracy
1	160000	0.7471	0.7892
		0.7879	0.8019
2	627369	0.9486	0.8057
		0.9506	0.8116
3	864678	0.9646	0.8141
		0.9656	0.8138
4	1030507	0.9696	0.8161
		0.9701	0.8159
5	1125981	0.9701	0.8166
		0.9704	0.8148

我一開始選出 20%的 data 作為 validation，之後設定 threshold 為 0.2，亦即將 predict 出的 unlabel data 的結果，小於 0.2 標記成 0，大於 0.8 標記則成 1，再重新加入我的 training data，總共跑 5 個 iteration，並且每次跑 2 個 epoch，雖然 training 的結果一直持續上升，但 validation 的結果並沒有很明顯的上升。最後在 kaggle 上的正確率為 0.81913，與沒有做 semi-supervised 的結果並沒有相差太多。