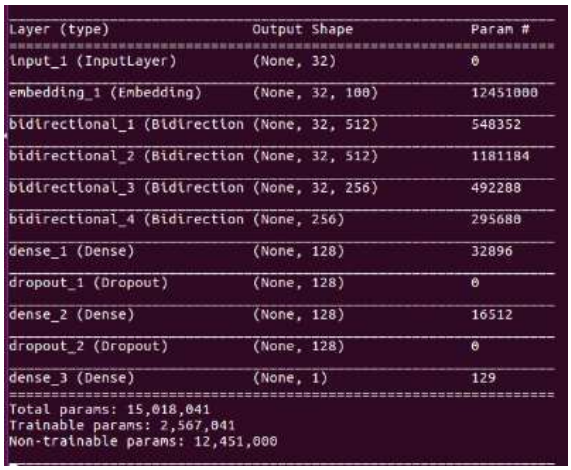
學號：B04701232 系級： 工管三 姓名：陳柔安

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：

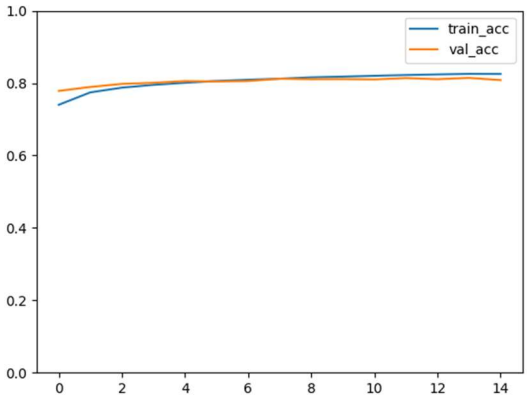
MAX_NUM_WORDS = 100000
MAX_SEQUENCE_LENGTH = 32
EMBEDDING_DIM = 100
DROPOUT = 0.4
EPOCHS = 15
BATCH = 128
Loss function ='binary_crossentropy'
Optimizer='rmsprop'



Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 32)	0
embedding_1 (Embedding)	(None, 32, 100)	12451000
bidirectional_1 (Bidirection	(None, 32, 512)	548352
bidirectional_2 (Bidirection	(None, 32, 512)	1181184
bidirectional_3 (Bidirection	(None, 32, 256)	492288
bidirectional_4 (Bidirection	(None, 256)	295680
dense_1 (Dense)	(None, 128)	32896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 128)	16512
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 1)	129

Total params: 15,018,041
Trainable params: 2,567,041
Non-trainable params: 12,451,000

- data preprocess：把 X_test, X_train 都一起放進 tokenizer 裡，並取 training data 中最後 20000 筆資料作為 validation data。
- model build：Embedding layer 使用 gensim 的 Word2Vec 函式 pretrain，RNN 的 layer 選用 Bidirectional LSTM。使用 callback 存下所有 model 取 valacc 最佳者為第 11 個 epoch，valacc 為 0.81280。
- 準確率：上傳至 kaggle 的最佳結果為 0.81353/0.81449。(private/public)



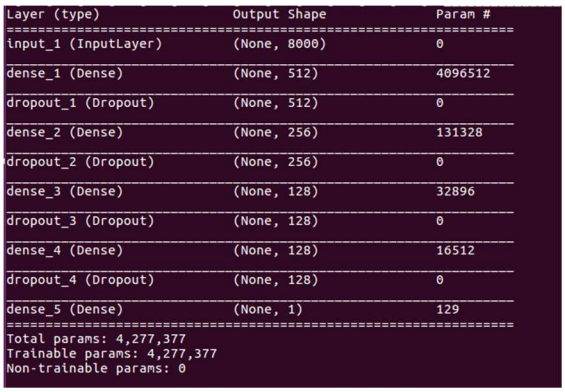
圖一-RNN model 訓練過程

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：

MAX_NUM_WORDS = 8000

MAX_SEQUENCE_LENGTH = 32
EMBEDDING_DIM = 100
DROPOUT = 0.4
EPOCHS = 15
BATCH = 128
Loss function ='binary_crossentropy'

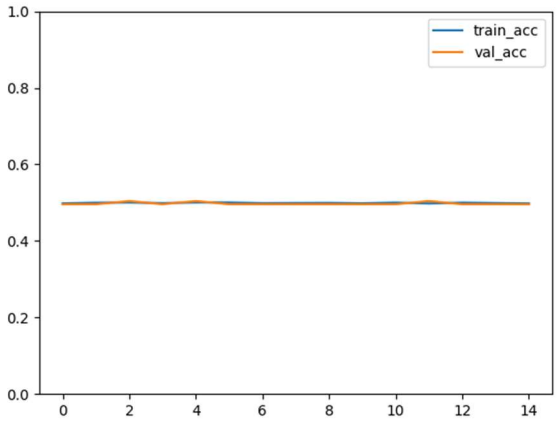


Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 8000)	0
dense_1 (Dense)	(None, 512)	4096512
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
dropout_3 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 128)	16512
dropout_4 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 1)	129

Total params: 4,277,377
Trainable params: 4,277,377
Non-trainable params: 0

Optimizer='rmsprop'

- data preprocess：同 RNN model，tokenizer 使用 keras 內建 texts_to_matrix()
- model build：將 bag of words 的 matrix 直接傳進 DNN，並使用 callback 存下所有 model。
- 準確率：幾乎所有 epoch 的準確率都在 0.5 左右，原因可能是在 train 的過程中因為 memory 太小只能取較少的 num_words，才導致結果像是用隨機猜測的。



圖二-BOW model 訓練過程

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：

RNN：0.60031992/0.99357611

BOW：0.50122428/0.50122428

猜測原因可能為 RNN 在 predict 時會考慮到詞語前後順序，BOW 不會。在這句話中，but 這個連接詞後才是所要強調的語意重點，RNN model 在 predict 上可看出 but 和 good 的先後順序關係，以此作為判斷情緒分數的依據，所以兩者分數會有差異，而 BOW 則否。

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：

以下準確率為選取 15 個 epoch 中，valacc 最高的 model 來對 testing data 預測，上傳 kaggle 所得之結果(private/public)。

有標點：0.81249/0.81449

無標點：0.81353/0.81449

可能原因推測為 tokenizer 會將標點視為字詞，但實際上標點並沒有顯著的影響語意，所以無標點的準確率稍低，但因為每次 train 都會有一點不同，而有無標點 train 出來的準確率並沒有非常大的差異，所以這個差異也可能來自於 random 的誤差。

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：

- 標記 label 方法：以先前 train 好的 best model（RNN）將 unlabeled data predict 一次，若 predict 的值大於 0.7 則標記 1，小於 0.3 則標記 0，將有標記的這些 unlabeled data 加入 training data 中，重新 train 一次。
- 準確率：
  semi-supervised 準確率：0.81249/0.81449
  supervised 準確率：0.81353/0.81449
- 原因探討：從此數據看起來並無顯著影響，探討原因可能為在 train semi-supervised 時因數據量多，training 時間較長，因此建立 model 時沒有 train 到 15 個 epoch，只從前 5 個 epoch 中選擇最好的，若 train 多一點 epoch 可能會有更好的效果。