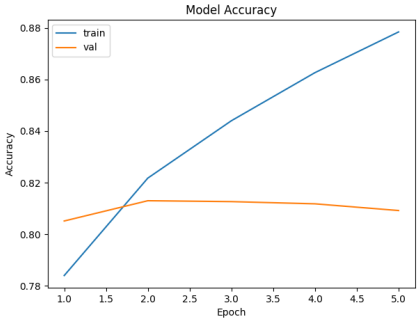
學號：R05525096 系級： 工海二 姓名：郭捷

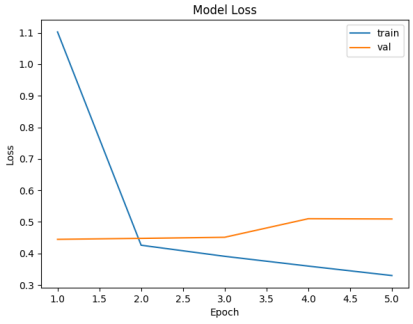
1. (1%) 請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何？
(Collaborators:)

答：

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 38)	0
embedding_1 (Embedding)	(None, 38, 256)	7680256
lstm_1 (LSTM)	(None, 512)	1574912
dense_1 (Dense)	(None, 256)	131328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 1)	257

Total params: 9,386,753
Trainable params: 9,386,753
Non-trainable params: 0



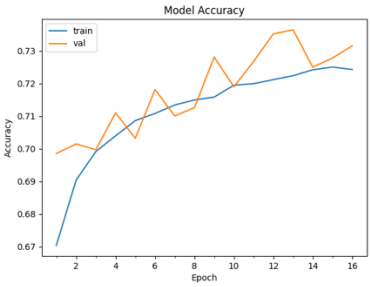


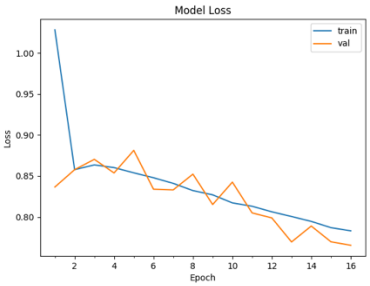
訓練細節:本次實作的 RNN model 先用 gensim 的 word2vec 訓練 training data 和 unlabel data 將訓練結果作為 embedding layer 的初始 weight。epoch 設定為 20，但是因為有 earlystop 所以大概在第 5 個 epoch 就會停下來。Optimizer 使用 adam 。loss function 為 binary_crossentropy。
在 training 的準確率為 0.8218 在 val 的準確率為 0.8130 在 kaggle public board 的準確率為 0.81624

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：

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 15000)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               3840256
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 257
=================================================================
Total params: 3,840,513
Trainable params: 3,840,513
Non-trainable params: 0
```





訓練細節:本次實作的 BOW model 由於 memory 的關係 vocab_size 設定為 15000 。epoch 設定為 20，但是因為有 earlystop 所以大概在第 16 個 epoch 就會停下來。Optimizer 使用 adam 。loss function 為 binary_crossentropy。
在 training 的準確率為 0.7224 在 val 的準確率為 0.7365 在 kaggle public board 的準確率為 0.73533

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：bow model 對兩句的情緒分數相等，

```
[[ 0.49893251]
 [ 0.49893251]]
```

RNN model 對兩句的情緒分數

```
[[ 0.25100937]
 [ 0.91962677]]
```

首先由於 bow model 忽略了句子的順序，因為這兩句話的單字數相同，所以對 bow model 來說是相同的句子，所以出來的分數是一樣的。而對於 RNN 的 model 來說這是兩個不同的句子，第一個句子顯然比較偏向負面一點，有點抱怨今天太熱，但是第二句則向表達雖然熱但依然是個好天氣，偏向正面。造成差異的最主要原因是 BOW 不考慮文法及詞的順序，導致其對於這樣的句子理解得不夠好。

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：在我實作的 Model 中,有包含標點符號（沒有過濾標點符號）的 tokenize 的方式準確率會略高於沒有包含標點符號（有過濾標點符號）的方式。我推測可能的原因是,.!這類標點符號在某種程度上也是可以表達情緒的，例如在情緒比較激動的時候常常會使用驚嘆號（！），在情緒比較平緩的時候比較常會使用，。這類標點符號。

   有過濾標點符號 kaggle public score : 0.81624

   沒有過濾標點符號 kaggle public score : 0.81685

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：本次實作的 semi-supervised 的方法，threshold 設定為 0.9，也就是當情緒分數小於 0.1 時會將這筆資料標記為負面，當情緒分數大於 0.9 的時候會標記為正面，在沒有使用 semi-surpervised training 的情況下準確率為 0.81624。使用 semi-surpervised training 的方法準確率達到 0.82247。可能的原因為 semi-surpervised 的方法將一些在 train data 中沒有見過的資料但是有信心可以分類正確的資料進行標記加入 train data 中增加 train data 的資料量，並且避免 overfit train data。並且將一些 predict 情緒分數處於 0.5 附近的情緒更好的區分其是屬於正面還是負面。