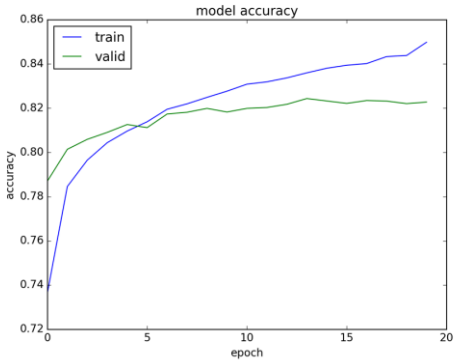
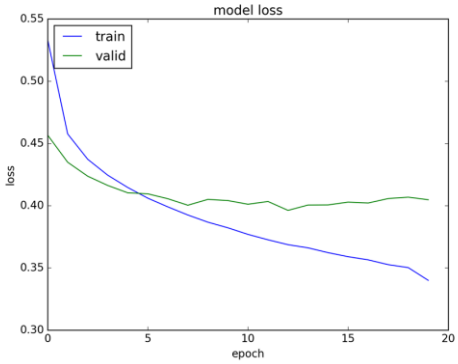
學號：R05943138 系級： 電子所碩二 姓名：賴又誠

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？
   我使用 gensim 去做 word embedding(每個 word 及標點符號轉換成 100 維的向量),之後再置入後方 RNN+NN 中學習字之間的順序和分類

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
masking_1 (Masking)          (None, 40, 100)           0
_________________________________________________________________
bidirectional_1 (Bidirection (None, 256)               175872
_________________________________________________________________
batch_normalization_1 (Batch (None, 256)               1024
_________________________________________________________________
dense_1 (Dense)              (None, 64)                16448
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 64)                0
_________________________________________________________________
batch_normalization_2 (Batch (None, 64)                256
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 65
=================================================================
Total params: 193,665
Trainable params: 193,025
Non-trainable params: 640
_________________________________________________________________
```

loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])



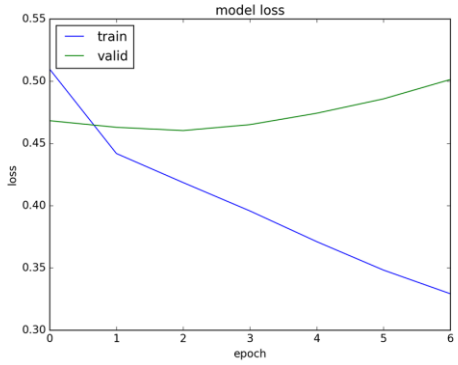


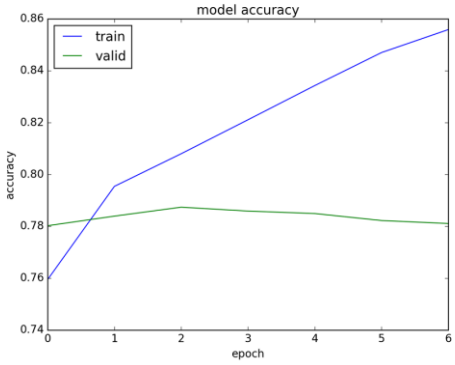
Accuracy(public)	Accuracy(private)
0.82511	0.82294

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_7 (Dense)              (None, 64)                256064
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 64)                0
_________________________________________________________________
batch_normalization_4 (Batch (None, 64)                256
_________________________________________________________________
dropout_4 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_8 (Dense)              (None, 1)                 65
=================================================================
Total params: 256,385
Trainable params: 256,257
Non-trainable params: 128
_________________________________________________________________
```

使用助教所提到 keras 的 Tokenizer.text_to_matrix,使用的 mode 為 tfidf,為了和第一題公平比較,上題一個 sentence 的參數量為 max_length*word_dim=(40*100),所以我用 bow 把一句 sentence 做成 4000 維度的向量去學習





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。

Sentence	Bag of word	RNN
today is a good day, but it is hot	0.56278945 (正面情緒)	0.44426772 (負面情緒)

today is hot, but it is a good day	0.56278945 (正面情緒)	0.98298484 (正面情緒)

最後一層是使用 sigmoid,所以結果>0.5 為 1;結果<0.5 為 0, 因為 bow 沒有順序的因素,所以兩者的分數是一樣的,而 RNN 會學習順序這件事使得句子正負情緒得以被學習出來.

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。

Model	Tokenize	Accuracy(public)
同第一題	無標點符號	0.8191
同第一題	有標點符號	0.82379

可以看到有標點符號是比較好的,我覺得是因為`!`或`...`稍微會透漏一點正負面的情感,故對於這次的 case 是有用的.

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。

使用手法為 self-training,先 train 完 label data 後把 no-label data 的前 10 萬筆用這個 model 先標上 label 並分為兩個 class,再加入 training data(20 萬筆中),再以剛剛的 model 繼續以這 30 萬筆 train 下去,在 public 上有明顯的進步,不使用全部 no-label data 是因為記憶體不足,若使用全部 data 電腦會當機.

Model	Topology	Number of training data	Accuracy(public)
同第一題 (無標點符號)	Supervised learning	200,000(label)	0.81919
同第一題 (無標點符號)	Semi-Supervised learning	200,000(label)+ 100,000(no-label)	0.82117