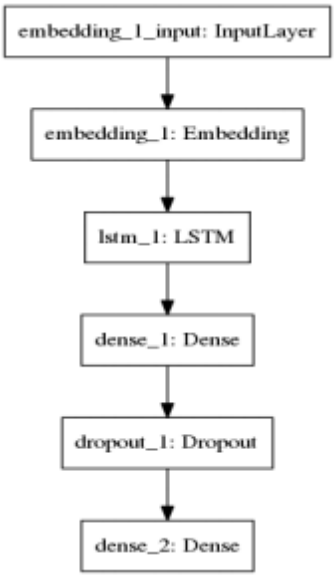
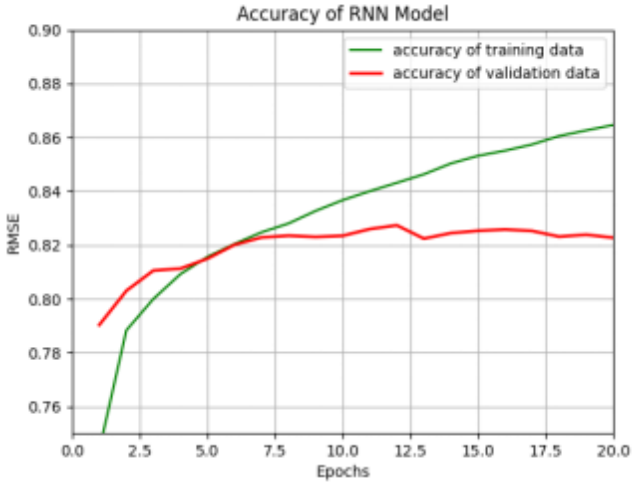
學號：R05922145 系級：資工碩二 姓名：郁錦濤

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：



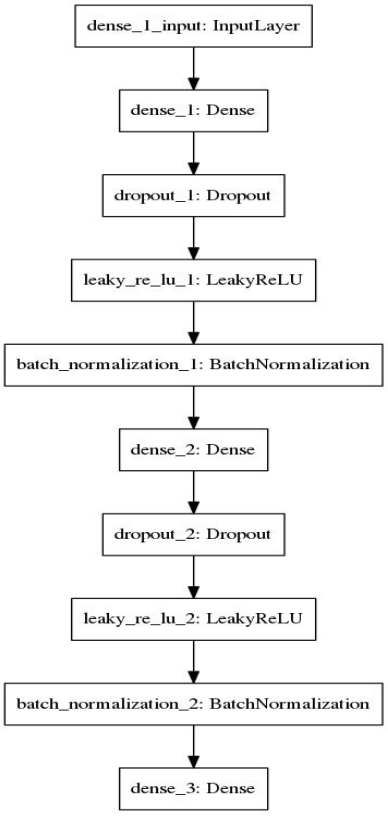

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 40, 100)           2000100
_________________________________________________________________
lstm_1 (LSTM)                (None, 512)               1255424
_________________________________________________________________
dense_1 (Dense)              (None, 32)                16416
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 33
=================================================================
Total params: 3,271,973
Trainable params: 1,271,873
Non-trainable params: 2,000,100
_________________________________________________________________
Train on 180000 samples, validate on 20000 samples
```



我的model採用了一層LSTM，然後dense和dropout，train的過程中將embedding也加入一起train，epoch爲20，optimizer爲adam，adam的參數爲默認值lr=1e-3，loss fuction採用binary_crossentropy。最後在private set上的準確率爲0.82122。

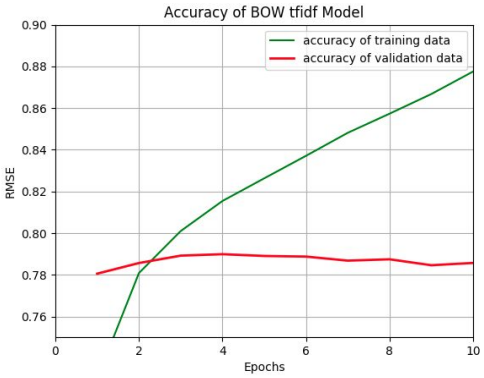
2. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何？ (Collaborators:)

答：



Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1024)	10241024
dropout_1 (Dropout)	(None, 1024)	0
leaky_re_lu_1 (LeakyReLU)	(None, 1024)	0
batch_normalization_1 (Batch	(None, 1024)	4096
dense_2 (Dense)	(None, 1024)	1049600
dropout_2 (Dropout)	(None, 1024)	0
leaky_re_lu_2 (LeakyReLU)	(None, 1024)	0
batch_normalization_2 (Batch	(None, 1024)	4096
dense_3 (Dense)	(None, 1)	1025

Total params: 11,299,841
Trainable params: 11,295,745



我實做的BOW model分別才用count和tfidf方式進行tokenize，都是先用label data/nolabel data/testing data通過tokenizer.fit_on_texts構建dict，然後採用

text_to_sequence中的mode選取count和tfidf兩種不同方式，將text表示成vector的樣子。然後將text放入DNN中，epoch=10, optimizer=Adam，loss fuction採用binary_crossentry。最終的結果如下表所示。

通過結果預測，發現在BOW model中，如果僅採用count方式進行表示text的vector的話，結果會很不理想，而採用tf-idf方式會又不錯的表現。我覺得BOW count表現不好的原因在於，我選擇的dict的dim因爲內存原因，我只能限制在10000，並且沒有處理stopwords所以表現不佳。

兩種方式的預測結果如下表：

BOW	public set	private set
count	0.48564	0.48361
tfidf	0.78891	0.78812

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：

Model	"today is a good day, but it is hot"	"today is hot, but it is a good day
BOW-count	0.48300534	0.48161653
BOW-tfidf	0.52564400	0.51569376
RNN	0.1712955	0.13643019

BOW的兩種model對於這兩句話的得分大致上是一致的，都在0.5左右，比較難判斷sentiment。

RNN的model對於這兩句話具有比較明顯的劃分，情緒分數差距較大。

BOW沒有考慮詞的順序對於句子情緒的影響。所以改變句子中詞語的順序對於判斷情緒的影響不大；而RNN中才用了Word2Vec，它考慮了詞與詞之間的關系，句子的詞序變化會對於情緒判斷有很大影響。

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：

我的model中取tokenize的方式是用keras中的Tokenizer。通過對Tokenize的參數filter進行修改來實現有無包含標點符號。通過比較，發現含有標點符號的tokenize方式在public set和private set上的performance跟好。仔細分析，我覺得標點符號對於人的情感是具有一定影響的，一個句子以感嘆號結尾和以問號結尾是不同，例如“我很高興。”和“我很高興？”，從字面意義上來看，這兩句話表達的情感是截然不同的，但是如果不包含標點符號的話，就會認爲兩句話是一致的，這樣就會產生誤差。

punctions	public	private
yes	0.82355	0.82122
no	0.82031	0.81785

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：

我的semi-supervised方法採用了助教的方式，首先用含有label的data進行train，得到用label的data訓練出來的模型，然後用不含label的data通過得到的model進行預測。得到的預測結果，選取合適的threshold值（我取得是0.3和0.7），把機率小於0.2和大於0.8的data放入training data中再進行fit操作。

按照经验，semi-surpervised training会增加training的资料量，對於準確率會有一定程度的提升，但是在我的model中，進行了semi-surpervised後準確度卻沒有提升，反而準確率有所降低。究其原因，我認爲threshold的取值很重要，對於模棱兩可的data只有選用更高的threshold才能更好的进行分類。另外，數據的相關性也很重要。