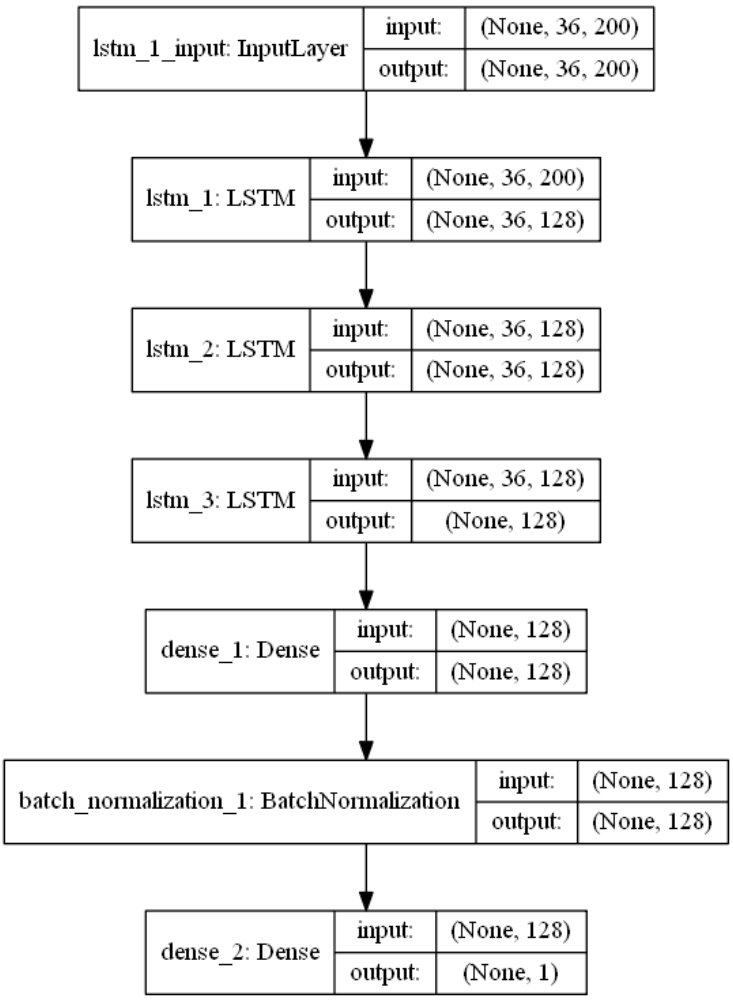
學號：B03902096 系級： 資工四 姓名：陳柏屹

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：首先將 training_label.txt 和 training_nolabel.txt 兩者合併後作處理，保留注音符號等，並將所有的數字換成同一個數字 0 (這是因為我認為數字大小不影響結果，統一為同個數字可以有較佳的 word2vec 結果)。再用 gensim 訓練我的 word2vec model(training algorithm 為 skip-gram，size=200，min_count=0, iter=10)。最後再丟入以下的 RNN 模型中。其中我使用的 activation function 皆為 sigmoid， epoch 數維持在每次訓練皆 15。而 loss function 則為 binary_crossentropy，optimizer 為 adam。Testing 時同樣地也要先實施同樣的前處理，只留下英文和統一後的數字。

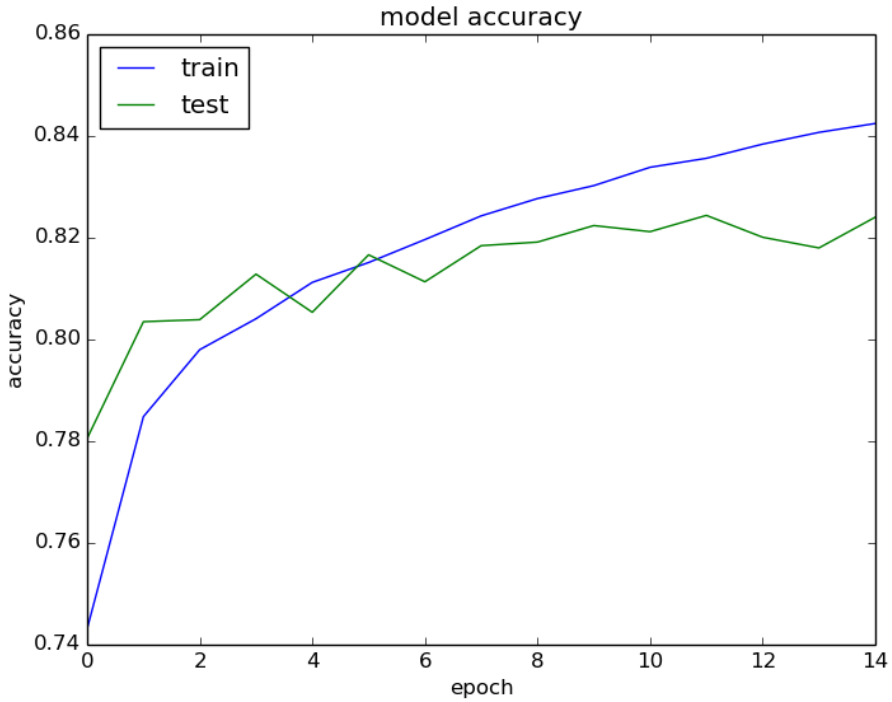


而最終準確率從 kaggle 得知如以下：

public testing set	private testing set

0.83132	0.83169

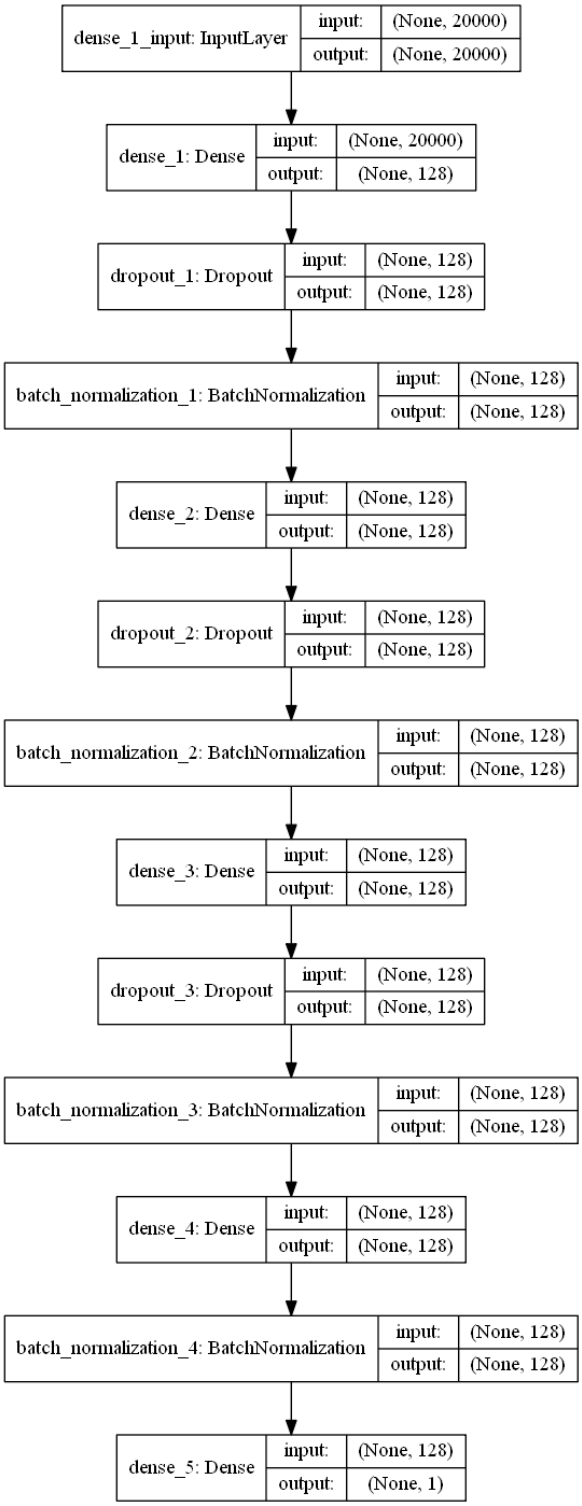
訓練過程中的 training set, validation set 準確率變化則如以下：



(圖中的 test 為佔整體 training data 的 10%validation set 測試結果)

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:)

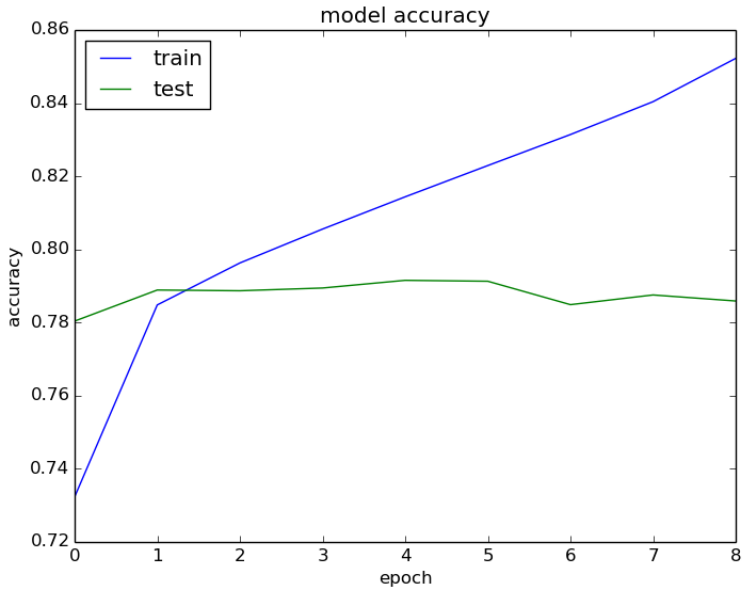
答：同樣地我將 training_label.txt 去除標點符號，並將數字統一為 0 後下去做，而 bag of word 的轉換我則利用 dictionary 手刻，並只留下出現次數前兩萬個的字詞。Dense 模型方面我則維持原本的 RNN model 的層數，並維持同樣的 dropout rate，只是用 Dense layer + BatchNormalization layer 來取代原本的 lstm layer。我使用的 activation function 皆為 sigmoid，並利用 early stopping 將 epoch 數維持在每次訓練皆 12 內。而 loss function 則為 binary_crossentropy，optimizer 為 adam。Testing 時同樣地也要先實施同樣的前處理，只留下英文和統一後的數字。



而最終準確率從 kaggle 得知如以下：

public testing set	private testing set
0.78634	0.78690

訓練過程中的 training set, validation set 準確率變化則如以下：



可以發現 validation 的準確率並沒有明顯的起伏。

(圖中的 test 為佔整體 training data 的 10%validation set 測試結果)

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：

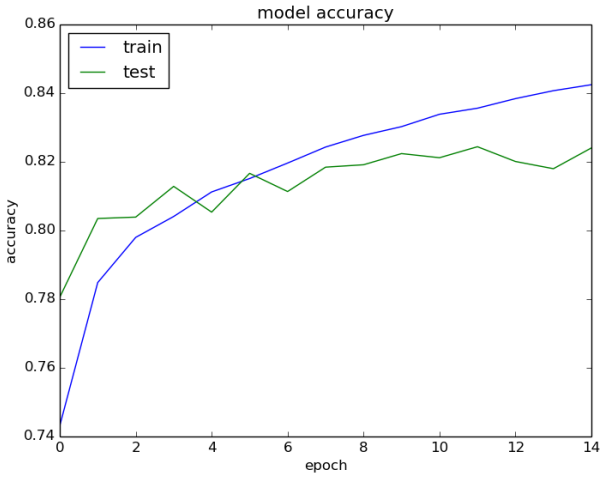
	today is a good day, but it is hot	today is hot, but it is a good day
RNN	0.22265944	0.99392265
BOW	0.98759651	0.98759651

我們可以發現 BOW 模型對於兩句話來說的得分是相同的，原因顯而易見的是因為他並沒有考慮文字間排列的位置，因此兩句話對它來說是相同的，猜測因為句子中包含 good，因此判斷接近正面。RNN 模型則給出兩句話不同趨近的 label，顯示它較能夠還原文意，不因為單純有 good 就將句子標籤為正面，而是會考慮 but 的位置再進行判斷。

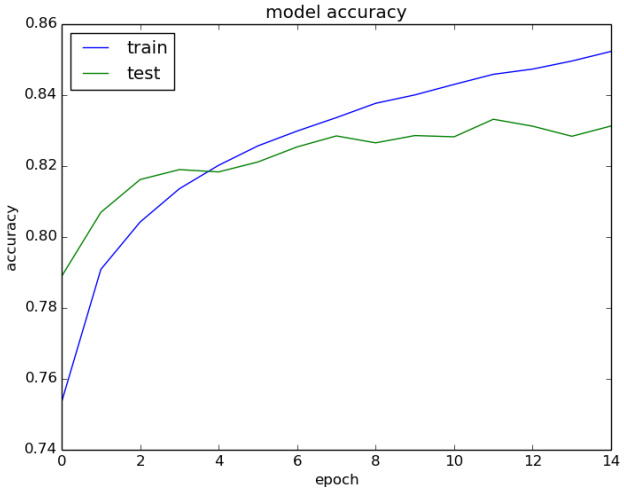
4. (1%) 請比較"有無"包含標點符號兩種不同 tokenize 的方式，並討論兩者對準確率的影響。
   (Collaborators:)
   答：首先將 training_label.txt 和 training_nolabel.txt 兩者合併後作處理，兩者差別

只在是否保留注音符號，並將所有的數字換成同一個數字 0 (這是因為我認為數字大小不影響結果，統一為同個數字可以有較佳的 word2vec 結果)。再用 gensim 訓練我的 word2vec model(training algorithm 為 skip-gram，size=200，min_count=0, iter=10)。從訓練過程中的準確率變化、kaggle test set 的準確率可以發現保留標點符號可以提升模型的準確率。

	public testing data accuracy	private testing data accuracy
有標點符號	0.83132	0.83169
無標點符號	0.82447	0.82287



去除標點符號



保留標點符號

(圖中的 test 為佔整體 training data 的 10%validation set 測試結果)

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：我同樣使用 1.的 RNN 模型來實作 self-training。然而我會先用 1.的 RNN 模型去計算 training_nolabel.txt 的分數，並將分數超過 0.92 和低於 0.08 的保留。並將他們加入 training data 後重新 train 過模型，再去 predict testing data。然而我發現資料量會變得很肥大很慢，且 validation 的 accuracy 也會異常地飆高(9x%)，這都是因為 no label 部分的加入，然而再經過 early stopping 後 epoch 數為 3，但顯然地使用 self training 並沒有提升模型準確率。

	public testing data accuracy	private testing data accuracy
without semi	0.83132	0.83169
semi	0.79328	0.79227