學號：b04901074 系級： 電機三 姓名：吳倉永

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:)

前處理：我在文字的前處理部分有參考以前同學的方法(github: victoresque)，我觀察了 twitter data，發現文字相當的髒，有很多疊字的情形出現，例如：tomorrowwww 跟 tomorrow…..等，其實都可以用一個字表示就好。因此我根據某個詞出現的頻率資訊，去修改我們的 data，建立自己的「辭典」(利用 labeled data 跟 unlabeled data 來建立)，也就是 conversion map，將 data 的字串優化成一個較 compact 的 corpus(文本)。

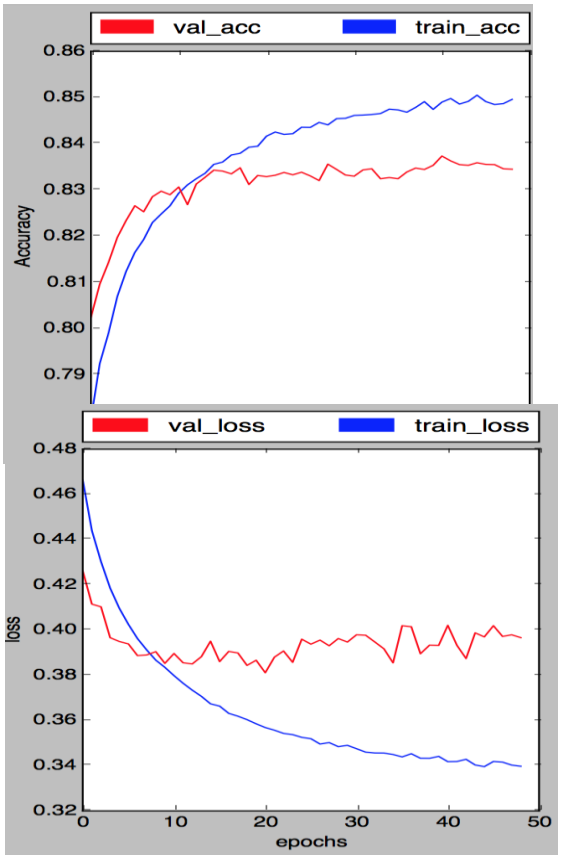
訓練過程：利用 adam，binary cross-entropy，epoch 50，batch 512 且有 earlystopping 跟 check-point 來訓練模型

模型架構：LSTM + GRU + FC * 3，每層都是有使用 Dropout (0.5)，且 FC 都會配 Batch-normalization，共有 parameters 42M。

結果：單一 model 準確率，public：83.45%, private：83.38%，利用 ensemble 5 個 model 後得到 public：83.9%，private：83.6%的準確率

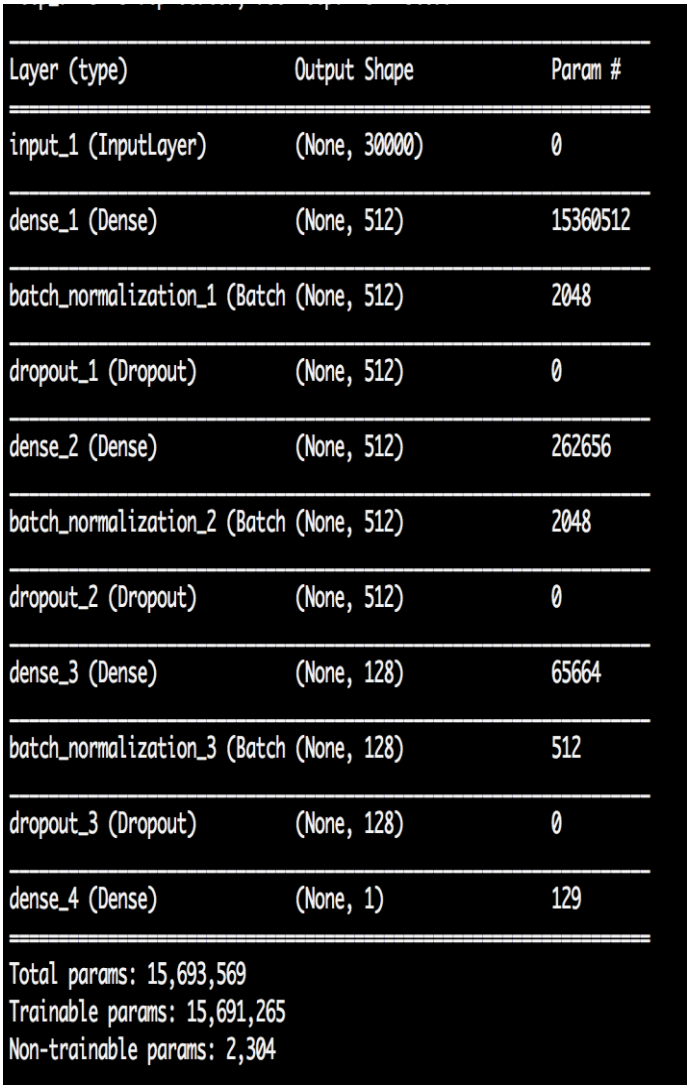
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 30, 256)	0
lstm_1 (LSTM)	(None, 30, 512)	1574912
gru_1 (GRU)	(None, 512)	1574400
dense_1 (Dense)	(None, 1024)	525312
batch_normalization_1 (Batch	(None, 1024)	4096
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524800
batch_normalization_2 (Batch	(None, 512)	2048
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 128)	65664
batch_normalization_3 (Batch	(None, 128)	512
dropout_3 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 1)	129

Total params: 4,271,873
Trainable params: 4,268,545
Non-trainable params: 3,328



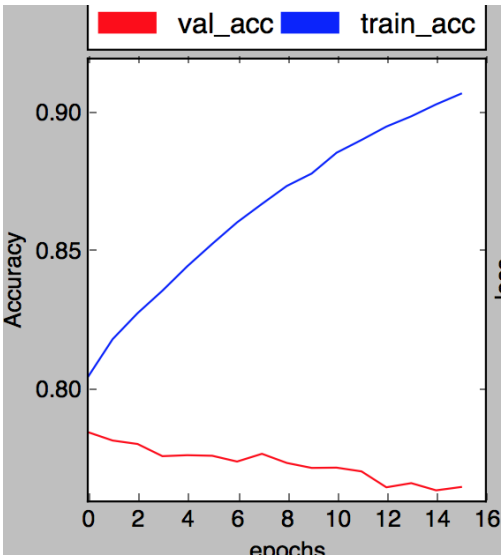
2. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何？

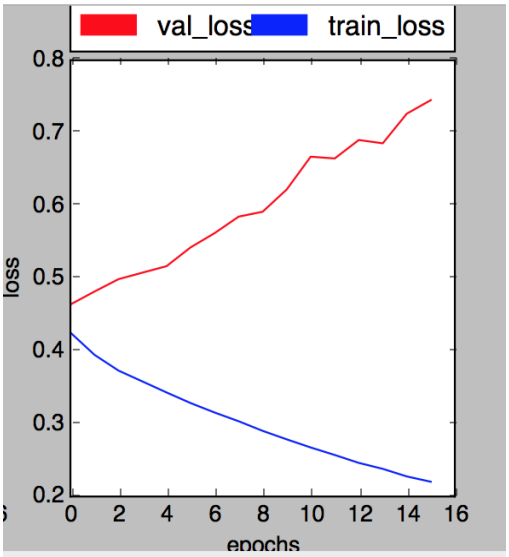
Data 前處理都跟 RNN 一樣，只差在最後利用 BOW 及 FC 架構來進行訓練，與 RNN 不同的是，此架構很快就 overfitting，可以想到模型只是在 fit training data，而無法學到文字語意上的精隨。



Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 30000)	0
dense_1 (Dense)	(None, 512)	15360512
batch_normalization_1 (Batch	(None, 512)	2048
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 512)	262656
batch_normalization_2 (Batch	(None, 512)	2048
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 128)	65664
batch_normalization_3 (Batch	(None, 128)	512
dropout_3 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 1)	129

Total params: 15,693,569
Trainable params: 15,691,265
Non-trainable params: 2,304





結果：public：78.76%, private：78.8%

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:)

BOW 第一句分數：62%，第二句分數：63%

RNN 第一句分數：22%，第二句分數：98.8%

可以發現 RNN 有考慮到文字的順序，造成上的語意差別，BOW 並沒有

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。

我將所有的標點符號(包含特殊符號、驚嘆號、問號、點…等)都移除，其他持不變，結果是比較差的，kaggle public：82.79%(83.45% with punctuations)，private：82.7%。有此可見標點符號像驚嘆號、問號，都對語氣的判斷上有影響

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。

我利用 self-training 的方法來做 semi-supervised learning。利用 model predict unlabeled data，且只要在 1, 0 的 threshold 附近(這邊我取 0.05，讓我的 semi data 可信度增加，以加強 data 的純度)，且在每 2 個 epoch 後，重新標記 unlabeled data，並進行 training。觀察 training 過程，可以發現很快就 overfitting，我認為是因為我們使用自己 predict 出來的 data 來進行 training，有 fitting data 的疑慮，造成 model 對沒看過的 data 沒有很大的辨別力。

結果：public：82.4%，private：82.3%，

就結果論準確率比 supervised 還差(public：83.45%，private：83.38%)