學號：B03901165　系級： 電機四　姓名：謝世暐

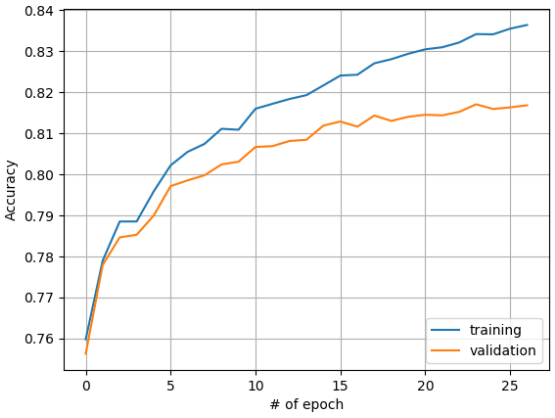
1. (1%) 請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何？
(Collaborators: b03901161 楊耀程 ; b03901001 廖宜倫)

答：

下圖是我使用的 RNN 模型架構和訓練過程，架構為 GRU(128)->GRU(128)->GRU(128)->Dropout(0.2)->Dense(512)->Dropout(0.2)->Dense(1,sigmoid)。Word Embedding 是用 gensim train，每個字 embed 成 32 維，使用所有 data (training、testing、unlabeled)來 train。Tokenize 方式為有包括標點符號。Validation 切 10%的 training data，Early Stopping 的 patience=2，Self-learning 用 prediction>0.9 和<0.1 的部分，用 predict 的 label 當作真實 label 將 unlabeled data 一起拿進來 train，使用的 optimizer 為 Adam，loss 為 binary_crossentropy。由於有用 early stopping，epoch 數即為 validation accuracy 第三次沒有繼續升高的時候。Train 完之後在作 Fine-Tuning 提高 validation 的準確率，Fine-Tuning 在每次 validation 正確率提高就會結束，我會重複跑個幾次直到無法得到更高的正確率。最後的 model 在 validation 正確率是 0.82635，public 的正確率是 0.82626。(下圖訓練過程的圖是 fine-tuning 之前的 acc-epoch 圖)

Layer (type)	Output Shape	Param #
gru_1 (GRU)	(None, 39, 128)	61824
gru_2 (GRU)	(None, 39, 128)	98688
gru_3 (GRU)	(None, 128)	98688
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 512)	66048
dropout_2 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 1)	513

Total params: 325,761
Trainable params: 325,761
Non-trainable params: 0



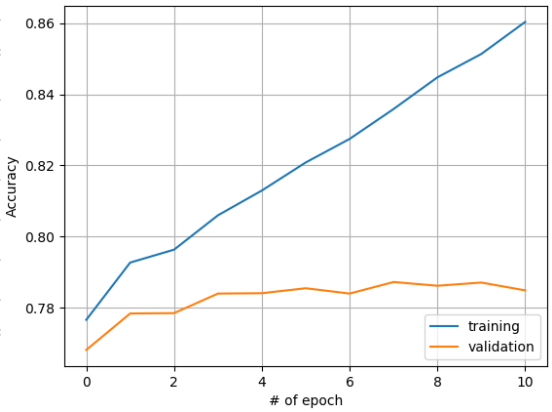
2. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何？
(Collaborators: 無)

答:

下圖是我的 BOG 模型架構和訓練過程，架構為 Dense(128)->Dropout(0.5)->Dense(256)->Dropout(0.5)-> Dense(256)->Dropout(0.5)-> Dense(1,sigmoid)。使用的 optimizer、loss、early stopping 方式皆與上題相同，但沒有再作 self-learning。Tokenizer 會只考慮前 2000 個最常出現的字(包含所有 data)，然後去幫每個句子數這兩千個字中每個字出現了幾次，因此每個句子的維度是 2000，轉換後的句子就直接丟進 DNN 裡 train。DNN 選用不會跟 RNN 的參數量差太多的模型，一樣 Validation 切 10%的 training data。Train 完在 validation 的正確率約 0.785。

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 128)	256128
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 256)	33024
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 256)	65792
dropout_3 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 1)	257

Total params: 355,201
Trainable params: 355,201
Non-trainable params: 0



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: 無)

答:

兩句的分數如下:

	today is a good day, but it is hot	today is hot, but it is a good day
RNN	0.03565 (negative)	0.97635 (positive)
BOG	0.82734 (positive)	0.82734 (positive)

兩句的預測在 RNN 有完全不同的結果，而在 BOG 如預期的結果完全相同。BOG 因為 encode 的方式是計算字出現的次數，只要字的個數、種類都相同結果就會一樣。RNN 的預測結果是前句負面、後句正面，若要試著解釋的話，以「雖然…，但是…」這樣的句型，通常語意上是想用雖然…的部分來強調但是…的部分。因此第一句想強調的是 hot，第二句是 good，也許 hot 比較負面而 good 比較正面，所以 RNN 預測出這樣的結果。BOG 的話由於是將句子中的字打包來看，因此可能就是某些字對於是否正面會有比較大的影響。我猜測應是 good 這個字，因此將 good 拿掉再嘗試預測，結果 BOG 的預測就變成了 0.40593(負面)，因此證明確實 good 這個字有很大的影響。

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: 無)

答:

就我的 model 和實驗結果而言，有包含標點符號比起沒有要來的好。以 fine tune 後的 training 結果，沒有包含標點的 model 達到了 validation 0.8136 的正確率，但有包含的則達到了 0.8248 左右的正確率(皆無做 semi-supervised learning 的情況)。若沒包含標點的再經過 self-learning，最高也只到 validation 0.8225 的正確率，而有包含標點的再經過 self-learning，則得到了 validation 0.82635 的正確率。雖然其中可能包含一些 random 因素，但總體上有包含會比沒有包含要好。

5. (1%) 請描述在你的 semi-supervised 方法是如何標記 label，並比較有無 semi-supervised training 對準確率的影響。
(Collaborators: 無)
答:

我用的 semi-supervised learning 是 self-learning。我首先先 train 一個沒有用 self-learning 的 model 版本(只用 training-data)，再用這個 model 找出在 unlabeled data 的 prediction>0.9 或是<0.1 的部分，將這部分的 prediction 當作 label 加進原本的 training data(最後共約 790000 筆)，再重新 train 一個新的 model(方式如前)。準確率上與沒有作 semi-supervised training 的 model 相比，沒有用標點的 model 從 0.8136 升到 0.82255，而有標點的從 0.8248 升到 0.82635，因此確實有所進步。