HW4: Text Sentiment Classification

學號：R06922030 系級： 資工碩一 姓名：傅敏桓

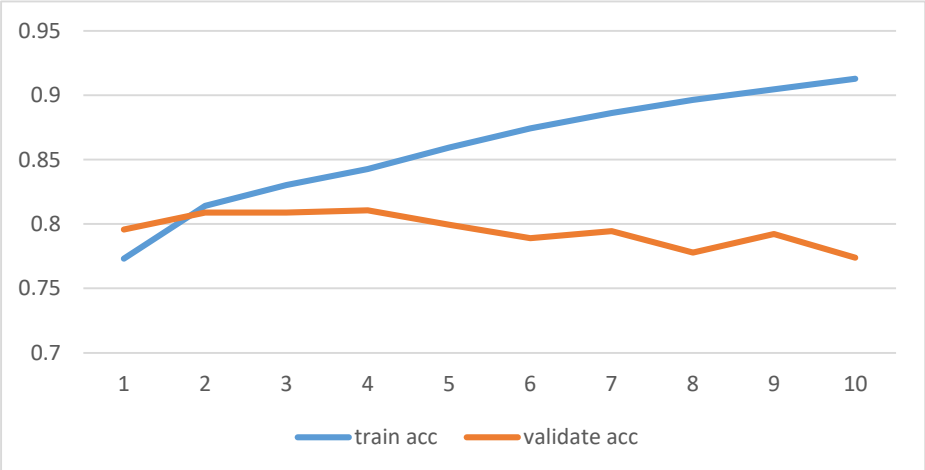
1. **請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何？**

(Collaborators: 無)

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 40, 100)	19195500
lstm_1 (LSTM)	(None, 40, 256)	365568
lstm_2 (LSTM)	(None, 256)	525312
dense_1 (Dense)	(None, 64)	16448
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65

本次作業實作的 RNN 模型如上圖所示。模型首先接收維度 40 的整數序列作為輸入向量，經過詞嵌入層後輸出維度 (40, 100) 的詞嵌入向量，再通過兩層 256 維的 LSTM 層、一層 64 維的全連接層和 Dropout 層 (Dropout 係數 = 0.5) 後，通過 sigmoid 函數得到最後的輸出。前處理的部分包含去除 3 個以上連續出現的字母、簡易的詞幹提取，但保留標點符號，最後以全部的訓練資料建立字典，詞彙數量總計為 191,955。

模型的訓練過程如下圖所示，切有標記訓練資料的最後 20%作為驗證集，在驗證集上最高準確率為 81.06%。模型使用交叉熵作為損失函數，以 RMSProp 進行參數更新，每次訓練的批大小設為 128，更新 10 個 eopchs 取在驗證集上準確率最高之模型參數。從訓練過程中準確率的變化，可以觀察到模型在訓練後期逐漸出現過度擬和之現象。



本次作業也使用未標記資料進行半監督式學習，得到新的訓練資料後再訓練 3 個新的架構相似的模型 (詳細作法參考 5.所述)，預測時透過 ensemble 投票的方式決定最後的預測結果。這些模型只有在 LSTM 維度的部分和原本的模型相異，維度分別是

160、192 和 224，訓練的方法同 1.所述，過程如下。可以觀察到模型在訓練資料上的準確率變得很高，推測是因為新加入的資料包含相似模型的預測結果，對新模型來說相對較容易準確預測。模型在 Kaggle 上最後得到 private 分數 0.81684。

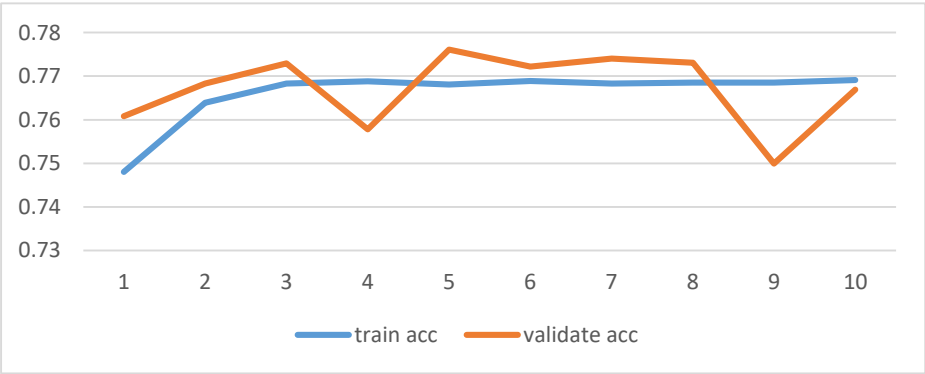
LSTM dim.	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
160	.959/.813	.972/.815	.974/.814	.975/.813	.976/.814
192	.959/.814	.972/.814	.974/.814	.975/.815	.976/.815
224	.959/.814	.972/.813	.974/.814	.975/.813	.975/.814

表一、半監督學習模型訓練過程（train acc / validate acc）

2. **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)

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 64)	1280064
dense_2 (Dense)	(None, 128)	8320
dense_3 (Dense)	(None, 256)	33024
dropout_1 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 1)	257

本次實作的 BOW 模型如上圖所示。模型接收維度 20,000 的 BOW 向量作為輸入，經過三層全連接層和 Dropout 層（Dropout 係數 = 0.5）後通過 sigmoid 函數得到最後的輸出。前處理部分和 1.雷同，惟考量記憶體大小限制，字典詞彙量上限設定為 20,000，遇到 OOV 會自動忽略，並採用計數方式產生 BOW 統計矩陣。訓練過程如下圖所示，訓練方法同 RNN 模型、也取同樣的驗證集，在驗證集上最高準確率為 77.61%。可以發現 BOW 模型雖然相較於 RNN 而言架構簡單不少，但在這次的問題上也能夠有不錯的表現。



3. **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)

以原始的 RNN 模型對上述句子進行預測，得到的分數分別為 0.525 以及 0.965；以 BOW 模型預測之結果則同為 0.627。BOW 模型只考慮各句出現的各個單字總數，而上述句子雖順序不同、但用字並無差異，也如預期在 BOW 模型獲得相同的分數。在 RNN 的部分兩者的分數則有顯著之差異，顯示 RNN 模型對於輸入序列的時序關係較為敏感，在進行預測時還會考慮時序較前的輸出回饋。觀察以上的例子，推測 RNN 模型可能有學到逆接關係的句子中，連接詞後的子句才是該句的重點所在，因此給後者更高的分數。(但可能因為句子包含 happy 一詞，以致模型仍然都偏向預測正向情緒)

4. **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)

上述之 RNN 模型在前處理時並未去除標點。以同樣的模型設定、前處理時不考慮標點的模型作為對照組，結果如下表所示。可以發現有考慮標點符號的模型準確率略高，推測某些標點可能常與特定情緒標籤共同出現；且驚嘆號、問號這類標點在文章中本來就有加強表達語氣或情緒的功能，尤其在非正式的文體中常被大量使用。

RNN (考慮標點)	81.06%
RNN (不考慮標點)	80.36%

5. **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: 無)

透過 1.所述之模型對未標記資料進行標記後，選輸出值大於 0.8 的資料作為正向、小於 0.2 的資料作為負向情緒的資料，得到有信心的標記結果共計 845,267 筆。在維持驗證集不變 (取訓練集末 20%，共 40,000 筆) 的狀況下，和原本的訓練資料合計 1,005,267 筆，再以這些資料訓練新的模型。

由於資料數目較為龐大，我們稍微減少模型的參數 (LSTM 維度分別降至 160、192 和 224) 以縮短訓練時間，訓練過程如上頁表一所示。原始 RNN 模型在同樣的驗證集上得到的準確率最高為 81.06%，相較之下半監督學習 RNN 模型準確率最高為 81.48%，顯示這樣的半監督學習方式可能對於模型的表現略有提升，但進步並不顯著。

RNN	81.06%
RNN + 半監督學習	81.48%

備註：在 Kaggle 實際分數最高的預測 (private 分數 0.81842) 是用 7 個模型做 ensemble 的結果，考量到運行時間限制，故最後繳交的作業只用 3 個模型做 ensemble。
使用套件：numpy, keras, genism 及 python standard library