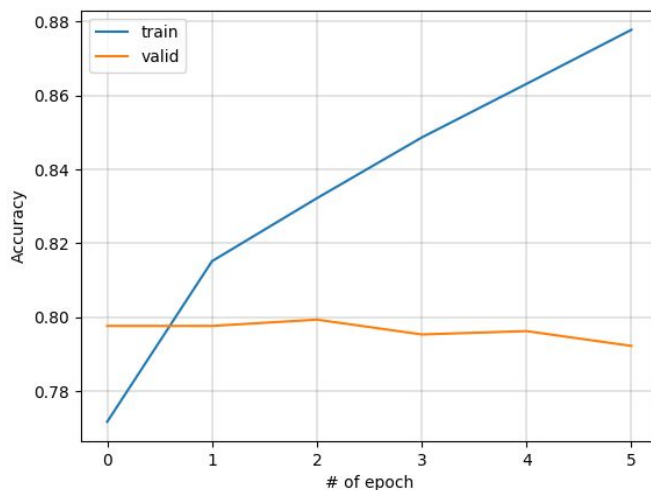


學號：B04501002 系級：土木三 姓名：謝至宥

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: )

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 40)	0
embedding_1 (Embedding)	(None, 40, 128)	2560000
lstm_1 (LSTM)	(None, 512)	1312768
dense_1 (Dense)	(None, 256)	131328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 1)	257
Total params: 4,004,353		
Trainable params: 4,004,353		
Non-trainable params: 0		

答：

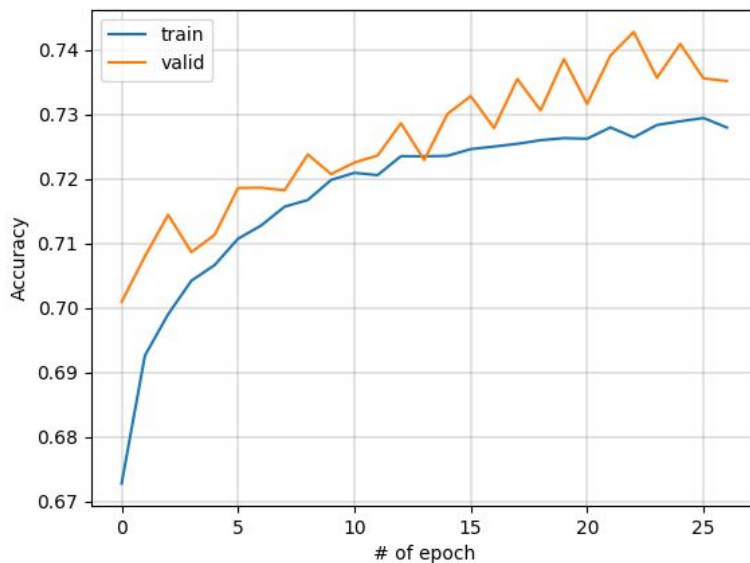


這是使用simple RNN LSTM模型，LSTM和Dense層之間的dropout rate都是0.3，單字共有2萬個，optimizer是adam，loss function使用keras的二元交叉熵，early stop patience為3，訓練時validation的正確率都不太會上升，0.8差不多是最高了，很容易就被early stopping中止，這個模型跑5個epoch後就停止了，有試著調各參數，但是都沒看到明顯的改善。

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: )

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 20000)	0
dense_1 (Dense)	(None, 256)	5120256
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 1)	257
Total params: 5,120,513		
Trainable params: 5,120,513		
Non-trainable params: 0		

答：



BOW的dropout rate、單字數量、optimizer、loss function還有early stop都和simple RNN一樣，總參數量還讓它比RNN多，有五百多萬個，訓練時正確率就像以前做的各種DNN一樣會持續上升，到第27個epoch時被early stopping中止，最後表現成果其實比RNN還要差，可見RNN具有記憶順序的特性是重要的。但是訓練RNN每個epoch所需的時間約是BOW

的7.5倍，可見LSTM梯度的計算量多很多。

比較特別的是觀察到validation的正確率幾乎都比train還高，應該是剛好取到語意正負評價差距都較大的validation資料。

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：使用BOW和RNN得到這兩句話的分數分別為(0.62, 0.62)和(0.32, 0.91)，因為BOW只會檢查哪些字出現過，這兩句話出現的單字完全一樣，對BOW而言輸入是一樣的。RNN輸入時保留了單字出現的順序，所以不同的順序就可以得到不同的輸出，就對這兩句話而言，其結果分別為負面和正面。

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：有標點符號的我是將keras的Tokenizer filters設為空的，其他各項設定和之前一樣，訓練兩個將正確率取平均為80.5675%，沒有標點符號的就用預設的filters，也是訓練兩個取平均，結果為79.925%。根據結果，保留標點符號可以提昇約0.6%的正確率，可能是標點符號其實也是包含情緒的，文章是從twitter蒐集來的，我們日常打字經常就會用各種符號來表達心情，可能是這個原因使正確率提高的。

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: )

答：訓練時threshold使用0.1，也就是把預估結果 $<0.1$ 和 $>0.9$ 的資料轉成0和1，再當成training data訓練模型，每次跑兩個epoch，結果正確率確實有提昇。我用第一題在第二個epoch得到的模型再做semi-supervised，public 和 private的正確率分別由0.80047上升到0.80333和0.80248上升到0.80324，有上升但是幅度極小。可能是無標籤和有標籤的資料很類似，所以就算有新的資料，實際上也沒有太多新的東西可以學習，所以只能提高一點點。