

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:黃禹程 R06944034、鄭克宣 R06921083、丁縉楷 R06922129)

- *LSTM+DNN*：架構細節如兩張截圖所示
- *Word2vec* 自行造 *embedding matrix* 所需之 *vector*
- 統一以 'i' 該 word 作為 *padding* 與出現次數過低之 word 的 *vector*
- 參數：*batch=128*、*epoch=2*、*opt=adam*、*droprate=0.5*、*validation_split=0.05*
- *ensemble* 兩 *model*：2 層 *LSTM* 之 *model*+3 層 *LSTM* 之 *model*
- 通過 *public* 與 *private* 之 *simple* 與 *strong baseline*

	PRIVATE 分數	PUBLIC 分數	平均
ENSEMBLE	0.82029	0.82204	0.82116
2 層 LSTM	0.81639	0.81763	0.81701
3 層 LSTM	0.81650	0.81853	0.81752

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, None, 100)	5529000
bidirectional_1 (Bidirection (None, None, 512))		731136
bidirectional_2 (Bidirection (None, 512))		1574912
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
dropout_3 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 512)	131584
dropout_4 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 512)	262656
dropout_5 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 1024)	525312
dropout_6 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 1024)	1049600
dropout_7 (Dropout)	(None, 1024)	0
dense_7 (Dense)	(None, 2)	2050
Total params: 10,003,370		
Trainable params: 10,003,370		
Non-trainable params: 0		

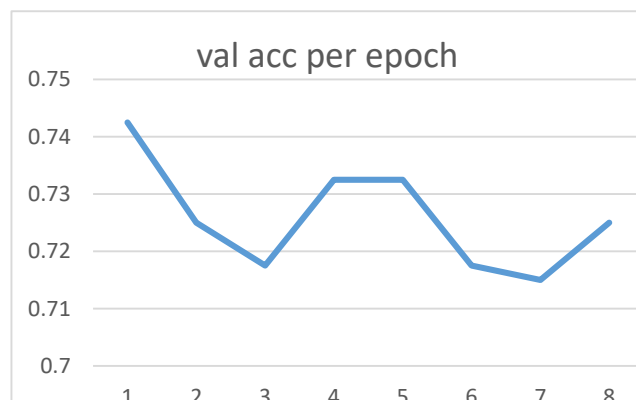
圖 1-1：2 層 LSTM

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, None, 100)	5529000
bidirectional_1 (Bidirection (None, None, 512))		731136
bidirectional_2 (Bidirection (None, None, 512))		1574912
bidirectional_3 (Bidirection (None, 512))		1574912
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
dropout_3 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 512)	131584
dropout_4 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 512)	262656
dropout_5 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 1024)	525312
dropout_6 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 1024)	1049600
dropout_7 (Dropout)	(None, 1024)	0
dense_7 (Dense)	(None, 2)	2050
Total params: 11,578,282		
Trainable params: 11,578,282		
Non-trainable params: 0		

圖 1-2：3 層 LSTM

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:黃禹程 R06944034、鄭克宣 R06921083、丁縉楷 R06922129)

EPOCH 數	VAL ACC
1	0.7425
2	0.725
3	0.7175
4	0.7325
5	0.7325
6	0.7175
7	0.715
8	0.725



- DNN 架構：架構細節如兩張截圖所示
- 參數皆與第一題相同，惟 epoch 數提升至 10 epochs (考慮其為 DNN 模型)
- 礙於 BOW 對顯卡記憶體太大，極易發生 memory error，故訓練時僅使用 8000 筆 train data，並以 val acc 代替 Kaggle 分數。準確率 74% 遜於 RNN 不少

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。

type	BOW	RNN
Good hot	[0.29307103 0.70692897]	[0.4754712 0.52545291]
Hot good	[0.29307103 0.70692897]	[0.16394244 0.83605754]

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 128)	32532352
activation_1 (Activation)	(None, 128)	0
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 128)	16512
activation_2 (Activation)	(None, 128)	0
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 128)	16512
activation_3 (Activation)	(None, 128)	0
dropout_3 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 2)	258
activation_4 (Activation)	(None, 2)	0
Total params: 32,565,634		
Trainable params: 32,565,634		
Non-trainable params: 0		

- BOW 兩句分數一樣，而 RNN 則略有差異，但其情緒判斷皆相同
- 原因：BOW 僅考慮一句話各個 word 所出現次數，並不在意順序所造成的差異，故兩句分數相同；反之，RNN 因初始設計即會考量 word 排列順序所影響

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: 黃禹程 R06944034、鄭克宣 R06921083、丁縉楷 R06922129)

有無標點符號	PRIVATE 分數	PUBLIC 分數	平均
有	0.82301	0.82460	0.82381
無	0.81639	0.81763	0.81701

- 模型架構、參數皆與第一題之 2 層 LSTM 相同 (礙於時間並沒有 implement ensemble)
- 惟 word2vector 與 tokenizer 之 filter 參數皆不去除任何標點符號
- 有標點符號之準確率高於無標點符號
- 原因：標點符號往往隱含表達意涵之轉折、變化，若囊括之則能較為精準判斷語意

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: 黃禹程 R06944034、鄭克宣 R06921083、丁縉楷 R06922129)

有無 SEMI	PRIVATE 分數	PUBLIC 分數	平均
有 SEMI	0.81471	0.81736	0.81604
無 SEMI	0.81639	0.81763	0.81701

- 模型架構、參數皆與第一題之 2 層 LSTM 相同(礙於時間並沒有 implement ensemble)
- 採最簡單之 semi-supervised，將所有 120 萬筆 unlabel data 預測出來(並無篩選 threshold)，並按照既有模型、參數 train 得到有 SEMI
- 有 SEMI 之效果略遜於無 SEMI，可能原因為沒有設 threshold，剔除較不明顯之 data