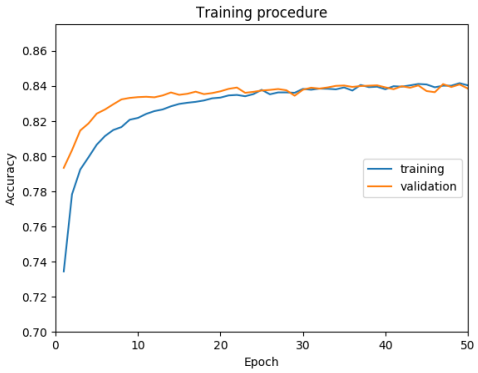
學號：b04901060 系級：電機三 姓名：黃文璁

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: 無) 答：

Layer (type)	Output Shape	Param #
gru_7 (GRU)	(None, 40, 512)	1181184
gru_8 (GRU)	(None, 512)	1574400
dense_10 (Dense)	(None, 512)	262656
dense_11 (Dense)	(None, 512)	262656
dense_12 (Dense)	(None, 1)	513

Total params: 3,281,409
Trainable params: 3,281,409
Non-trainable params: 0



此為單一模型中結果最好的，首先大致對文字資料進行下列預處理：

1. 只保留 “.?!” 三種標點符號和英文、數字。
2. 處理疊字問題。由於文字來源是 Twitter 故有許多疊字出現。
3. 處理基本的 stemming 和詞頻低於一定值的詞。

預處理後利用 gensim 的 **word2vec** 進行 word embedding，維度為 256。

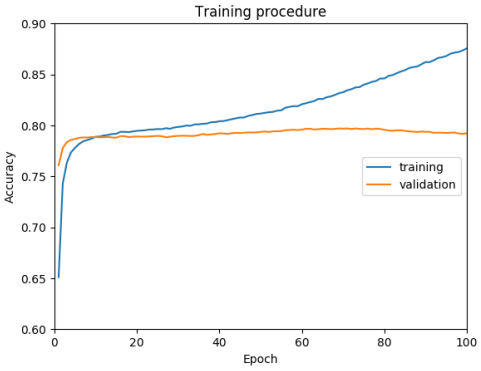
RNN 架構為：
2x GRU: units=512, dropout=0.5, recurrent_dropout=0.5
2x Dense: units=512, activation=selu
optimizer=adam, loss=binary_crossentropy

由上圖，training 和 validation accuracy 大概都在 25 個 epoch 時收斂到約 0.835。
Kaggle 上準確率為： **Public**: 0.83533 **Private**: 0.83348 | **平均**：0.83441
此外還實作了 10 個模型的 ensemble，其他模型和上述的模型參數有些許不同。
Kaggle 上準確率為： **Public**: 0.83947 **Private**: 0.83840 | **平均**：0.83894

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：

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 64)	273664
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65

Total params: 273,729
Trainable params: 273,729
Non-trainable params: 0



預處理大致同 RNN，另外還處理了 skipwords，BOW 的維度約為 4000。

BOW 架構為：
Dense: units=64, activation=relu
Dropout: rate=0.2
optimizer=adam, loss=binary_crossentropy

由上圖，validation accuracy 大概在 5 個 epoch 時收斂到約 0.79。
另外由圖中可以觀察到和 RNN 相比，BOW 的 overfit 情況更明顯。
本機測試 1/10 資料 validation 的準確率為：**0.79235**

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：

就我們看來，第一個句子稍偏負面，而第二個句子則較明確為正面。

		Prediction	Label
使用 **RNN** 來預測兩句話，得到的結果為：	第一句：	0.165759	0
	第二句：	0.992874	1
使用 **BOW** 來預測兩句話，得到的結果為：	第一句：	0.680686	1
	第二句：	0.680686	1

可以觀察到 RNN 對這兩句話的情緒都較為肯定（預測結果很接近 0 或 1）而 BOW 則顯然無法分辨這兩句話的情緒差異，這是由於兩個句子中的詞頻率相同。對 RNN 來說，會考慮到前後關係，故應能根據 'but' 的語氣轉折預測出正確的結果。而 BOW 則無法判斷前後文，故語氣有轉折的句子就有可能判斷錯誤。

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：

使用的 Model 同第一題。分別考慮標點符號的有無：

有標點符號： **Public**: 0.83406 **Private**: 0.83261 | 平均：0.83333

無標點符號： **Public**: 0.82390 **Private**: 0.82390 | 平均：0.82390

這是由於某些標點符號對文字的情緒有決定性的影響，最明顯的為 ! 和 ? 兩種。

例如有些句子去掉標點後會幾乎看不出情緒成分，但若加上驚嘆號的話就可以較明顯的看出情緒。而問號則關係到句子是否為疑問句，也可能對預測產生影響。除了這兩種外，刪節號 … 等符號也可能有一些影響。

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。**
(Collaborators: 無) 答：

從 unlabeled data 中取出 40 萬筆和 training data 不重複的句子，先利用 Kaggle 上準確率為 0.83349 / 0.83193 (平均 0.83271) 的模型對這 40 萬筆資料做預測得到 prediction，再保留預測值 >0.95 / <0.05 的句子，分別標上 pseudo label 1 / 0，再將這些句子和原本的 20 萬筆 training data 一起訓練和原本相同的模型。

經過上述處理後，semi-supervised training 時大概共有 47 萬筆資料，訓練後在 Kaggle 上的準確率為 0.83499 / 0.83411 (平均 0.83455)。平均下來大概進步了 0.2%。