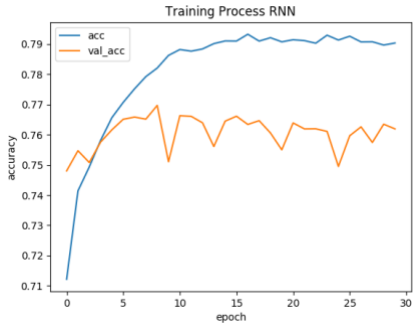
Machine Learning HW6 Report

學號：B05901003 系級：電機二 姓名：徐敏倩

1. (1%) 請說明你實作之 RNN 模型架構及使用的 word embedding 方法，回報模型的正確率並繪出訓練曲線*。

RNN 的模型，我疊了三層 bidirectional 的 LSTM，再加上三層的 Dense。前兩層 bidirectional LSTM 的 return_sequences 為 True，即輸出為三維，可再作為下一層 LSTM 的輸入，而第三層的 return_sequences 為 False，即輸出為二維，作為下一層 Dense 的輸入；且每一層 bidirectional LSTM 皆設其 dropout 為 0.3、recurrent_dropout 為 0.4。每兩層 Dense 之間會夾一層的 Dropout。Batch size 為 128、epoch 為 30、optimizer 使用 keras 內建的'rmsprop'，最後選擇在 validation set 上正確率最高的 model 來預測。

Word embedding 的部分，則是先用 jieba 將一句話分為一個一個詞，再將整個 train_x 和 test_x 一起當作 word2vec 的 input，設 size 為 128、min_count 為 1、iter 為 100。用此 embedding model 將句子中的詞轉為 vector，並截長補短使每一個句子的長度為 36。



Layer (type)	Output Shape	Param #
bidirectional_1 (Bidirection	(None, 36, 512)	788480
bidirectional_2 (Bidirection	(None, 36, 256)	656384
bidirectional_3 (Bidirection	(None, 256)	394240
dense_1 (Dense)	(None, 128)	32896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
dropout_2 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

Total params: 1,880,321
Trainable params: 1,880,321
Non-trainable params: 0

（左圖為模型正確率的訓練曲線、右圖為 RNN 模型的架構）

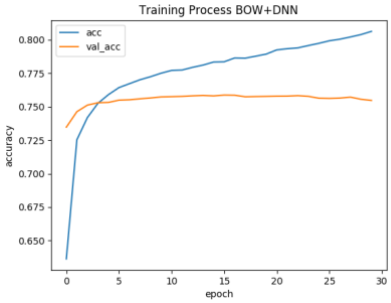
由上左圖可知，training set 的正確率會隨 epoch 的上升而增加，大約可到 80%左右，但在 validation set 的部分，正確率大約在 epoch 為 8 的時候就到了最大值，且整體的正確率大約都會在 75%~77%之間。Kaggle 上的 private score 為 0.76240、public score 為 0.75900。

2. (1%) 請實作 BOW+DNN 模型，敘述你的模型架構，回報模型的正確率並繪出訓練曲線*。

先訓練一個 word2vec 的 model，藉由 model 中的 function 將一個一個詞轉為 index；將每一句話轉為長度為 11695 的 list，其每一個數為該 index 所對應到的字在這句話中所出現的次數。

DNN 的模型，疊了四層的 Dense，每兩層之間夾一層的 Dropout(0.5)，每一層的 activation function 皆使用 sigmoid，每一層的 units 分別為 1024, 512, 128, 32, 1。Batch size 為 128、epoch 為 30、

optimizer 使用 keras 內建的‘rmsprop’，最後選擇在 validation set 上正確率最高的 model 來預測。



Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 512)	5988352
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 128)	65664
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 32)	4128
dropout_3 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 1)	33

Total params: 6,058,177
Trainable params: 6,058,177
Non-trainable params: 0

（左圖為模型正確率的訓練曲線、右圖為 DNN 模型的架構）

由上左圖可知，training set 的正確率會隨 epoch 的上升而增加，在 epoch 為 30 時已超過 80%，但在 validation set 的部分，正確率大約在 epoch 為 3 的時候就到了最大值，且整體的正確率都維持在 75%左右。Kaggle 上的 private score 為 0.75240、public score 為 0.75420。

3. (1%) 請敘述你如何 improve performance（preprocess, embedding, 架構等），並解釋為何這些做法可以使模型進步。

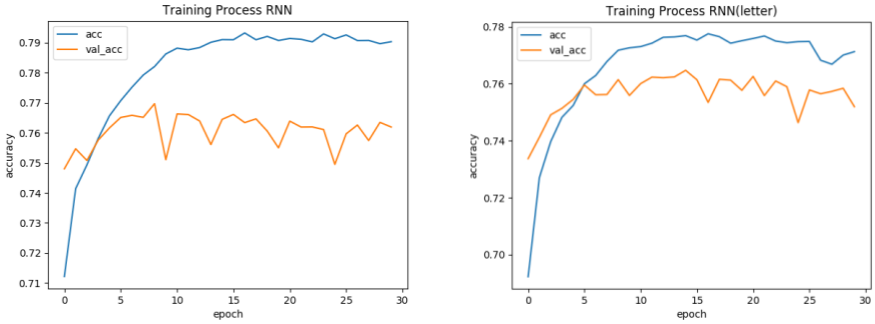
以第一題所述之 word embedding 的方式，建構六個不同的 RNN 模型，加上第二題所述之 DNN + BOW 的模型，分別對於 test_x 做預測，對這七個模型所預測出來的數據取平均，將大於 0.5 的資料歸為 class 1，其餘則歸到 class 0。

以 ensemble 七個相異 model（Kaggle 上的預測皆在 0.75 左右）的方式來提高預測的準確率（Kaggle 上的預測有 0.76），相較於 majority vote，取平均的方式，大概可使預測的準確率提升 0.001（Kaggle score）左右。

以這種 ensemble 的方式之所以能使預測的結果提升，來自於原本模型中預測出的結果較接近 0.5 的資料。對於一個模型預測出結果接近 0 或 1 的資料而言，ensemble 七個正確率皆在 0.75 的模型，其實並不怎麼會影響到其預測的結果；但是對於原本模型無法分辨的資料而言，ensemble 七個模型，即是考慮了來自不同的意見，再以此推斷出最合了分類，因此便可提高模型預測的準確率。

4. (1%) 請比較不做斷詞 (e.g., 以字為單位) 與有做斷詞，兩種方法實作出來的效果差異，並解釋為何有此差別。

以第一題所述之 RNN 模型，僅改變 word embedding 的方式，一個以詞為單位，一個以字為單位，可得以下兩種不同的結果。



（左圖為以詞為單位正確率的訓練曲線、右圖為以字為單位正確率的訓練曲線）

由上圖可知，在 validation set 上，兩種方式預測出來的結果相差並不大，皆在 0.75~0.76 左右，但在 training set 上，便可發現以詞為單位的模型訓練出來的結果較以字為單位的模型訓練出來的結果稍高一些。而在 Kaggle 的 score 上面，以詞為單位的模型不論在 private 抑或是 public score 上皆較以詞為單位的模型高約 0.005 左右。

由以上的結果可知，有無做斷句其實對於預測的結果影響並不大，其可能源於中文字本身一個字即可代表所表示的意義，因此以詞或是以字並不會大幅度地影響到預測的結果；但也有可能是因爲在使用 jieba 做斷句的時候，其實並沒有做得很好，導致兩種方式對預測的結果影響不大。

5. (1%) 請比較 RNN 與 BOW 兩種不同 model 對於 "在說別人白痴之前，先想想自己"與"在說別人之前先想想自己，白痴" 這兩句話的分數（model output），並討論造成差異的原因。

分別以第一題和第二題所述的模型預測題目所述的句子，可得以下的結果：

	RNN	BOW + DNN
第一句	0.51402760	0.58844334
第二句	0.42656228	0.58844334

由上表可發現，對於 BOW+DNN 的模型而言，兩句話所預測出的結果一模一樣，因為在做 BOW 時，我們只考慮句子中出現過的詞，而不考慮他們出現的順序，因此這兩句話預測出的結果並不會有任何的不同；但是觀察 RNN 模型時，便可知第二句的結果較第一句稍低，而這不同即是來自於在做 word embedding 時，一句話中字詞出現的前後順序，會產生出不同的 input，因此預測出的結果也會有所不同。