Machine Learning HW6 Report

學號：B05502145 系級：電機三 姓名：林禹丞

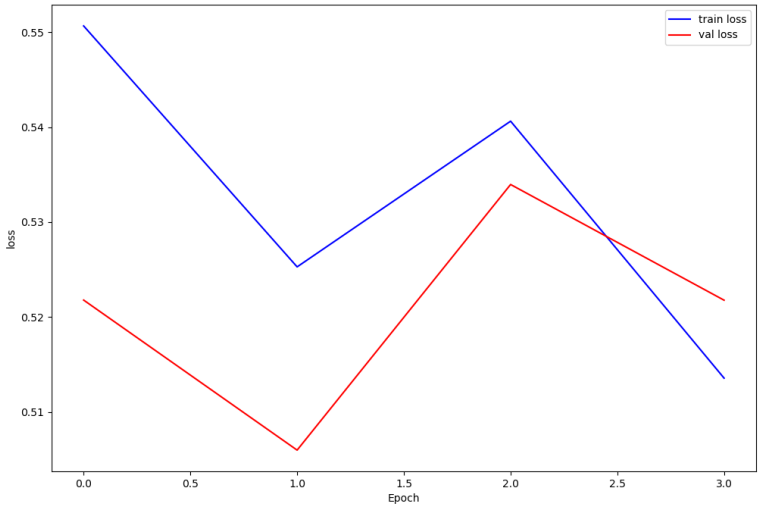
1. (1%) 請說明你實作之 RNN 模型架構及使用的 word embedding 方法，回報模型的正確率並繪出訓練曲線*

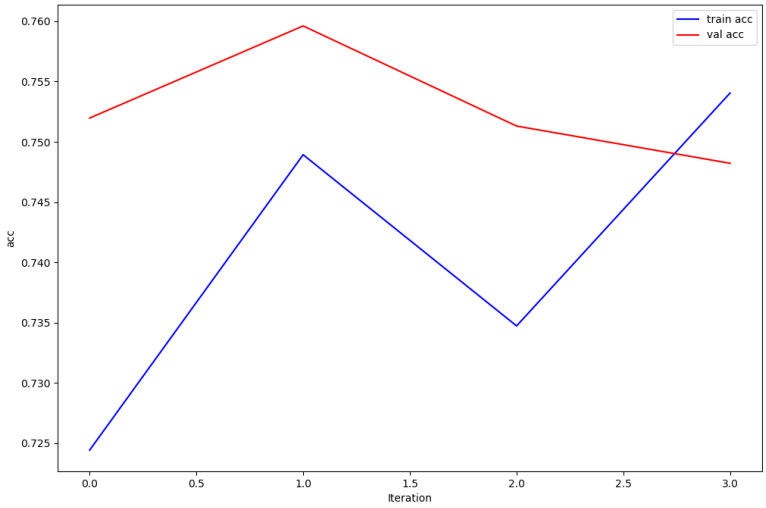
   RNN 模型架構：

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, None, 250)	7164000
bidirectional_1 (Bidirection	(None, None, 512)	778752
bidirectional_2 (Bidirection	(None, None, 512)	1181184
time_distributed_1 (TimeDist	(None, None, 256)	131328
dropout_1 (Dropout)	(None, None, 256)	0
dense_2 (Dense)	(None, None, 128)	32896
dropout_2 (Dropout)	(None, None, 128)	0
dense_3 (Dense)	(None, None, 64)	8256
dropout_3 (Dropout)	(None, None, 64)	0
dense_4 (Dense)	(None, None, 32)	2080
dropout_4 (Dropout)	(None, None, 32)	0
dense_5 (Dense)	(None, None, 16)	528
dropout_5 (Dropout)	(None, None, 16)	0
dense_6 (Dense)	(None, None, 2)	34

Total params: 9,299,058
Trainable params: 2,135,058
Non-trainable params: 7,164,000

Word embedding 方法：
我採用 gensim 套件，調整一些參數把 embedding 先 train 好，然後在 RNN 的架構裡面調成 nontrainable。
訓練曲線：





Kaggle 正確率：public(0.760)、private(0.75790)

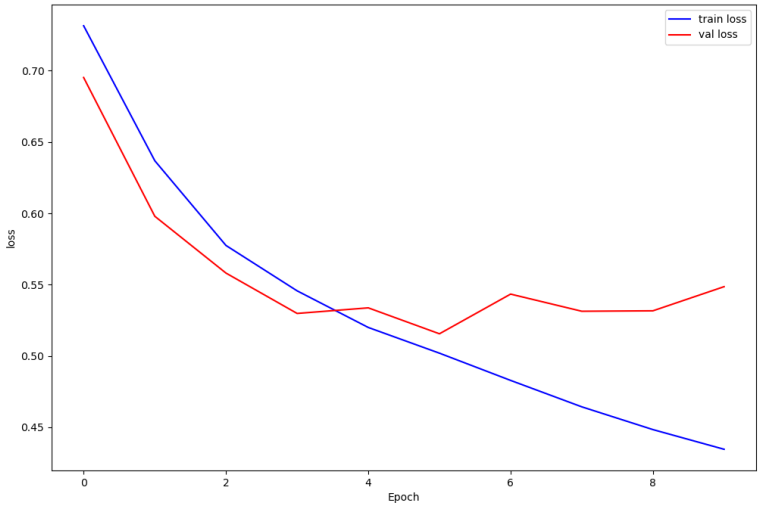
2. (1%) 請實作 BOW+DNN 模型，敘述你的模型架構，回報模型的正確率並繪出訓練曲線*。

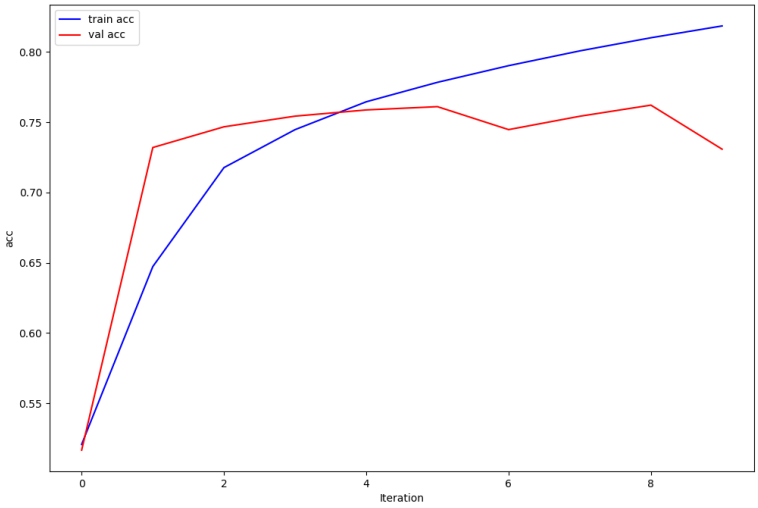
BOW+DNN 模型架構：

Layer (type)	Output Shape	Param #

dense_1 (Dense)	(None, 1, 256)	7380480
batch_normalization_1 (Batch	(None, 1, 256)	1024
dropout_1 (Dropout)	(None, 1, 256)	0
dense_2 (Dense)	(None, 1, 128)	32896
batch_normalization_2 (Batch	(None, 1, 128)	512
dropout_2 (Dropout)	(None, 1, 128)	0
dense_3 (Dense)	(None, 1, 64)	8256
batch_normalization_3 (Batch	(None, 1, 64)	256
dropout_3 (Dropout)	(None, 1, 64)	0
dense_4 (Dense)	(None, 1, 16)	1040
batch_normalization_4 (Batch	(None, 1, 16)	64
dropout_4 (Dropout)	(None, 1, 16)	0
dense_5 (Dense)	(None, 1, 2)	34

訓練曲線：





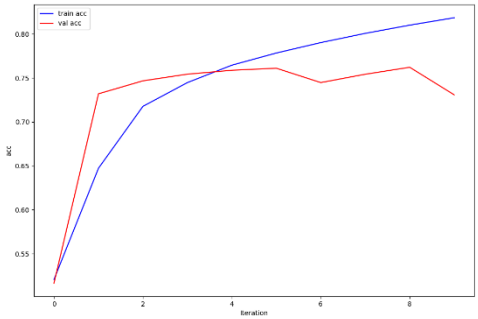
Kaggle 正確率：public(0.75140)、private(0.75500)

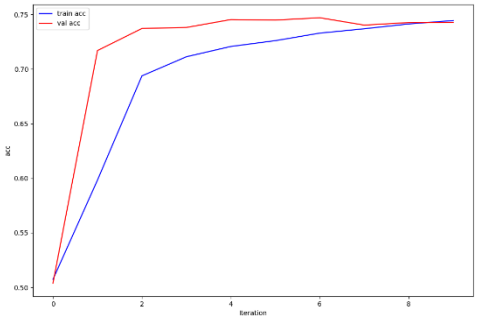
3. (1%) 請敘述你如何 improve performance（preprocess, embedding, 架構等），並解釋為何這些做法可以使模型進步。

   preprocessing：
   利用 re 套件，把重複兩個以上的符號刪掉；例如：？？變成？。結果發現訓練結果差不多，稍微變差。我認為符號也代表某種意義，如果把符號刪掉或許會損失了一點判斷的依據，導致訓練結果變差。
   GRU & LSTM：
   比較兩種不同的 Gate，發現結果沒什麼太大的差別。

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。

   在 BOW+DNN 架構上面比較這兩種方式可以發現：有作斷詞會有比較好的效果。可以觀察到左邊是有作斷詞右邊是以字為單位的訓練曲線(兩個的 word2vec 是步一樣的，其餘兩個 model 和參數都一樣)。





在 BOW 的方法下，以詞為單位比較能找出哪個字詞是帶有惡意的，以字為單位可能會有惡意和善意留言有同一個字在內的狀況，導致訓練結果不佳。另外，社群網站可能會有很多新的字詞，可能無法被準確切割，也導致結果不佳。

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。

Input / Model	在說別人白痴之前，先想想自己	在說別人之前先想想自己，白痴
RNN	0 with prob. 0.5376	1 with prob. 0.6266
BOW	1 with prob. 0.5617	1 with prob. 0.5617

因為 GRU 的特性，所以 RNN model 有辦法考慮到句子先後順序。因此 RNN 判斷出第一句不是人身攻擊而第二句則是人身攻擊。
BOW 的 model 只考慮了各種字詞的有無，所以對 BOW 來說這兩個 input 是一樣的。而因為這個句子有”白癡”，所以兩個句子都被判斷為人身攻擊。