

## Machine Learning HW6 Report

學號：b06902028 系級：資工二 姓名：林柏劭

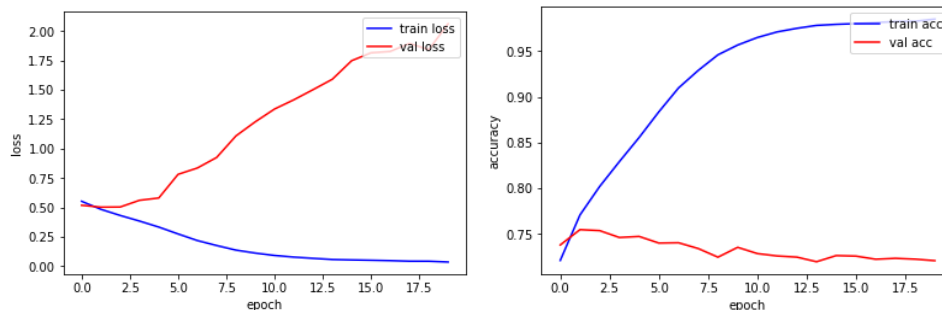
1. (1%) 請說明你實作之 RNN 模型架構及使用的 word embedding 方法，回報模型的正確率並繪出訓練曲線\*

RNN 模型架構如下

Embedding(trainable=True)
Conv1D(256,3,relu)
MaxPooling1D(2)
Dense(256,relu)
LSTM(256)
Dense(128,relu)
Dropout(0.5)
Dense(1,sigmoid)

Word embedding 我基本上是使用最基本的方法，先使用 jieba 分詞後，使用 word2vec 建立 gensim 的 model，使用這個 model 去建立一個 dictionary 與 embedding matrix，並將 embedding matrix 放入 RNN model 中一起 train。我將每一個詞的向量維度設為 200，一個句子最多取前面 100 個詞。

將 batch\_size 設為 256 後，訓練曲線如下



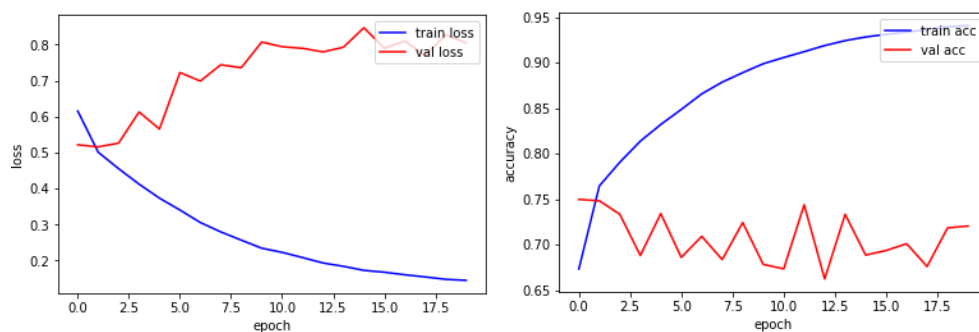
從上右圖可觀察到，大概 epoch > 2 以後，便開始 overfitting 了。最後我將 train 2 個 epoch 的上傳，得到 public score = 0.75880, private score = 0.75960 後來我將此 model 與其他先前較弱的 model 去做 ensemble，最後可以得到 public score = 0.76410, private score = 0.76310。

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\*

模型架構如下

Dense(512)
BatchNormalization
Activation(relu)
Dropout(0.5)
Dense(256)
BatchNormalization
Activation(relu)
Dropout(0.5)
Dense(128)
BatchNormalization
Activation(relu)
Dropout(0.5)
Dense(64)
BatchNormalization
Activation(relu)
Dropout(0.5)
Dense(1,sigmoid)

將 batch\_size 設為 128 後，訓練曲線如下



最後我將 train 2 個 epoch 的上傳，得到 public score = 0.74440, private score = 0.74370

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。

當初 hw3 作圖片辨識一樣, 每一個圖片是一個 3 維向量, 各自有不同的 feature, 可運用 convolution layer 去提取。而現在的句子轉為一個個 2 維向量後, 我認為這就跟圖片一樣(少一維), 每個句子都有不同的 feature, 因此我在 LSTM layer 前建構 CNN(圖片為 3 維, 句子為 2 維, 因此使用 Conv1D 與 Maxpoolong1D)。

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

我使用了一個簡易的 RNN model 來觀察

	public score	private score
不做斷詞	0.75480	0.75350
有做斷詞	0.74750	0.74030

可發現, 不做斷詞的結果會比有做斷詞差。推測是因為中文有一些詞, 若將詞中的一個字做改變, 便會大大改變整個詞的意思。譬如說「智障」與「智慧」, 前者是惡意, 後者是非惡意。

舉例來說, 假設現在 train data 中有很多「智障」這個詞出現, 若不做斷詞, 以字為單位, 則 model 會變成說看到每一個句子有「智」或「障」時便會認為這句為惡意。但若現在有一個 test data 為「你真有智慧」, 正常來說應是一個非惡意的句子, 但 model 此時看到句子中出現了「智」, 因此有可能會錯把此句誤認為惡意的。

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。

"在說別人白痴之前, 先想想自己" => 句 1

"在說別人之前先想想自己, 白痴" => 句 2

	句 1	句 2
RNN	0.45446083	0.6856756
BOW	0.45117775	0.45117775

對於句 1, 我的兩個 model 都沒判斷成功, 但可看到 RNN 表現還是稍比 BOW 好一點。但在句 2, RNN 模型便可成功判斷, 猜想或許是因為白痴一詞單獨出現, 對於判斷是否為惡意有幫助。但因為句 2 是句 1 重新排列後的成果, 因此對於 BOW 來說, 句 1 會等於句 2, 判斷的結果一模一樣