

Machine Learning HW6 Report

學號：B06902066 系級：資工二 姓名：蔡秉辰

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*

(1) model 架構：

我是使用 5 個 model 去做 ensemble，這 5 個 model 為：

第一個 model 如下：

Bidirectional(CuDNNGRU(128))

Bidirectional(CuDNNGRU(128)) + Dropout(0.1)

PReLU(units = 64) + Dropout(0.5)

PReLU(units = 32) + Dropout(0.5)

第二個 model：將兩個 CuDNNGRU 改成 CuDNNLSTM

第三個 model：將第一個 model 中的第一個 CuDNNGRU 改成 CuDNNLSTM，第二個維持不變

第四個 model：將第一個 model 中的第二個 CuDNNGRU 改成 CuDNNLSTM，另一個維持不變

第五個 model：將第一個 model 中的 DNN 的 activation 改成 selu。

以上所有 model 均有使用 ModelCheckpoint()和 ReduceLROnPlateau()。

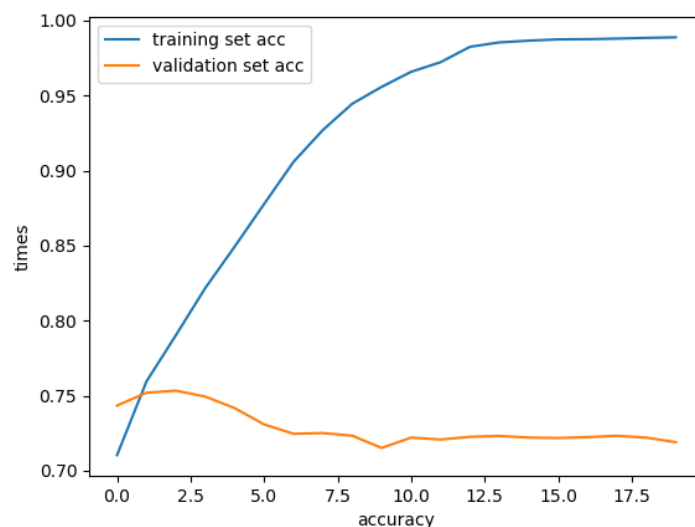
(2) model 正確率：

Public accuracy：0.76200

Private accuracy：0.75960

(3) 訓練曲線：

選第一個 model 去繪出其訓練曲線：



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*。

(1) DNN 建法為：

Dense(128, relu) + Dropout(0.5)

Dense(128, relu) + Dropout(0.5)

Dense(64, relu) + Dropout(0.5)

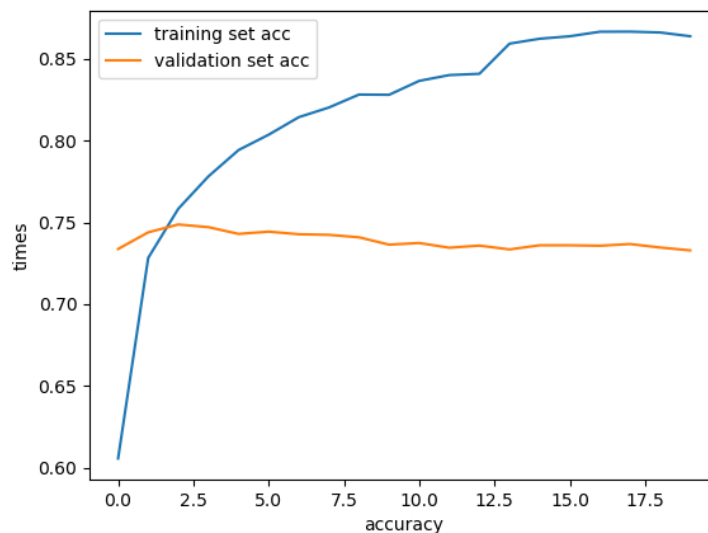
Dense(32, relu) + Dropout(0.5)

一樣有使用 ModelCheckpoint()和 ReduceLROnPlateau()

(2) 正確率：

	Validation acc	Private acc	Public acc
BOW + DNN	0.74875	0.74760	0.74780

(3) 訓練過程：



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。

(1) 從疊一層 LSTM/GRU 改成疊兩層 LSTM/GRU，約莫會使 performance 提升 0.001。此為 trial and error 得出之結果。

(2) 將 embedding layer 的 trainable 設成 True。舉例而言，可能原本 word2vec 做出來的兩個其認為的相近詞，但其實一個並非惡意言論，另一個卻是惡意言論，此時若將 embedding layer 變成 trainable，則會使表現好上 0.01~0.02。E.g.在我做出來的 word2vec model 中，若找和"拉基"的同義詞，"文組"會在第三名的位置。但明顯前者帶有惡意，但後者多數時候並非惡意言論。

(3) 使用 ensemble，會使預測準確率更高。

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。

因多數詞彙均為 1~2 個字，故我將原本 code 的 PADDING_LENGTH 從 66 增加成 1.5 倍，也就是 99 個字。(希望能維持取到"字"的數量相近)

以下為不做斷詞與其中一個有做斷詞的準確率比較：

	Validation acc	Private acc	Public acc
有斷詞	0.76040	0.75640	0.75780
無斷詞	0.75475	0.74710	0.75000

準確率有較低一點點，但沒有低到非常多。我認為是因為是我 model 是使用 RNN，且為 bidirectional，故仍然能大略判斷詞彙。但仍然比先切詞彙，再丟去 RNN 來得略劣一點。

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。

結果如下：(括號內表示 model 認為偏向{非惡意, 惡意}的比率)

RNN(僅取其中一個 model)：前句---(0.396, 0.604)，後句---(0.309, 0.691)

BOW + DNN：前句---(0.308, 0.692)，後句---(0.308, 0.692)

由以上結果可看出：

- (1) 因為 BOW 只在乎每個詞彙出現過幾次，故對於兩個句子而言，其判定都是一樣的，均認為此較偏向惡意留言，且機率高達 0.69。
- (2) 而對於 RNN 而言，因為考慮過詞彙順序，故其認為後者為惡意留言之機率高於前者許多。不過猜測是因為有"白痴"一詞，故縱使我認為前者應該算不上是惡意留言，但 RNN model 仍然將前者誤判為惡意留言。