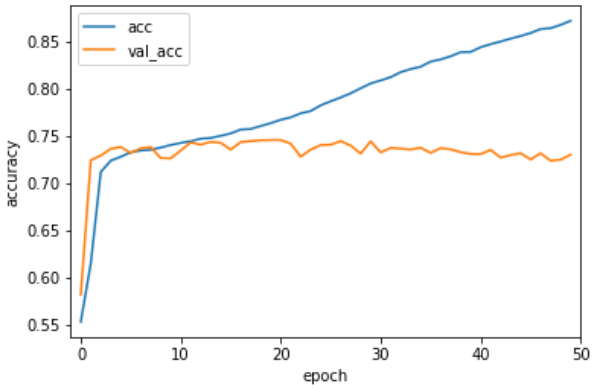
Machine Learning HW6 Report

學號：R07922108　系級：資工碩一　　姓名：陳鎰龍

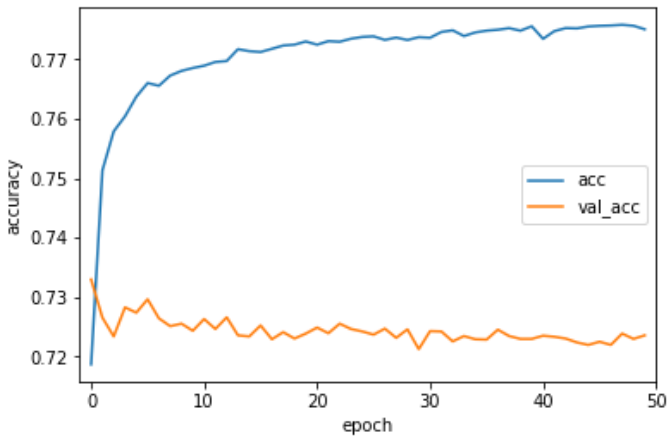
1. (1%) 請說明你實作之 RNN 模型架構及使用的 word embedding 方法，回報模型的正確率並繪出訓練曲線*

   RNN 模型架構為兩層 256 unit 的 STM，再接上兩層 256unit 的 Dense，最後再接上 1 個 neuron 輸出做 binary cross entropy。而我 word embedding 先將所有 train data 和 test data 用 jieba 斷句後做 padding，再丟進 Word2Vec API 建 model，word_dim 為 100、利用 skip-gram，最後將每個字用 word2vec 內建的函數轉成向量。正確率在 kaggle 上為 0.7453。



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*。

   DNN 模型架構為 Input – Dense 512 – Dense 256 – Dense 64 – Dense 1，因為詞種類太多所以我設定字典字數目最大為 10000，最後在 kaggle 上的正確率為 0.736，小輸 RNN 一些。



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。

   我在 preprocess 的時候發現有許多留言就是瘋狂打重複字，例如說大葉>>>(超多

個)>>>台大之類的留言，但是多餘的字詞對於模型判斷完全沒有幫助，所以對於大量出現的字詞，我有加以刪去，對於超過 padding 大小的字詞來說可以捕捉到更多有用的字詞。

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。

   使用 RNN 的情況下，val_acc 最高達到了 0.573，但 DNN 卻達到 0.665，可能是因為如果以字為單位，RNN 做 padding 後能看到的東西變少惹，但 DNN 還是能夠抓到特定帶有惡意的字，所有才有如此差異。

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。

   對於 RNN 來說，分數為 0.635 / 0.605，這是因為字的順序會影響 Model 的預測，所以分數會有差異，但是用人腦判斷我覺得後面那句應該比較惡意，所以理想來看後面應該要拿比較高的分數。對於 BOW 來講，兩者的分數都是 0.653，因為他是統計字出現的次數，字的順序對其並沒有影響，所以分數一模一樣。