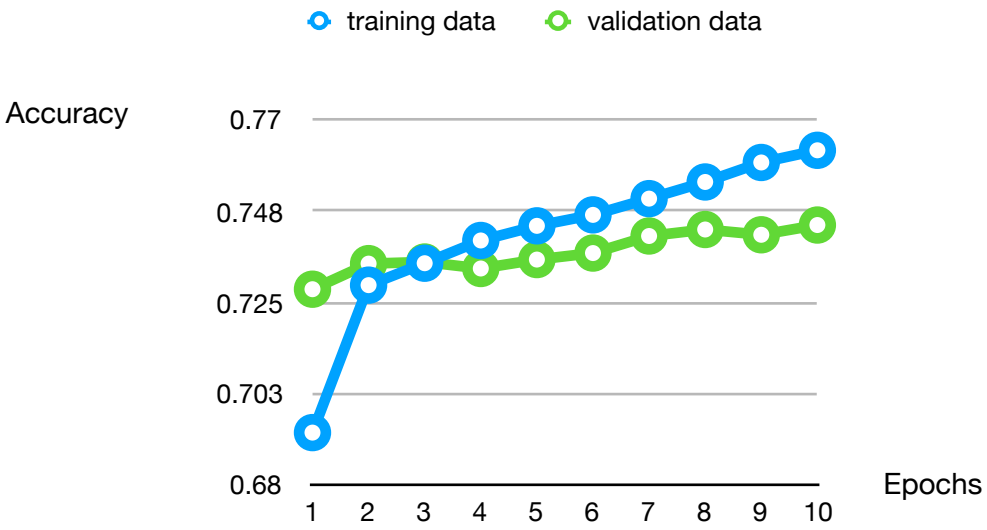
Machine Learning HW6 Report

學號：B06705057　系級：資工二　　姓名：黃資翔

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*

先利用 jieba 將 文字切成一個個的詞，再利用 word2vec 將每個詞找到一個對應的向量 (250 維度)，向量越近代表該詞語越相關。

模型架構 : GRU (64 個 neuron) -> Dense (128 -> 256 -> 512 -> 1)

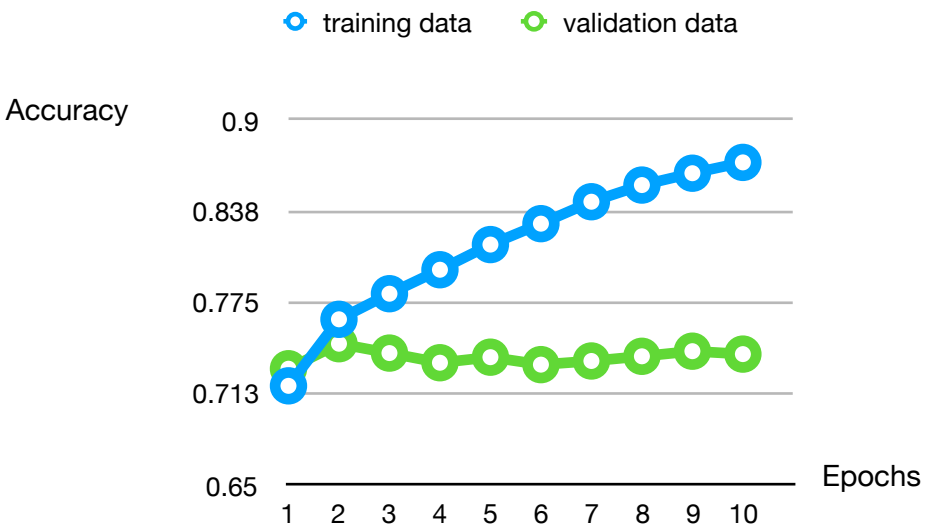


Public score : 0.74600　　Private score : 0.74700

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 * 。

只考慮頻率最高的 9000 個字存入 bag 中

模型架構 : Dense(32 -> 1)



Public score : 0.72530　　Private score : 0.72810

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. Embedding : 訓練不同模型進行投票，最終答案依照多數決
2. Preprocess : 因為 RNN 的關係，要讓每一個資料有相同的長度，因此短的句子可以前面補零，後面補零，或重複句子中的詞語。長的句子可以只保留前面的詞或是將句子拆成許多子句子進行預測，最終結果以分數平均為準。也可以改變 word2vec 中每個字的維度大小。這些方法配合 embedding 由於可以互相彌補長度要固定的限制，向量擷取更穩定，準確率可以再提升一些。

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。

以相同模型架構 GRU (64 個 neuron) -> Dense (128 -> 256 -> 512 -> 1)

	Public score	Private score
有做斷詞	0.74600	0.74700
沒做斷詞	0.73650	0.73790

不做斷詞的話準確率會略低一點，這結果蠻合理的，因為中文是一個詞為單位，很多單字不成意，或是一個字可以組合成數個截然不同語意的詞。

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。

句子一：在說別人白痴之前，先想想自己
句子二：在說別人之前先想想自己，白痴

	句子一	句子二
RNN	0.5319	0.6555
BOW + DNN	0.6190	0.6190

RNN 模型：由於模型輸出的分數跟詞語順序有關，句子二可以看出，白癡前面為逗號，逗號為停頓語氣，所以白痴可以獨立為一個句子，因此認為 ”白痴” 為罵人的意思。而句子一中的白痴，透過上下文順序關係，模型認為該句子沒有像句子二中的 “白痴” 那麼粗魯。

BOW + DNN 模型：由於模型輸出的分數跟詞語順序無關，只統計每個詞語的出現次數，兩個句子有相同的詞頻，因此分數會相同。而分數偏高可能是 “白痴” 為不雅字眼。