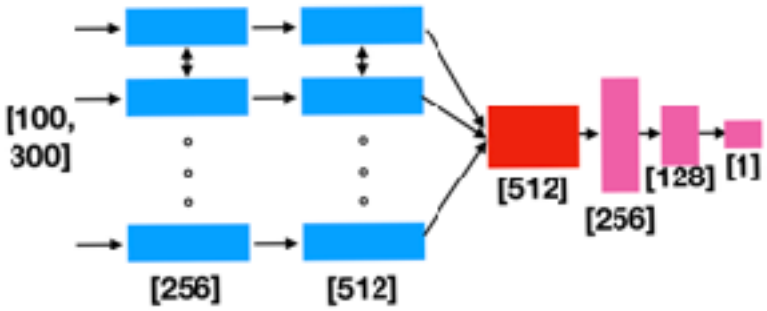
Homework4 Report

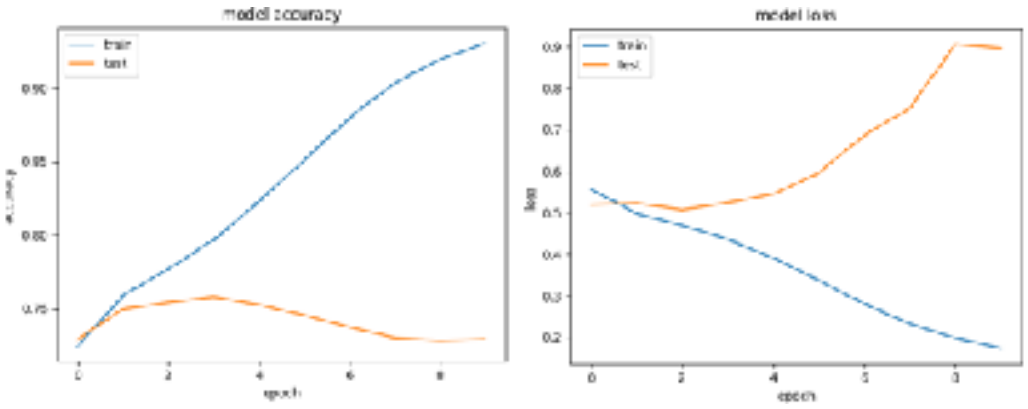
姓名：劉兆鵬 學號：r07921052 系級: 電機碩一

1. **(0.5%) 請說明你實作的之 RNN 模型及使用 word embedding 方法，回報模型的正確率並繪出訓練曲線．(0.5%)請實作 BOW+DNN 模型，敘述你的模型架構，回報正確率並匯出訓練曲線．**

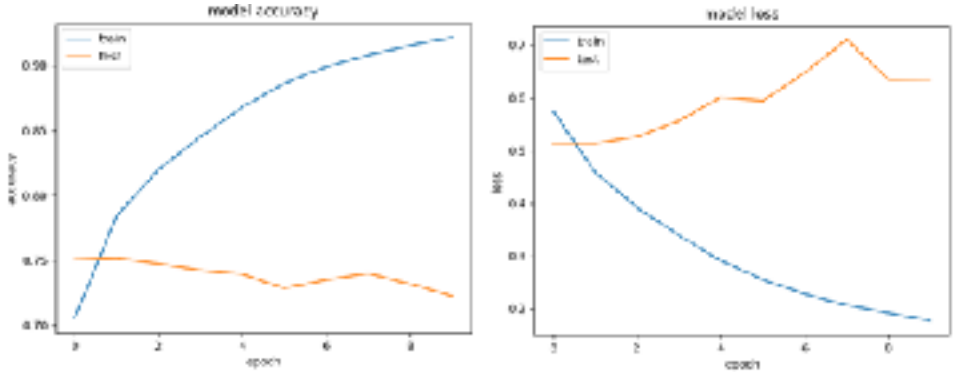
	RNN+embedding	Bow+DNN
Public Accuracy	0.75835	0.75260
Private Accuracy		



我使用的 word embedding 為 Word2Vec，首先我先通過 jieba 將訓練資料中的每句話進行斷詞，接著我使用Word2Vec來學習每個詞所代表的詞向量，其中我採用大小為 300 的向量，因為我看網路上最常使用的 Glove 等pretrained embedding 都是使用 300 維的詞向量能夠得到最好的結果．故我先使用 Word2Vec 將詞轉為向量並作為 keras embedding layer 的初始權重，並將此 layer 設定為 non-trainable，接著我經過兩層 bi-directional LSTM，因為我認為一句話的意思應該是需要經過其前後文才能得出來的．而由於一句話中，並非每個詞都能夠表示其為惡意的，故我在第二層 LSTM 的輸出經由 Attention model 來給予一句話中每一個詞的權重，來加強每一個詞語的意義，模型如上圖一所示．經過此模型設計可使準確率於 publiuc 為 0.75835， private 為 XXXXX，訓練曲線如下圖所示．



在實作 Bow 的部分我一樣先將句子使用 jieba 進行斷詞後，接著使用一個字典來存取每一個詞的位置，接著將一句話中每個詞出現的次數作為其句子的向量．故我將其訓練資料進行 Bow 轉換後共為 120000*30000，因為我的字典為出現最多的前 30000 個詞．接著再將其資料經由三層 Fully connected layer 進行訓練，其準確率在 public 為 0.75260，private 為 XXX，其訓練曲線如下圖所示．



2．(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．

在 preprocessing 的部分，一開始我先將文字進行斷詞後，保留了所有的文字作為訓練資料，而斷詞後的詞大概有 15 萬個左右，將此資料進行訓練後會發現結果不如預期，在將每個詞所出現的次數印出來看後，發現有許多的詞只出現過一次兩次而已，而出現十次以上的詞也只有二萬三千個詞而已，故有許多詞與是沒有訓練價值的，而我也將字典詞的數量從全部改為只使用前三萬字就能夠達到不錯的效果．另外，因為 jieaba 的斷詞辭典中，對於一些較新的詞語並無法辨認．例如姜太公就無法準確地進行切割，故我於程式碼中，將 jieba 加入一些我認為有意義的詞語就也能夠獲得好一點的效果．

而在一句話的長度方面，將所有的句子長度印出來可以發現最長的句子到一萬多個詞而最短的只有個位數個詞，且可以發現詞數很多的句子內容都是重複的內容，並沒有其實質的意義，故我將句子的長度從 300 逐漸往下降，發現在長度為100時能夠獲得較好的效果．

在 embedding 的部分，我使用 word2vec 作為詞的非監督式學習，因為將所有的字典經由 keras embedding layer 進行訓練會使得模型參數量過多而無法訓練起來．故根據文獻中所使用的 300 維向量來訓練 word2vec，再將此 word2vec model 所轉換出來的詞向量作為 embedding layer 的初始權重，就能夠減少 RNN 模型訓練的參數量，而達到更好的效果．

而在架構的部分，因為我認為一句話的語意是以一整句的內容來評斷，故我使用 bi-directional LSTM 而非使用單向的 LSTM，在經過實驗後，也能夠得到預期的結果．而因為在資料中一句話能夠表達其是否為惡意的句子時．我認為有些較有攻擊性的詞語應該獲得較多的分數，故我在 bi-directional LSTM 的輸出後面加上 Attention model，主要是給予每一個 LSTM 一個權重讓模型能夠更加注意哪一個部分能夠作為判斷的依據，最後再經由 DNN 來進行類別的分類，且因為此類別只有 0 與 1，故最後我使用 sigmoid 作為 Activation function．

3．(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．

	斷詞	以字為單位
Public Accuracy	0.75835	0.75607
Private Accuracy		

為了能夠公平的比較斷詞與不斷詞的方法，故我維持其相同的 preprocess 方法，embedding 以及 model 的架構，唯一改變的部分只有將使用 jieba 的部分改為以每一個字為單位做為訓練資料．發現使用斷詞作為訓練資料的準確率會較高一點點，並沒有太大的差別，我認為是因為對於中文字而言，每一個字都有其所代表的意義，且使用 LSTM 會記憶其前面所傳遞的訓練，故使用斷詞以及使用單一個字為單位並不會有太大的影響．而斷詞的結果比較高的原因，我想是因為有時候必須要是完整的一個詞才能夠表達其意義，但在做字典時，會因為單一個字出現數量較多，而其詞所配對的字並無被加入至字典中，而導致模型無法完全的理解應有的意義，而導致斷詞後的效果比較好．

4．(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)，並討論造成其差異的原因．

	RNN	BOW
第一句	0.32791	0.51508
第二句	0.39782	0.61516

在這兩句話的比較中，我的 RNN 無法準確的辨識這兩句話，我想可能是 RNN 在看過全部的訓練資料後，認為這兩句話並不代表為惡意的句子，可能是為要求別人而並非惡意，但是因為 BOW 是看句子中每個詞出現的次數來作為判斷的依據，我想因為 BOW 認定為說只要有 “白痴” 出現的句子可能都會判定為是惡意的，故 BOW 會將此兩句話作為惡意句子而 RNN 並沒有認定此二句為惡意．

----------Handwritten question----------

手寫題從下頁開始