

Machine Learning HW6 Report

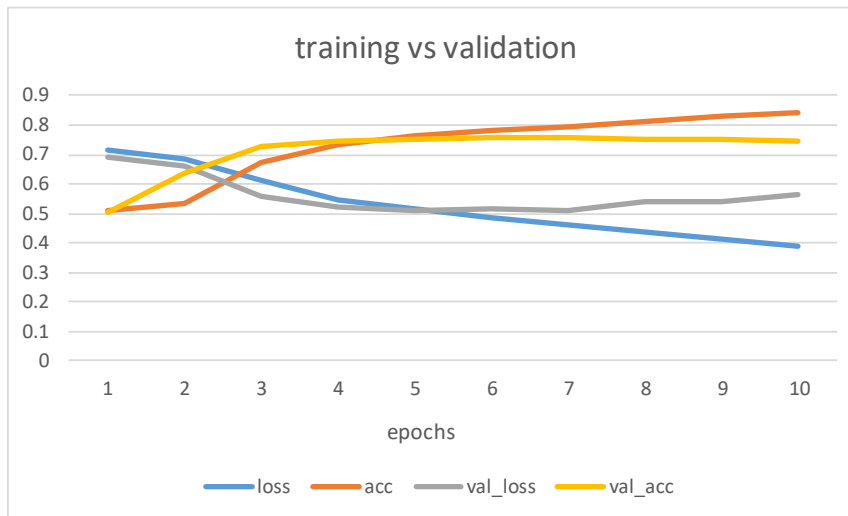
學號：B06705058 系級：資管二

姓名：劉品桢

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*

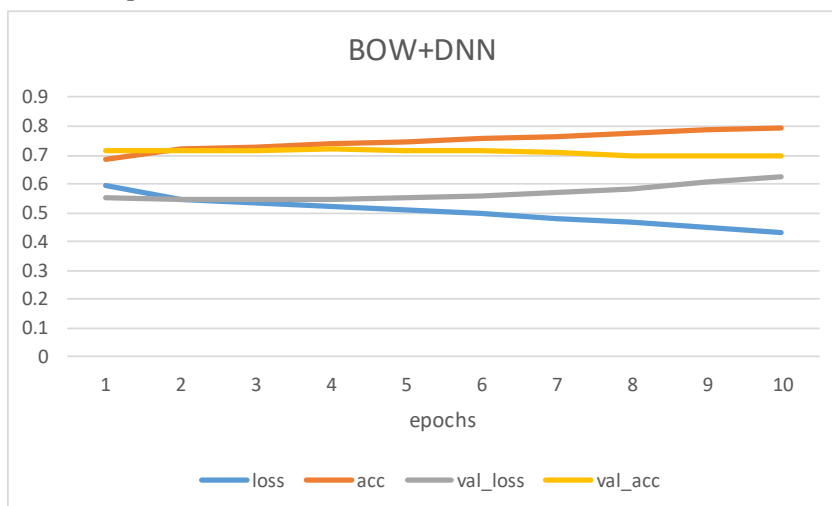
RNN 中使用兩層 GRU 分別為 64、128，一層 Dense 並且在中間進行 Dropout 避免 overfitting，正確率為 0.75880。

而 word embedding 則是使用 word2Vec 先進行詞向量訓練，然後將 trainable 設置為 true，使得 embedding 可以與模型一起訓練，缺點是非常容易 overfitting。



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*。

先做斷詞，然後將使用 tokenizer 進行 BOW，堆疊三層 DNN，分別為 256、512、1024，epochs = 10。



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。

一開始過 simple baseline 時並沒有將 embedding 跟著 model 一起 train , 在進行 strong baseline 時曾經調過各種 LSTM、GRU 的參數以及 Dense 的堆疊, 都沒有起很大的效果, 直到將 embedding 跟著 model 一起 train 才終於把準確率拉起來。原因可能是因為現在人講話很沒邏輯, 將 word embedding 與 model 一起 train 才能起到效果。

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。

RNN 不斷詞的準確率剩下 0.74430, 而且 training 時 loss 下降的速度變慢, 不斷詞代表是一個字一個字下去 train, 無法理解詞的意思, 因此準確率較差。

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。

BOW 中, 兩者的分數是完全相同的(都是 0.5671986)。

而在 RNN 之中, 前者的分數為 0.48316336、後者為 0.4295534。

再 BOW 之中因為沒有考慮文法、詞的順序, 因此對 BOW 來說兩個句子是完全一樣的, 因此會有相同的分數。

而 RNN 則有考慮順序及文法, 因此分數不同(雖然結果不盡理想)。