# Machine Learning HW5 Report

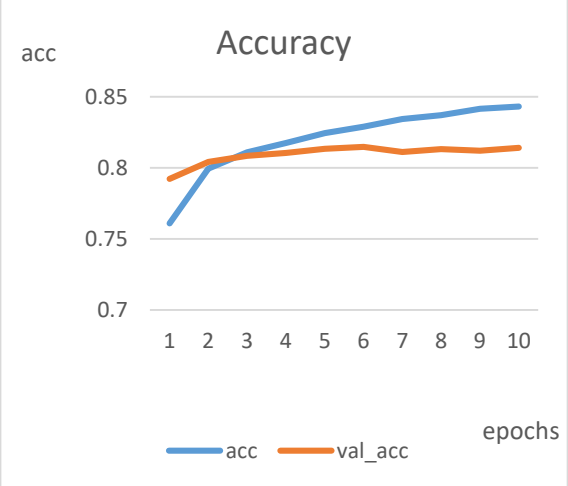
學號：B05611038 系級：生機二 姓名：張育堂

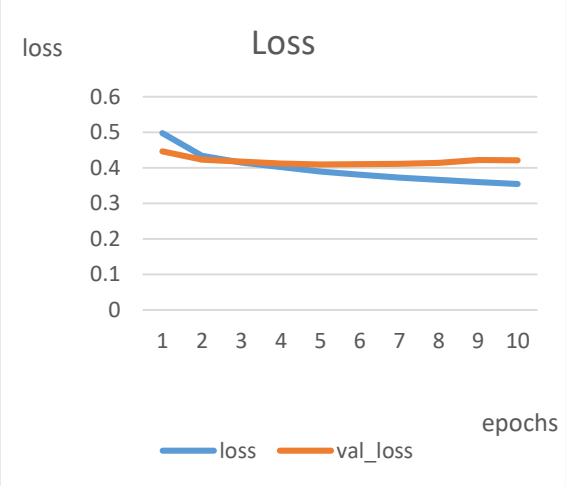
1. (1%) 請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何？
(Collaborators: No )

原本該在最上層的 Embedding layer，使用了 gensim 的 word2vec model 代替了，所以 input 是使用將每個字表示成 200 維的 vector，並且每句固定為 39 個字，所以 input 是(batch_size, 39, 200)的矩陣，並且沒有使用 semi-supervised learning。

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_1 (InputLayer) | (None, 39, 200) | 0 |
| gru_1 (GRU) | (None, 39, 512) | 1095168 |
| dropout_1 (Dropout) | (None, 39, 512) | 0 |
| gru_2 (GRU) | (None, 39, 512) | 1574400 |
| dropout_2 (Dropout) | (None, 39, 512) | 0 |
| flatten_1 (Flatten) | (None, 19968) | 0 |
| dense_1 (Dense) | (None, 1024) | 20448256 |
| dropout_3 (Dropout) | (None, 1024) | 0 |
| dense_2 (Dense) | (None, 512) | 524800 |
| dropout_4 (Dropout) | (None, 512) | 0 |
| dense_3 (Dense) | (None, 2) | 1026 |

Total params: 23,643,650
Trainable params: 23,643,650
Non-trainable params: 0



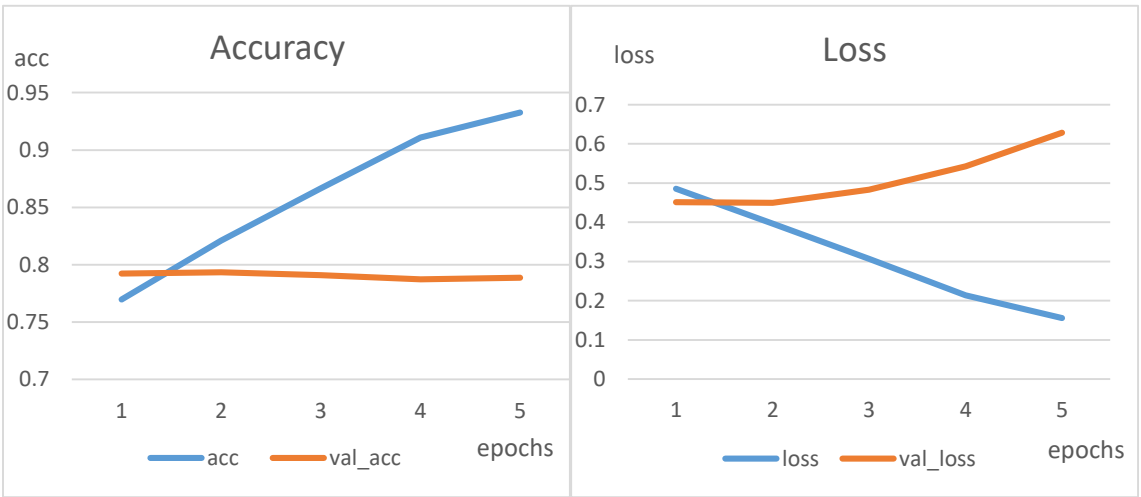
2. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何？
(Collaborators: b04611003 林宏揚)

BOW model 我使用了 keras 中 Tokenizer，先用 Tokenizer 建立字典並用其最常出現的 10000 個字當作 input vector，並使用一個單字在字句中出現的次數的形式訓練 BOW model。

其實由 training 的紀錄來看，自然語言的 model 常常有 overfitting 的現象，雖然 BOW model 的正確率在還可以接受的範圍，但是觀察 loss 就發現其實關於自然語言處理的 model 是不好訓練的。

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_1 (InputLayer) | (None, 10000) | 0 |
| dense_1 (Dense) | (None, 1024) | 10241024 |
| dropout_1 (Dropout) | (None, 1024) | 0 |
| dense_2 (Dense) | (None, 512) | 524800 |
| dropout_2 (Dropout) | (None, 512) | 0 |
| dense_3 (Dense) | (None, 256) | 131328 |
| dropout_3 (Dropout) | (None, 256) | 0 |
| dense_4 (Dense) | (None, 2) | 514 |

Total params: 10,897,666
Trainable params: 10,897,666
Non-trainable params: 0



3. (1%) 請比較 bag of word 與 RNN 兩種不同 model 對於"today is a good day, but it is hot"與"today is hot, but it is a good day"這兩句的情緒分數，並討論造成差異的原因。

(Collaborators: No)

RNN

```
Sentence: 'today is a good day, but it is hot' label : 1
Sentence: 'today is hot, but it is a good day' label : 1
```

BOW

```
Sentence: 'today is a good day, but it is hot' label : 1
Sentence: 'today is hot, but it is a good day' label : 1
```

不管是 RNN 還是 BOW model，這兩句話的 label 都是屬於正向的，我想實際上在 training 比較好的 RNN model 是可以將這 2 句話的情緒作分隔，因為第一句雖然講話委婉，但實際上情緒應該是屬於負向的。

至於 BOW model 是永遠也無法將 2 句話分成不同 label，因為它的原理是計算正向的字的次數以及負向的字的次數來進行預測，所以就算 BOW model 訓練的再好，也無法將 2 句字的組成一模一樣的句子做不同的 label，因為他們的 input 是一樣的。

4. (1%) 請比較"有無"包含標點符號兩種不同 tokenize 的方式，並討論兩者對準確率的影響。

(Collaborators: No)

| | 有包含驚嘆號、問號 | 無標點符號 |
|---|---|---|
| Public Score | 0.82265 | 0.82165 |

在做這題的時候，在做有標點的時候，我原本是想包含所有的標點符號的，但是我想在文字的標點符號中，通常也只有驚嘆號還有問號會有語氣上表達的不同，因此我在保留標點符號的時候，我選擇保留問號驚嘆號。

而實際上雖然差異非常微小，但是的確有保留標點符號的分數比較高，正如同上一段所預期的。而這也可以應用在以後要對自然語言處理時，不應該只考慮字和字的關聯，而且同時應該思考標點符號對余文本中所有句子語句有不同的影響。

5. (1%) 請描述在你的 semi-supervised 方法是如何標記 label，並比較有無 semi-surpervised training 對準確率的影響。

(Collaborators: No)

| | Supervised Learning | Semi-supervised Learning |
|---|---|---|
| Public Score | 0.81416 | 0.79074 |

我在 semi-supervised 中，是將當時最好的 model 將沒有 label 的資料進行預測之後，再將預測出的資料連同有 label 的資料進行 training。我想結果表現較差是可以預期的，畢竟在 label 的時候無法確認那些 label 的正確與否，我是有想過用助教建議的連續幾個 epoch 預測都相同的數據才進行 label 跟 training，但是因為這樣 training 曠日廢時，以後有機會遇到再試試看。