Machine Learning HW5 Report

學號：B06507002　系級：材料三　　姓名：林柏勳

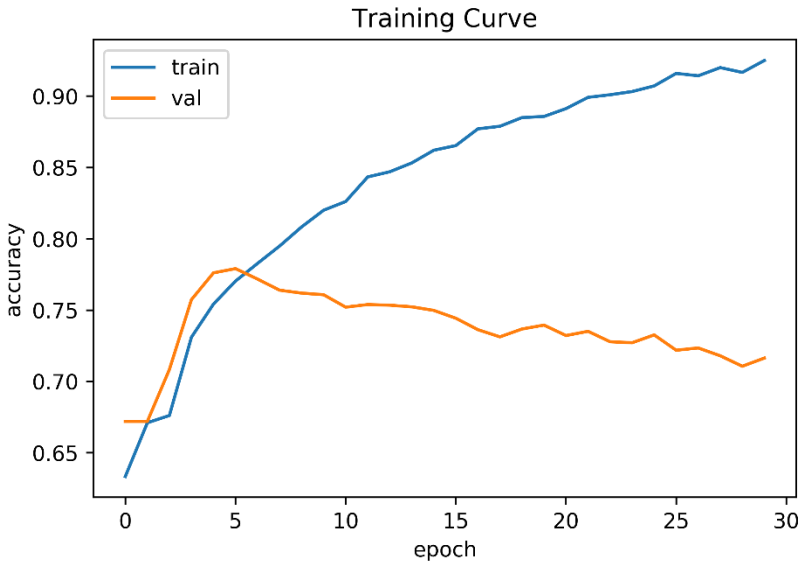
1. (1%) 請說明你實作之 RNN 模型架構及使用的 word embedding 方法，回報模型的正確率並繪出訓練曲線*

正確率: 0.779，模型架構如下，其中 bidirectional 部分是使用 LSTM 進行實作。

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 50)	0
embedding_1 (Embedding)	(None, 50, 300)	750000
bidirectional_1 (Bidirection	(None, 50, 16)	19840
global_max_pooling1d_1 (Glob	(None, 16)	0
dense_1 (Dense)	(None, 4)	68
dropout_1 (Dropout)	(None, 4)	0
dense_2 (Dense)	(None, 1)	5

模型參考: https://www.kaggle.com/sudalairajkumar/a-look-at-different-embeddings/notebook

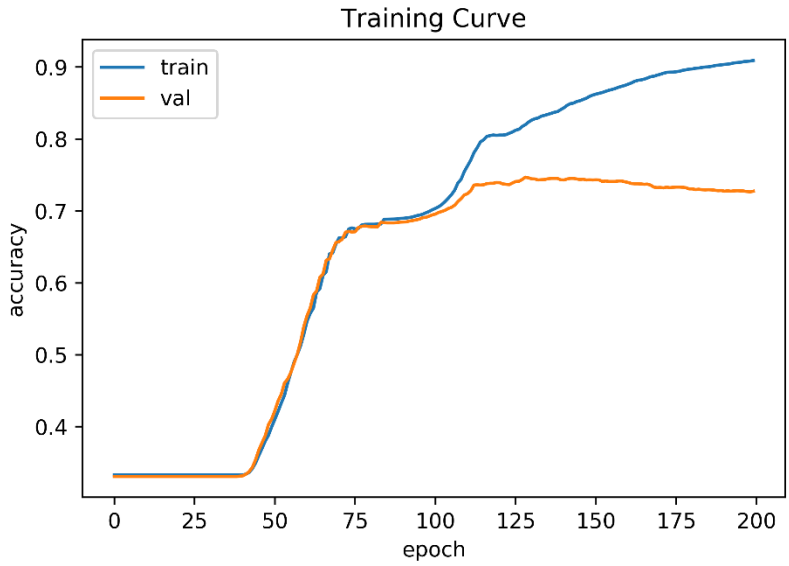
embedding 訓練方法為放在 classifier 的第一層跟 classifier 一起訓練，並沒有特別使用 word to vector 的方法，設定 embedding 的維度為 300。



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*。

正確率 0.747，模型架構如下

```
Net(
  (fc1): Sequential(
    (0): Linear(in_features=2324, out_features=32, bias=True)
    (1): LeakyReLU(negative_slope=0.01)
  )
  (fc2): Sequential(
    (0): Linear(in_features=32, out_features=16, bias=True)
    (1): LeakyReLU(negative_slope=0.01)
  )
  (fc3): Sequential(
    (0): Linear(in_features=16, out_features=8, bias=True)
    (1): Sigmoid()
  )
  (out): Linear(in_features=8, out_features=2, bias=True)
)
```



3.

以 LSTM 為例，當我將資料讀進來後直接用空白做分割時，訓練出來的 validation accuracy 為 0.756，而如果做進一步的資料處理，如: 刪除標點符號，刪除冗餘的字眼，如:@user 或 url 等等，可以將 validation accuracy 提高到 0.779 左右。

而我也試著實作 genism 的 word embedding 後，發現效果並沒有非常顯著，所以還是將 embedding 和 classifier 一起訓練。另外是嘗試不同的模型架構，發現 GRU 跟

LSTM 的結果是差不多的，不同的 RNN 模型並沒有對結果有顯著的差異，當然 RNN 模型會比 BOW+DNN 來的好的原因在於它有考慮字詞的前後順序。最後是用多個 model 做 ensemble，結果會比較好，原因在於不同 model 之間都有 variance，ensemble 可以降低這樣的 variance。

4. (1%) 請比較不做斷詞 (e.g.,用空白分開) 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。

	不做斷詞	有做斷詞(以 LSTM 為參照)
Validation accuracy	0.756	0.779
Public score	0.78837	0.79767

比較有無使用斷詞的模型分數，可以發現，有做斷詞的分數都比較高，原因在於去除標點符號與移除沒有意義的符號，可以有效去除雜訊，使得訓練起來的效果會比較好。

5. (1%) 請比較 RNN 與 BOW 兩種不同 model 對於 "Today is hot, but I am happy." 與"I am happy, but today is hot." 這兩句話的分數（model output），並討論造成差異的原因。

	"Today is hot, but I am happy."	"I am happy, but today is hot."
RNN	0.3296	0.3269
BOW	0.2456	0.2456

因為我在 RNN 是使用 binary cross entropy，而 BOW 中一般的 cross entropy，所以我將 BOW 的 output 通過 softmax 後再取第一維的資訊進行呈現，分數的意義均代表是惡意言論的機率。

由 output 的結果可以看出兩個 model 均判斷這兩句話都不是惡意言論，BOW 代表不受到字句的前後順序的影響，所以兩句話的分數一樣，而也因為兩句話都不是惡意言論，所以 RNN 的分數看起來差不多。

6. (2%)Refer to math problem

1.

```
import numpy as np
x1=[0,1,0,3]
x2=[1,0,1,-2]
x3=[1,1,1,4]
x4=[0,1,1,0]
x5=[0,1,0,2]
x6=[0,0,1,-4]
x7=[1,1,1,1]
x8=[1,0,1,2]
X=[x1,x2,x3,x4,x5,x6,x7,x8]
w=np.array([[0,0,0,1],[100,100,0,0],[-100,-100,0,0],[0,0,100,0]])
b=np.array([[0],[-10],[110],[-10]])
print("w=\n",w,"\nb=\n",b)
```

```
w=
 [[   0    0    0    1]
 [ 100  100    0    0]
 [-100 -100    0    0]
 [   0    0  100    0]]
b=
 [[  0]
 [-10]
 [110]
 [-10]]
```

將 $x_1,x_2,\ldots,x_8$ 的值輸入完畢後，使用將題目給定的w, w_i, w_f, w_o寫成一個 4*4 的矩陣 W，
b, b_i, b_f, b_o寫成一個 4*1 的矩陣 B。

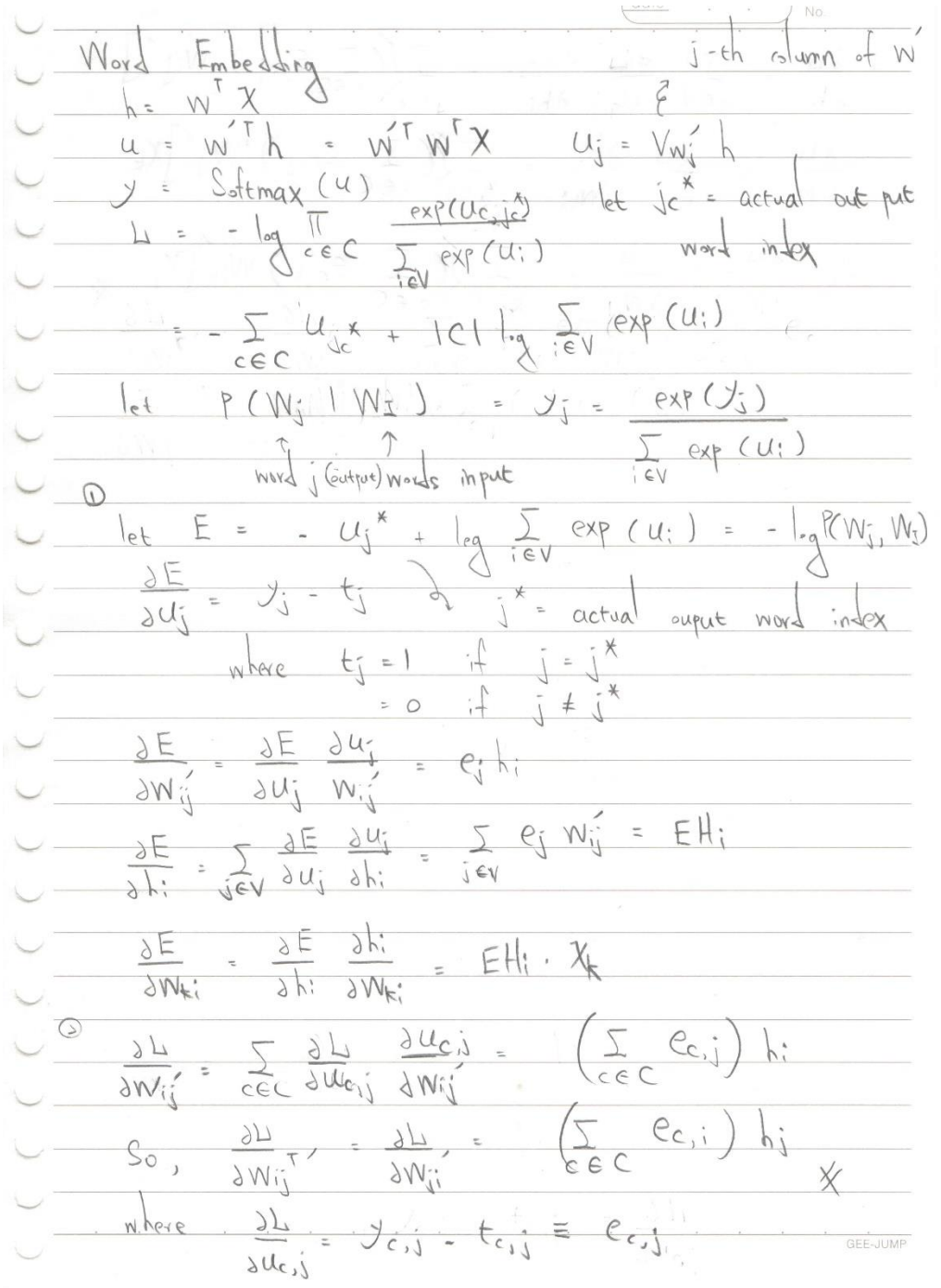
```
def f(x):
    return 1/(1+np.exp(-x))
def g(x):
    return x
def h(x):
    return x
c=0
for i in range(len(X)):
    Z=np.matmul(W,np.array(X[i]).reshape(-1,1))+B
    z,zi,zf,zo=Z
    c=f(zi)*g(z)+c*f(zf)
    y=f(zo)*h(c)
    print("At t= %d z= %d zi= %d zf= %d zo= %d c= %d y= %d"%(i+1,z,zi,zf,zo,np.round(c),np.round(y)))
```

```
At t= 1 z= 3 zi= 90 zf= 10 zo= -10 c= 3 y= 0
At t= 2 z= -2 zi= 90 zf= 10 zo= 90 c= 1 y= 1
At t= 3 z= 4 zi= 190 zf= -90 zo= 90 c= 4 y= 4
At t= 4 z= 0 zi= 90 zf= 10 zo= 90 c= 4 y= 4
At t= 5 z= 2 zi= 90 zf= 10 zo= -10 c= 6 y= 0
At t= 6 z= -4 zi= -10 zf= 110 zo= 90 c= 6 y= 6
At t= 7 z= 1 zi= 190 zf= -90 zo= 90 c= 1 y= 1
At t= 8 z= 2 zi= 90 zf= 10 zo= 90 c= 3 y= 3
```

根據題目給定的初始條件 c=0

定義函式 f、g、h 後，每一步的計算就可以轉變為$Z = Wx^t + B$，之後再根據題目定義的方法更新每一步的 c 值，就可以在進一步算出每一步 ouput 的值 y。
所以 ouput sequence 為 0,1,4,4,0,6,1,3，這邊假設通過 activation function 後的值非零即一，所以在打印出結果時，直接四捨五入每一步的 c 跟 y 值。

2.



$$\frac{\partial L}{\partial h_i} = \sum_{j \in V} \frac{\partial L}{\partial u_{c,j}} \frac{\partial u_{c,j}}{\partial h_i} = \sum_{j \in V} \left[\left(\sum_{c \in C} e_{c,j} \right) w_{ij}' \right]$$

$$\frac{\partial L}{\partial w_{ki}} = \frac{\partial L}{\partial h_i} \frac{\partial h_i}{\partial w_{ki}} = \sum_{j \in V} \left[\left(\sum_{c \in C} e_{c,j} \right) w_{ij}' \right] x_k$$

$$\therefore \frac{\partial L}{\partial w_{ij}'} = \frac{\partial L}{\partial w_{ji}} = \sum_{k \in V} \left[\left(\sum_{c \in C} e_{c,k} \right) w_{ik}' \right] x_j \quad \ast$$