

Homework 4 Report Problem Set

Professor Pei-Yuan Wu

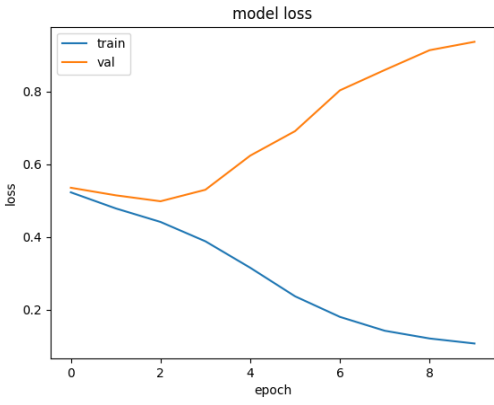
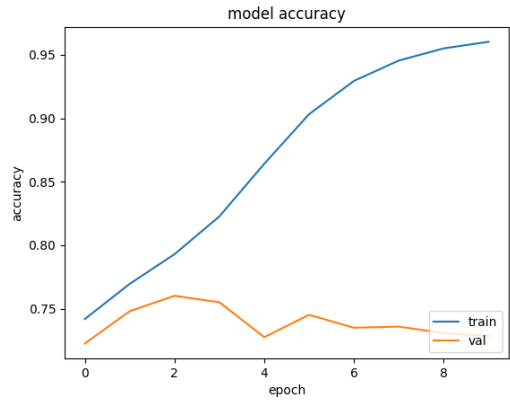
EE5184 - Machine Learning

R06942141 電信碩一 詹鈞皓

Problem 1. (0.5%) 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*。(0.5%) 請實作 BOW+DNN 模型，敘述你的模型架構，回報正確率並繪出訓練曲線。* 訓練曲線 (Training curve)：顯示訓練過程的 loss 或 accuracy 變化。橫軸為 step 或 epoch，縱軸為 loss 或 accuracy。

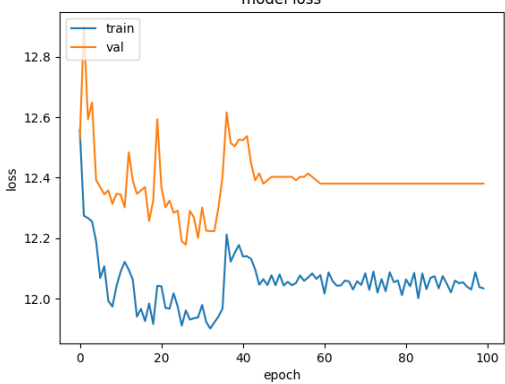
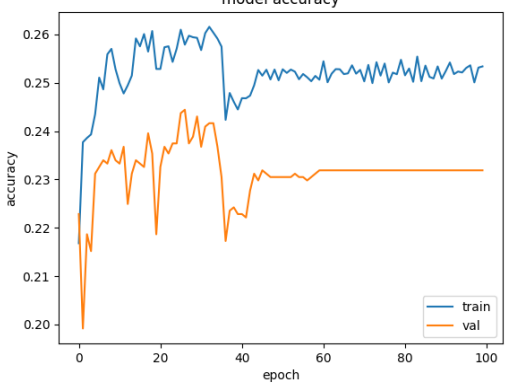
RNN 模型：

這次作業我用的是 LSTM，在前處理部份，先使用 jieba 將文句斷詞，接著使用 gensim 的 word2vec，將詞轉換成 300 維的向量，接著用 tokenizer 來做出字典，並且將詞轉換成對應字典的數字，例如字典中第一個字是“姜太公”，那麼“姜太公”就會轉換成 1，然後將這串數字當成 LSTM 的輸入。在訓練 network 部份，我疊了 2 層 Bidirectional LSTM layers，然後接一層 attention layers，最後再接 dense layer 做分類，因為這次收斂速度比較快，大概在第 5 個 epoch 就可以知道結果，所以就只設 10 個 epoch。

LSTM																																											
 <p>The graph titled 'model loss' shows training (train) and validation (val) loss over 10 epochs. The x-axis is 'epoch' (0 to 10) and the y-axis is 'loss' (0.2 to 0.8). The train loss (blue line) decreases from approximately 0.55 to 0.1. The val loss (orange line) starts at 0.55, dips slightly at epoch 2, and then increases to approximately 0.95 by epoch 10.</p> <table><tr><th>epoch</th><th>train loss</th><th>val loss</th></tr><tr><td>0</td><td>0.55</td><td>0.55</td></tr><tr><td>2</td><td>0.45</td><td>0.50</td></tr><tr><td>4</td><td>0.35</td><td>0.65</td></tr><tr><td>6</td><td>0.25</td><td>0.80</td></tr><tr><td>8</td><td>0.18</td><td>0.90</td></tr><tr><td>10</td><td>0.10</td><td>0.95</td></tr></table>	epoch	train loss	val loss	0	0.55	0.55	2	0.45	0.50	4	0.35	0.65	6	0.25	0.80	8	0.18	0.90	10	0.10	0.95	 <p>The graph titled 'model accuracy' shows training (train) and validation (val) accuracy over 10 epochs. The x-axis is 'epoch' (0 to 10) and the y-axis is 'accuracy' (0.75 to 0.95). The train accuracy (blue line) increases steadily from approximately 0.74 to 0.96. The val accuracy (orange line) starts at 0.72, peaks at epoch 2 (0.76), dips at epoch 4 (0.73), and then fluctuates around 0.74.</p> <table><tr><th>epoch</th><th>train accuracy</th><th>val accuracy</th></tr><tr><td>0</td><td>0.74</td><td>0.72</td></tr><tr><td>2</td><td>0.79</td><td>0.76</td></tr><tr><td>4</td><td>0.86</td><td>0.73</td></tr><tr><td>6</td><td>0.93</td><td>0.74</td></tr><tr><td>8</td><td>0.95</td><td>0.74</td></tr><tr><td>10</td><td>0.96</td><td>0.74</td></tr></table>	epoch	train accuracy	val accuracy	0	0.74	0.72	2	0.79	0.76	4	0.86	0.73	6	0.93	0.74	8	0.95	0.74	10	0.96	0.74
epoch	train loss	val loss																																									
0	0.55	0.55																																									
2	0.45	0.50																																									
4	0.35	0.65																																									
6	0.25	0.80																																									
8	0.18	0.90																																									
10	0.10	0.95																																									
epoch	train accuracy	val accuracy																																									
0	0.74	0.72																																									
2	0.79	0.76																																									
4	0.86	0.73																																									
6	0.93	0.74																																									
8	0.95	0.74																																									
10	0.96	0.74																																									
public score	private score																																										
0.76890	0.76375																																										

BOW + DNN 模型：

在實作 bow 部份，我用的是 Keras 內建的 Tokenizer 的 function: `texts_to_matrix`，他可以將輸入的文字依照不同種的模式轉換成矩陣，要實作出 BOW 只要使用這個 function，然後設定成 count mode，他就會去算詞出現的頻率然後轉換成矩陣。不過礙於記憶體容量問題，我們得把字典的詞數量設定在 22000，才不會讓記憶體不夠。在訓練模型部份，除了 input layer 之外，我只疊了 2 層 dense layer 就做分類，epoch 數目也和上述一樣。

BOW + DNN	
 <p>model loss</p> <p>The plot shows training loss (blue line) and validation loss (orange line) over 100 epochs. Training loss starts at approximately 12.7 and decreases to about 12.05. Validation loss starts at approximately 12.75, drops to 12.4, and then stabilizes around 12.4 after epoch 40.</p>	 <p>model accuracy</p> <p>The plot shows training accuracy (blue line) and validation accuracy (orange line) over 100 epochs. Training accuracy starts at approximately 0.20 and increases to about 0.255. Validation accuracy starts at approximately 0.20, peaks at 0.245 around epoch 20, and then stabilizes around 0.23 after epoch 40.</p>
public score	private score
0.75327	0.74782

Problem 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。

在這個作業中，文字處理的前處理的作法會對整個預測結果產生重大影響，除了使用 jieba 斷詞之外，比較大的影響就是怎麼訓練 word2vec 了，我會將 min_count 設為 5，這就可以刪去沒出現多少次的詞，避免影響預測，然後 word2vec 維度設為 300，因為有去試過 50, 100, 300 的效果，300 的表現最好。接著就是討論一下要如何 padding，因為文句中可能出現很長很長的句子，或是短短幾個字的句子，因為要讓所有長度都一樣，所以就要截長補短，而所有句子的句子長度就由統計來判斷，我統計出所有句子長度的中位數, 75%, 90, 來做實驗，最後取長度 100 表現的最好。

在訓練模型方面，因為句子中的字詞彼此一定會互相影響，相互作用，除了用 LSTM 之外，我選用的是 Bidirectional LSTM 來做訓練，讓每一個字詞影響彼此，經過 2 層 Bidirectional LSTM 之後就用 attention layer 把比較重要的部份給抓出來，然後就丟進 dense layer 做預測。

最後再挑出表現比較好的 Model 作 ensemble，當成預測結果

Problem 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。

由於在中文字中，有許多字詞是要兩個字連在一起才有意義的，像是“葡萄”，“葡”與“萄”本身是沒意義的，連在一起作為詞使用才會有意義。因此，對於 jieba 斷詞而言，就可以把“葡萄”這個詞給切分出來，不會讓他們分散成無意義的兩個字，這在訓練中會有很大的幫助。

但是這種詞不是很常出現，而在斷詞方面也有可能會有斷詞不準確導致影響訓練，所以不使用 jieba 斷詞，光以一個字一個字切分，還是有一定的預測準確度。而實作的成果如下，我分別寫了兩種資料前處理的函式做區別，由 kaggle 上的分數可見，用了 jieba 斷詞的效果還是比較好的。

	public score	private score
with parsing	0.76420	0.76155
without parsing	0.75370	0.75310

Problem 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)，並討論造成差異的原因。

這題是一個很有趣的實驗，以我自己主觀判斷，也是需要經過思考才能下決定，要讓機器去分辨模稜兩可的句子可能會有點複雜。不過實作的結果如下表所示，LSTM model 將前者歸類為 0，後者為 1，而 BOW model 則是將兩者皆歸類為 1，實際上，我主觀認為前者為 0 後者為 1，所以以結果來說 LSTM model 是比較符合我們的期待。

因為 BOW model 因為是算字詞的出現機率，所以對這兩個句子而言 BOW 會認為兩者是差不多的，而其句子又有出現”白痴”這項負面的字詞，所以才會都判定為 1。但以 LSTM 而言，他會會串聯整句話出現的字詞，所以可以判斷出前者句子中出現的”白痴”並不是再辱罵別人，而後者才是。

	在說別人白痴之前，先想想自己	在說別人之前先想想自己，白痴
LSTM	0.41756	0.60433
BOW	0.60544	0.60544

Problem 5.

5.

x	0	1	2	3	4	5	6	7	8	9
y	+	-	+	+	+	-	-	+	-	-
u_1	1	1	1	1	1	1	1	1	1	1
u_2	$\frac{1}{2}$	2	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	2	$\frac{1}{2}$	$\frac{1}{2}$
u_3	0.74	1.35	0.33	0.33	0.33	0.74	0.74	1.35	0.74	0.74

$t=1$ 切在 (4,5) 間，左為 +，右為 -

$\epsilon_1 = \frac{2}{10} = 0.2$ $\alpha_1 = \ln 2 = 0.6931$ $d_1 = \sqrt{\frac{0.8}{0.2}} = 2$	$\epsilon_2 = 0.3125$ $\alpha_2 = 0.392$ $d_2 = 1.48$
--	---

$t=2$ ，令 $d_s(i,j)$ 代表切在 i,j 之間的 error rate, for $i \in [0,8], j=i+1$

切左為 -, 右為 +: $d_s(0,1) = \frac{0.5 \times 5 + 2}{8} = 0.5625$ $d_s(3,4) = 0.4375$ $d_s(6,7) = 0.375$

$\vee d_s(1,2) = 0.3125$ $d_s(4,5) = 0.5$ $d_s(7,8) = 0.625$

$d_s(2,3) = 0.375$ $d_s(5,6) = 0.4375$ $d_s(8,9) = 0.5625$

$t=3$, tn 左為+, 右為-

$\checkmark d_s(0,1) = \frac{0.33 \times 3 + 1.35}{9.39} = 0.3166$
 $d_s(3,4) = 0.41$
 $d_s(6,7) = 0.5656$
 $\Sigma_3 = 0.3166$
 $d_s(1,2) = 0.4993$
 $d_s(4,5) = 0.3653$
 $d_s(7,8) = 0.3829$
 $d_3 = 1.46$
 $d_s(2,3) = 0.4546$
 $d_s(5,6) = 0.4654$
 $d_s(8,9) = 0.483$
 $\alpha_3 = \ln 1.46 = 0.37$

Final classifier:

$$H(x) = \text{sign}(0.69 \cdot f_1(x) + 0.39 \cdot f_2(x) + 0.37 \cdot f_3(x))$$

0	1	2	3	4	5	6	7	8	9
+	-		+				-		

$0.69 - 0.39 + 0.37$
 $0.69 - 0.39 - 0.37$
 $0.69 + 0.39 - 0.37$
 $-0.69 + 0.39 - 0.37$
 $\therefore \text{final error} = 0.1$

Problem 6.

6. $t=1$,

$$W^* = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 100 & 100 & 0 & 0 \\ -100 & -100 & 0 & 0 \\ 0 & 0 & 100 & 0 \end{bmatrix}, \quad X_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \end{bmatrix}, \quad b^* = \begin{bmatrix} 0 \\ -10 \\ 110 \\ -10 \end{bmatrix}$$

$z = \begin{bmatrix} 3 \\ 90 \\ 10 \\ -10 \end{bmatrix}$
 $c' = f(z_1)g(3) + c f(-10) = 3$
 $z_1 = 90$
 $y = f(-10)h(3) = 0, \quad y' = 0$
 $z_f = 10$
 $z_0 = -10$

$t=2, \quad c=3$

$$W^* [1 \ 0 \ 1 \ -2]^T + b^* = [-2, 90, 10, 90]$$

$$c' = f(90)g(-2) + 3f(10) = -2 + 3 = 1$$

$$y = f(90)h(1) = 1, \quad y^2 = 1$$

$t=3, \quad c=1$

$$W^* [1 \ 1 \ 1 \ 4]^T + b^* = [4, 190, -90, 90]$$

$$c' = f(190)g(4) + 1 \cdot f(-90) = 4 + 0 = 4$$

$$y = f(90)h(4) = 4, \quad y^3 = 4$$

$$t=4, C=4$$

$$w^* [0 \ 1 \ 10] + b^* = [0 \ 90 \ 10 \ 90]$$

$$C' = f(90)g(0) + 4 \cdot f(10) = 0 + 4 = 4$$

$$y = f(90)h(4) = 4, \quad y^4 = 4 \#$$

$$t=5, C=4$$

$$w^* [0 \ 1 \ 0 \ 2] + b^* = [2, 90, 10, -10]$$

$$C' = f(90)g(2) + 4 \cdot f(10) = 2 + 4 = 6$$

$$y = f(-10)h(6) = 0, \quad y^5 = 0 \#$$

$$t=6, C=6$$

$$w^* [0 \ 0 \ 1 \ -4] + b^* = [-4, -10, 110, 90]$$

$$C' = f(-10)g(-4) + 6f(110) = 0 + 6 = 6$$

$$y = f(90) \cdot h(6) = 6, \quad y^6 = 6 \#$$

$$t=7, C=6$$

$$w^* [1, 1, 1, 1] + b^* = [1, 190, -90, 90]$$

$$C' = f(190)g(1) + 6 \cdot f(-90) = 1 + 0 = 1$$

$$y = f(90)h(1) = 1, \quad y^7 = 1 \#$$

$$t=8, C=1$$

$$w^* [10 \ 1 \ 2] + b^* = [2, 90, 10, 90]$$

$$C' = f(90)g(2) + 1 \cdot f(10) = 2 + 1 = 3$$

$$y = f(90)h(3) = 3, \quad y^8 = 3 \#$$

$$\therefore Y = [0, 1, 4, 4, 0, 6, 1, 3]$$