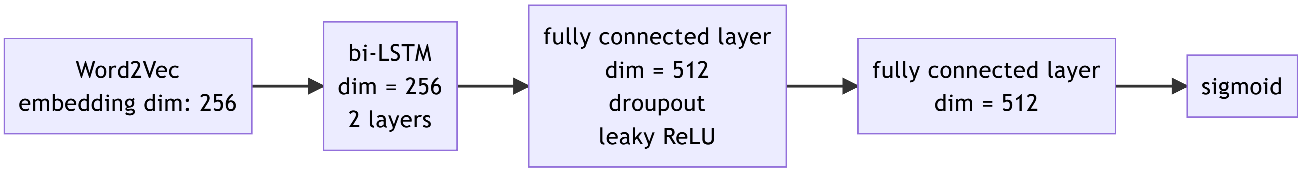
EEML HW4

B06702064 會計五 林聖硯

RNN model連結: https://drive.google.com/file/d/1Sry-e09ia5yjy5alD5Qndy74Xhok9nlC/view?usp=sharing (https://drive.google.com/file/d/1Sry-e09ia5yjy5alD5Qndy74Xhok9nlC/view?usp=sharing)
w2v model連結: https://drive.google.com/file/d/1TTgk4kyqUrL1PGiAw1glW-25-uoLUwym/view?usp=sharing (https://drive.google.com/file/d/1TTgk4kyqUrL1PGiAw1glW-25-uoLUwym/view?usp=sharing)

1. (1%) 請以block diagram或是文字的方式說明這次表現最好的 model 使用哪些layer module(如 Conv/RNN/Linear 和各類 normalization layer) 及連接方式(如一般forward 或是使用 skip/residual connection)，並概念性逐項說明選用該 layer module 的理由。

我的模型之block diagram，以下分別詳細說明每一個block的設計原理。



Word2Vec參數設定

- embedding dimension: 256
- hierachical smapling
- training iteration: 2
- min count: 2
- negative sampling size: 5
- window size: 5

在w2v的部分，我沒有特別著墨太多，只有在助教給的範例code裡面微調參數，這與後面我會將模型的embedding從non-trainable變成trainable有關。後面我發現在accuracy沒辦法上升時，將模型embedding變成trainable，做end-to-end learning能夠再次提升accuracy，但也因此一定會改變w2v的embedding，所以在w2v的部分沒有花太多時間調整。(這部分剛好也在我調整w2v超參數的時候驗證，不管怎麼樣調整w2v的參數，我的模型accuracy都沒有提升太多)

bi-LSTM參數設定

- nubmer of LSTM layer: 2
- bi-directional: True
- dropout: 0.5

在LSTM的部分，我是拿bi-LSTM最後一層、LSTM兩個方向的hidden state做concatenation，再傳入後面的fully conected layer，因為我認為這樣子得到的features比單純只有output state的更有代表性。(但後來發現兩者好像是等價的)

FC layer設定

在FC layers的部分，我只有兩層FC layers，因為在訓練的過程中其實發現到一層FC layers就足夠讓模型fit得很好了，所以沒有特別再加得很深。為了增加模型的generalize能力，所以才多加一層FC layer，讓第一層FC可以加上dropout以及batch norm的layer。

2. (1%) 請比較 word2vec embedding layer 初始設為 non-trainable/trainable 的差別，列上兩者在 validation/public private testing 的結果，並嘗試在訓練過程中設置一策略改變 non-trainable/trainable 設定，描述自己判斷改變設定的機制以及該結果。

我在這次的作業中嘗試以下三種embedding trainable的方式，以下設置為

1. 在每一個epoch中，embedding都是trainable
2. 在每一個epoch中，embedding都是non-trainable
3. 一開始設置為non-trainable，在valid loss開始沒有下降的**第一個**epoch後改為trainable
4. 一開始設置為trainable，在valid loss開始沒有下降的**第一個**epoch後改為non-trainable

實作方法	valid acc	public acc	private acc	訓練備註
1	80.550%	79.810%	81.440%	到第三個epoch就overfit，valid loss往下降
2	79.971%	80.700%	80.170%	
3	**81.680%**	**82.58%**	**82.08%**	表現最好
4	81.106%	80.720%	81.770%	

切換的時機都是在loss沒有下降的那一個epoch就開始切換，因為在訓練的途中我觀察到一個現象，在train RNN時，valid loss只要上升通常都不會再下降了。而最後的結果是第三種實作方法表現最好，我認為可能的原因是w2v給header的資訊量有限，當今天valid loss不再下降時，header已經能將w2v的資訊運用地很完善了(對generability來說)，再訓練下去就會overfit；這時若我們將embedding變成trainable，變成end-to-end learning反而能讓valid loss再次下降，代表w2v這時才算是調整成我們適合這個任務最好的embedding。這時發生一個很有趣的現象，切

換成trainable的那一個epoch的valid accuracy會最高(valid loss會最低)，之後變開始下降，顯示若將embedding調成trainable極度容易讓整個模型overfit training dataset。

3. (1%) 請敘述你如何對文字資料進行前處理，並概念性的描述你在資料中觀察到什麼因此你決定採用這些處理，並描述使用這些處理時作細節，以及比較其實際結果，該結果可以不用具備真正改進。如果你沒有作任何處理，請給出一段具體描述來說服我們為什麼不做處理可以得到好的結果，這個理由不能是因為表現比較好。

在資料集中，我觀察到以下幾個現象

1. 有comment會用重複的字來表達強烈情緒，比如forevaaaaaaaaaaahhhhhhhh
2. 很多句子會用"!!!"來表達強烈情緒
3. unlabel dataset有別的語言的句子
4. 很多句子會含有http網址，而且網址並不是連在一起的，中間會有各種奇怪的空白，所以沒辦法直接用一個regular expression解決

根據以上觀察我作了以下的資料前處理和助教原始的code(只用空白切詞)比較。

- 將超過三次重複的字詞縮減成最多三次，e.g.forevaaaaaaaaaaahhhhhhhh變成foreeevaaah
- 將http的前綴去除，並且移除特定且重複出現多次的網站名字如"http : bit . ly、"http : cli . gs "、"http : tinyurl . com"等等
- 在word2vec訓練中，將min_count(最低出現次數改成5次)，藉以移除網址後面隨機生成的亂碼
- 將除了"!"以外的標點符號去除

結果比較

附註: 以下模型都是用problem1裡面的模型搭配調整過後的超參數訓練

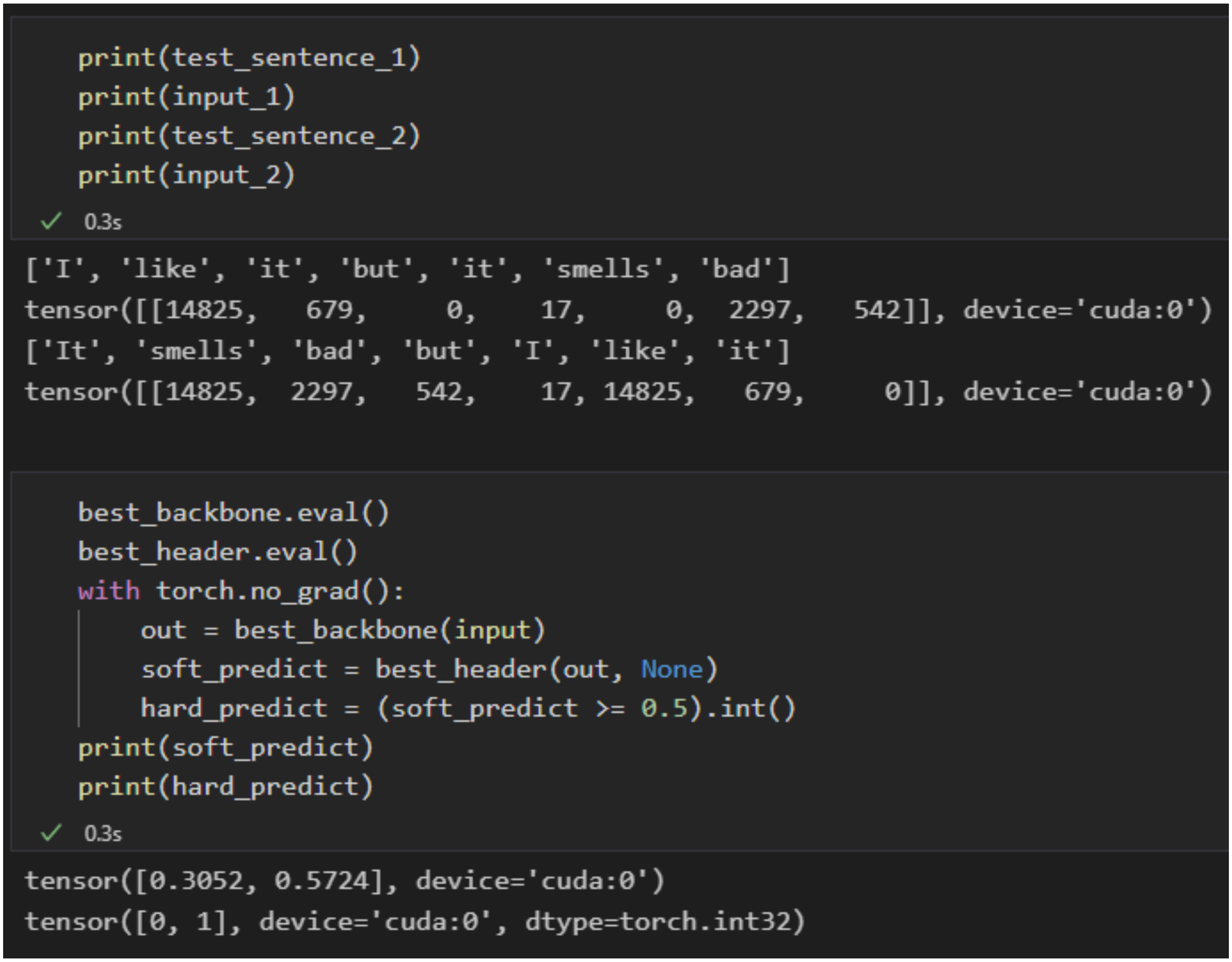
text preprocessing	valid acc	public acc	private acc
單純用空白切割	**82.168%**	**83.160%**	81.920%
使用自訂之資料前處理	82.136%	82.580%	**82.080%**

使用上述資料處理後，word embedding的input文字數從**30,885**下降至**14,745**，在public accuracy下降了0.6%左右，而在private accuracy的部分上升了0.16%，用整個testing set的分數來看，模型的表現是小幅的下降。我認為應該是因為text preprocessing的第一個部份以及第四個步驟把一些重要表達強烈情緒的字詞去掉了，導致模型判斷失準，但其實這麼微小的程度有可能只是因為weight initialization的差異所造成的。

4. (1%) 請「自行設計」兩句具有相同單字但擺放位置不同的語句，使得你表現最好的模型產生出不同的預測結果，例如 “Today is hot, but I am happy” 與 “I am happy, but today is hot”，並討論造成差異的原因。

我拿自行設計的句子丟入word embedding以及model中，得到的output分別為。
“I like it, but it smells bad”: 0
“It smells bad, but I like it”: 1

我認為我的模型將這兩個句子分別分類為負面以及正面是正常的，而且由於我的模型是bi-directional LSTM，他有學到"but"這個單字造成整個句子語氣的影響。通常在講一句話的時候，我們習慣強調放在"but"後面的子句，所以如果我用人工判斷，我也會把第一個句子分類為負面，第二個句子分類為正面。



```
print(test_sentence_1)
print(input_1)
print(test_sentence_2)
print(input_2)
```

✓ 0.3s

```
['I', 'like', 'it', 'but', 'it', 'smells', 'bad']
tensor([[14825,   679,     0,    17,     0,  2297,   542]], device='cuda:0')
['It', 'smells', 'bad', 'but', 'I', 'like', 'it']
tensor([[14825,  2297,   542,    17, 14825,   679,     0]], device='cuda:0')
```

```
best_backbone.eval()
best_header.eval()
with torch.no_grad():
    out = best_backbone(input)
    soft_predict = best_header(out, None)
    hard_predict = (soft_predict >= 0.5).int()
print(soft_predict)
print(hard_predict)
```

✓ 0.3s

```
tensor([0.3052, 0.5724], device='cuda:0')
tensor([0, 1], device='cuda:0', dtype=torch.int32)
```

EEML HW4 Math

Problem 1

1.

$$|d| = \sum_{i=1}^{N} c(w_i)$$

$$P(d|\theta_1, \cdots, \theta_N) = \binom{|d|}{c(w_1) \dots c(w_N)} \prod_{i=1}^{N} \theta_i^{c(w_i)}, \theta_i = P(w_i), \sum_{i=1}^{N} \theta_i = 1.$$

$$= (|d|)! \prod_{i=1}^{N} \frac{\theta_i^{c(w_i)}}{(c(w_i))!}$$

$$\log P(d|\theta_1, \cdots, \theta_N) = \log(|d|)! + \sum_{i=1}^{N} c(w_i) \log \theta_i - \sum_{i=1}^{N} \log(c(w_i))!$$

$$MLE = \underset{\theta_1, \cdots, \theta_N}{\arg\max} \{ P(d|\theta_1, \cdots, \theta_N) \} \quad s.t. \sum_{i=1}^{N} \theta_i = 1.$$

$$(i.e. \sum_{i=1}^{N} \theta_i - 1 = 0)$$

Assume $\mathcal{L} = \log P(d|\theta_1, \cdots, \theta_N) + \lambda (\sum_{i=1}^{N} \theta_i - 1)$

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{c(w_i)}{\theta_i} + \lambda = 0 \Rightarrow \theta_i = -\frac{c(w_i)}{\lambda}$$

we know that $\sum_{i=1}^{N} \theta_i = 1$

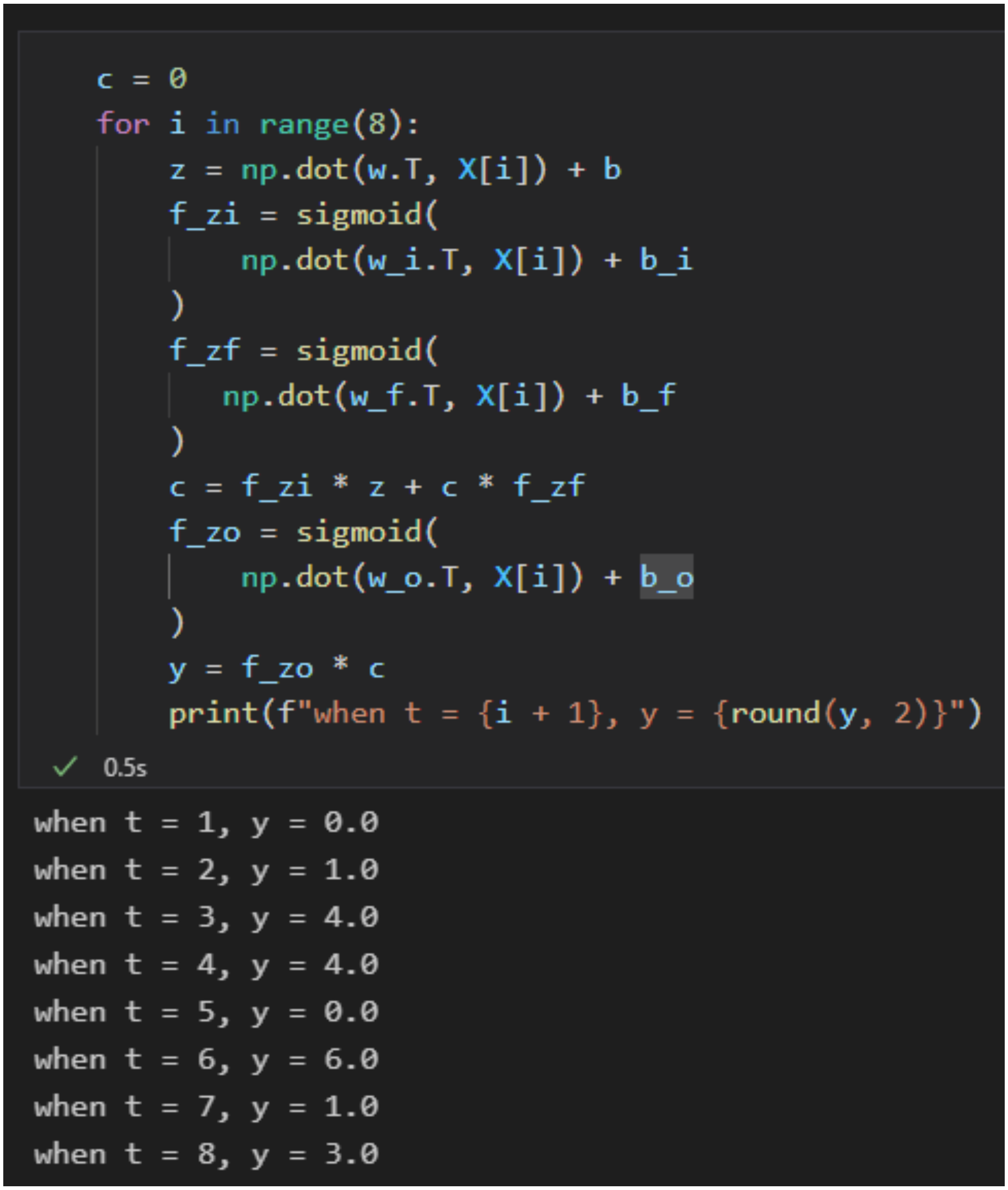
$$\Rightarrow \sum_{i=1}^{N} -\frac{c(w_i)}{\lambda} = 1$$

$$\Rightarrow \lambda = -\sum_{i=1}^{N} c(w_i) = -|d|$$

$$\therefore \theta_i = \frac{c(w_i)}{|d|} \quad \#.$$

Problem 2

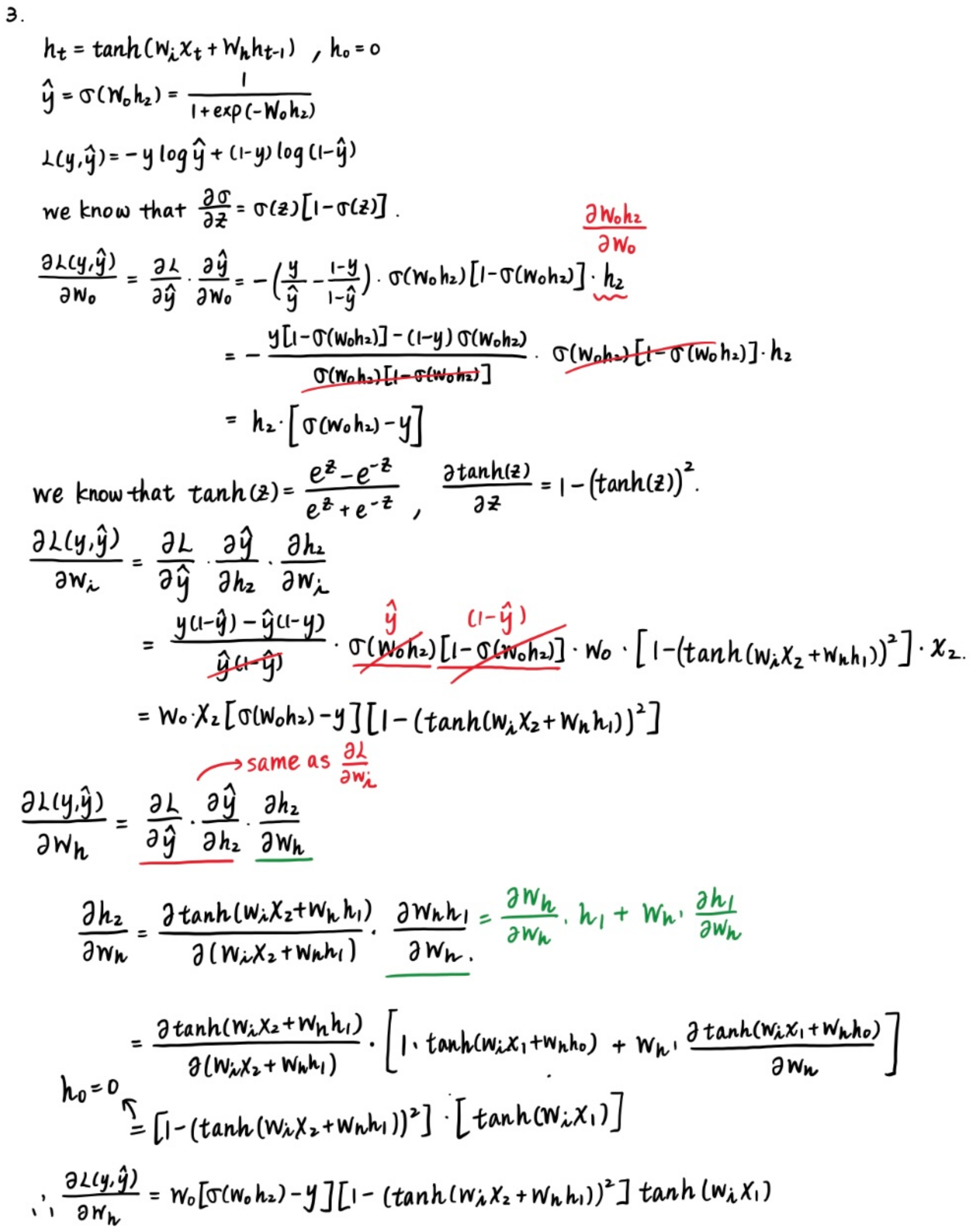
I wrote a program to solve the problem.



```python
c = 0
for i in range(8):
    z = np.dot(w.T, X[i]) + b
    f_zi = sigmoid(
        np.dot(w_i.T, X[i]) + b_i
    )
    f_zf = sigmoid(
       np.dot(w_f.T, X[i]) + b_f
    )
    c = f_zi * z + c * f_zf
    f_zo = sigmoid(
        np.dot(w_o.T, X[i]) + b_o
    )
    y = f_zo * c
    print(f"when t = {i + 1}, y = {round(y, 2)}")
```

```
when t = 1, y = 0.0
when t = 2, y = 1.0
when t = 3, y = 4.0
when t = 4, y = 4.0
when t = 5, y = 0.0
when t = 6, y = 6.0
when t = 7, y = 1.0
when t = 8, y = 3.0
```

Problem 3



3.

$$h_t = \tanh(W_i x_t + W_h h_{t-1}) \quad , h_0 = 0$$

$$\hat{y} = \sigma(W_o h_2) = \frac{1}{1+\exp(-W_o h_2)}$$

$$L(y,\hat{y}) = -y \log \hat{y} + (1-y) \log (1-\hat{y})$$

we know that $\frac{\partial \sigma}{\partial z} = \sigma(z)[1-\sigma(z)]$.

$$\frac{\partial L(y,\hat{y})}{\partial W_o} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial W_o} = -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right) \cdot \sigma(W_o h_2)[1-\sigma(W_o h_2)] \cdot \overset{\frac{\partial W_o h_2}{\partial W_o}}{h_2}$$

$$= -\frac{y[1-\sigma(W_o h_2)] - (1-y)\sigma(W_o h_2)}{\cancel{\sigma(W_o h_2)[1-\sigma(W_o h_2)]}} \cdot \cancel{\sigma(W_o h_2)[1-\sigma(W_o h_2)]} \cdot h_2$$

$$= h_2 \cdot \left[\sigma(W_o h_2) - y\right]$$

we know that $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$, $\frac{\partial \tanh(z)}{\partial z} = 1 - (\tanh(z))^2$.

$$\frac{\partial L(y,\hat{y})}{\partial W_i} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2} \cdot \frac{\partial h_2}{\partial W_i}$$

$$= \frac{y(1-\hat{y}) - \hat{y}(1-y)}{\cancel{\hat{y}(1-\hat{y})}} \cdot \overset{\hat{y}}{\cancel{\sigma(W_o h_2)}} \overset{(1-\hat{y})}{\cancel{[1-\sigma(W_o h_2)]}} \cdot W_o \cdot \left[1-(\tanh(W_i X_2 + W_h h_1))^2\right] \cdot X_2.$$

$$= W_o \cdot X_2 [\sigma(W_o h_2) - y][1-(\tanh(W_i X_2 + W_h h_1))^2]$$

$$\frac{\partial L(y,\hat{y})}{\partial W_h} = \overbrace{\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2}}^{\text{same as } \frac{\partial L}{\partial W_i}} \cdot \underline{\frac{\partial h_2}{\partial W_h}}$$

$$\frac{\partial h_2}{\partial W_h} = \frac{\partial \tanh(W_i X_2 + W_h h_1)}{\partial (W_i X_2 + W_h h_1)} \cdot \underline{\frac{\partial W_h h_1}{\partial W_h}} . = \frac{\partial W_h}{\partial W_h} \cdot h_1 + W_h \cdot \frac{\partial h_1}{\partial W_h}$$

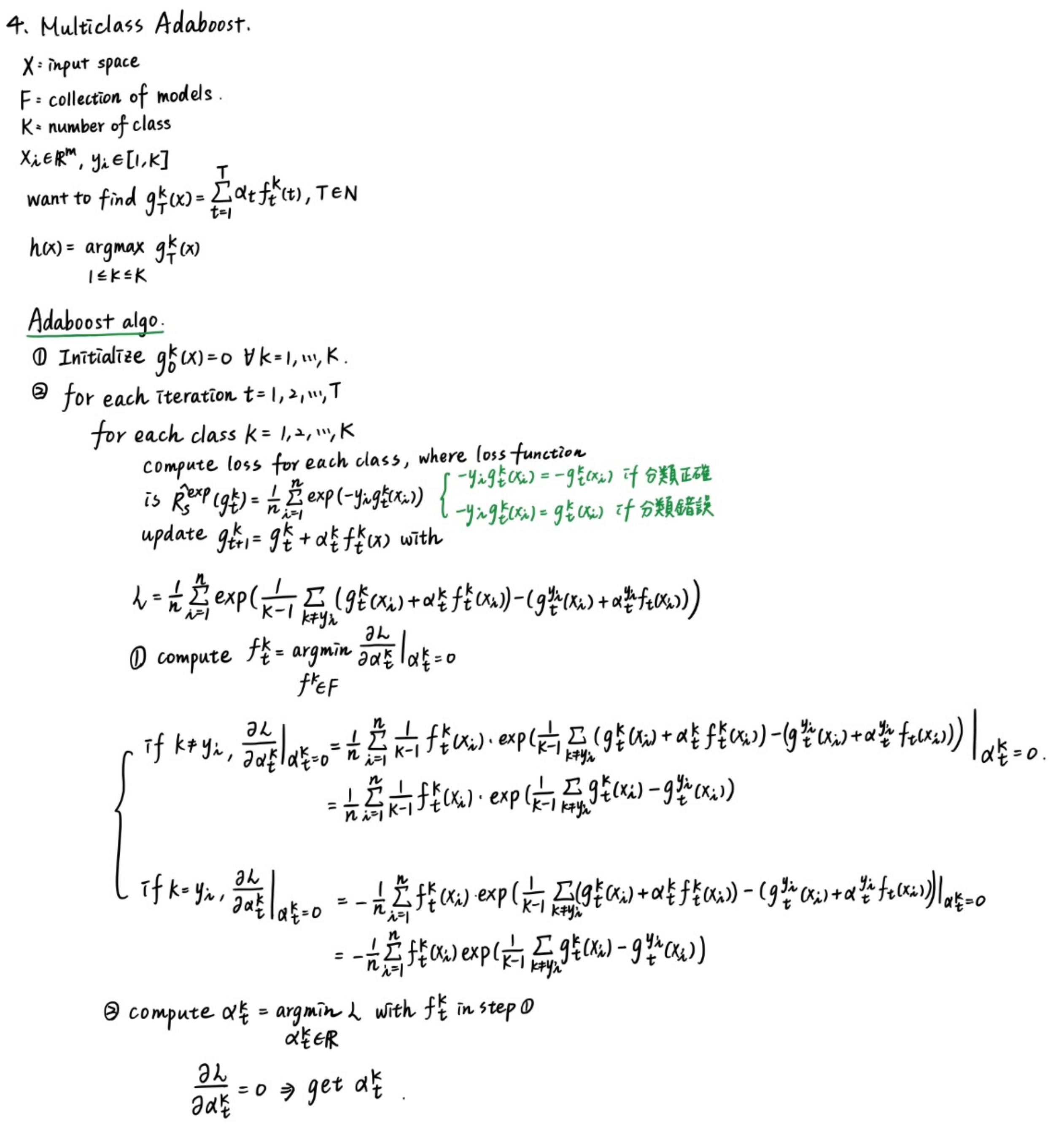
$$= \frac{\partial \tanh(W_i X_2 + W_h h_1)}{\partial (W_i X_2 + W_h h_1)} \cdot \left[1 \cdot \tanh(W_i X_1 + W_h h_0) + W_h \cdot \frac{\partial \tanh(W_i X_1 + W_h h_0)}{\partial W_h}\right]$$

$h_0 = 0$

$$\overset{\uparrow}{=} [1-(\tanh(W_i X_2 + W_h h_1))^2] \cdot [\tanh(W_i X_1)]$$

$$\therefore \frac{\partial L(y,\hat{y})}{\partial W_h} = W_o [\sigma(W_o h_2) - y][1-(\tanh(W_i X_2 + W_h h_1))^2] \tanh(W_i X_1)$$

Problem 4



4. Multiclass Adaboost.

X: input space

F = collection of models.

K = number of class

$X_i \in \mathbb{R}^m$, $y_i \in [1, K]$

want to find $g_T^k(x) = \sum_{t=1}^{T} \alpha_t f_t^k(t)$, $T \in N$

$$h(x) = \underset{1 \le k \le K}{\arg\max}\, g_T^k(x)$$

Adaboost algo.

① Initialize $g_0^k(x) = 0$ $\forall k = 1, \dots, K$.

② for each iteration $t = 1, 2, \dots, T$

for each class $k = 1, 2, \dots, K$

compute loss for each class, where loss function is $\hat{R}_S^{exp}(g_t^k) = \frac{1}{n}\sum_{i=1}^{n} \exp(-y_i g_t^k(x_i))$ $\begin{cases} -y_i g_t^k(x_i) = -g_t^k(x_i) & \text{if 分類正確} \\ -y_i g_t^k(x_i) = g_t^k(x_i) & \text{if 分類錯誤} \end{cases}$

update $g_{t+1}^k = g_t^k + \alpha_t^k f_t^k(x)$ with

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \exp\Big(\frac{1}{K-1}\sum_{k \ne y_i}\big(g_t^k(x_i) + \alpha_t^k f_t^k(x_i)\big) - \big(g_t^{y_i}(x_i) + \alpha_t^{y_i} f_t(x_i)\big)\Big)$$

① compute $f_t^k = \underset{f^k \in F}{\arg\min} \left.\frac{\partial \mathcal{L}}{\partial \alpha_t^k}\right|_{\alpha_t^k = 0}$

$$\begin{cases} \text{if } k \ne y_i, \left.\frac{\partial \mathcal{L}}{\partial \alpha_t^k}\right|_{\alpha_t^k=0} = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{K-1} f_t^k(x_i) \cdot \exp\Big(\frac{1}{K-1}\sum_{k \ne y_i}\big(g_t^k(x_i) + \alpha_t^k f_t^k(x_i)\big) - \big(g_t^{y_i}(x_i) + \alpha_t^{y_i} f_t(x_i)\big)\Big)\Big|_{\alpha_t^k = 0}. \\ \qquad = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{K-1} f_t^k(x_i) \cdot \exp\Big(\frac{1}{K-1}\sum_{k \ne y_i} g_t^k(x_i) - g_t^{y_i}(x_i)\Big) \\ \text{if } k = y_i, \left.\frac{\partial \mathcal{L}}{\partial \alpha_t^k}\right|_{\alpha_t^k=0} = -\frac{1}{n}\sum_{i=1}^{n} f_t^k(x_i) \cdot \exp\Big(\frac{1}{K-1}\sum_{k \ne y_i}\big(g_t^k(x_i) + \alpha_t^k f_t^k(x_i)\big) - \big(g_t^{y_i}(x_i) + \alpha_t^{y_i} f_t(x_i)\big)\Big)\Big|_{\alpha_t^k=0} \\ \qquad = -\frac{1}{n}\sum_{i=1}^{n} f_t^k(x_i) \exp\Big(\frac{1}{K-1}\sum_{k \ne y_i} g_t^k(x_i) - g_t^{y_i}(x_i)\Big) \end{cases}$$

② compute $\alpha_t^k = \underset{\alpha_t^k \in \mathbb{R}}{\arg\min}\, \mathcal{L}$ with f_t^k in step ①

$\frac{\partial \mathcal{L}}{\partial \alpha_t^k} = 0 \Rightarrow$ get α_t^k.