Applied Deep Learning - Homework 3

資工碩一 R10922005 李澤諺

May 16, 2022

Problem 1-1

本次作業中所使用的 model 爲 mT5，其 model architecture 使用了 Transformer 的 encoder-decoder，其首先會將 input sequence (即新聞內容) 輸入 encoder，接著將 encoder 的 hidden state 和 BOS token 輸入 decoder，之後不斷將 decoder 在上一個時間點的 hidden state 和上一個時間點所輸出的 token 輸入至 decoder 之中，直到 decoder 輸出 EOS token 爲止，而過程中 decoder 所輸出的 token sequence 即爲 inference 的結果 (即新聞摘要)。

Problem 1-2

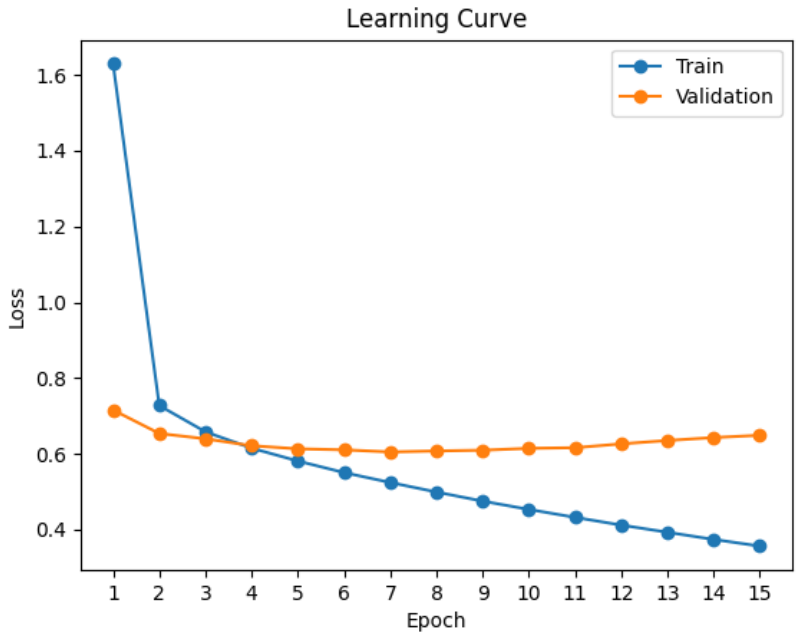
我會將 input sequence 使用 tokenizer 進行 tokenization，其中我所使用的 tokenizer 的參數爲：truncation 爲 True，padding 爲'max_length'，而在對新聞內容做 tokenization 時，max_length 會設爲 256，在對新聞摘要做 tokenization 時，max_length 則會設爲 64。

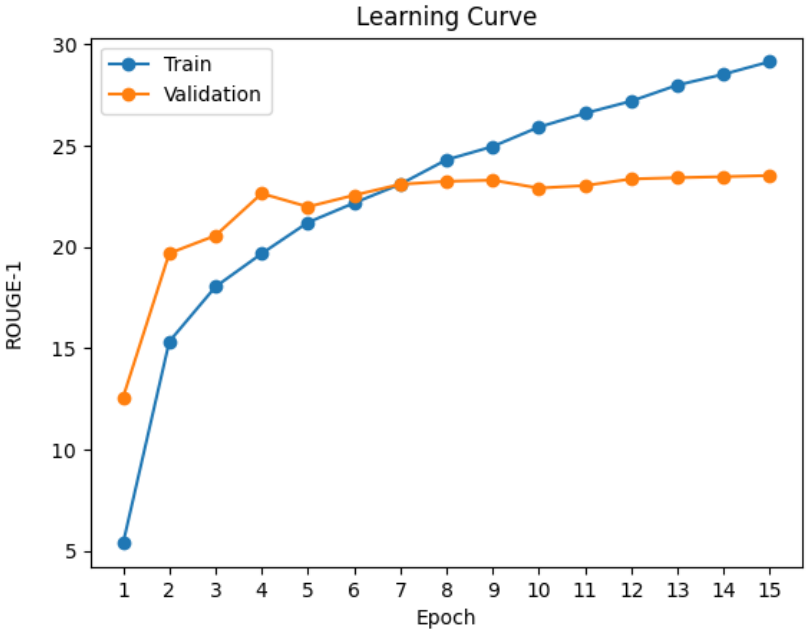
Problem 2-1

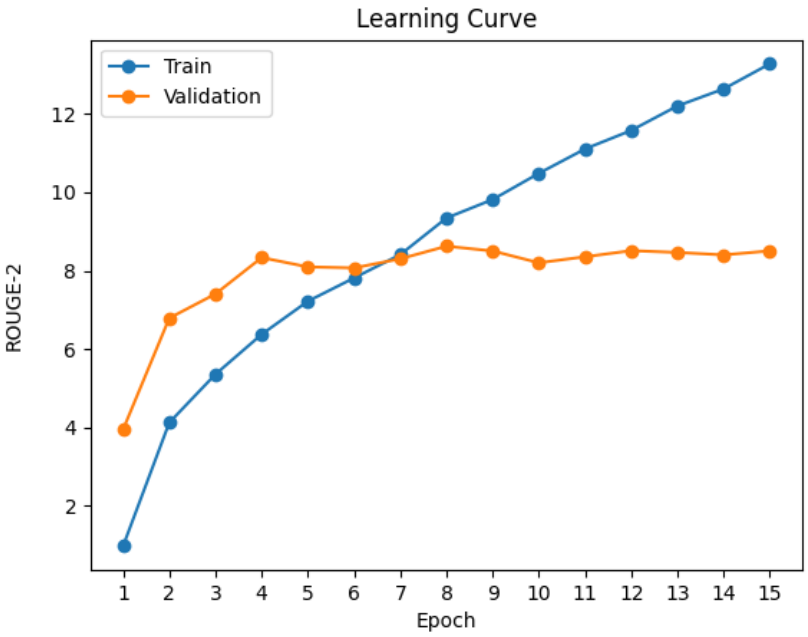
根據 validation 的結果，我最後所使用的 hyper-parameter 爲：batch size 爲 16，learning rate 爲 0.0005，訓練了 15 個 epoch，並且在訓練過程中，我每個 epoch 結束之後就會進行一次 validation，並儲存在 validation dataset 上 ROUGE-1、ROUGE-2、ROUGE-L 皆最高的 model。

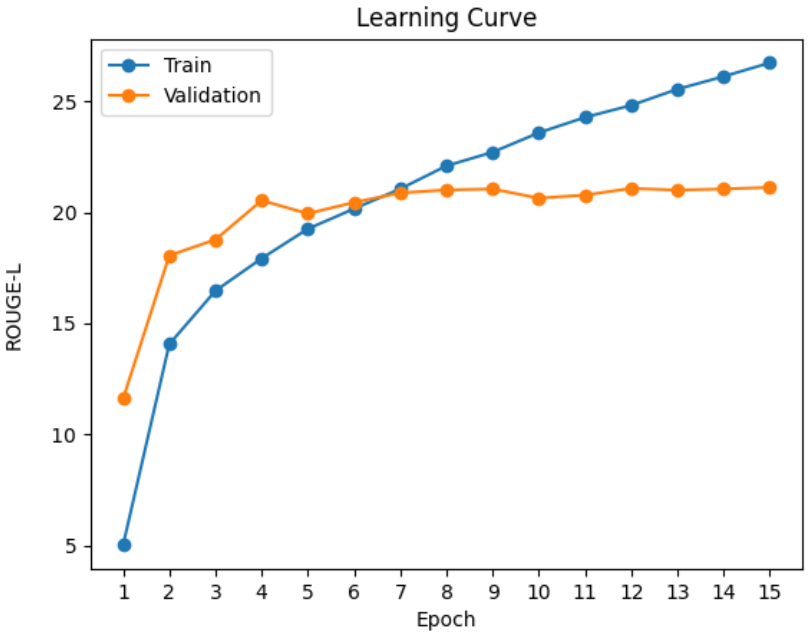
Problem 2-2

下圖爲訓練過程中所得到的 learning curve：


Learning Curve
Train
Validation
Loss
Epoch
1.6
1.4
1.2
1.0
0.8
0.6
0.4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15


Learning Curve
Train
Validation
ROUGE-1
Epoch
30
25
20
15
10
5
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15


Learning Curve
Train
Validation
ROUGE-2
Epoch
2
4
6
8
10
12
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15


Learning Curve
Train
Validation
ROUGE-L
Epoch
5
10
15
20
25
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Problem 3-1

Greedy Search

意即在每個時間點，model 會直接選取當前預測機率最高的 token 作爲該時間點的輸出，以此產生 output sequence。

Beam Search

意即在每個時間點，model 會保留到目前爲止預測機率最高的 N 個 sequence 作爲最終可能的 output sequence 候選。

Top-k Sampling

意即在每個時間點，model 會找出當前預測機率最大的 k 個 token，並將機率重新分佈在這些 token 上，以此對這些 token 進行 sampling，並將被 sample 到的 token 作爲該時間點的 output。

Top-p Sampling

意即在每個時間點，model 會找出機率總和至少爲 p 且數量最少的 token set，並將機率重新分佈在這些 token 上，以此對這些 token 進行 sampling，並將被 sample 到的 token 作爲該時間點的 output。

Temperature

意即在每個時間點，model 會將輸出的 logit 除以一個常數 T，再將 logit 通過 Softmax 得到機率分佈，當 $T > 1$ 時，機率分佈會變得更 smooth，而當 $0 < T < 1$ 時，機率分佈則會變得更 sharp。

Problem 3-2

下表爲使用不同的 beam size 並且不做任何的 sampling 時所得到的結果：

Beam Size	ROUGE-1	ROUGE-2	ROUGE-L
1 (Greedy Search)	23.5	8.2	21.0
2	24.5	9.1	21.9
4	25.0	9.6	22.3
8	25.2	9.8	22.5
16	25.1	9.8	22.4

下表爲使用 top-k sampling 所得到的結果，其中 beam size 爲 1，top-p 爲 1.0，temperature 爲 1.0：

top-k	ROUGE-1	ROUGE-2	ROUGE-L
50	19.5	6.3	17.2
60	19.2	6.1	16.9
70	19.0	6.0	16.9
80	18.9	6.0	16.8
90	18.8	5.9	16.6
100	18.8	5.9	16.7

下表爲使用 top-p sampling 所得到的結果，其中 beam size 爲 1，top-k 爲 50，temperature 爲 1.0：

top-p	ROUGE-1	ROUGE-2	ROUGE-L
1.0	19.3	6.1	17.1
0.9	20.3	6.7	18.1
0.8	21.0	7.0	18.6
0.7	21.8	7.4	19.4
0.6	22.4	7.6	19.8
0.5	22.6	7.8	20.2

下表爲使用 temperature 所得到的結果，其中 beam size 爲 1，top-k 爲 50，top-p 爲 1.0：

Temperature	ROUGE-1	ROUGE-2	ROUGE-L
0.1	23.5	8.2	21.0
0.5	22.6	7.8	20.2
1.0	19.4	6.1	17.2
5.0	5.2	0.2	4.5
10.0	4.6	0.1	4.0

由以上表格可以看出，當 beam size 越大時，所得到的 performance 亦會越高，但其仍會有一個上限，而在進行 sampling 時，當 top-k 和 top-p 越小，意即對越少個機率較高的 token 進行 sample 時，所得到的 performance 越高，而當 temperature 越大時，會使得機率分佈變得更 smooth，此時 performance 亦會下降。考慮到 peformance 和 efficiency，我最後所使用的 generation strategy 爲設 beam size 爲 4，且不使用任何的 sampling。

Bonus

我參考了這篇論文，並實作了當中的 self-critic policy gradient algorithm，其所使用的 loss function 如下：

$$\mathcal{L}_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T} -log\pi_\theta(\hat{y}_{n,t}|\hat{y}_{n,t-1}, s_{n,t})\times \Big(r(\hat{y}_{n,1}, \hat{y}_{n,2}, \cdots, \hat{y}_{n,T}) - r(\hat{y}^g_{n,1}, \hat{y}^g_{n,2}, \cdots, \hat{y}^g_{n,T})\Big)$$

其中 N 爲 batch size，T 爲 output sequence 的長度，而對於第 n 個 input sequence，model 會 sample 一個 output sequence (使用任何一種 generation strategy 皆可)，將其記爲 $\hat{y}_{n,1}$、$\hat{y}_{n,2}$、$\cdots$、$\hat{y}_{n,T}$，並計算其 reward (在此是將 ROUGE-1、ROUGE-2、ROUGE-L 的平均作爲 reward)，即上式中的 $r(\hat{y}_{n,1}, \hat{y}_{n,2}, \cdots, \hat{y}_{n,T})$，此外，model 還會使用 greedy search 產生另外一個 output sequence，將其記爲 $\hat{y}^g_{n,1}$、$\hat{y}^g_{n,2}$、$\cdots$、$\hat{y}^g_{n,T}$，並計算其 reward 作爲 baseline，即上式中的 $r(\hat{y}^g_{n,1}, \hat{y}^g_{n,2}, \cdots, \hat{y}^g_{n,T})$，最後會將每一個 sampled sequence 之中的每一個 token 其被 sample 到的機率，即上式中的 $\pi_\theta(\hat{y}_{n,t}|\hat{y}_{n,t-1}, s_{n,t})$，將其取對數並取負號再乘以該 sampled sequence 的 reward 與 baseline 之差後，作爲該 sampled sequence 的 loss。當該 sampled sequence 的 reward 大於 baseline 時，gradient descent 會使得該 sequence 被 sample 到的機率上升，反之，若該 sampled sequence 的 reward 小於 baseline，gradient descent 則會使得該 sequence 被 sample 到的機率下降，以此提升 model 的 performance。

我有嘗試將 supervised learning 所訓練出來的 model 使用 reinforcement learning 進行 fine-tune，此時發現 loss 會趨近於負無限大，且不論輸入爲何，model 皆會輸出同樣的結果，ROUGE-1、ROUGE-2、ROUGE-L 也幾乎都是 0，我也有嘗試將 supervised learning 的 loss 和 reinforcement learning 的 loss 以不同的 weight 相加進行訓練，希望能用 supervised learning 來引導 reinforcement learning，然而最後所得到的 performance 都不如只使用 supervised learning 來訓練時所得到的 performance，其可能是我的實作上有誤，或是 hyper-parameter 不佳所致，但研究許久我目前仍找不到 performance 不佳的原因。