ADL HW3 Report

editer: M11015Q02 柯元豪

Q1 (Model)

Model

Architecture

config


```
{
  "_name_or_path": "google/mt5-small",
  "architectures": [
    "MT5ForConditionalGeneration"
  ],
  "d_ff": 1024,
  "d_kv": 64,
  "d_model": 512,
  "decoder_start_token_id": 0,
  "dropout_rate": 0.1,
  "eos_token_id": 1,
  "feed_forward_proj": "gated-gelu",
  "initializer_factor": 1.0,
  "is_encoder_decoder": true,
  "layer_norm_epsilon": 1e-06,
  "model_type": "mt5",
  "num_decoder_layers": 8,
  "num_heads": 6,
  "num_layers": 8,
  "pad_token_id": 0,
  "relative_attention_num_buckets": 32,
  "tie_word_embeddings": false,
  "tokenizer_class": "T5Tokenizer",
  "torch_dtype": "float32",
  "transformers_version": "4.17.0",
  "use_cache": true,
  "vocab_size": 250100
}
```


describe

- T5 為 Transfer Text-to-Text Transformer 的簡寫，它最重要的貢獻是給整個 NLP 預訓練模型領域提供了一個通用框架，將所有 NLP 任務都轉化成 Text-to-Text （文字到文字）任務。 而所謂的 T5 模型其實就是個 Transformer 的 Encoder-Decoder 模型。
- MT5 則為多國語言版的 T5，繼承了 T5 原有的優點，激活函數改用 gated-gelu

- text summarization 是 T5 模型中能夠 handle 的任務之一，transformer 主要會以 data 的 text 去生成 summarization。

Preprocessing

```
step 1:
    將 jsonl 的 data 轉換成 json format，只存 'id', 'title', 'maintext'(rename as
'text') 這三個 column
step 2:
    指定 text columns 作為 inputs、title columns 作為 targets，而 inputs 每個 text
都會加上 prefix "summarize: "
step 3:
    將 inputs & targets 分別依照指定的 max_length 交給 tokenizer 做 tokenize，取得
tokenize 後的 inputs & labels
```

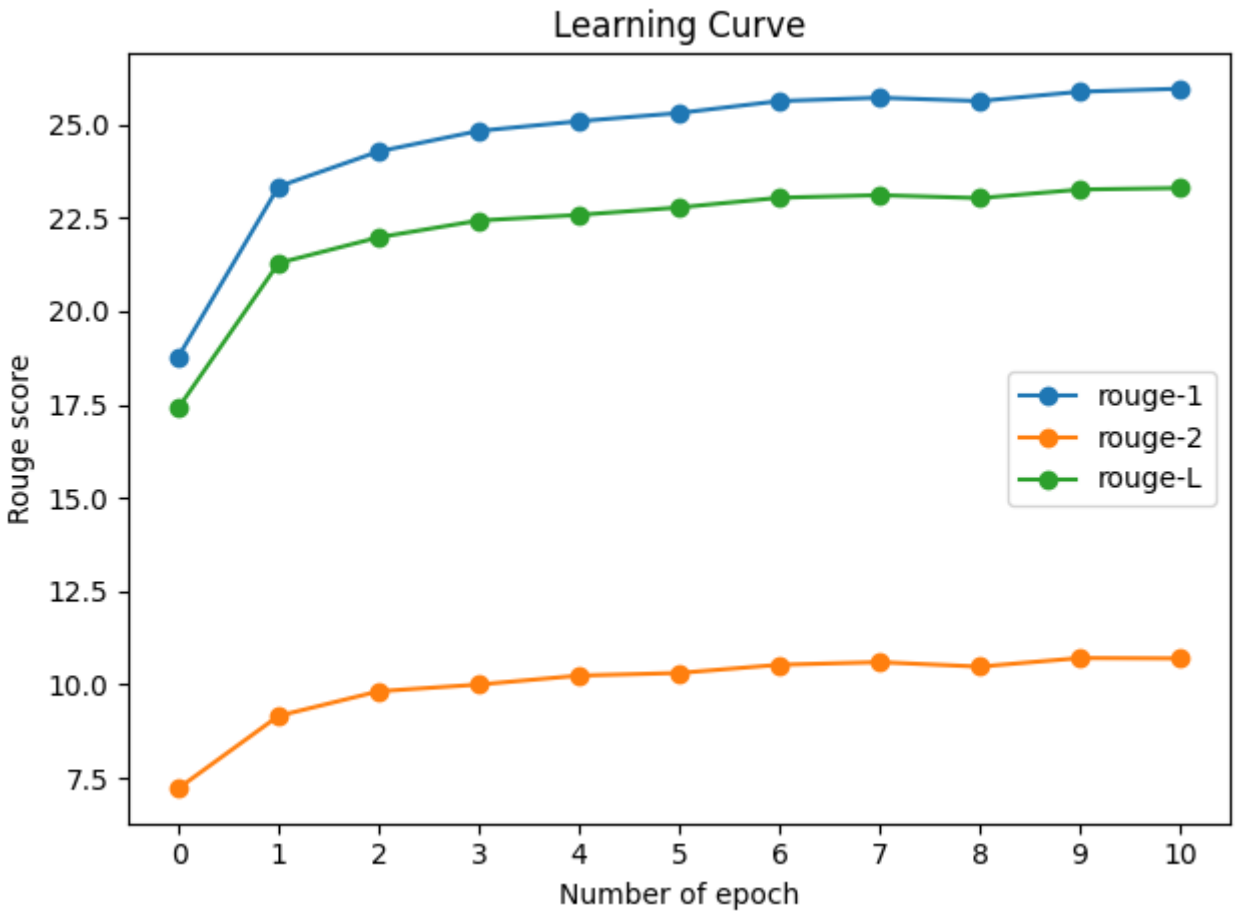
Q2 (Training)

Hyperparameter

```
- learning_rate: 4e-05
- train_batch_size: 1
- eval_batch_size: 2
- seed: 42
- optimizer: Adam with betas=(0.9,0.999) and epsilon=1e-08
- lr_scheduler_type: linear
- num_epochs: 10.0

- model_name_or_path google/mt5-small
- source_prefix "summarize: "
- max_source_length 256
- max_target_length 64
```

Learning Curves



Q3 (Generation Strategies)

Strategies

Greedy Search

Greedy search 是單純在所有可能做為下一個字的清單中，找出最高機率的字。公式如下:
$w_t = argmax_w P(w|w_{1:t-1})$ at each timestep t

Beam Search

Beam Search 與 Greedy search 原理相同，都是找最高機率的下個字。(num_beams = 1 為 greedy) 但 Beam Search 透過 num_beams 的不同，在每個 timestep 都會留下 top-num_beams 的字，直到最後一個 timestep，將每一束 word sequences 的每個字機率相乘並得到最終的機率，以此選擇最高機率的 word sequences，藉此可降低丟失隱藏的高機率 word sequences 的風險。

Sampling

正常的 Sampling，選擇下一個字 w_t 時只根據條件機率分布: w_t $P(w|w_{1:t-1})$

Top-k Sampling

每次 sampling 時都將 top-k 個最高機率的字機率加總後再重新分配機率，並只能從這些字中挑選下一個字。由於 k 是固定值，這往往會導致一些問題發生。若遇到字數長短不一且語境不同的句子，可能就需要將 k 調高或調低。

Top-p Sampling

top-p 不僅是從 top-k 個字中做 sampling，而是從累積機率超過 p 的最小可能字集中進行選擇，然後在這組字中重新分配機率。這樣字集合的大小便可以根據下一個字的機率分布動態增加或減少。

Temperture

softmax 在執行前，會先將 logits 除以 temperature。 因此透過降低 temperature 能夠使 $P(w|w_{1:t-1})$ 更銳利 (增加高機率詞的可能性，降低低機率詞的可能性)。

Hyperparameters

Greedy Search

```
<Fixed hyperparameters>
* do_sample = False, num_beams = 1, top_k = None, top_p = None, \
  temperature = None
<Variable hyperparameters>
* generation_max_length
```

	rouge-1	rouge-2	rouge-L
max_length = 64	24.66	9.35	22.16
max_length = 32	24.63	9.33	22.11

Beam Search

```
<Fixed hyperparameters>
* generation_max_length = 64, do_sample = False, top_k = None, top_p = None, \
  temperature = None
<Variable hyperparameters>
* num_beams
```

	rouge-1	rouge-2	rouge-L
num_beams = 2	25.67	10.25	23.88
num_beams = 5	25.96	10.70	23.30
num_beams = 10	25.91	10.78	23.26

Top-k Sampling

```
<Fixed hyperparameters>
* generation_max_length = 64, num_beams = None, do_sample = True, top_p = None, \
  temperature = None
<Variable hyperparameters>
* top_k
```

	rouge-1	rouge-2	rouge-L
top_k = 5	23.14	8.28	20.54
top_k = 20	21.19	7.08	18.78
top_k = 50	19.72	6.49	17.61

Top-p Sampling

```
<Fixed hyperparameters>
* generation_max_length = 64, num_beams = None, do_sample = True, top_k = 0, \
  temperature = None
<Variable hyperparameters>
* top_p
```

	rouge-1	rouge-2	rouge-L
top_p = 1	15.69	4.96	14.15
top_p = 0.9	17.57	5.96	15.81
top_p = 0.8	18.86	6.45	16.86

Temperture

```
<Fixed hyperparameters>
* generation_max_length = 64, num_beams = None, do_sample = True, top_k = 0, \
  top_p = None
<Variable hyperparameters>
* temperature
```

	rouge-1	rouge-2	rouge-L
temperature = 0.7	21.67	7.70	19.37
temperature = 0.4	24.26	9.07	21.69
temperature = 0.15	24.60	9.35	22.12

Final generation strategy

```
<Hyperparameters>
generation_max_length = 64,
num_beams = 5,
do_sample = False,
top_k = None,
top_p = None,
temperature = None
```

	rouge-1	rouge-2	rouge-L
Final score	25.96	10.70	23.30

tags: ADL