ADL HW1 Report

姓名：陳竹欣

學號：B10902005

系級：資工三

Q1: Data processing

Tokenizer

原始輸入：

```
"伊利諾大學是哪一個國家的大學?"
```

Step 1. 把字串拆成一個一個字元：

```
["伊", "利", "諾", "大", "學", "是", "哪", "一", "個", "國", "家", "的", "大", "學",
"?"]
```

Step 2. 把字元轉換成token：

```
[823, 1164, 6330, 1920, 2119, 3221, 1525, 671, 943, 1751, 2157, 4638, 1920, 2119,
136]
```

Step 3. 加入特殊token，例如`[CLS]: 101`，`[SEP]: 102`。

因為在訓練四選一的模型的時候，會先把問題跟四個候選段落分別接在一起，再判斷哪一個比較合理，訓練QA模型的時候也是會先把問題跟答案接在一起。所以經過 tokenization 之後輸出的串列會是：

```
[101] + [question token list] + [102] + [context token list] + [102]
```

Step 4. Padding

因為我們有設定`max_seq_length=512`，如果該串列的長度小於512，那麼後面都會補`0`；反之，如果該串列的長度大於512，那麼後面的部份都會被砍掉，但串列的最後一項仍然是`[102]`。也就是說，

如果長度不足：

```
[101] + [question token list] + [102] + [context token list] + [102] + [many of
0]
```

如果長度過長：

```
[101] + [question token list] + [102] + [cut context token list] + [102]
```

最後輸出的串列長度皆為512。

Answer Span

1. 每一個token都會有兩個label，分別是start和end，如果那個token是答案的第一個字，那麼該token的start就會是1；如果那個token是答案的最後一個字，那麼該token的end就會是1。

   舉例來說，如果某一筆資料的答案在index=102與index=105，那麼index=102的token的(start, end) = (1, 0)，index=105的token的(start, end) = (0, 1)，其他token的(start, end) = (0, 0)。

   再舉個特別的例子，如果某一筆資料的答案只有一個字，在index=49的地方，那麼index=49的token的(start, end) = (1, 1)，其他token的(start, end) = (0, 0)。

2. 假設n_best=20，代表模型會選出20個最有可能為開頭的index以及20個最有可能為結尾的index，如果我們直接從開頭和結尾中選擇機率最高的index的話，有可能產生 (1) 結尾在開頭前面 (2) 結尾和開頭離太遠 這兩種情況發生。為了避免以上兩種情況發生，我們要窮舉出所有可能，並將以上兩種情況排除，最後從剩下的可能中選出機率最高的那組開頭和結尾作為輸出的答案。

Q2: Modeling with BERTs and their variants

Describe

My model: bert-base-chinese

The performance of your model

public score	private score
0.75045	0.75429

loss function: cross entropy loss

The optimization algorithm: Adam

learning rate: 3e-5

batch size:

paragraph selection	span selection
2	4

epoch:

paragraph selection	span selection
1	3

Try another type of pre-trained LMs and describe

My model: `shibing624/text2vec-base-chinese` and `hfl/chinese-roberta-wwm-ext` [1]

在 paragraph selection 的部份，我使用了三種模型和相同的參數，訓練出三個模型，再透過投票選出最有可能的答案。

BERT 使用的 hyper parameter 在上一題有提到，另外兩個模型使用的 hyper parameter 如下：

arguments	value
`--max_seq_length`	`512`
`--model_name_or_path`	`shibing624/text2vec-base-chinese` or `hfl/chinese-roberta-wwm-ext`
`--per_device_train_batch_size`	`1`
`--per_device_eval_batch_size`	`1`
`--learning_rate`	`3e-5`
`--num_train_epochs`	`1`
`--gradient_accumulation_steps`	`2`

在 question answering 的部份，我使用了兩種模型以及兩種參數，訓練出三個模型，再透過投票選出最有可能的答案。

第一、二種：（code裡面的代稱為 `e5b4l1_t2v` 和 `e5b8l1_t2v`）

arguments	value
`--max_seq_length`	`512`
`--model_name_or_path`	`shibing624/text2vec-base-chinese`
`--per_device_train_batch_size`	`4` or `8` (b4 和 b8 差在這裡)
`--per_device_eval_batch_size`	`1`
`--learning_rate`	`1e-5`
`--num_train_epochs`	`5`
`--gradient_accumulation_steps`	`2`

第三種：（code裡面的代稱為 `e5b8l1_hfl`）

arguments	value
`--max_seq_length`	`512`
`--model_name_or_path`	`hfl/chinese-roberta-wwm-ext`
`--per_device_train_batch_size`	`8`
`--per_device_eval_batch_size`	`1`
`--learning_rate`	`1e-5`
`--num_train_epochs`	`5`
`--gradient_accumulation_steps`	`2`

The performance of your model

public score	private score
0.78752	0.79223

The difference between pre-trained LMs

RoBERTa (Robustly optimized BERT approach)：

和BERT的差異主要有以下四點：

1. Dynamic masking: 不同的epoch會有不同的 masking 的方式，能夠學習更多樣的資訊 (BERT 為 static)。
2. 更大的batch size（256->2K->8K）
3. 只訓練 full length sequences
4. 更多 pre-trained data（比BERT多10倍）

WWM (whole word masking)：

因為中文不像英文會把一個字拆成很多字根，所以可以把整個字遮起來。

EXT (extending data)：

因為Chinese Wikipedia的data太少了，所以作者再加入了 encyclopedia, news, 和 question answering web 等資料，比於來的Chinese Wikipedia多了快10倍。

text2vec:

用CoSENT方法訓練，並且基於`hfl/chinese-macbert-base`在中文STS-B數據訓練得到的模型。

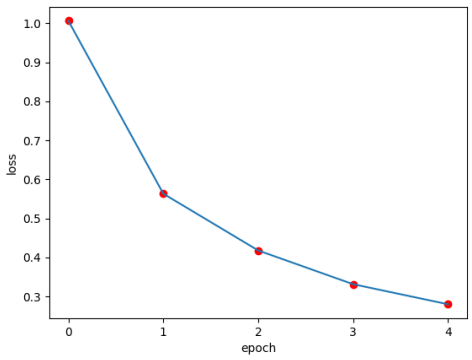
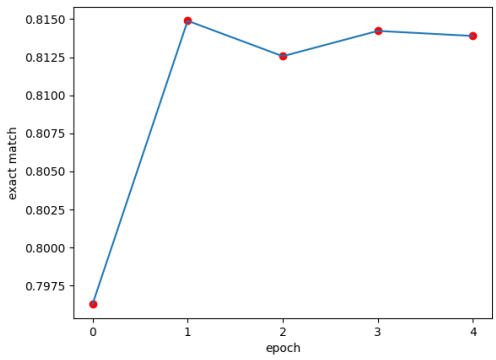
Q3: Curves

這裡使用的 model 和 hyper parameter 如下：

arguments	value
`--max_seq_length`	`512`
`--model_name_or_path`	`shibing624/text2vec-base-chinese`
`--per_device_train_batch_size`	`4`
`--per_device_eval_batch_size`	`1`
`--learning_rate`	`1e-5`
`--num_train_epochs`	`5`
`--gradient_accumulation_steps`	`2`

結果：

	0	1	2	3	4
loss	1.0062	0.5632	0.4177	0.3313	0.2799
exact_match	79.6278	81.4889	81.2562	81.4224	81.3892

Learning curve of the loss	Learning curve of the EM
	

此處的 loss 為 train loss = total train loss / num of train data， exact match 為 validation data 的 exact match 。

Q4: Pre-trained vs Not Pre-trained

我選擇拿四選一的部份（paragraph selection）做比較。程式碼異動的部份如下：

1. Config

```
# Before
config = AutoConfig.from_pretrained(args.model_name_or_path,
trust_remote_code=args.trust_remote_code)
# After
config = BertConfig()
```

2. Model

```
# Before
model = AutoModelForMultipleChoice.from_pretrained(
            args.model_name_or_path,
            from_tf=bool(".ckpt" in args.model_name_or_path),
            config=config,
            trust_remote_code=args.trust_remote_code,
        )
# After
model = AutoModelForMultipleChoice.from_config(config,
trust_remote_code=args.trust_remote_code)
```

以下為 training 使用的 hyper parameter 。

arguments	Pre-trained	Not Pre-trained
`--max_seq_length`	`512`	`512`
`--model_name_or_path`	`bert-base-chinese`	None
`--per_device_train_batch_size`	`1`	`1`
`--per_device_eval_batch_size`	`1`	`1`
`--learning_rate`	`3e-5`	`3e-5`
`--num_train_epochs`	`1`	`1`
`--gradient_accumulation_steps`	`2`	`2`

以下為 performance 的比較：

	Pre-trained	Not Pre-trained
eval accuracy	0.9511465603190429	0.47291458956463944

Q5: Bonus

輸入處理：

1. context 的部份為四個候選段落接起來
2. 如果 relevant 的段落不是第一段，answer start 的 index 要再加上前面段落的字數才會是正確的。

參數調整：

1. 因為 context 從一段變成四段，所以 max_seq_length 應該也要設成四倍，也就是 512 * 4 = 2048。（但後來發現如果使用bert-base-chinese的話，max_seq_length會自動被調回512，但還是能順利訓練完模型，因為tokenize的時候會以 `doc_stride` 把 context 切開）

以下是 training 使用的 hyper parameter：

arguments	**value**
`--train_file`	`train-1-step.json`
`--validation_file`	`valid-1-step.json`
`--max_seq_length`	`2048`
`--model_name_or_path`	`bert-base-chinese`
`--per_device_train_batch_size`	`4`
`--per_device_eval_batch_size`	`1`
`--learning_rate`	`3e-5`
`--num_train_epochs`	`3`
`--gradient_accumulation_steps`	`2`
`--output_dir`	`ffinal`

以下是 testing 使用的 hyper parameter：

arguments	value
`--test_file`	`test-1-step.json`
`--model_name_or_path`	`ffinal`

上述的 `*-1-step.json` 都是經過輸入處理過後的檔案。

另外，loss function 和 Q2 BERT 的部份一樣是 cross entropy loss，optimization algorithm 一樣是 Adam 。

以下為 performance 的比較，除了輸入和 max_seq_length 不同之外，其他的 hyper parameter 都相同：

	public score	private score
paragraph selection + span selection pipeline approach	0.75045	0.75429
end-to-end transformer-based model	0.70886	0.7028

Reference

[1] Revisiting Pre-trained Models for Chinese Natural Language Processing

https://arxiv.org/pdf/2004.13922.pdf