ADL hw2 Report

R10546017 朱瑋民

1. Data preprocessing

(1) 我是用 transformer 的 AutoTokenizer 這個套件，從 pretrained weight (context 用 bert base chinese, question answering 用 albert chinese)，而 bert 的 chinese 的 tokenizer 會將中文的句子 encode 成 Bert 所需要的編號，每個編號對應一個字，最前方為 [CLS]，最後方為 [SEP]，而 pretrain model 裡面沒有的字會用 [UNK] 符號遮蔽起來，會有一個 max_sequence_length 的參數來對 tokenize 完的句子長度做切割，並且利用 padding 將每個輸入的句子的長度變成一樣長，而因為 padding 的 token id 也有被考慮進去，因此 Tokenizer 也會將這些詞經由 Attention mask 的方式來做忽略。

(2) (a) 因為前面會多一個 CLS , 後面也會多一個 SEP，還會有 PAD 做 padding 等等，因此在 encode 與 decode 的過程中會需要作些微的調整來因應位置的不同，而每個字詞會對應到一個 token id，因此只需要對應過去每個字詞所對應到的 token id 在對應回來即可。

(b) 利用 beam search 的演算法來做選取，用 np.argmax 的方式，將出來的 logit 最大的詞取出來，並反覆迭代來找到最終的結果，並且將

這個詞對應回去原來的 context，而這個最終答案的長度會被 max_answer_length 所限制，且最初所選擇概率最大的也會被 start_n_top 所影響。

2. Modeling with BERTs and their variant

(1) (a) 最初訓練是使用 bert base Chinese 的 pretrain model，configuration 如下：

"attention_probs_dropout_prob": 0.1,

"classifier_dropout": null,

"directionality": "bidi",

"hidden_act": "gelu",

"hidden_dropout_prob": 0.1,

"hidden_size": 768,

"initializer_range": 0.02,

"intermediate_size": 3072,

"layer_norm_eps": 1e-12,

"max_position_embeddings": 512,

"model_type": "bert",

"num_attention_heads": 12,

"num_hidden_layers": 12,

"pad_token_id": 0,

"pooler_fc_size": 768,

"pooler_num_attention_heads": 12,

"pooler_num_fc_layers": 3,

"pooler_size_per_head": 128,

"pooler_type": "first_token_transform",

"position_embedding_type": "absolute",

"torch_dtype": "float32",

"transformers_version": "4.18.0",

"type_vocab_size": 2,

"use_cache": true,

"vocab_size": 21128

(b) 而這個 model 最終在 kaggle 上面 public leaderboard 的成績為 0.72513 private leaderboard 的成績為 0.73080

(c) Loss function 為 Cross Entropy loss

(d) Optimizer 使用 Adam，learning rate 為 0.00003，batch size 為 3

(2) (a) 在 Context selection 的 case 上面保持使用 bert base Chinese，因為 Accuracy 已經有 0.9 多，已經算是還不錯了，而在 question answering 上面，則把 model 改成 albert 的 chinese large，configuration 如下：

"attention_probs_dropout_prob": 0,

"bos_token_id": 2,

"classifier_dropout_prob": 0.1,

"down_scale_factor": 1,

"embedding_size": 128,

"eos_token_id": 3,

"gap_size": 0,

"hidden_act": "relu",

"hidden_dropout_prob": 0,

"hidden_size": 1024,

"initializer_range": 0.02,

"inner_group_num": 1,

"intermediate_size": 4096,

"layer_norm_eps": 1e-12,

"layers_to_keep": [],

"max_position_embeddings": 512,

"model_type": "albert",

"net_structure_type": 0,

"num_attention_heads": 16,

"num_hidden_groups": 1,

"num_hidden_layers": 24,

"num_memory_blocks": 0,

"output_past": true,

"pad_token_id": 0,

"position_embedding_type": "absolute",

"tokenizer_class": "BertTokenizer",

"torch_dtype": "float32",

"transformers_version": "4.18.0",

"type_vocab_size": 2,

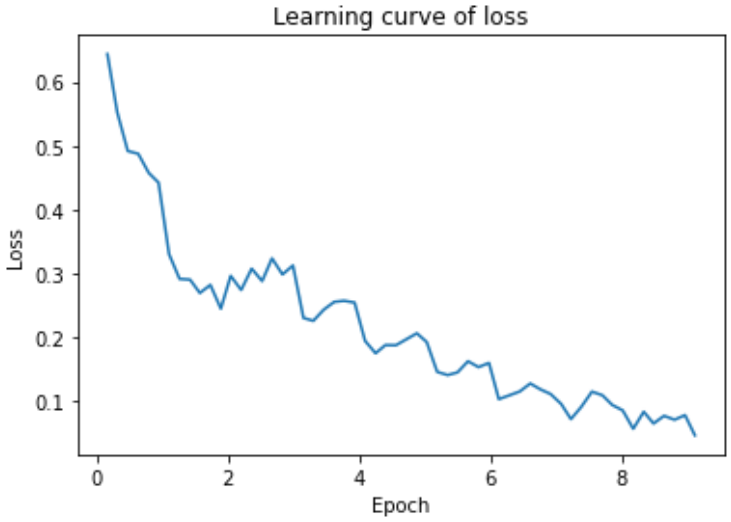
"vocab_size": 21128

(b) 這個 model 在 public leader board 上面的成績為 0.77486，

private leader board 的成績為 0.76784，明顯提升了許多

(c) 換成 albert 之後，由於是 large，網路結構較大，且有較多的 pretrain data，並搭配 cross layer sharing 的機制，結果明顯好了許多，且速度較其他的 large model 快了一點 。

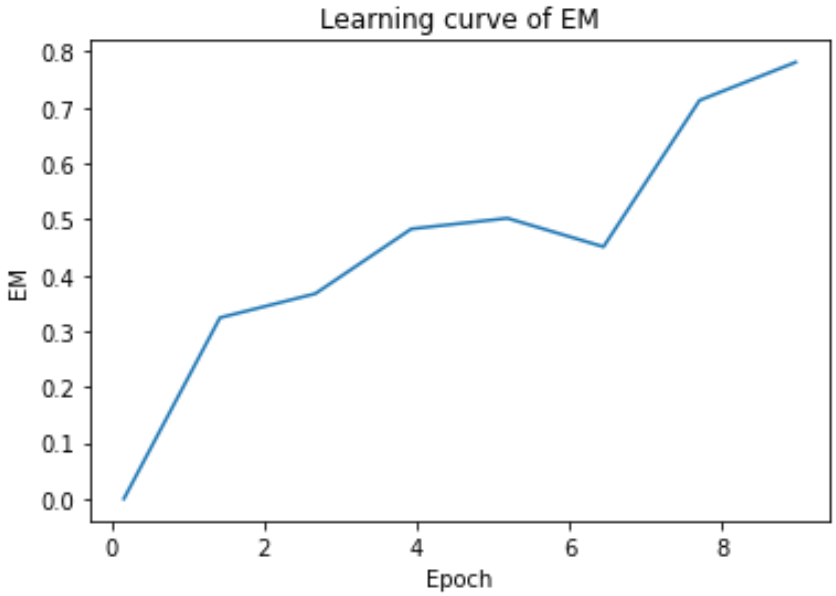
3. (a)



(b)

EM 的計算方法為 True count / data 的長度，其中 true count 為完全符合 label 的 answer 的字詞個數

結果如下：



4. Train 以一個 bert base chinese 的 model 的 config 來訓練 QA，所以內部的 hideen layer , attention head , hidden size 等等都與 bert base chinese 的 model 架構一樣，且 Tokenizer 也是用 bert base chinese 的 tokenizer，並以相同 learning rate : 0.00003，batch size : 3 來做訓練，epoch 設為 10，也就是幾乎一樣的架構只是把 load the pretrained weights 的部分省略，然而實驗結果發現 EM 的效果趨近於 0，效果極差，可能原因為 training 的 epoch 不夠多，可見 pretrained weight 的重要性。
5. 自己是用 bert 的 multiple choice ，直接將格式改成類似 swag 的格式之後，放入做 intent classification，在 private leader board 可以來到 0.954 左右，維持一樣的 category cross entropy 與 RAdam 的 optimizer，learning rate 也是設為 0.0001，效果便比之前的 BiGRU 來的好了一些，可見 BERT 的強大。