資工四陳柏均
B02902019

HW4 Translate and NLG Report

這次作業使用的是 seq2seq model based 的程式,用來處理語言間翻譯以及自然語言生成等問題.一般利用 seq2seq 做 training 時常常會遇到一個問題,如果把每個 training data 直接丟進去 train,不同長度的 input 與 output 會在 tensorflow 的架構下形成許多不同的 subgraph,儘管他們之間的差異並不是非常的大,因此整個 graph 的架構就會變得相當肥大不必要,我們利用 bucketing and padding 的 tradeoff,將 input 與 output 分成四種範圍統一做規範,利用 PAD 把不足的長度補起來.統一在 training 時 input & output data 資料長度不足所會引起的問題.另外根據 2014 Sequence to Sequence Learning with Neural Networks 這篇 paper 所闡述的,如果把 padded input 倒轉過來 train 會有更好的效果,translate.py 就是這樣做的.

另外在 tokenizing sentence 時利用 WMT'15 的 `tokenizer.perl` 也會讓 `performance` 變更好,我就跑去研究怎麼樣用 `python call perl` 的函式,但是最重要的是,其實只改一個 `optimizer` 大概就讓平均分數衝了 `0.2` 左右.因為原本的 `translate.py` 實在是還滿基礎的版本,其實參數在官網上都說是沒有 `tuned` 過的.另外在 `NLG` 的部分也做過一點小實驗,其實一個 `input` 給兩個 `output` 的設計上我覺得會 `confuse` 自己的 `model`,所以我做了一點小比較,第一個是複製 `input` 一次讓他對到兩個 `output`,第二個是兩個 `output` 任取一個當作 `training set`,另外一個就捨棄.以下是我 `translate & NLG` 的實驗結果.

Translate	
6800	0.332981
10000	0.365751
13700	0.386855
14200	0.377411
14700	0.363238
15400	0.391567
18100	0.397043
21200	0.305375
29400	0.321943
34500	0.329535
39800	0.335028

NLG 1 對 2	
20100	0.665246
22500	0.646659
33250	0.664973
33300	0.650461
39600	0.648290
44000	0.657850
52300	0.656587
54000	0.668680

NLG 1 對 1	
5200	0.657035
8300	0.661949
10400	0.651822
11800	0.657951
19000	0.662993
21400	0.663611

結論:

其實不改 optimizer 只做 data process 沒有特別的差別,改了 optimizer NLG 差不多在 0.7 以上,translate 也是從 0.4 起跳,以上.

Model 存放位址

https://drive.google.com/drive/folders/0B1ICo5FbEcqqU0FuamF1VF83QTA?usp=sharing

translate 與 NLG 各三份檔案,包含 checkpoint, XXX.ckpt, XXX.ckpt.meta 存到各自的資料夾,還有各自的 data,就可以跑 script 了.

特別跟助教說明的一件事情,我忘記存我在 coda lab 上好 performance 的檔案,不過現在提交的兩份其實都有過 strong baseline.造成不便不好意思.