Lab5

113062573 余侞璇

我們使用的RTX3070的記憶體只有8G，所以參數設太大的情況下會發生OOM，你可以嘗試比較小的參數，也可以實驗看看在什麼情況下pytorch impl的記憶體用量會超過8G

Heads = 64會超過

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令 | forward | backward | F\_b | MB |
| --batch 16  --seq 1024  --#heads 32  --emb 256  --impl Flash2  --causal  --repeat 30  --output benchmark\_result.json | Time :  0.0011807646602392196 | 0.0036186744769414266 | 0.004799439137180646 | 132.00048828125 |
| Flops :    7.27489133207942 | 5.934448267408387 | 6.264225925711201 |
| --batch 16  --seq 1024  --#heads 32  --emb 256  --impl Pytorch  --causal  --repeat 30  --output benchmark\_result.json | 0.018689655140042304 | 0.026574663817882538 | 0.04526431895792484 | 4184.25 |
| 0.459609047231492 | 0.8080943799390313 | 0.6642046486979406 |
| --batch 16  --seq 512  --#heads 32  --emb 256  --impl Flash2  --causal  --repeat 30  --output benchmark\_result.json | 0.003348046789566676 | 0.0011760104447603227 | 0.0045240572343269985 | 66.00048828125 |
| 0.6414138699292012 | 4.565188297365991 | 1.661383218357562 |
| --batch 16  --seq 512  --#heads 32  --emb 256  --impl Pytorch  --causal  --repeat 30  --output benchmark\_result.json | 0.0042635375012954075 | 0.0068192501862843835 | 0.011082787687579792 | 1076.25 |
| 0.5036858822861349 | 0.7872873077450847 | 0.6781861188609805 |
| --batch 16  --seq 256  --#heads 32  --emb 256  --impl Flash2  --causal  --repeat 30  --output benchmark\_result.json | 0.00011854097247123718 | 0.0004377997169891993 | 0.0005563406894604364 | 33.00048828125 |
| 4.528990279122828 | 3.065733548733911 | 3.377513504939542 |
| --batch 16  --seq 256  --#heads 32  --emb 256  --impl Pytorch  --causal  --repeat 30  --output benchmark\_result.json | 0.00113937146961689 | 0.0018041191001733144 | 0.0029434905697902046 | 290.25 |
| 0.47119918860222215 | 0.7439515938116628 | 0.6383741165285703 |
| --batch 16  --seq 1024  --#heads 16  --emb 256  --impl Flash2  --causal  --repeat 30  --output benchmark\_result.json | 0.003171409914890925 | 0.001852172240614891 | 0.005023582155505816 | 98.00048828125 |
| 2.7085538680027224 | 11.594405751848814 | 5.984727658738335 |
| --batch 16  --seq 1024  --#heads 16  --emb 256  --impl Pytorch  --causal  --repeat 30  --output benchmark\_result.json | 0.009639596318205197 | 0.01350617321829001 | 0.023145769536495206 | 2136.25 |
| 0.8911093689449607 | 1.590001559503091 | 1.2989315833545836 |
| --batch 16  --seq 1024  --#heads 8  --emb 256  --impl Flash2  --causal  --repeat 30  --output benchmark\_result.json | 0.00031337899466355644 | 0.0009550636013348897 | 0.0012684425959984462 | 81.00048828125 |
| 27.41069037260187 | 22.485242291701496 | 23.702114046662643 |
| --batch 16  --seq 1024  --#heads 8  --emb 256  --impl Pytorch  --causal  --repeat 30  --output benchmark\_result.json | 0.00501799335082372 | 0.006972300633788109 | 0.011990293984611829 | 1112.25 |
| 1.7118266190189222 | 3.0800215894208427 | 2.5074256820211995 |
| --batch 16  --seq 1024  --#heads 32  --emb 128  --impl Flash2  --causal  --repeat 30  --output benchmark\_result.json | 0.001349905381600062 | 0.00370534285902977 | 0.005055248240629832 | 148.00048828125 |
| 3.1816802529589996 | 2.897820430795857 | 2.9736196563370187 |
| --batch 16  --seq 1024  --#heads 32  --emb 128  --impl Pytorch  --causal  --repeat 30  --output benchmark\_result.json | 0.018850571538011233 | 0.026448387776811917 | 0.04529895931482315 | 4148.25 |
| 0.2278428156589 | 0.4059762859879804 | 0.3318483639221478 |
| --batch 16  --seq 1024  --#heads 32  --emb 512  --impl Flash2  --causal  --repeat 30  --output benchmark\_result.json | 0.003115176161130269 | 0.0037004832178354264 | 0.006815659378965696 | 196.00048828125 |
| 5.5148949193828845 | 11.606503916297486 | 8.822263379177983 |
| --batch 16  --seq 1024  --#heads 32  --emb 512  --impl Pytorch  --causal  --repeat 30  --output benchmark\_result.json | 0.019109233841300012 | 0.027041939894358318 | 0.04615117373565833 | 4256.25 |
| 0.8990349548640639 | 1.5882615347784448 | 1.3028821864511193 |

我總共實驗了三種不同的參數變化，分別是seq、heads與emb。基礎指令為

--batch 16

--seq 1024

--heads 32

--emb 256

--impl Flash2 / Pytorch

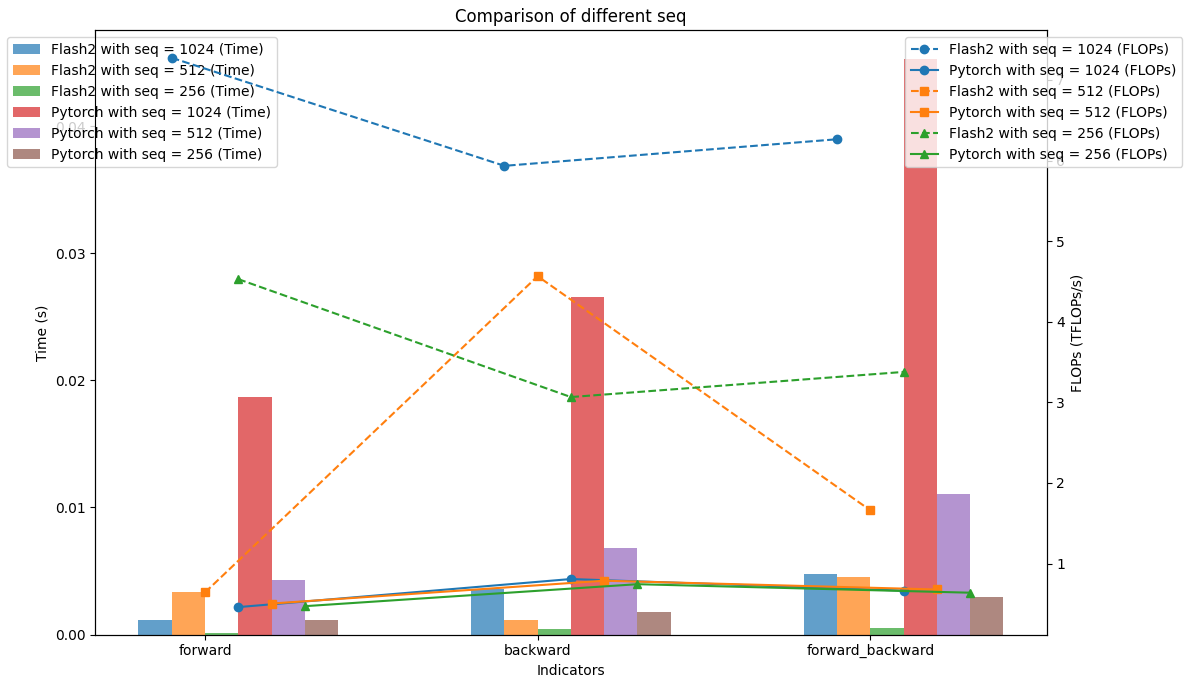
--causal

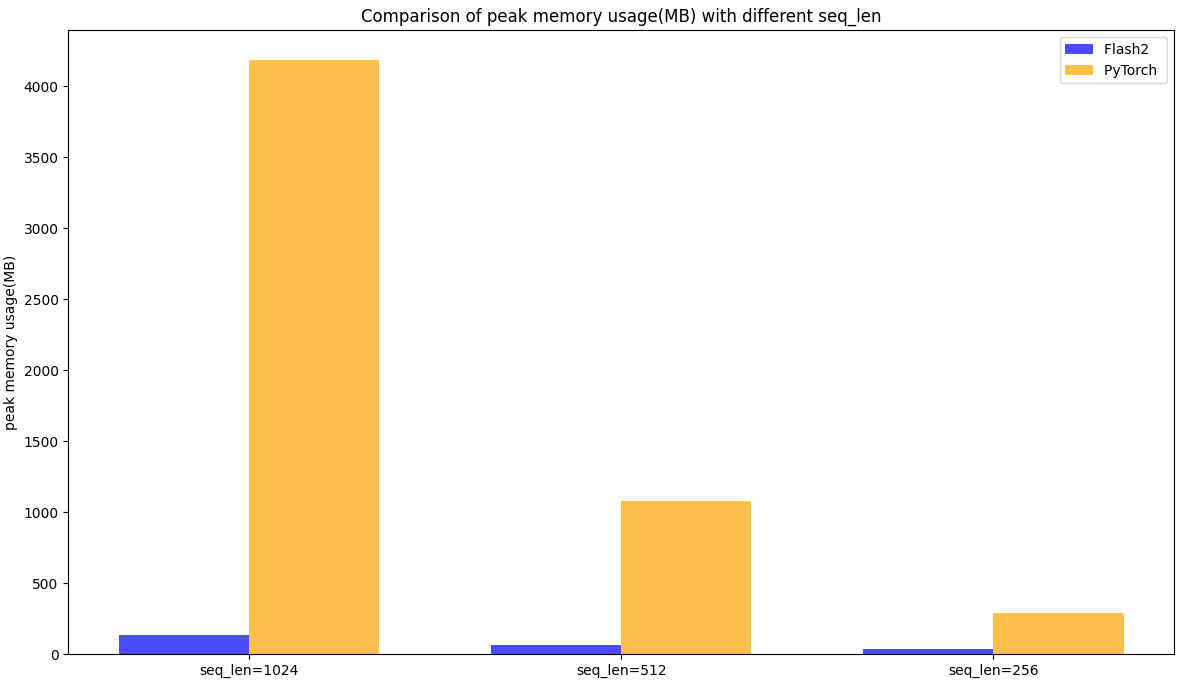
--repeat 30

基於上述指令去修改seq, heads與emb，觀察Flash2和Pytorch的變化。

* seq

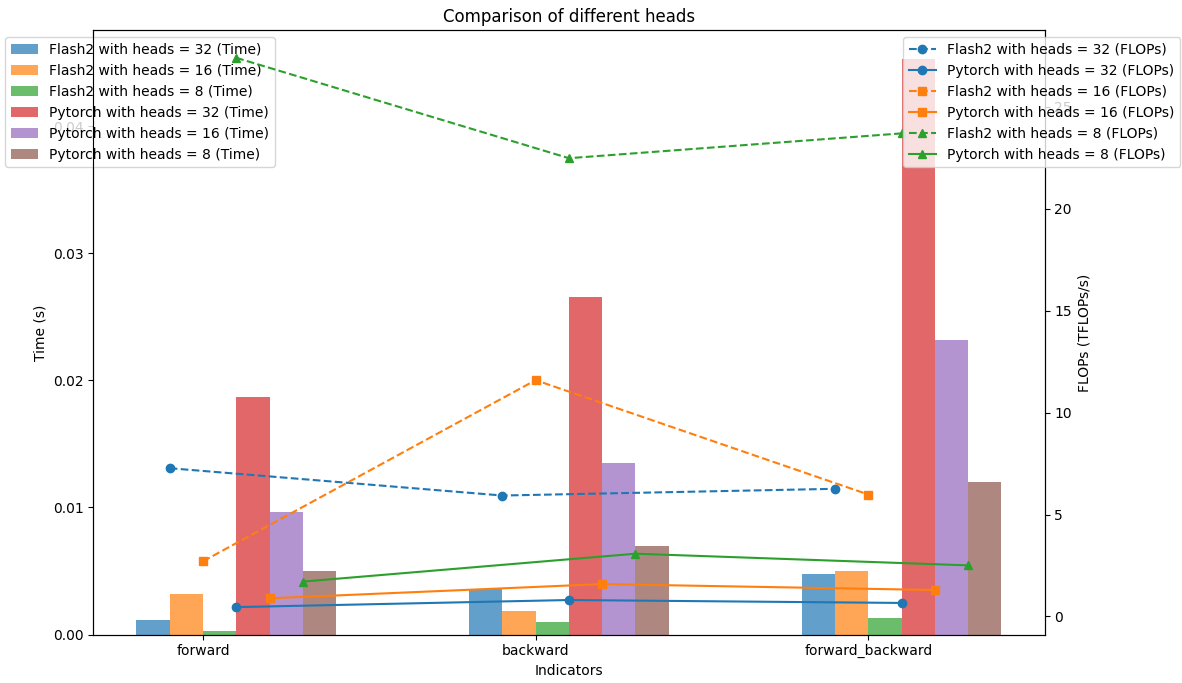
首先是seq長度分別設成1024、512、256，可以發現flash2所花的execution time遠低於pytorch，隨著需要處理的seq長度變小，時間大致上也越小。不過Flash2 的 FLOPs高於pytorch，可能是因為tiling等方式，會多次計算中間值與加權和，增加了不少計算量，但由於減少了對記憶體的access次數，故整體的execuation time仍然低於pytorch。

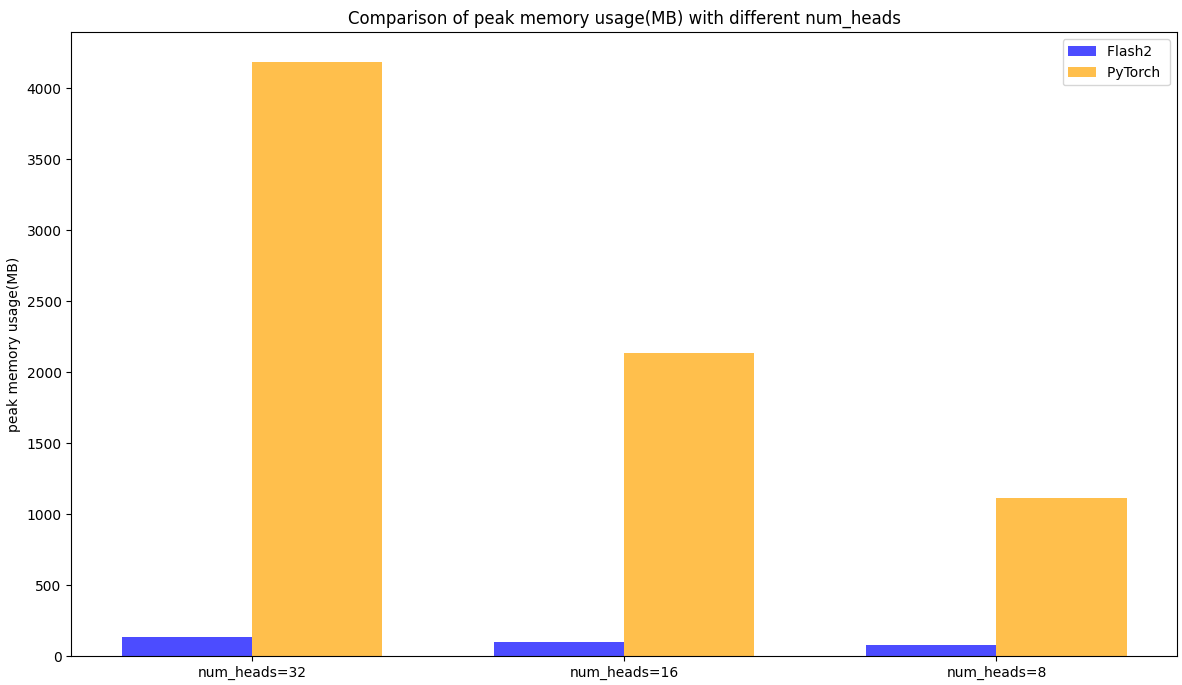




* heads

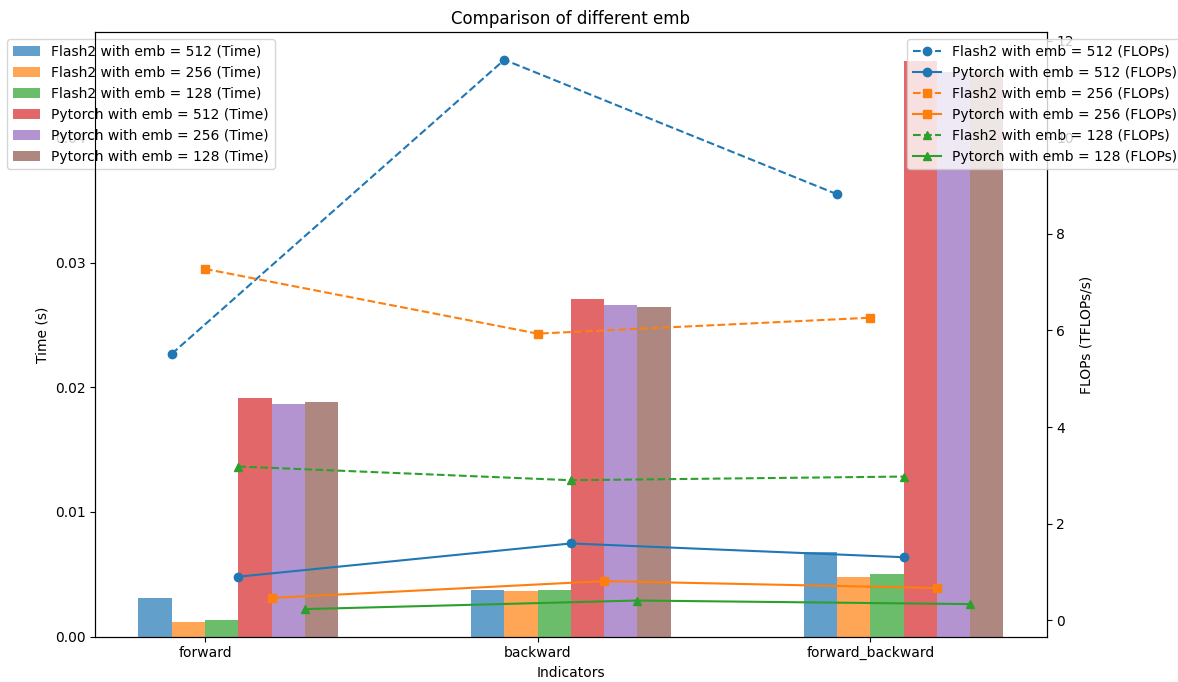
number of heads分別設成32、16、8。Execution time很明顯仍是flash2遠低於pytorch，不管head數量。至於FLOPs則是heads數量越少會變越大，可能是因為heads變少導致維度增加，矩陣計算的FLOPs也隨之成長。如果不想要太多的計算量，取得適中的heads數量非常重要。可以發現Flash2的memory usage隨著head數量變化並不大，幾乎持平，可能是因為每個block處理的數據量沒有改變，故memory usage差異不明顯。

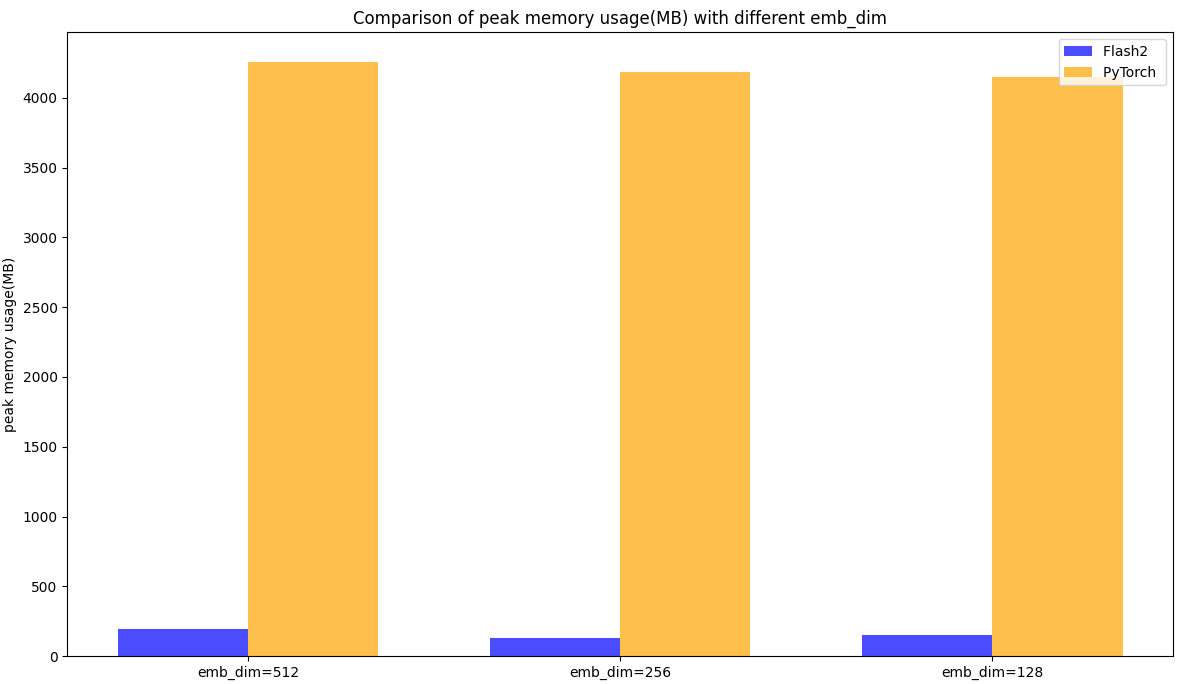




* emb

emb分別設成512、256、128。Execution time很明顯仍是flash2遠低於pytorch，FLOPs亦然，因為矩陣運算的複雜度就是直接依賴emb\_dim。這裡的memory usage就有隨著emb\_dim變小而跟著變化了。





一開始使用spec提供的範例指令：python lab5.py \ --batch\_size 32 \ --seq\_len 1024 \ --num\_heads 32 \ --emb\_dim 2048 \ --impl Flash2 \ --causal \ --repeats 30 \ --output benchmark\_result.json，但遇到了記憶體不夠用的情況。後來嘗試了調小batch\_size、seq\_len與num\_heads等組合，才終於找到現在使用的基本指令，不會記憶體不夠用。

綜合上述，Flash2使用了tiling分塊處理matrix，故可以在seq\_len很長的情況下顯著縮短執行時間，同時也降低內存的壓力；另外， Flash2可以有效地將每個head的計算平行化，所以head數量增加時，Flash2的 execution time 增長速度也比 Pytorch慢；最後是emb\_dim，Flash2會將大matrix分塊處理以減少memory overhead，並且對softmax等步驟進行了inlined optimization。

不過，Flash2的 FLOPs都比 Pytorch高，但因其memory access的效率比Pytorch來的好，故整體執行時間更短。

Flash2在sequence length長、multi-heads與 emb\_dim大的情況下，可以顯著降低執行時間與記憶體需求。