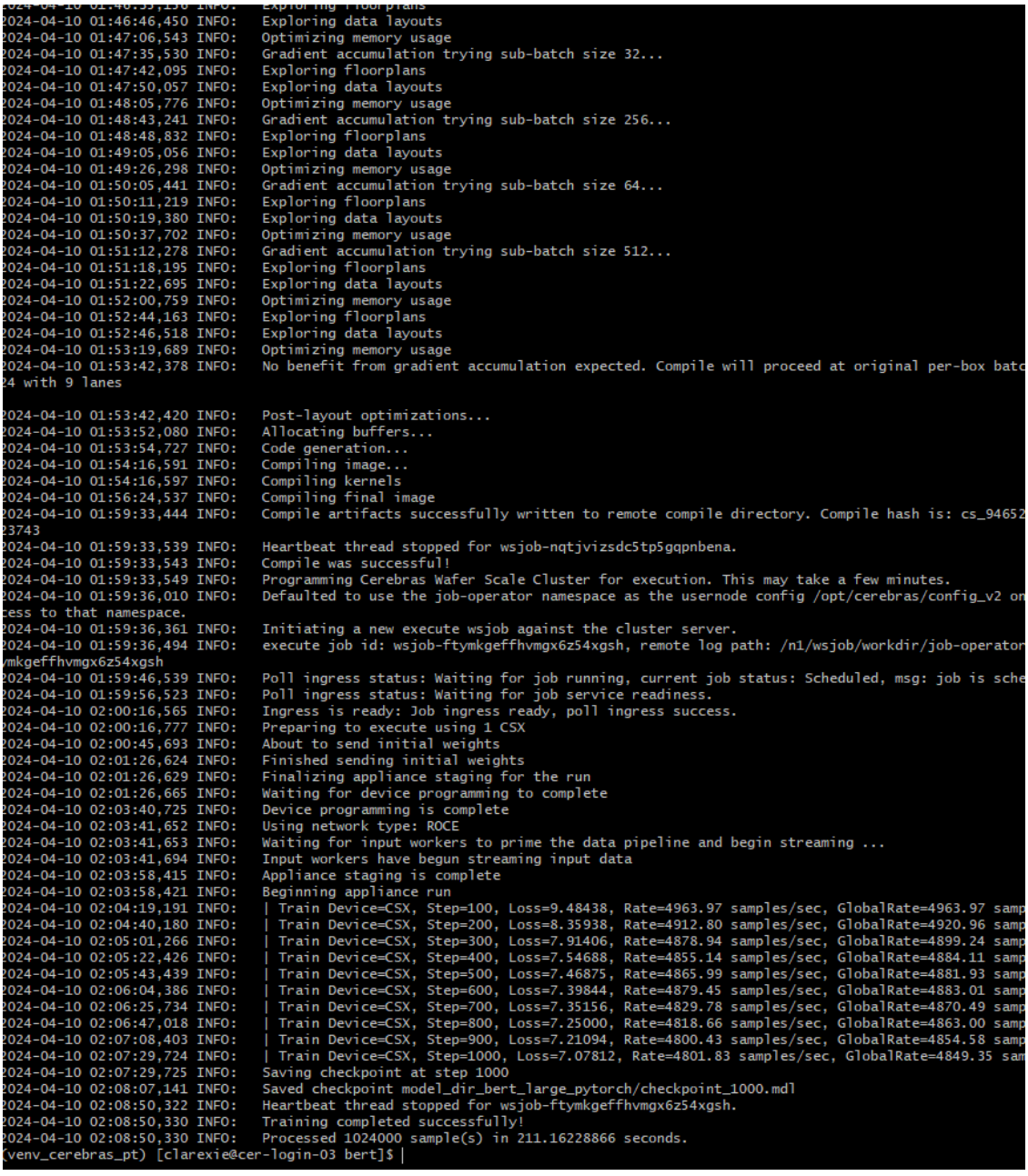
Cerebras

Homework

Run BERT example with different batch sizes like 512, 2048 and observe the performance difference, results on Polaris, it ran faster it fit into the cache more effectively, because data loading is expensive. (here is the result when I varied the context window instead from 128 to 512 and observed very interesting performance differences).



The above is 128

Next we will do 512:

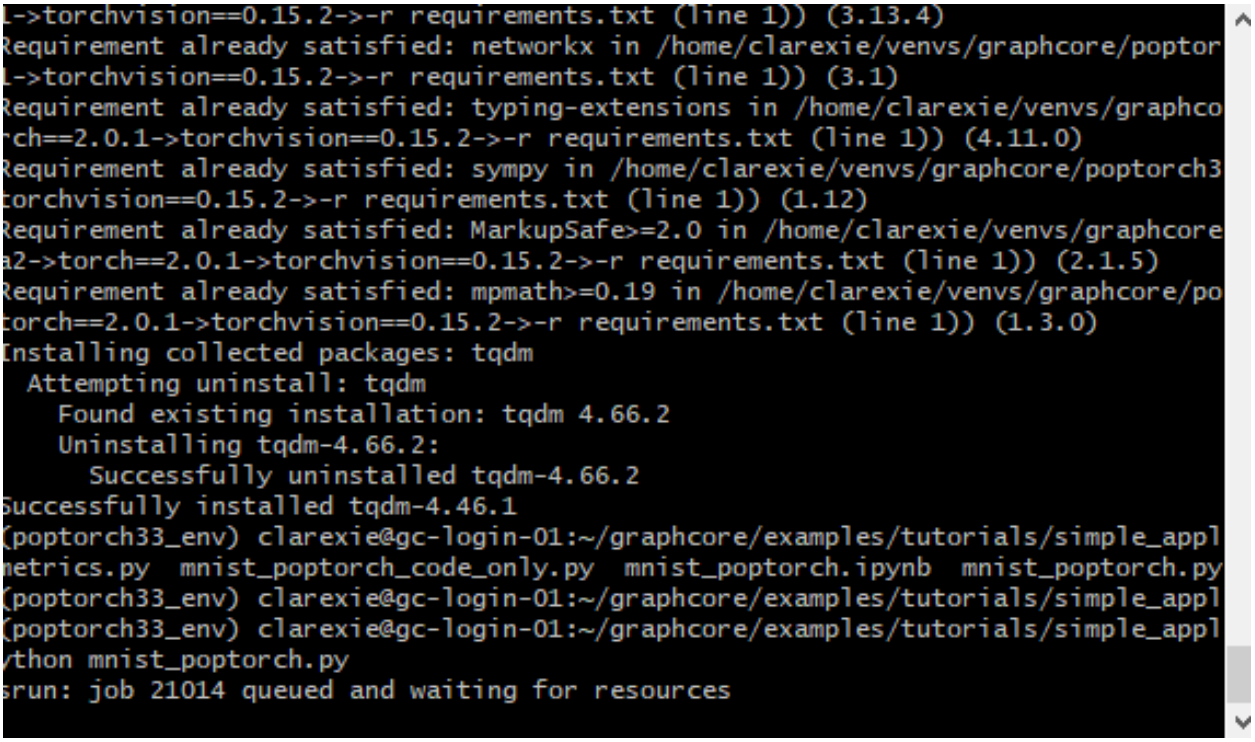
```
2024-04-10 02:30:03,658 INFO:   Gradient accumulation trying sub-batch size 128...
2024-04-10 02:30:10,341 INFO:   Exploring floorplans
2024-04-10 02:30:21,874 INFO:   Exploring data layouts
2024-04-10 02:30:55,002 INFO:   Optimizing memory usage
2024-04-10 02:32:10,138 INFO:   Gradient accumulation trying sub-batch size 32...
2024-04-10 02:32:16,668 INFO:   Exploring floorplans
2024-04-10 02:32:23,780 INFO:   Exploring data layouts
2024-04-10 02:32:41,871 INFO:   Optimizing memory usage
2024-04-10 02:33:10,583 INFO:   Gradient accumulation trying sub-batch size 256...
2024-04-10 02:33:17,494 INFO:   Exploring floorplans
2024-04-10 02:33:35,254 INFO:   Exploring data layouts
2024-04-10 02:34:05,529 INFO:   Optimizing memory usage
2024-04-10 02:35:05,254 INFO:   Gradient accumulation trying sub-batch size 64...
2024-04-10 02:35:11,330 INFO:   Exploring floorplans
2024-04-10 02:35:19,597 INFO:   Exploring data layouts
2024-04-10 02:35:38,426 INFO:   Optimizing memory usage
2024-04-10 02:36:23,703 INFO:   Gradient accumulation trying sub-batch size 512...
2024-04-10 02:36:29,658 INFO:   Exploring floorplans
2024-04-10 02:36:33,619 INFO:   Exploring data layouts
2024-04-10 02:37:12,202 INFO:   Optimizing memory usage
2024-04-10 02:38:10,893 INFO:   Exploring floorplans
2024-04-10 02:38:12,928 INFO:   Exploring data layouts
2024-04-10 02:38:53,426 INFO:   Optimizing memory usage
2024-04-10 02:39:46,356 INFO:   Gradient accumulation picked sub-batch size 512 with 11 lanes

2024-04-10 02:39:46,423 INFO:   Post-layout optimizations...
2024-04-10 02:39:56,924 INFO:   Allocating buffers...
2024-04-10 02:39:59,608 INFO:   Code generation...
2024-04-10 02:40:12,420 INFO:   Compiling image...
2024-04-10 02:40:12,425 INFO:   Compiling kernels
2024-04-10 02:42:31,832 INFO:   Compiling final image
2024-04-10 02:45:14,838 INFO:   Compile artifacts successfully written to remote compile directory. Compile hash is: cs_25301974003835
11140
2024-04-10 02:45:14,907 INFO:   Heartbeat thread stopped for wsjob-j3v7kj7w9tzyibxkt9ynqp.
2024-04-10 02:45:14,911 INFO:   Compile was successful!
2024-04-10 02:45:14,917 INFO:   Programming Cerebras Wafer Scale Cluster for execution. This may take a few minutes.
2024-04-10 02:45:17,247 INFO:   Defaulted to use the job-operator namespace as the usernode config /opt/cerebras/config_v2 only has ac
cess to that namespace.
2024-04-10 02:45:17,637 INFO:   Initiating a new execute wsjob against the cluster server.
2024-04-10 02:45:17,785 INFO:   execute job id: wsjob-62bb2sg873fruutuunbw3s, remote log path: /n1/wsjob/workdir/job-operator/wsjob-62
bb2sg873fruutuunbw3s
2024-04-10 02:45:27,852 INFO:   Poll ingress status: Waiting for job running, current job status: Scheduled, msg: job is scheduled.
2024-04-10 02:45:37,872 INFO:   Poll ingress status: Waiting for job service readiness.
2024-04-10 02:45:57,916 INFO:   Ingress is ready: Job ingress ready, poll ingress success.
2024-04-10 02:45:58,098 INFO:   Preparing to execute using 1 CSX
2024-04-10 02:46:24,816 INFO:   About to send initial weights
2024-04-10 02:47:06,288 INFO:   Finished sending initial weights
2024-04-10 02:47:06,292 INFO:   Finalizing appliance staging for the run
2024-04-10 02:47:06,343 INFO:   Waiting for device programming to complete
2024-04-10 02:49:28,922 INFO:   Device programming is complete
2024-04-10 02:49:29,916 INFO:   Using network type: ROCE
2024-04-10 02:49:29,917 INFO:   Waiting for input workers to prime the data pipeline and begin streaming ...
2024-04-10 02:49:29,966 INFO:   Input workers have begun streaming input data
2024-04-10 02:49:46,885 INFO:   Appliance staging is complete
2024-04-10 02:49:46,889 INFO:   Beginning appliance run
2024-04-10 02:50:53,525 INFO:   | Train Device=CSX, Step=100, Loss=10.17188, Rate=1538.38 samples/sec, GlobalRate=1538.38 samples/sec
2024-04-10 02:52:00,479 INFO:   | Train Device=CSX, Step=200, Loss=9.14062, Rate=1533.00 samples/sec, GlobalRate=1533.88 samples/sec
2024-04-10 02:53:07,531 INFO:   | Train Device=CSX, Step=300, Loss=8.39062, Rate=1529.51 samples/sec, GlobalRate=1531.65 samples/sec
2024-04-10 02:54:14,446 INFO:   | Train Device=CSX, Step=400, Loss=8.15625, Rate=1529.97 samples/sec, GlobalRate=1531.30 samples/sec
2024-04-10 02:55:21,168 INFO:   | Train Device=CSX, Step=500, Loss=7.90625, Rate=1532.83 samples/sec, GlobalRate=1531.99 samples/sec
2024-04-10 02:56:28,011 INFO:   | Train Device=CSX, Step=600, Loss=7.89062, Rate=1532.30 samples/sec, GlobalRate=1531.98 samples/sec
2024-04-10 02:57:34,929 INFO:   | Train Device=CSX, Step=700, Loss=7.84375, Rate=1531.06 samples/sec, GlobalRate=1531.73 samples/sec
2024-04-10 02:58:41,828 INFO:   | Train Device=CSX, Step=800, Loss=7.77344, Rate=1530.83 samples/sec, GlobalRate=1531.60 samples/sec
2024-04-10 02:59:48,490 INFO:   | Train Device=CSX, Step=900, Loss=7.67188, Rate=1533.99 samples/sec, GlobalRate=1532.10 samples/sec
2024-04-10 03:00:55,599 INFO:   | Train Device=CSX, Step=1000, Loss=7.46875, Rate=1529.12 samples/sec, GlobalRate=1531.47 samples/sec
2024-04-10 03:00:55,600 INFO:   Saving checkpoint at step 1000
2024-04-10 03:01:33,909 INFO:   Saved checkpoint model_dir_bert_large_pytorch_512/checkpoint_1000.mdl
2024-04-10 03:02:52,525 INFO:   Heartbeat thread stopped for wsjob-62bb2sg873fruutuunbw3s.
2024-04-10 03:02:52,540 INFO:   Training completed successfully!
2024-04-10 03:02:52,541 INFO:   Processed 1024000 sample(s) in 668.637653252 seconds.
```

Graphcore

Homework

Run MNIST example by changing values of the input parameters like batch-size, learning rate and number of epochs trained and observe and report the performance implications.

Unfortunately, I've been trying forever. This job just is in queue forever:

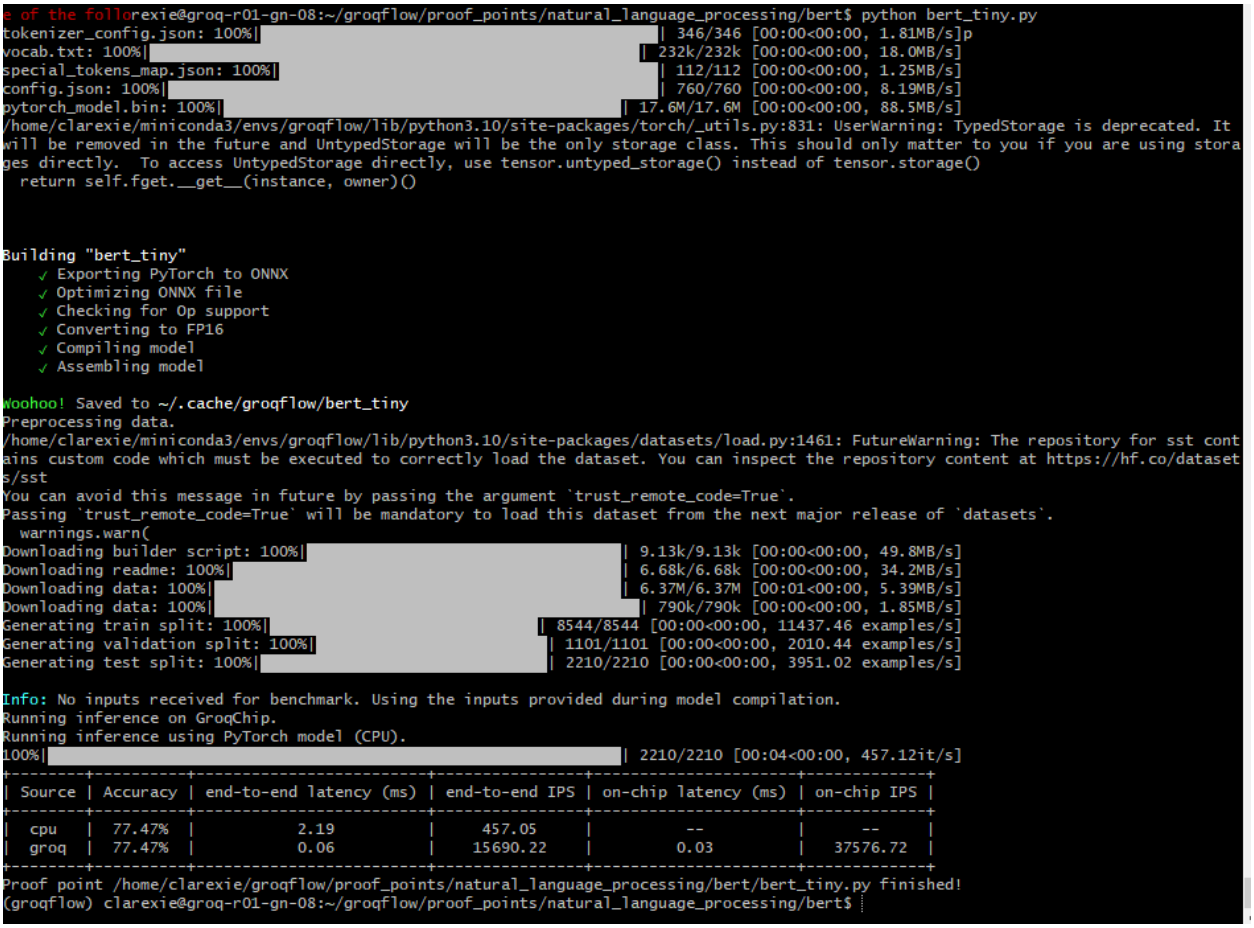


Groq

Homework

Run BERT example with custom input instead of dummy input.

Notes: I found that there's some issue with requirements.txt that results in a bunch of extra work. They need to specify the pandas version to be 1.4.4 or something close to that because the required numpy version doesn't work with the most recent pandas verison that requirments installs.



this is what I switched the inputs to be:

```
inputs = {
    "input_ids":torch.arange(max_seq_length, dtype=torch.long).expand(batch_size, max_seq_length),
    "attention_mask": torch.eye(batch_size, max_seq_length, dtype=torch.bool),
}
```

still pretty arbitrary, but at least this varys the index and move the mask.

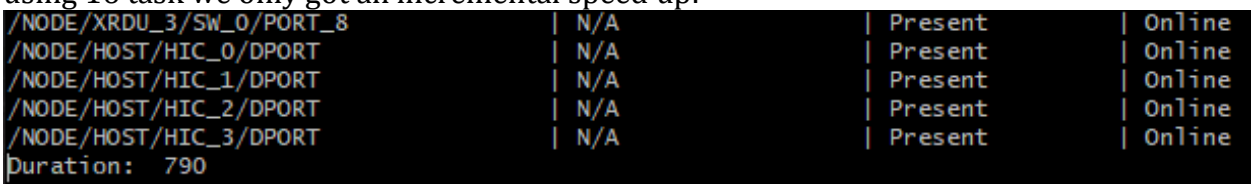
Sambanova

Homework

For BERT example, understand flags used in the script. Change values for flag `--ntasks` and measure its effect on performance.

Answer:

using 16 task we only got an incremental speed up:



this is likely because of all the communication necessary

here is the sambnova log file path ~/ai-science-training-series/07_AITestbeds/Sambanova/bert, we call them task8 and task16 respectively