|  |  |  |
| --- | --- | --- |
| Cores in Pool | chuncksize | Processing time |
| 64 | 4 | 1.54 sec |
| 25 | 12 | 0.88sec |

**Multi-Processing**

**Writing time**

|  |  |
| --- | --- |
| **Packages** | **Writing time of image** |
| Opencv | 0.2 sec |
| PIL | 0.5 sec |
| matplotlib | 1.5 sec |

|  |  |  |  |
| --- | --- | --- | --- |
| **Cores in pool** | **Model inference time** | **Post processing time** | **Pool time** |
| 64 | 3.14 sec | 5.1 sec | 4.15sec |
| 25 | 5.59 sec | 2.62 sec | 2.1 sec |

|  |  |
| --- | --- |
| **Multiprocessing** | **Pool time** |
| IBM code | 1.54 sec |
| Spawn | 1.4sec |
| Ray | 3.1 sec |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Model inference time** | **Post processing time** | **Pool time** |
| IBM | 3.14 sec | 5.1 sec | 4.15sec |
| Ray+Batch size (1) | 4.4 sec | 1.6 sec | 1 sec |
| Ray+ Batch size(6) | 3.62sec | 2.94sec | 2.86sec |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Model Inference time | Post processing time | Total ED |
| Ray and batch size 6 | 3.62sec | 2.94sec | 7.5 sec |

Procedure for onnx conversion and triton loading.

DEV\_000F310380BA-30224118383000

DEV\_000F310380BB-30224118383000

DEV\_000F310380BC-30224118383000

DEV\_000F310380BD-30224118383000

DEV\_000F310380BE-3 0224118383000

DEV\_000F310380BF-30224118383000

DEV\_000F310380BG-30224118383000

DEV\_000F310380BH-30224118383000

DEV\_000F310380BI-30224118383000

DEV\_000F310380BJ-30224118383000

DEV\_000F310380BK-30224118383000

DEV\_000F310380BL-30224118383000

DEV\_000F310380BM-30224118383000

DEV\_000F310380BN-30224118383000

DEV\_000F310380BO-30224118383000

Python dali\_grpc\_client.py --model\_name “Human\_pre” --input\_name “DALI\_INPUT\_0” --output\_name “DALI\_OUTPUT\_0” --img “/home/bashfaq/code/Triton\_client\_server/dali/dali\_backend/client/data/DEV\_000F310380BC-30224118383000.jpg”

|  |  |
| --- | --- |
|  | Time saved |
| Before Optimization | 488.8ms |
| After Optimization | 56.801 ms |

|  |  |
| --- | --- |
|  | Inference time |
| Before Optimization | 15.747ms |
| After Optimization | 13.45ms |

|  |  |  |
| --- | --- | --- |
| Instance Group | FP16 | FP32 |
| 1 | throughput: 74.2 infer/sec, latency 13731 usec | throughput: 39.4 infer/sec, latency 25739 usec |
| 2 | throughput: 73.6 infer/sec, latency 13781 usec | throughput: 40.4 infer/sec, latency 25146 usec |

FP 16: count 2

Inferences/Second vs. Client Average Batch Latency

Concurrency: 1, throughput: 75.2 infer/sec, latency 13275 usec

Concurrency: 2, throughput: 86 infer/sec, latency 23223 usec

Concurrency: 3, throughput: 107.4 infer/sec, latency 27901 usec

Concurrency: 4, throughput: 110 infer/sec, latency 36302 usec

FP32:

Inferences/Second vs. Client Average Batch Latency

Concurrency: 1, throughput: 39.2 infer/sec, latency 25483 usec

Concurrency: 2, throughput: 53.4 infer/sec, latency 37375 usec

Concurrency: 3, throughput: 52.8 infer/sec, latency 56705 usec

Concurrency: 4, throughput: 53.2 infer/sec, latency 75682 usec

|  |  |  |
| --- | --- | --- |
|  |  | **Throughput** |
| Before optimization | FP32 model | 53 infer/sec |
| After optimization | FP16 model | 110 infer/sec |
| After optimization | TensorRT backend on FP16 model | 217 infer/sec |

Pytorch model code debug:

training\_dir =”/home/Ashfaq/Classic” #Line 336

checkpoint\_path =”/home/Ashfaq/Classic/19/19\_model.pth” # Line 338

--is\_testing =False

|  |  |
| --- | --- |
| **Modules** | **Time consumed in secs (Avg)** |
| Ig | 1.15 |
| ED | 3.52 |
| EV | 1.3 |
| DM | 0.057 |
| **Total** | **6.02** |

1. Move some of the pre-processing and post-processing steps to GPU, 🡪 optimize ~1.2 secs
2. Deploy the trained model to triton (pt or onnx) –> urgent and important
3. Quantize models in PyTorch / Triton –> saves 50% of inference time w.r.t FP16
4. Improve on batching of Triton servers
5. Vectorize pre-processing, post-processing steps (get rid of for loops)
6. Train the model with higher resolution of tile images.

Optimization:

ED Module :

**Model:**

Tensorflow :

1. Dynamic Batching- Increased extra overhead, NA Refer #91708
2. Batching- Optimised at batch 6,

<https://docs.nvidia.com/deeplearning/tensorrt/developer-guide/index.html#streaming>

1. Tensorflow -TensorRT backend – Custom layers created problem, Refer 95520
2. Quantization – Tensorflow v2.0 doesn’t have quantization support so need to downgrade the model to v1.0 and tried with onnx model but it didn’t work due to custom layer support in onnx , Refer 91689
3. TensorRT – custom layers are not supported in TensorRT
4. TF-TRT- falling back to tensorflow model, NA

Pytorch:

1. Torchscript conversion – Failed due to custom layer are not supported in JIT, Refer 102455
2. TensorRT – Pytorch 1.9-> Onnx-> TensorRT ,

<https://blog.xmartlabs.com/2020/06/01/how-to-speed-up-inference-in-your-deep-learning-model/>

1. Pruning in Pytorch : Quanlity of predicted image is reduced, Applied on last layer, some weights are zero

Post -processing:

1. Ray with GPU – Increased Extra overhead, NA
2. Dali - Need to code in cuda c++
3. Numba JIT – Extra overhead, NA

dynamic\_axes={'input' : {0 : 'batch\_size'},    # variable length axes

                                '492' : {0 : 'batch\_size'},

                                '448':{0 : 'batch\_size'},

                                '451':{0 : 'batch\_size'},

                                '466':{0 : 'batch\_size'},

                                '478':{0 : 'batch\_size'},

                                '490':{0 : 'batch\_size'}}

🡪 17\*4 = 68hrs

torch.onnx.export(model, x\_ft\_gpu, "19\_model\_dexined.onnx", input\_names=['input'],output\_names = ['output\_1','output\_2','output\_3','output\_4','output\_5','output\_6','output\_7'],

dynamic\_axes={'input' : {0 : 'batch\_size'}, 'output\_7' : {0 : 'batch\_size'}}, export\_params=True,verbose=True)

Single-GPU ----> 1d 20h 20m 20s --> 24+20 = 44hrs

**Experiment 1:**

Batch size = 128

Lr = 1.6e-3

**Multi-GPU --🡪 11hr 29 min 6sec**

**Experiment 2:**

Batch – 96

Lr=1.2e-3

Note: Predictions are not good

**Experiment 3:**

Batch= 64

Lr = 8e-4

**Multi-GPU --🡪 12hrs 13min 6sec**

B1,UD1,B2,UD2,B3

<https://www.bestbuy.com/site/asus-rog-zephyrus-m15-15-6-4k-ultra-hd-gaming-laptop-intel-core-i7-16gb-memory-nvidia-geforce-rtx-2060-1tb-ssd-prism-black/6403817.p?skuId=6403817>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Tensorflow version of ED | Pytorch without dynamic batching | Pytorch with dynamic batching | Pytorch FP16 with out dynamic batching |
| Pre-processing time | 600.08ms | 990.1ms | 2146.9ms | 947.5ms |
| **Inference time** | **1722.1ms** | **2397.0ms**  **(sending+generating+inference)** | **12602.6ms** | **1597.8ms** |
| Post-processing time | 1275.4ms | 1465.6ms | 2664.0ms | 2129.4ms |
| **Total Time** | **3599.4ms** | **4853.6ms** | **17414.2ms** | **4677.1ms** |

**100ms**

**Used 4-gpus for tensorflow and pytorch**

|  |  |  |
| --- | --- | --- |
| **Request** | Currency 1 | Currency 5 |
| Generator | 1064.9 (+- 0.1) ms |  |
| sending | 1.5ms (min=0.7 , max=4.3) |  |
| **Total request time** | **1066.4 ms** |  |
| **Actual inference time(Response)** | **81.4 ms (+- 3.1)** |  |
| **Total inference time** | **1153.4 ms** |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Model inference time** | **Post processing time** | **Pool time** |
| Multi-processing | 3.14 sec | 5.1 sec | 4.15sec |
| Ray | 4.4 sec | 1.6 sec | 1 sec |

|  |  |  |
| --- | --- | --- |
|  | **Pytorch without dynamic batching** | **Pytorch FP16 with out dynamic batching** |
| Pre-processing time | 990.1ms | 947.5ms |
| **Inference time** | **2397.0ms** | **1597.8ms** |
| Post-processing time | 1465.6ms | 2129.4ms |
| **Total Time** | **4853.6ms** | **4677.1ms** |

Pytorch:

|  |  |  |
| --- | --- | --- |
|  | **Currency 1-pytorch (ms)** | **Currency 1-tensorflow(ms)** |
| Total\_request\_time | 898.3 | 448.32 |
| Tile sending time(sum) | 296.8 | 282.1 |
| **Actual inference time** | **610.42** | 166.1 |
| **Response\_time** | **39.08** | 413.4 |
| **Total inference time** | **939.75** | 867.3 |

Action plan

1. Run experiment with pytorch with 4 -gpu on dataset.
2. Reduce model with FP16 needs to test on 5 teams.
3. Instance group needs to check with perf\_analyser and at deployment.
4. On pre-processing and post-processing.

What I have tried?

1. I have updated the triton client version to 2.19.0.
2. Converted to pytorch to torch script format .

Please let me know if you have any idea regarding this issue.

Regards,

Ashfaq Baig

1. What is latency?
2. Rerun the model\_analyser and save best config . Rerun with previous best config.
3. Measure s  **: (sending+generating+inference) on 10 payloads for tensorflow and pytorch for currency 5 & 1.**

## Environment

**GPU Name**: NVIDIA A100-PCIE-40GB  
**Nvidia Driver Version**: 510.47.03  
**CUDA Version**: 11.6  
**Operating System + Version**: Ubuntu 18.04  
**Python Version (if applicable)**: 3.8.5  
**Triton client (if applicable)**: 2.19  
**PyTorch Version (if applicable)**: 1.9.1+cu111:  
\*\* torchvision version: 0.10.1+cu111  
\*\* python library onnxruntime-gpu version \*\*: 1.10.0  
**Triton Server**: [nvcr.io/nvidia/tritonserver:22.02-py3](http://nvcr.io/nvidia/tritonserver:20.10-py3)

I have converted pytorch keypoint rcnn to onnx model and loaded in triton server   
When ran triton client against the onnx model, it managed to load the model. But when the model tried predicting on a image with session.run(), it returns the following error:

I exported torchvision's keypoint-RCNN as an ONNX model and loaded in triton server  to run inference with it. This works fine for a single image. Running a list of images, one at a time, through the same triton client works fine using the CPU. When using the GPU to load the model inside triton server, however, I get a CUDNN error if the first image has zero detections and the next image has at least some detections.

CUDNN failure 3: CUDNN\_STATUS\_BAD\_PARAM ; GPU=2 ; hostname=9a41efbf088d ; expr=cudnnFindConvolutionForwardAlgorithmEx( s\_.handle, s\_.x\_tensor, s\_.x\_data, s\_.w\_desc, s\_.w\_data, s\_.conv\_desc, s\_.y\_tensor, s\_.y\_data, 1, &algo\_count, &perf, algo\_search\_workspace.get(), max\_ws\_size);  
2022-03-14 06:01:11.911637097 [E:onnxruntime:, sequential\_executor.cc:346 Execute] Non-zero status code returned while running FusedConv node. Name:'Conv\_2152\_Relu\_2153' Status Message: CUDNN error executing cudnnFindConvolutionForwardAlgorithmEx( s\_.handle, s\_.x\_tensor, s\_.x\_data, s\_.w\_desc, s\_.w\_data, s\_.conv\_desc, s\_.y\_tensor, s\_.y\_data, 1, &algo\_count, &perf, algo\_search\_workspace.get(), max\_ws\_size)  
2022-03-14 06:02:09.256138000 [E:onnxruntime:log, cuda\_call.cc:118 CudaCall] CUDNN failure 3: CUDNN\_STATUS\_BAD\_PARAM ; GPU=2 ; hostname=9a41efbf088d ; expr=cudnnFindConvolutionForwardAlgorithmEx( s\_.handle, s\_.x\_tensor, s\_.x\_data, s\_.w\_desc, s\_.w\_data, s\_.conv\_desc, s\_.y\_tensor, s\_.y\_data, 1, &algo\_count, &perf, algo\_search\_workspace.get(), max\_ws\_size);

Please let me know if you have any idea regarding this issue.

Regards,

Ashfaq