ECE 408 REPORT
Project Milestone 1

team_foveon
Zuodong Liang / zuodong2 | Fengze Xie / fengzex2

I. Kernels that collectively consume more than 90% of the program time

Name	Time(%)
[CUDA memcpy HtoD]	38.97
void cudnn::detail::implicit_convolve_sgemm<float, float, int=1024, int=5, int=5, int=3, int=3, int=3, int=1, bool=1, bool=0, bool=1>(int, int, int, float const *, int, float*, cudnn::detail::implicit_convolve_sgemm<float, float, int=1024, int=5, int=5, int=3, int=3, int=3, int=1, bool=1, bool=0, bool=1>*, kernel_conv_params, int, float, float, int, float, float, int, int)	20.81
volta_cgemm_64x32_tn	12.2
void op_generic_tensor_kernel<int=2, float, float, float, int=256, cudnnGenericOp_t=7, cudnnNanPropagation_t=0, cudnnDimOrder_t=0, int=1>(cudnnTensorStruct, float*, cudnnTensorStruct, float const *, cudnnTensorStruct, float const *, float, float, float, float, dimArray, reducedDivisorArray)	7.21
volta_sgemm_128x128_tn	5.76
void fft2d_c2r_32x32<float, bool=0, bool=0, unsigned int=1, bool=0, bool=0>(float*, float2 const *, int, int, int, int, int, int, int, int, int, float, float, cudnn::reduced_divisor, bool, float*, float*, int2, int, int)	5.74
Total Time %	**90.69**

II. CUDA API calls that collectively consume more than 90% of the program time

Name	Time(%)
cudaStreamCreateWithFlags	41.7
cudaMemGetInfo	34.01
cudaFree	21.27
Total Time %	**96.98**

III. Explanation of the difference between kernels and API calls

API, as a programming interface, allow the host to interface with the GPU. APIs are executed on the host. The kernels, on the other hand, are the code that actually running on the GPU.
CUDA APIs are defined by NVIDIA, kernels are codes that written by programmers.

IV. Output of rai running MXNet on the CPU

```
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8236}
8.81user 3.67system 0:05.11elapsed 244%CPU (0avgtext+0avgdata
2469544maxresident)k
0inputs+2824outputs (0major+669134minor)pagefaults 0swaps
```

Run time: 5.11 s

V. Output of rai running MXNet on the GPU

```
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8236}
4.35user 3.41system 0:04.30elapsed 180%CPU (0avgtext+0avgdata
2836344maxresident)k
0inputs+4552outputs (0major+661560minor)pagefaults 0swaps
```

Run time: 4.30 s

Project Milestone 2

```
* Running /usr/bin/time python m2.1.py 10000
Loading fashion-mnist data... done
Loading model... done
New Inference
Op Time: 2.446374
Op Time: 7.487101
Correctness: 0.8397 Model: ece408
15.36user 4.47system 0:11.54elapsed 171%CPU (0avgtext+0avgdata
1619204maxresident)k
0inputs+2824outputs (0major+618017minor)pagefaults 0
swaps
```

```
* Running /usr/bin/time python m2.1.py 1000
Loading fashion-mnist data... done
Loading model... done
New Inference
Op Time: 0.247094
Op Time: 0.747553
Correctness: 0.852 Model: ece408
4.44user 2.77system 0:01.98elapsed 362%CPU (0avgtext+0avgdata
330676maxresident)k
0inputs+2824outputs (0major+110770minor)pagefaults 0swaps
```

```
* Running /usr/bin/time python m2.1.py 100
Loading fashion-mnist data... done
Loading model... done
New Inference
Op Time: 0.032041
Op Time: 0.073482
Correctness: 0.84 Model: ece408
3.10user 2.87system 0:01.05elapsed 564%CPU (0avgtext+0avgdata
203712maxresident)k
0inputs+2824outputs (0major+62399minor)pagefaults 0swaps
```

I. Whole program execution time

When number of images is 10,000, from the execution result, we can see that the whole program execution time is 11.54 s.
When number of images is 1,000, the whole program execution time is 1.98 s.
When number of images is 100, the whole program execution time is 1.05 s.

II. Op Times

When number of images is 10,000, the Op Time for layer 1 is 2.446374 s, the Op Time for layer 2 is 7.487101 s.
When number of images is 1,000, the Op Time for layer 1 is 0.247094 s, the Op Time for layer 2 is 0.747553 s.
When number of images is 100, the Op Time for layer 1 is 0.032041s, the Op Time for layer 2 is 0.073482s.