| **Name:** | *<name>* |
| --- | --- |
| **NetID:** | *<netid>* |
| **Section:** | *<class section>* |

**ECE 408/CS483 Milestone 2 Report**

| 1. Show output of rai running Mini-DNN on the basic GPU convolution implementation for batch size of 1k images. This can either be a screen capture or a text copy of the running output. Please do not show the build output. (The running output should be everything including and after the line "*Loading fashion-mnist data...Done*"). |
| --- |
| *Test batch size: 1000*  *Loading fashion-mnist data...Done*  *Loading model...Done*  *Conv-GPU==*  *Layer Time: 94.2767 ms*  *Op Time: 1.63159 ms*  *Conv-GPU==*  *Layer Time: 73.3287 ms*  *Op Time: 6.26002 ms*  *Test Accuracy: 0.886* |
| 1. For the basic GPU implementation, list Op Times, whole program execution time, and accuracy for batch size of 100, 1k, and 10k images. |
| | Batch Size | Op Time 1 | Op Time 2 | Total Execution Time | Accuracy | | --- | --- | --- | --- | --- | | 100 | *0.174753 ms* | *0.634484 ms* | *0m1.337s* | *0.86* | | 1000 | *1.63159 ms* | *6.26002 ms* | *0m9.761s* | *0.886* | | 10000 | *14.565 ms* | *57.0254 ms* | *1m35.155s* | *0.8714* | |
| 1. List all the kernels that collectively consumed more than 90% of the kernel time and what percentage of the kernel time each kernel did consume (start with the kernel that consumed the most time, then list the next kernel, until you reach 90% or more). |
| *Time(%) Total Time Instances Average Minimum Maximum Name*  *100.0 79285376 2 39642688.0 16169943 63115433 conv\_forward\_kernel*  *no surprise here.* |
| 1. List all the CUDA API calls that collectively consumed more than 90% of the API time and what percentage of the API time each call did consume (start with the API call that consumed the most time, then list the next call, until you reach 90% or more). |
| *CUDA API Statistics (nanoseconds)*  *Time(%) Total Time Calls Average Minimum Maximum Name*  *84.1 1573401240 10 157340124.0 17122 626385810 cudaMemcpy*  *11.5 215410540 8 26926317.5 86825 206835502 cudaMalloc*  *its greater than 90… interesting to see that malloc and memcpy are so painful* |
| 1. Explain the difference between kernels and CUDA API calls. Please give an example in your explanation for both. |
| *CUDA API is what allows us to talk to the gpu to tell it what to do.. whether we want more memory or move memory etc.*  *The Kernel holds the instructions we want the gpu to actually do.* |
| 1. Show a screenshot of the GPU SOL utilization |
| *<nsight output here>*  *Running ight-cu-cli --section '.\*' -o analysis\_file ./m2 \\ Output will appear after run is complete.*  *Test batch size: 10000*  *Loading fashion-mnist data...Done*  *Loading model...Done*  *Conv-GPU==*  *==PROF== Connected to process 556 (/build/m2)*  *==PROF== Profiling "prefn\_marker\_kernel()" - 1: 0%....50%....100% - 73 passes*  *==PROF== Profiling "conv\_forward\_kernel" - 2: 0%....50%....100% - 74 passes*  *==PROF== Profiling "do\_not\_remove\_this\_kernel()" - 3: 0%....50%....100% - 73 passes*  *Layer Time: 11127.1 ms*  *Op Time: 10238.8 ms*  *Conv-GPU==*  *==PROF== Profiling "prefn\_marker\_kernel()" - 4: 0%....50%....100% - 73 passes*  *==PROF== Profiling "conv\_forward\_kernel" - 5: 0%....50%....100% - 74 passes*  *==PROF== Profiling "do\_not\_remove\_this\_kernel()" - 6: 0%....50%....100% - 73 passes*  *Layer Time: 30116.8 ms*  *Op Time: 29449.1 ms*  *Test Accuracy: 0.8714*  *==PROF== Disconnected from process 556*  *==PROF== Report: /build/analysis\_file.ncu-rep* |