|  |  |
| --- | --- |
| **Name:** | *Ruiqi Li* |
| **NetID:** | *Ruiqili4* |
| **Section:** | *AL1* |

**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*"). |
| *<output here>*  Loading model...Done  Conv-GPU==  Layer Time: 100.138 ms  Op Time: 4.65247 ms  Conv-GPU==  Layer Time: 96.732 ms  Op Time: 29.7178 ms  Test Accuracy: 0.886  real 0m9.782s  user 0m9.404s  sys 0m0.304s |
| 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 | *0.48717 ms* | *2.93471 ms* | *0m1.287s* | *0.86* | | 1000 | 4.65247 ms | 29.7178 ms | 0m9.782s | 0.886 | | 10000 | *44.8455 ms* | *276.037 ms* | *1m38.762s* | *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). |
| conv\_forward\_kernel – 100% time |
| 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). |
| cudaMemcpy – 75.3% time  cudaDeviceSynchronize – 15.2% time  cudaMalloc – 8.6% time |
| 1. Explain the difference between kernels and CUDA API calls. Please give an example in your explanation for both. |
| Kernels are device code which can only be invoked by device, while CUDA API calls can be invoked by host code. Therefore, kernels can be designed for many specific tasks, while CUDA API calls play a role of building blocks to call the kernels. For example, cudaMalloc is a basic API call, and conv\_forward\_kernel in this milestone is the kernel for a specific convolutional layer. |
| 1. Show a screenshot of the GPU SOL utilization |
|  |