ECE408 Project Report

Milestone 1

Team Name: tensorjammed
Team Members:

Name	NetID	School
Jingning Tang	jtang10	UIUC
Monil Pathak	mpathak2	UIUC
Xiang Li	xiangl12	UIUC

1. Include a list of all kernels that collectively consume more than 90% of the program time.

Time (%)	Time (ms)	Name
~~36.84%~~	~~37.628~~	~~CUDA memcpy HtoD~~
22.70%	23.185	volta_scudnn_128x32_relu_interior_nn_v1
20.80%	21.248	cudnn::detail::implicit_convolve_sgemm
7.39%	7.546	volta_sgemm_128x128_tn
7.24%	7.391	cudnn::detail::activation_fw_4d_kernel
4.31%	4.404	cudnn::detail::pooling_fw_4d_kernel
99.28% (in total)	101.402 (in total)	

2. Include a list of all CUDA API calls that collectively consume more than 90% of the program time.

Time (%)	Time (ms)	Name
40.47%	37.628	cudaStreamCreateWithFlags
33.77%	23.185	cudaMemGetInfo
22.04%	21.248	cudaFree
96.28% (in total)	82.061 (in total)	

The memcpy operation appears in both GPU activities and API calls. From our understanding, it is obviously an API call provided by CUDA. It can be part of the GPU activities but might not be part of the kernel actions as we commonly define. As described in the following question, kernel is usually defined as the programmer-defined function running on different threads on GPU.

3. Include an explanation of the difference between kernels and API calls

Kernels are programmer-defined functions called by the host running on multiple GPU threads in parallel, whereas API calls are CUDA-defined functions called by the host (CPU).

To be more specific, kernel is the code defined by the programmer and loaded onto N different threads GPU. In the baseline model, they include convolution, non-linear activations (ReLU and tanh), pooling and matrix multiplications (sgemm).

API calls are the functions provided by CUDA and called on the host side for operations including device query, cudaMalloc, cudaMemcpy, and cudaLaunchKernel, etc.

4. Show output of rai running MXNet on the CPU

```
* Running /usr/bin/time python m1.1.py
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8177}
20.09user 4.03system 0:13.44elapsed 179%CPU (0avgtext+0avgdata
5955964maxresident)k
0
inputs+2856outputs (0major+1585730minor)pagefaults 0swaps
```

5. List program run time

user: 20.09s, system: 4.03s, elapsed: 13.44s

6. Show output of rai running MXNet on the GPU

```
* Running /usr/bin/time python m1.2.py
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8177}
4.30user 2.70system 0:04.58elapsed 153%CPU (0avgtext+0avgdata
2849680maxresident)k
0inputs+
4568outputs (0major+708272minor)pagefaults 0swaps
```

7. List program run time

user: 4.30s, system: 0.04s, elapsed: 4.58s

Milestone 2

1. **List whole program execution time**

user: 132.44s, system: 4.41s, elapsed: 2min 6.72s

2. **List op time**

Op time: 21.280921

Op time: 101.222768