ECE 408 Final Project

School: UIUC
Team Name: Shrek
Team: Alex Fan (zhenfan3), Brandon Van (jbvan2), Joshua Song (jssong3)

Milestone 1

Top 10 time consuming kernels:

1. volta_scudnn_128x32_relu_interior_nn_v1
2. 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)
3. volta_sgemm_128x128_tn
4. void cudnn::detail::activation_fw_4d_kernel<float, float, int=128, int=1, int=4, cudnn::detail::tanh_func<float>>(cudnnTensorStruct, float const *, cudnn::detail::activation_fw_4d_kernel<float, float, int=128, int=1, int=4, cudnn::detail::tanh_func<float>>, cudnnTensorStruct*, float, cudnnTensorStruct*, int, cudnnTensorStruct*)
5. void cudnn::detail::pooling_fw_4d_kernel<float, float, cudnn::detail::maxpooling_func<float, cudnnNanPropagation_t=0>, int=0, bool=0>(cudnnTensorStruct, float const *, cudnn::detail::pooling_fw_4d_kernel<float, float, cudnn::detail::maxpooling_func<float, cudnnNanPropagation_t=0>, int=0, bool=0>, cudnnTensorStruct*, cudnnPoolingStruct, float, cudnnPoolingStruct, int, cudnn::reduced_divisor, float)
6. void mshadow::cuda::MapPlanLargeKernel<mshadow::sv::saveto, int=8, int=1024, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>, mshadow::expr::Plan<mshadow::expr::ScalarExp<float>, float>>(mshadow::gpu, unsigned int, mshadow::Shape<int=2>, int=2, int)
7. void mshadow::cuda::SoftmaxKernel<int=8, float, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>>(mshadow::gpu, int=2, unsigned int)
8. void mshadow::cuda::MapPlanKernel<mshadow::sv::saveto, int=8, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>, mshadow::expr::Plan<mshadow::expr::ScalarExp<float>, float>>(mshadow::gpu, unsigned int, mshadow::Shape<int=2>, int=2)
9. Volta_sgemm_32x32_sliced1x4_tn
10. void mshadow::cuda::MapPlanKernel<mshadow::sv::plusto, int=8, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>, mshadow::expr::Plan<mshadow::expr::Broadcast1DExp<mshadow::Tensor<mshadow::gpu, int=1, float>, float, int=2, int=1>, float>>(mshadow::gpu, unsigned int, mshadow::Shape<int=2>, int=2)

Top 10 time-consuming API calls:

1. cudaMemGetInfo
2. cudaFree
3. cudaFuncSetAttribute
4. cudaMemcpy2DAsync
5. cudaStreamSynchronize
6. cudaMalloc
7. cudaGetDeviceProperties
8. cuDeviceGetAttribute
9. cudaEventCreate
10. cudaEventCreateWithFlags

Difference between kernels and API calls:

The API calls are the CUDA calls that provide an extension to the C language. They facilitate configuration of the parallel computing device - actions include allocation of memory and transfer of data to and from. A

kernel function is code intended to run on the parallel device. Upon calling, it launches multiple threads to process different parts of the data in parallel.

Running MXNet on the CPU:

Output:

```
✱ Running /usr/bin/time python m1.1.py
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8177}
```

Run time:

```
20.95user 6.05system 0:14.19elapsed 190%CPU (0avgtext+0avgdata 5954620maxresident)k
0inputs+2856outputs (0major+1580062minor)pagefaults 0swaps
```

Running MXNet on the GPU:

Output:

```
✱ Running /usr/bin/time python m1.2.py
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8177}
```

Run time:

```
4.24user 2.57system 0:04.62elapsed 147%CPU (0avgtext+0avgdata 2846512maxresident)k
0inputs+4568outputs (0major+706410minor)pagefaults 0swaps
```

Milestone 2

Output:

```
✱ Running /usr/bin/time python m2.1.py 10000
Loading fashion-mnist data... done
Loading model... done
New Inference
Op Time: 26.134832
Op Time: 154.410258
Correctness: 0.8171 Model: ece408
191.63user 6.42system 3:05.15elapsed 106%CPU (0avgtext+0avgdata 5953104maxresident)k
0inputs+2856outputs (0major+2264713minor)pagefaults 0swaps
```

Op Times:

```
Op Time: 26.134832
Op Time: 154.410258
```

Execution Time:

```
191.63user 6.42system 3:05.15elapsed 106%CPU (0avgtext+0avgdata 5953104maxresident)k
0inputs+2856outputs (0major+2264713minor)pagefaults 0swaps
```

Milestone 3

Output:

Number of Images: 100

```
Loading fashion-mnist data... done
Loading model... done
New Inference
Op Time: 0.000651
Op Time: 0.001620
Correctness: 0.85 Model: ece408
```

Number of Images: 1000

```
Loading fashion-mnist data... done
Loading model... done
New Inference
Op Time: 0.006198
Op Time: 0.016068
Correctness: 0.827 Model: ece408
```

Number of Images: 10000

```
Loading fashion-mnist data... done
Loading model... done
New Inference
Op Time: 0.064318
Op Time: 0.147558
Correctness: 0.8171 Model: ece408
```

Performance:

According to `forward1_analysis.nvprof` and `forward2_analysis.nvprof`,

- The performance of the kernel is most likely limited by the latency of arithmetic or memory operations;
- There are low global memory load efficiency and low warp execution issues;
- Global memory load and store are not properly aligned, which leads to inefficient use of memory bandwidth;
- Divergence branches lower warp execution efficiency, which leads to inefficient use of the GPU's compute resources.