ECE 408
Final Project Report

Group Name

ConvolutionallyUnsampledDiscreteAlgorithm

Members

Zain Paya
Aditya Bhargava
Arvind Kamal

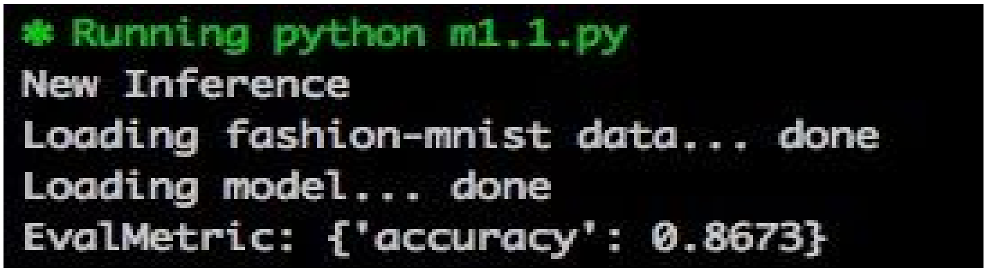
Contents

MILESTONE 1: Getting Started

MILESTONE 2: New CPU Layer in MXNet

MILESTONE 1: Getting Started

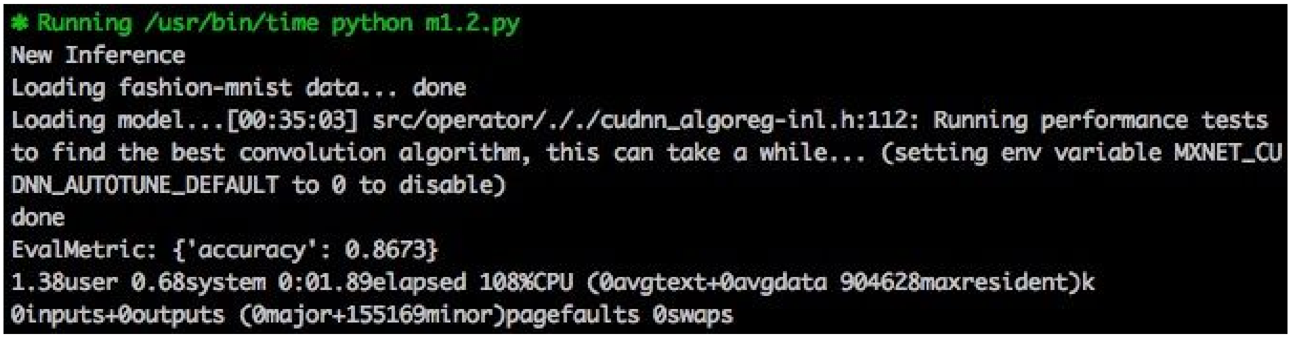
1.1 RUN THE BASELINE FORWARD PASS



```
* Running python m1.1.py
New Inference
Loading fashion-mnist data... done
Loading model... done
EvalMetric: {'accuracy': 0.8673}
```

Fig 1.1: Output for CPU run on m1.1.py

1.2 RUN THE BASELINE GPU IMPLEMENTATION

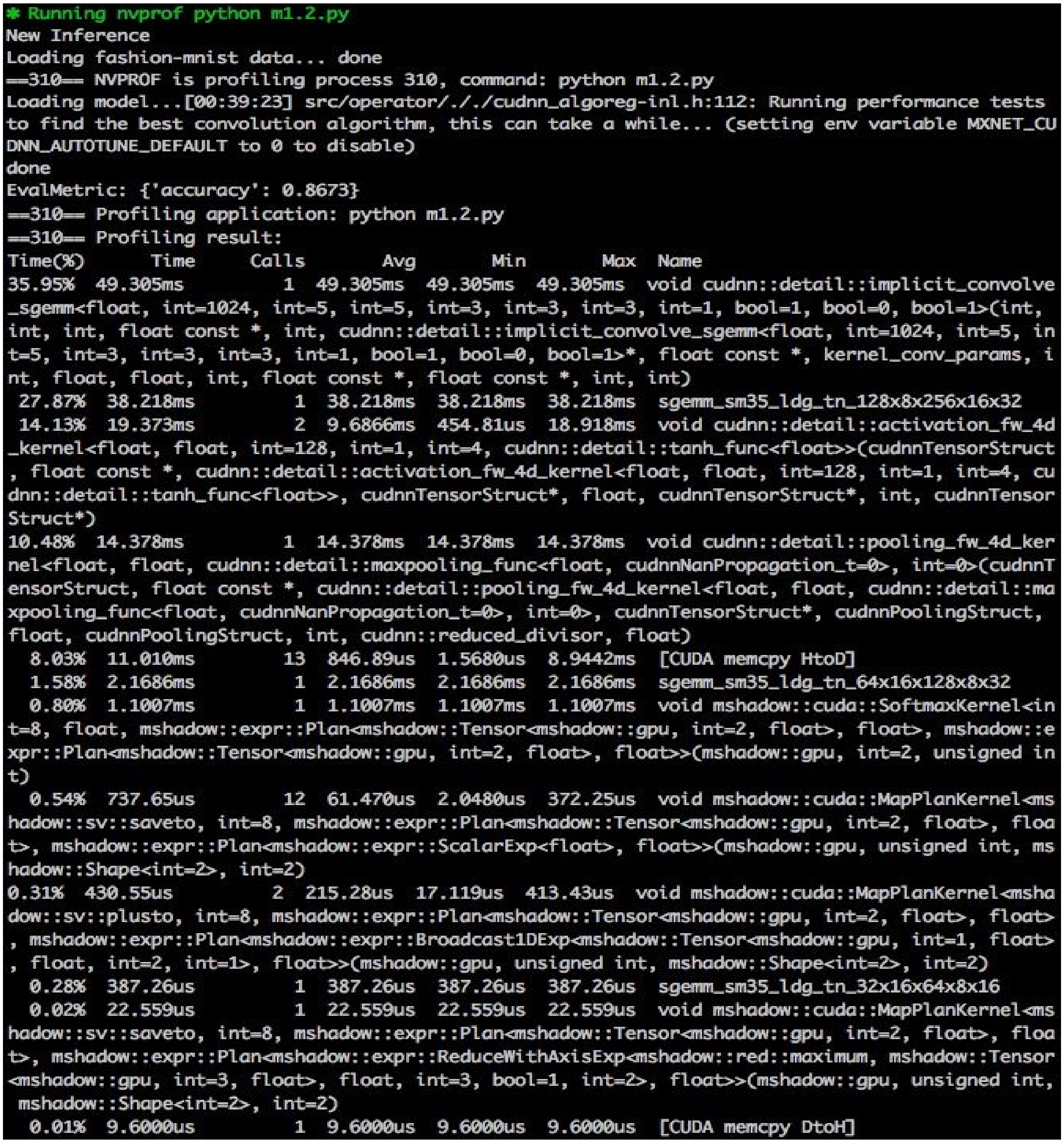


```
* Running /usr/bin/time python m1.2.py
New Inference
Loading fashion-mnist data... done
Loading model...[00:35:03] src/operator/././cudnn_algoreg-inl.h:112: Running performance tests
to find the best convolution algorithm, this can take a while... (setting env variable MXNET_CU
DNN_AUTOTUNE_DEFAULT to 0 to disable)
done
EvalMetric: {'accuracy': 0.8673}
1.38user 0.68system 0:01.89elapsed 108%CPU (0avgtext+0avgdata 904628maxresident)k
0inputs+0outputs (0major+155169minor)pagefaults 0swaps
```

Fig 1.2: Output for GPU run on m1.2.py with time outputs (real, user, sys)

1.3 GENERATE A NVPROF PROFILE

- Top 2 time consuming kernels: ***implicit_convolve_sgemm, pooling_fw_4d_kernel***

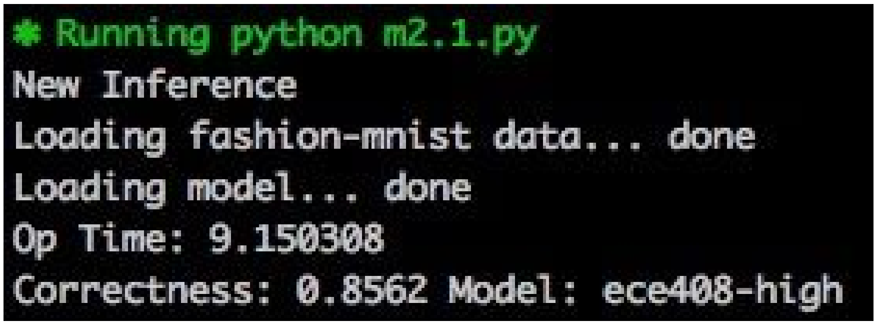


```
* Running nvprof python m1.2.py
New Inference
Loading fashion-mnist data... done
==310== NVPROF is profiling process 310, command: python m1.2.py
Loading model...[00:39:23] src/operator/././cudnn_algoreg-inl.h:112: Running performance tests
to find the best convolution algorithm, this can take a while... (setting env variable MXNET_CU
DNN_AUTOTUNE_DEFAULT to 0 to disable)
done
EvalMetric: {'accuracy': 0.8673}
==310== Profiling application: python m1.2.py
==310== Profiling result:
Time(%)      Time     Calls       Avg       Min       Max  Name
35.95%  49.305ms         1  49.305ms  49.305ms  49.305ms  void cudnn::detail::implicit_convolve
_sgemm<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, cudnn::detail::implicit_convolve_sgemm<float, int=1024, int=5, in
t=5, int=3, int=3, int=3, int=1, bool=1, bool=0, bool=1>*, float const *, kernel_conv_params, i
nt, float, float, int, float const *, float const *, int, int)
 27.87%  38.218ms         1  38.218ms  38.218ms  38.218ms  sgemm_sm35_ldg_tn_128x8x256x16x32
 14.13%  19.373ms         2  9.6866ms  454.81us  18.918ms  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, cu
dnn::detail::tanh_func<float>>, cudnnTensorStruct*, float, cudnnTensorStruct*, int, cudnnTensor
Struct*)
10.48%  14.378ms         1  14.378ms  14.378ms  14.378ms  void cudnn::detail::pooling_fw_4d_ker
nel<float, float, cudnn::detail::maxpooling_func<float, cudnnNanPropagation_t=0>, int=0>(cudnnT
ensorStruct, float const *, cudnn::detail::pooling_fw_4d_kernel<float, float, cudnn::detail::ma
xpooling_func<float, cudnnNanPropagation_t=0>, int=0>, cudnnTensorStruct*, cudnnPoolingStruct,
float, cudnnPoolingStruct, int, cudnn::reduced_divisor, float)
  8.03%  11.010ms        13  846.89us  1.5680us  8.9442ms  [CUDA memcpy HtoD]
  1.58%  2.1686ms         1  2.1686ms  2.1686ms  2.1686ms  sgemm_sm35_ldg_tn_64x16x128x8x32
  0.80%  1.1007ms         1  1.1007ms  1.1007ms  1.1007ms  void mshadow::cuda::SoftmaxKernel<in
t=8, float, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>, mshadow::e
xpr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>>(mshadow::gpu, int=2, unsigned in
t)
  0.54%  737.65us        12  61.470us  2.0480us  372.25us  void mshadow::cuda::MapPlanKernel<ms
hadow::sv::saveto, int=8, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, floa
t>, mshadow::expr::Plan<mshadow::expr::ScalarExp<float>, float>>(mshadow::gpu, unsigned int, ms
hadow::Shape<int=2>, int=2)
0.31%  430.55us         2  215.28us  17.119us  413.43us  void mshadow::cuda::MapPlanKernel<msha
dow::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)
  0.28%  387.26us         1  387.26us  387.26us  387.26us  sgemm_sm35_ldg_tn_32x16x64x8x16
  0.02%  22.559us         1  22.559us  22.559us  22.559us  void mshadow::cuda::MapPlanKernel<ms
hadow::sv::saveto, int=8, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, floa
t>, mshadow::expr::Plan<mshadow::expr::ReduceWithAxisExp<mshadow::red::maximum, mshadow::Tensor
<mshadow::gpu, int=3, float>, float, int=3, bool=1, int=2>, float>>(mshadow::gpu, unsigned int,
 mshadow::Shape<int=2>, int=2)
  0.01%  9.6000us         1  9.6000us  9.6000us  9.6000us  [CUDA memcpy DtoH]
```

Fig 1.3: Output table for running times of all kernels via nvprof

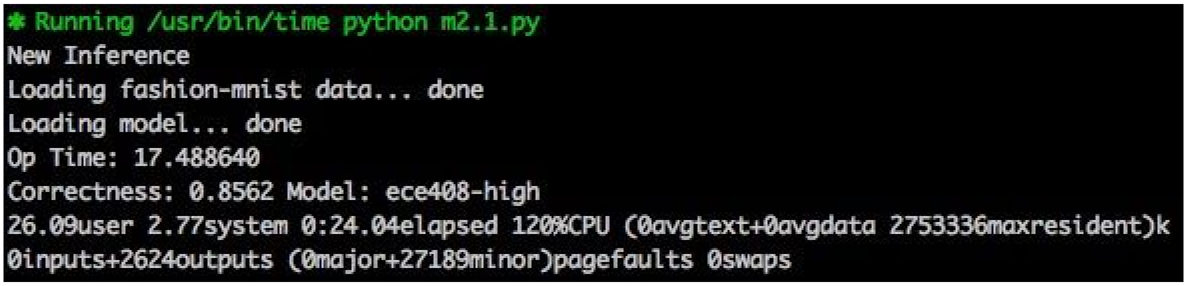
MILESTONE 2: New CPU Layer in MXNet

2.1 ADD CPU FORWARD IMPLEMENTATION



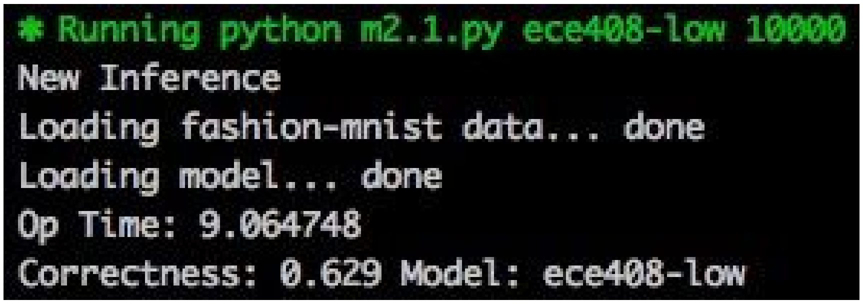
```
* Running python m2.1.py
New Inference
Loading fashion-mnist data... done
Loading model... done
Op Time: 9.150308
Correctness: 0.8562 Model: ece408-high
```

Fig 2.1: Output for CPU Forward Convolution on m2.1.py (default)



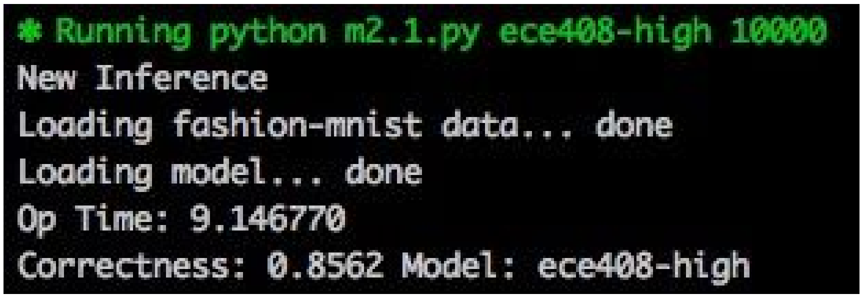
```
* Running /usr/bin/time python m2.1.py
New Inference
Loading fashion-mnist data... done
Loading model... done
Op Time: 17.488640
Correctness: 0.8562 Model: ece408-high
26.09user 2.77system 0:24.04elapsed 120%CPU (0avgtext+0avgdata 2753336maxresident)k
0inputs+2624outputs (0major+27189minor)pagefaults 0swaps
```

Fig 2.2: Output for CPU Forward Convolution on m2.1.py with time outputs (elapsed, user, system)



```
* Running python m2.1.py ece408-low 10000
New Inference
Loading fashion-mnist data... done
Loading model... done
Op Time: 9.064748
Correctness: 0.629 Model: ece408-low
```

Fig 2.3: Comparative output for ece-408-low of size 10000



```
✱ Running python m2.1.py ece408-high 10000
New Inference
Loading fashion-mnist data... done
Loading model... done
Op Time: 9.146770
Correctness: 0.8562 Model: ece408-high
```

Fig 2.4: Comparative output for ece-408-high of size 10000

TEAM MEMBERS' ROLES

- **Zain Paya**
  - Implemented Forward Convolution with an extra layer for Milestone 2

- **Aditya Bhargava**
  - Implemented Forward Convolution with an extra layer for Milestone 2

- **Arvind Kamal**
  - Implemented Forward Convolution with an extra layer for Milestone 2