

HPML HW2 Results
Shashvat Shah (sss9772)

C1.

Code files attached. Code written for Resnet 18 with images of size 32*32 with 3 input channels.

C2.

Batch file –

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=8
#SBATCH --time=3:00:00
#SBATCH --mem=16GB
#SBATCH --job-name=hw2
#SBATCH --output=/scratch/sss9772/hw2/hw2_C2.out

SRCDIR=$SCRATCH/hw2

module purge

singularity exec \
    --overlay /scratch/sss9772/hw2/torch.ext3:ro \
    /scratch/work/public/singularity/cuda11.6.124-cudnn8.4.0.27-devel-
ubuntu20.04.4.sif\
    /bin/bash -c "source /ext3/env.sh; python $SRCDIR/C2.py --device='cpu' --
optim='SGD' -workers=1 --datapath=$SRCDIR'/data'"
```

Output –

```
already downloaded and verified
already downloaded and verified
ing epoch: 0 started
h: 0 : Train Loss: 1.932 : Accuracy : 30.242

ing epoch: 1 started
h: 1 : Train Loss: 1.466 : Accuracy : 45.990

ing epoch: 2 started
h: 2 : Train Loss: 1.143 : Accuracy : 58.816

ing epoch: 3 started
h: 3 : Train Loss: 0.906 : Accuracy : 67.768
```

```
ing epoch: 4 started  
h: 4 : Train Loss: 0.745 : Accuracy : 73.774  
  
ing Done  
  
Loader Time for Each Epoch: [0.7538565173745155, 0.8563365563750267,  
63870292901993, 0.7313348799943924, 0.7004451006650925]  
  
ing Time for Each Epoch: [293.63443315029144, 301.7336898781359,  
6671611815691, 307.99317280203104, 306.48903711140156]  
  
Time for Each Epoch: [294.4454376325011, 302.6880677342415, 314.50189058855176,  
82165357097983, 307.28359688073397]  
Data Loading time for 1 worker = 3.7783600836992264
```

C3.

I)

Batch file –

```
#!/bin/bash  
#SBATCH --nodes=1  
#SBATCH --ntasks-per-node=1  
#SBATCH --cpus-per-task=8  
#SBATCH --time=3:00:00  
#SBATCH --mem=16GB  
#SBATCH --job-name=hw2  
#SBATCH --output=/scratch/sss9772/hw2/hw2_C3.out  
  
SRCDIR=$SCRATCH/hw2  
  
module purge  
  
singularity exec \  
    --overlay /scratch/sss9772/hw2/torch.ext3:ro \  
    /scratch/work/public/singularity/cuda11.6.124-cudnn8.4.0.27-devel-  
ubuntu20.04.4.sif\  
    /bin/bash -c "source /ext3/env.sh; python $SRCDIR/C3.py --device='cpu' --  
optim='SGD' --datapath=$SRCDIR'/data'" >> hw2_C3_${SLURM_JOB_ID}.out
```

Output –

Files already downloaded and verified

starting train loop
Training with 0 worker(s) started

training epoch: 0 started
Epoch: 0 : Train Loss: 1.979 : Accuracy : 29.908

training epoch: 1 started
Epoch: 1 : Train Loss: 1.483 : Accuracy : 45.068

training epoch: 2 started
Epoch: 2 : Train Loss: 1.258 : Accuracy : 54.292

training epoch: 3 started
Epoch: 3 : Train Loss: 1.050 : Accuracy : 62.766

training epoch: 4 started
Epoch: 4 : Train Loss: 0.896 : Accuracy : 68.236

Training with 0 worker(s) ended

Output with 0 worker ---

Total Dataloading time with 0 worker(s) 70.80041855573654

Total Train time with 0 worker(s) for 5 epochs is [261.7845182046294,

307.7180707268417, 306.9846892505884, 307.5020462796092, 309.16493060439825]

Total run time with 0 worker(s) for 5 epochs is [276.0325861722231, 321.9138206169009,

321.1848204024136, 321.68340768665075, 323.34078665450215]

Training with 4 worker(s) started

training epoch: 0 started
Epoch: 0 : Train Loss: 2.041 : Accuracy : 27.372

training epoch: 1 started
Epoch: 1 : Train Loss: 1.523 : Accuracy : 43.670

training epoch: 2 started
Epoch: 2 : Train Loss: 1.260 : Accuracy : 54.190

training epoch: 3 started
Epoch: 3 : Train Loss: 1.048 : Accuracy : 62.498

training epoch: 4 started
Epoch: 4 : Train Loss: 0.915 : Accuracy : 67.682

Training with 4 worker(s) ended

Output with 4 worker ---

Total Dataloading time with 4 worker(s) 4.548905618488789

Total Train time with 4 worker(s) for 5 epochs is [317.4099959433079, 317.63107965886593, 317.868261564523, 317.3138878494501, 317.35995181649923]
Total run time with 4 worker(s) for 5 epochs is [318.3739269301295, 318.6212253011763, 318.87277107313275, 318.33198035880923, 318.36796951293945]
Training with 8 worker(s) started

training epoch: 0 started
Epoch: 0 : Train Loss: 1.937 : Accuracy : 30.900

training epoch: 1 started
Epoch: 1 : Train Loss: 1.428 : Accuracy : 47.592

training epoch: 2 started
Epoch: 2 : Train Loss: 1.183 : Accuracy : 57.948

training epoch: 3 started
Epoch: 3 : Train Loss: 0.982 : Accuracy : 65.222

training epoch: 4 started
Epoch: 4 : Train Loss: 0.819 : Accuracy : 71.374

Training with 8 worker(s) ended

Output with 8 worker ---
Total Dataloading time with 8 worker(s) 6.348907221108675
Total Train time with 8 worker(s) for 5 epochs is [316.41548673808575, 312.4138615503907, 316.61687901988626, 316.4066286124289, 315.57025334611535]
Total run time with 8 worker(s) for 5 epochs is [317.8048660233617, 313.7666309066117, 317.962209995836, 317.79715433716774, 316.95580115169287]
Optimized number of workers is 8

ii)

Code written to vary the workers from 0, 4, 8, and incrementally add till IO performance time doesn't decrease any further.

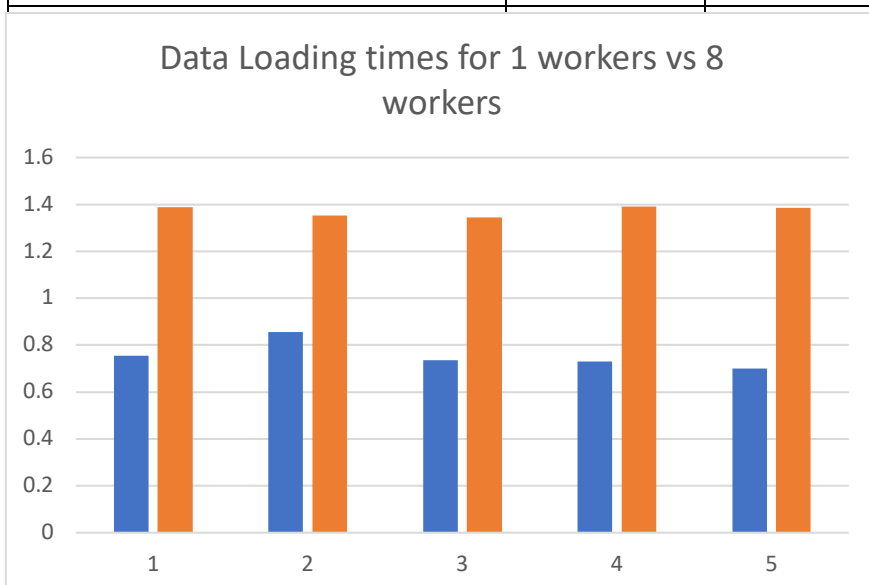
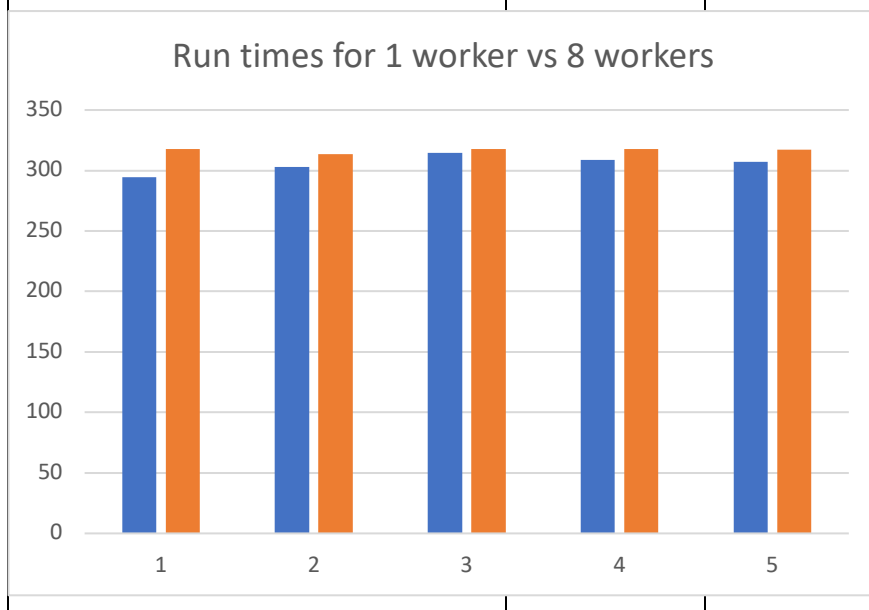
We are using a 8 core CPU, so ideally we should get peak performance at 8 workers, i.e. 1 worker per core.

Hence we get optimized performance for 8 workers for a 8 core system

C4.

Train Times (s) (1 vs 8 workers)		
293.6344332	316.4154867	
301.7336899	312.4138616	
313.6671612	316.616879	
307.9931728	316.4066286	

306.4890371	315.5702533	
Run Times(s) (1 vs 8 workers)		
294.4454376	317.804866	
302.6880677	313.7666309	
314.5018906	317.96221	
308.8216536	317.7971543	
307.2835969	316.9558012	
Dataloading time(s) (1 vs 8 workers)		
0.753856517	1.389379285	
0.856336556	1.352769356	
0.736387029	1.345330976	
0.73133488	1.390525725	
0.700445101	1.385547806	



- Comparing the time performance of 1 vs 8 workers, we notice that 1 worker performs better than 8.
- Although in C3, we say that for workers, 0, 4, 8, 12 -> 8 performs the best
- The data Loading time when worker is 1 is less than data loading time when worker is 8. Hence there is an overhead due to which Data cannot be read at a rate at which it is processed.

C5.

- Batch file for GPU execution

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --gres=gpu:1
#SBATCH --cpus-per-task=8
#SBATCH --time=3:00:00
#SBATCH --mem=16GB
#SBATCH --job-name=hw2
#SBATCH --output=hw2_C5.out

SRCDIR=$SCRATCH/hw2

module purge
echo "Hostname: $(hostname)"
echo "RAM: $(free -h | grep Mem: | awk '{print $4}')"
echo "GPU: $(nvidia-smi -q | grep 'Product Name')"

singularity exec --nv \
    --overlay /scratch/sss9772/hw2/torch.ext3:ro \
    /scratch/work/public/singularity/cuda11.6.124-cudnn8.4.0.27-devel-
ubuntu20.04.4.sif\
    /bin/bash -c "source /ext3/env.sh; python $SRCDIR/C5.py --device='gpu' --
optim='SGD' --workers=8 --datapath=$SRCDIR'/data'"
```

Output For GPU job –

```
Hostname: gv001.hpc.nyu.edu
RAM: 84Gi
GPU:   Product Name           : Tesla V100-PCIE-32GB
Using gpu device as per torch
Files already downloaded and verified
Files already downloaded and verified
training epoch: 0 started
Epoch: 0 : Train Loss: 1.953 : Accuracy : 29.754

training epoch: 1 started
Epoch: 1 : Train Loss: 1.442 : Accuracy : 46.978
```

training epoch: 2 started
Epoch: 2 : Train Loss: 1.192 : Accuracy : 56.840

training epoch: 3 started
Epoch: 3 : Train Loss: 0.994 : Accuracy : 64.522

training epoch: 4 started
Epoch: 4 : Train Loss: 0.852 : Accuracy : 69.980

Training Done

DataLoader Time for Each Epoch: [1.0764153935015202, 1.0881811417639256, 1.113740924745798, 1.1328192725777626, 1.1524167954921722]

Training Time for Each Epoch: [11.44190339371562, 3.7459254674613476, 4.095083698630333, 3.763152163475752, 3.755140606313944]

Run Time for Each Epoch: [28.78697108849883, 21.1373727992177, 21.14442566409707, 21.186986841261387, 21.211393550038338]

Total Data Loading time for 8 worker = 5.563573528081179

Comparison –

Train Times (s) (CPU vs GPU)

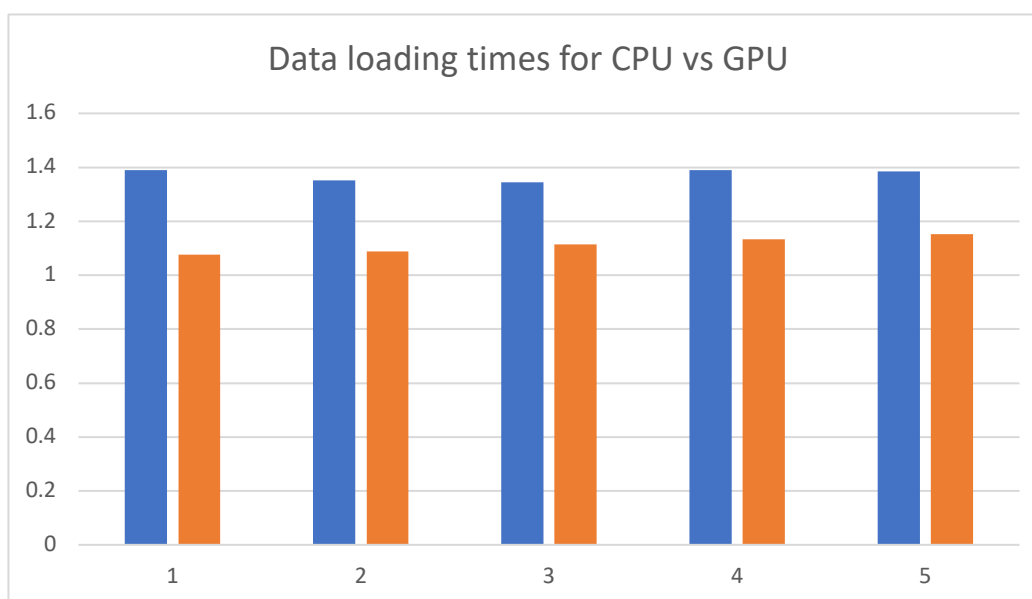
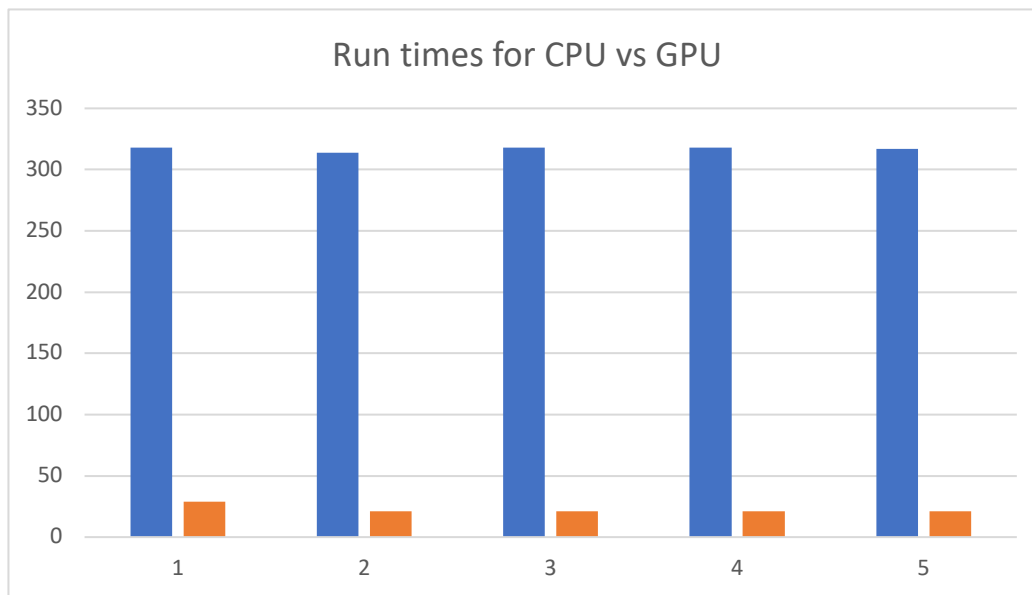
316.4154867	11.44190339
312.4138616	3.745925467
316.616879	4.095083699
316.4066286	3.763152163
315.5702533	3.755140606

Run Times(s) (CPU vs GPU)

317.804866	28.78697109
313.7666309	21.1373728
317.96221	21.14442566
317.7971543	21.18698684
316.9558012	21.21139355

Dataloading time(s) (CPU vs GPU)

1.389379285	1.076415394
1.352769356	1.088181142
1.345330976	1.113740925
1.390525725	1.132819273
1.385547806	1.152416795



From above results we can see that the average run time in-
CPU – 316.857s
GPU – 22.69 s

C6.

Batch -File

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --gres=gpu:1
```



```

#SBATCH --cpus-per-task=8
#SBATCH --time=3:00:00
#SBATCH --mem=16GB
#SBATCH --job-name=hw2
#SBATCH --output=hw2_C6.out

SRCDIR=$SCRATCH/hw2

module purge
echo "Hostname: $(hostname)"
echo "RAM: $(free -h | grep Mem: | awk '{print $4}')"
echo "GPU: $(nvidia-smi -q | grep 'Product Name')"

singularity exec --nv \
    --overlay /scratch/sss9772/hw2/torch.ext3:ro \
    /scratch/work/public/singularity/cuda11.6.124-cudnn8.4.0.27-devel-
ubuntu20.04.4.sif\
    /bin/bash -c "source /ext3/env.sh;
    python $SRCDIR/C6.py --device='gpu' --optim='SGD' --workers=8 --
datapath=$SRCDIR'/data';
    python $SRCDIR/C6.py --device='gpu' --optim='sgd_nest' --workers=8 --
datapath=$SRCDIR'/data';
    python $SRCDIR/C6.py --device='gpu' --optim='adagrad' --workers=8 --
datapath=$SRCDIR'/data';
    python $SRCDIR/C6.py --device='gpu' --optim='adadelta' --workers=8 --
datapath=$SRCDIR'/data';
    python $SRCDIR/C6.py --device='gpu' --optim='adam' --workers=8 --
datapath=$SRCDIR'/data'"

```

Output –

```

Hostname: gv017.hpc.nyu.edu
RAM: 176Gi
GPU:   Product Name           : Tesla V100-SXM2-16GB
Optimizer used SGD
Using gpu device as per torch
Files already downloaded and verified
Files already downloaded and verified
training epoch: 0 started
Epoch: 0 : Train Loss: 2.208 : Accuracy : 24.842

training epoch: 1 started
Epoch: 1 : Train Loss: 1.559 : Accuracy : 41.804

training epoch: 2 started
Epoch: 2 : Train Loss: 1.278 : Accuracy : 53.620

training epoch: 3 started

```

Epoch: 3 : Train Loss: 1.032 : Accuracy : 63.156

training epoch: 4 started

Epoch: 4 : Train Loss: 0.893 : Accuracy : 68.338

Training Done

DataLoader Time for Each Epoch: [1.2212976478040218, 1.2940411660820246, 1.3187661468982697, 1.3827914670109749, 1.4153028931468725]

Training Time for Each Epoch: [12.181377079337835, 4.087892062962055, 4.067207042127848, 4.08948677405715, 4.112500282004476]

Run Time for Each Epoch: [27.9237460475415, 19.901506315916777, 19.94115119613707, 20.01136606372893, 20.06460434384644]

Total Data Loading time for 8 worker = 6.6321993209421635

Optimizer used sgd_nest

Using gpu device as per torch

Files already downloaded and verified

Files already downloaded and verified

training epoch: 0 started

Epoch: 0 : Train Loss: 1.841 : Accuracy : 32.778

training epoch: 1 started

Epoch: 1 : Train Loss: 1.332 : Accuracy : 51.166

training epoch: 2 started

Epoch: 2 : Train Loss: 1.060 : Accuracy : 62.014

training epoch: 3 started

Epoch: 3 : Train Loss: 0.904 : Accuracy : 67.884

training epoch: 4 started

Epoch: 4 : Train Loss: 0.772 : Accuracy : 72.748

Training Done

DataLoader Time for Each Epoch: [1.454762365669012, 1.2878957092761993, 1.3603150080889463, 1.3828104808926582, 1.3613102044910192]

Training Time for Each Epoch: [5.586087474599481, 4.267902310937643, 4.242701470851898, 4.255950532853603, 4.250671358779073]

Run Time for Each Epoch: [21.507476832717657, 20.039288077503443, 20.11332199536264, 20.13096071407199, 20.1192623116076]

Total Data Loading time for 8 worker = 6.847093768417835

Optimizer used adagrad

Using gpu device as per torch

Files already downloaded and verified
Files already downloaded and verified
training epoch: 0 started
Epoch: 0 : Train Loss: 1.679 : Accuracy : 38.094

training epoch: 1 started
Epoch: 1 : Train Loss: 1.161 : Accuracy : 58.024

training epoch: 2 started
Epoch: 2 : Train Loss: 0.914 : Accuracy : 67.638

training epoch: 3 started
Epoch: 3 : Train Loss: 0.751 : Accuracy : 73.724

training epoch: 4 started
Epoch: 4 : Train Loss: 0.633 : Accuracy : 77.786

Training Done

DataLoader Time for Each Epoch: [1.3287613410502672, 1.391565766185522, 1.3055110536515713, 1.371166942641139, 1.400342082604766]

Training Time for Each Epoch: [5.873166296631098, 4.540004760026932, 4.525269310921431, 4.516920607537031, 4.5189011842012405]

Run Time for Each Epoch: [21.35176182538271, 20.126656241714954, 20.02910310961306, 20.104795563966036, 20.145651964470744]
Total Data Loading time for 8 worker = 6.7973471861332655

Optimizer used adadelat
Using gpu device as per torch
Files already downloaded and verified
Files already downloaded and verified
training epoch: 0 started
Epoch: 0 : Train Loss: 1.382 : Accuracy : 49.222

training epoch: 1 started
Epoch: 1 : Train Loss: 0.875 : Accuracy : 68.968

training epoch: 2 started
Epoch: 2 : Train Loss: 0.684 : Accuracy : 76.114

training epoch: 3 started
Epoch: 3 : Train Loss: 0.577 : Accuracy : 79.784

training epoch: 4 started
Epoch: 4 : Train Loss: 0.505 : Accuracy : 82.482

Training Done

DataLoader Time for Each Epoch: [1.3130199667066336, 1.327890556305647, 1.3137208819389343, 1.3972876891493797, 1.3510730229318142]

Training Time for Each Epoch: [6.5973216239362955, 5.279841348528862, 5.293136408552527, 5.2822334934026, 5.2792941480875015]

Run Time for Each Epoch: [21.93282081373036, 20.672384329140186, 20.658783147111535, 20.74530495889485, 20.71976149082184]

Total Data Loading time for 8 worker = 6.702992117032409

Optimizer used adam

Using gpu device as per torch

Files already downloaded and verified

Files already downloaded and verified

training epoch: 0 started

Epoch: 0 : Train Loss: 2.257 : Accuracy : 23.926

training epoch: 1 started

Epoch: 1 : Train Loss: 1.845 : Accuracy : 29.366

training epoch: 2 started

Epoch: 2 : Train Loss: 1.820 : Accuracy : 30.208

training epoch: 3 started

Epoch: 3 : Train Loss: 1.816 : Accuracy : 30.144

training epoch: 4 started

Epoch: 4 : Train Loss: 1.820 : Accuracy : 30.340

Training Done

DataLoader Time for Each Epoch: [1.4555851928889751, 1.4000249467790127, 1.3176694847643375, 1.368088275194168, 1.3942363634705544]

Training Time for Each Epoch: [6.601624088361859, 5.283381650224328, 5.281179498881102, 5.232320327311754, 5.302466429769993]

Run Time for Each Epoch: [21.8229441922158, 20.465376436710358, 20.388609929010272, 20.443466562777758, 20.495607128366828]

Total Data Loading time for 8 worker = 6.935604263097048

C7.

Code run with Resnet 18 without batch norm

Batch File – Same as C2

Output –

Hostname: gv017.hpc.nyu.edu

RAM: 176Gi

GPU: Product Name : Tesla V100-SXM2-16GB

Optimizer used SGD

Using gpu device as per torch

Files already downloaded and verified

Files already downloaded and verified

training epoch: 0 started

Epoch: 0 : Train Loss: 1.916 : Accuracy : 27.724

training epoch: 1 started

Epoch: 1 : Train Loss: 1.516 : Accuracy : 44.402

training epoch: 2 started

Epoch: 2 : Train Loss: 1.271 : Accuracy : 54.604

training epoch: 3 started

Epoch: 3 : Train Loss: 1.096 : Accuracy : 61.440

training epoch: 4 started

Epoch: 4 : Train Loss: 0.944 : Accuracy : 67.326

Training Done

DataLoader Time for Each Epoch: [1.3426127433776855, 1.3257606457918882, 1.3194668274372816, 1.3763357512652874, 1.3544038739055395]

Training Time for Each Epoch: [8.961337175220251, 2.5326761342585087, 2.6423807088285685, 2.638436581939459, 2.608920520171523]

Run Time for Each Epoch: [24.5952619779855, 18.230498924851418, 18.233347184956074, 18.28760041296482, 18.263216579332948]

Total Data Loading time for 8 worker = 6.718579841777682

Q1.

18 Convolutional layers are there in Resnet18

Q2.

Last layer dimension is (512*1)

Q3.

```
import torch
from torchsummary import summary
from torchvision.models import resnet18
```

```

model = resnet18(pretrained=True)
if torch.cuda.is_available():
    model.cuda()
summary(model, (3, 32, 32))

```

Layer (type)	Output Shape	Param #

Conv2d-1	[-1, 64, 16, 16]	9,408
BatchNorm2d-2	[-1, 64, 16, 16]	128
ReLU-3	[-1, 64, 16, 16]	0
MaxPool2d-4	[-1, 64, 8, 8]	0
Conv2d-5	[-1, 64, 8, 8]	36,864
BatchNorm2d-6	[-1, 64, 8, 8]	128
ReLU-7	[-1, 64, 8, 8]	0
Conv2d-8	[-1, 64, 8, 8]	36,864
BatchNorm2d-9	[-1, 64, 8, 8]	128
ReLU-10	[-1, 64, 8, 8]	0
BasicBlock-11	[-1, 64, 8, 8]	0
Conv2d-12	[-1, 64, 8, 8]	36,864
BatchNorm2d-13	[-1, 64, 8, 8]	128
ReLU-14	[-1, 64, 8, 8]	0
Conv2d-15	[-1, 64, 8, 8]	36,864
BatchNorm2d-16	[-1, 64, 8, 8]	128
ReLU-17	[-1, 64, 8, 8]	0
BasicBlock-18	[-1, 64, 8, 8]	0
Conv2d-19	[-1, 128, 4, 4]	73,728
BatchNorm2d-20	[-1, 128, 4, 4]	256
ReLU-21	[-1, 128, 4, 4]	0
Conv2d-22	[-1, 128, 4, 4]	147,456
BatchNorm2d-23	[-1, 128, 4, 4]	256
Conv2d-24	[-1, 128, 4, 4]	8,192
BatchNorm2d-25	[-1, 128, 4, 4]	256
ReLU-26	[-1, 128, 4, 4]	0
BasicBlock-27	[-1, 128, 4, 4]	0
Conv2d-28	[-1, 128, 4, 4]	147,456
BatchNorm2d-29	[-1, 128, 4, 4]	256
ReLU-30	[-1, 128, 4, 4]	0
Conv2d-31	[-1, 128, 4, 4]	147,456
BatchNorm2d-32	[-1, 128, 4, 4]	256
ReLU-33	[-1, 128, 4, 4]	0
BasicBlock-34	[-1, 128, 4, 4]	0
Conv2d-35	[-1, 256, 2, 2]	294,912
BatchNorm2d-36	[-1, 256, 2, 2]	512
ReLU-37	[-1, 256, 2, 2]	0
Conv2d-38	[-1, 256, 2, 2]	589,824
BatchNorm2d-39	[-1, 256, 2, 2]	512
Conv2d-40	[-1, 256, 2, 2]	32,768
BatchNorm2d-41	[-1, 256, 2, 2]	512
ReLU-42	[-1, 256, 2, 2]	0
BasicBlock-43	[-1, 256, 2, 2]	0
Conv2d-44	[-1, 256, 2, 2]	589,824
BatchNorm2d-45	[-1, 256, 2, 2]	512

ReLU-46	[-1, 256, 2, 2]	0
Conv2d-47	[-1, 256, 2, 2]	589,824
BatchNorm2d-48	[-1, 256, 2, 2]	512
ReLU-49	[-1, 256, 2, 2]	0
BasicBlock-50	[-1, 256, 2, 2]	0
Conv2d-51	[-1, 512, 1, 1]	1,179,648
BatchNorm2d-52	[-1, 512, 1, 1]	1,024
ReLU-53	[-1, 512, 1, 1]	0
Conv2d-54	[-1, 512, 1, 1]	2,359,296
BatchNorm2d-55	[-1, 512, 1, 1]	1,024
Conv2d-56	[-1, 512, 1, 1]	131,072
BatchNorm2d-57	[-1, 512, 1, 1]	1,024
ReLU-58	[-1, 512, 1, 1]	0
BasicBlock-59	[-1, 512, 1, 1]	0
Conv2d-60	[-1, 512, 1, 1]	2,359,296
BatchNorm2d-61	[-1, 512, 1, 1]	1,024
ReLU-62	[-1, 512, 1, 1]	0
Conv2d-63	[-1, 512, 1, 1]	2,359,296
BatchNorm2d-64	[-1, 512, 1, 1]	1,024
ReLU-65	[-1, 512, 1, 1]	0
BasicBlock-66	[-1, 512, 1, 1]	0
AdaptiveAvgPool2d-67	[-1, 512, 1, 1]	0
Linear-68	[-1, 1000]	513,000

```

=====
Total params: 11,689,512
Trainable params: 11,689,512
Non-trainable params: 0
-----
Input size (MB): 0.01
Forward/backward pass size (MB): 1.29
Params size (MB): 44.59
Estimated Total Size (MB): 45.90
-----

```

As per the code and it's output, the number of trainable parameter for Resnet18 with input size of (3,32,32) is 11,689,512 ~ 11.5M parameters

Q4.

In Adam, for each parameter, the learning rate of optimizer is adapted which results in more number of trainable parameters which in our case will be approximately 2 times – i.e. 22347924.