

Lab 2

Introduction to High Performance Machine Learning

Aditya Wagh

amw925

Problem C2

Results for SGD with given parameters and two workers.

```
=====
Running the computations for Question: C2
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 1.960 | Training Accuracy: 30.032
Epoch: 2 | Training Loss: 1.471 | Training Accuracy: 45.546
Epoch: 3 | Training Loss: 1.191 | Training Accuracy: 57.026
Epoch: 4 | Training Loss: 0.993 | Training Accuracy: 64.808
Epoch: 5 | Training Loss: 0.847 | Training Accuracy: 70.098
Finished training using sgd optimizer.
Epoch 1 | Data Loading Time: 0.557 sec | Training Time: 5.429 sec | Running Time: 25.527 sec
Epoch 2 | Data Loading Time: 0.520 sec | Training Time: 3.547 sec | Running Time: 23.631 sec
Epoch 3 | Data Loading Time: 0.516 sec | Training Time: 3.577 sec | Running Time: 23.743 sec
Epoch 4 | Data Loading Time: 0.500 sec | Training Time: 3.545 sec | Running Time: 23.830 sec
Epoch 5 | Data Loading Time: 0.513 sec | Training Time: 3.588 sec | Running Time: 23.895 sec
```

Problem C3

Results for SGD with given parameters and **no** workers.

```

=====
Running the computations for Question: C3
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 1.991 | Training Accuracy: 29.344
Epoch: 2 | Training Loss: 1.493 | Training Accuracy: 45.014
Epoch: 3 | Training Loss: 1.217 | Training Accuracy: 56.006
Epoch: 4 | Training Loss: 0.983 | Training Accuracy: 65.040
Epoch: 5 | Training Loss: 0.840 | Training Accuracy: 70.298
Finished training using sgd optimizer.
Total Data loading time for 0 workers: 62.254564471542835 sec
=====

```

Results for SGD with given parameters and **four** workers.

```

=====
Running the computations for Question: C3
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 2.071 | Training Accuracy: 27.166
Epoch: 2 | Training Loss: 1.542 | Training Accuracy: 42.838
Epoch: 3 | Training Loss: 1.340 | Training Accuracy: 51.052
Epoch: 4 | Training Loss: 1.136 | Training Accuracy: 59.408
Epoch: 5 | Training Loss: 0.973 | Training Accuracy: 65.292
Finished training using sgd optimizer.
Total Data loading time for 4 workers: 3.14421858638525 sec
=====

```

Results for SGD with given parameters and **eight** workers.

```

=====
Running the computations for Question: C3
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 1.923 | Training Accuracy: 30.358
Epoch: 2 | Training Loss: 1.424 | Training Accuracy: 47.332
Epoch: 3 | Training Loss: 1.182 | Training Accuracy: 57.570
Epoch: 4 | Training Loss: 0.990 | Training Accuracy: 64.862
Epoch: 5 | Training Loss: 0.844 | Training Accuracy: 70.256
Finished training using sgd optimizer.
Total Data loading time for 8 workers: 4.507385556586087 sec
=====

```

Results for SGD with given parameters and **twelve** workers.

```

=====
Running the computations for Question: C3
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 1.903 | Training Accuracy: 30.816
Epoch: 2 | Training Loss: 1.456 | Training Accuracy: 46.142
Epoch: 3 | Training Loss: 1.201 | Training Accuracy: 56.384
Epoch: 4 | Training Loss: 1.003 | Training Accuracy: 64.022
Epoch: 5 | Training Loss: 0.863 | Training Accuracy: 69.218
Finished training using sgd optimizer.
Total Data loading time for 12 workers: 5.804826465435326 sec

```

Results for SGD with given parameters and **sixteen** workers.

```

=====
Running the computations for Question: C3
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 2.065 | Training Accuracy: 27.370
Epoch: 2 | Training Loss: 1.555 | Training Accuracy: 42.344
Epoch: 3 | Training Loss: 1.320 | Training Accuracy: 51.854
Epoch: 4 | Training Loss: 1.108 | Training Accuracy: 60.114
Epoch: 5 | Training Loss: 0.942 | Training Accuracy: 66.702
Finished training using sgd optimizer.
Total Data loading time for 16 workers: 7.273995839059353 sec

```

We can thus conclude that the number of optimal workers is 4.

Problem C4

```

=====
Running the computations for Question: C4
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 1.908 | Training Accuracy: 31.074
Epoch: 2 | Training Loss: 1.382 | Training Accuracy: 49.168
Epoch: 3 | Training Loss: 1.118 | Training Accuracy: 59.686
Epoch: 4 | Training Loss: 0.946 | Training Accuracy: 66.516
Epoch: 5 | Training Loss: 0.797 | Training Accuracy: 71.800
Finished training using sgd optimizer.
Total Data loading time for 1 workers: 2.282497084699571 sec

```

```

=====
Running the computations for Question: C4
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 1.725 | Training Accuracy: 36.756
Epoch: 2 | Training Loss: 1.268 | Training Accuracy: 53.752
Epoch: 3 | Training Loss: 1.006 | Training Accuracy: 64.104
Epoch: 4 | Training Loss: 0.827 | Training Accuracy: 70.992
Epoch: 5 | Training Loss: 0.705 | Training Accuracy: 75.430
Finished training using sgd optimizer.
Total Data loading time for 4 workers: 3.088451717980206 sec

```

We can thus observe that a single worker is better than 4 workers. This might be due to the fact that the data loading is very fast here. SO adding more workers would create synchronisation overheads.

Problem C5

GPU Results

```

=====
Running the computations for Question: C5
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 1.874 | Training Accuracy: 31.982
Epoch: 2 | Training Loss: 1.448 | Training Accuracy: 46.868
Epoch: 3 | Training Loss: 1.200 | Training Accuracy: 56.848
Epoch: 4 | Training Loss: 0.975 | Training Accuracy: 65.214
Epoch: 5 | Training Loss: 0.803 | Training Accuracy: 71.782
Finished training using sgd optimizer.
Average running time for 5 epochs on a GPU: 24.183283765427767 sec

```

CPU Results

```
=====
Running the computations for Question: C5
Files already downloaded and verified
Files already downloaded and verified
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 2.057 | Training Accuracy: 27.970
Epoch: 2 | Training Loss: 1.533 | Training Accuracy: 43.180
Epoch: 3 | Training Loss: 1.269 | Training Accuracy: 53.604
Epoch: 4 | Training Loss: 1.046 | Training Accuracy: 62.486
Epoch: 5 | Training Loss: 0.909 | Training Accuracy: 67.644
Finished training using sgd optimizer.
Average running time for 5 epochs on a CPU: 209.24727779850363 sec
```

Problem C6

SGD

```
=====
Running the computations for Question: C6
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 1.924 | Training Accuracy: 30.968
Epoch: 2 | Training Loss: 1.419 | Training Accuracy: 47.652
Epoch: 3 | Training Loss: 1.146 | Training Accuracy: 58.736
Epoch: 4 | Training Loss: 0.957 | Training Accuracy: 65.944
Epoch: 5 | Training Loss: 0.794 | Training Accuracy: 72.118
Finished training using sgd optimizer.
Average training time per epoch for SGD: 3.8602296579629183 sec
Average loss per epoch for SGD: 1.2479867635332809
Average top-1 training per epoch for SGD: 55.083600000000004
```

SGD with Nesterov

```
=====
Running the computations for Question: C6
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGDNEST optimizer.
Epoch: 1 | Training Loss: 1.830 | Training Accuracy: 33.990
Epoch: 2 | Training Loss: 1.310 | Training Accuracy: 52.014
Epoch: 3 | Training Loss: 1.030 | Training Accuracy: 63.126
Epoch: 4 | Training Loss: 0.862 | Training Accuracy: 69.454
Epoch: 5 | Training Loss: 0.731 | Training Accuracy: 74.424
Finished training using sgdnest optimizer.
Average training time per epoch for SGDNEST: 4.18667245451361 sec
Average loss per epoch for SGDNEST: 1.1525002196583214
Average top-1 training per epoch for SGDNEST: 58.601600000000005
```

Adagrad

```
=====
Running the computations for Question: C6
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using ADAGRAD optimizer.
Epoch: 1 | Training Loss: 2.166 | Training Accuracy: 25.310
Epoch: 2 | Training Loss: 1.637 | Training Accuracy: 39.012
Epoch: 3 | Training Loss: 1.408 | Training Accuracy: 48.276
Epoch: 4 | Training Loss: 1.210 | Training Accuracy: 55.906
Epoch: 5 | Training Loss: 1.065 | Training Accuracy: 61.612
Finished training using adagrad optimizer.
Average training time per epoch for ADAGRAD: 4.128054190985859 sec
Average loss per epoch for ADAGRAD: 1.4970790893751746
Average top-1 training per epoch for ADAGRAD: 46.0232
```

Adadelata

```
=====
Running the computations for Question: C6
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using ADADELTA optimizer.
Epoch: 1 | Training Loss: 1.387 | Training Accuracy: 48.924
Epoch: 2 | Training Loss: 0.890 | Training Accuracy: 67.996
Epoch: 3 | Training Loss: 0.689 | Training Accuracy: 75.932
Epoch: 4 | Training Loss: 0.579 | Training Accuracy: 79.926
Epoch: 5 | Training Loss: 0.508 | Training Accuracy: 82.296
Finished training using adadelata optimizer.
Average training time per epoch for ADADELTA: 5.299018644541502 sec
Average loss per epoch for ADADELTA: 0.8107364524383934
Average top-1 training per epoch for ADADELTA: 71.0148
```

Adam

```

=====
Running the computations for Question: C6
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using ADAM optimizer.
Epoch: 1 | Training Loss: 2.145 | Training Accuracy: 24.354
Epoch: 2 | Training Loss: 1.842 | Training Accuracy: 29.172
Epoch: 3 | Training Loss: 1.815 | Training Accuracy: 30.318
Epoch: 4 | Training Loss: 1.803 | Training Accuracy: 30.982
Epoch: 5 | Training Loss: 1.783 | Training Accuracy: 31.614
Finished training using adam optimizer.
Average training time per epoch for ADAM: 4.814706747792661 sec
Average loss per epoch for ADAM: 1.8776330136522954
Average top-1 training per epoch for ADAM: 29.288

```

Problem C7

No Batch Norm

```

=====
Running the computations for Question: C7
Files already downloaded and verified
Files already downloaded and verified
GPU is available. Training on Quadro RTX 8000
Started training using SGD optimizer.
Epoch: 1 | Training Loss: 2.080 | Training Accuracy: 20.308
Epoch: 2 | Training Loss: 1.651 | Training Accuracy: 38.364
Epoch: 3 | Training Loss: 1.391 | Training Accuracy: 49.288
Epoch: 4 | Training Loss: 1.203 | Training Accuracy: 57.156
Epoch: 5 | Training Loss: 1.061 | Training Accuracy: 62.732
Finished training using sgd optimizer.
Average training loss per epoch without batch normalization layers: 1.477071694026188
Average top-1 training accuracy per epoch without batch normalization layers: 45.569599999999994

```

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 32, 32]	1,728
BatchNorm2d-2	[-1, 64, 32, 32]	128
Conv2d-3	[-1, 64, 32, 32]	36,864
BatchNorm2d-4	[-1, 64, 32, 32]	128
Conv2d-5	[-1, 64, 32, 32]	36,864
BatchNorm2d-6	[-1, 64, 32, 32]	128
BasicBlock-7	[-1, 64, 32, 32]	0
Conv2d-8	[-1, 64, 32, 32]	36,864

BatchNorm2d-9	[-1, 64, 32, 32]	128
Conv2d-10	[-1, 64, 32, 32]	36,864
BatchNorm2d-11	[-1, 64, 32, 32]	128
BasicBlock-12	[-1, 64, 32, 32]	0
Conv2d-13	[-1, 128, 16, 16]	73,728
BatchNorm2d-14	[-1, 128, 16, 16]	256
Conv2d-15	[-1, 128, 16, 16]	147,456
BatchNorm2d-16	[-1, 128, 16, 16]	256
Conv2d-17	[-1, 128, 16, 16]	8,192
BatchNorm2d-18	[-1, 128, 16, 16]	256
BasicBlock-19	[-1, 128, 16, 16]	0
Conv2d-20	[-1, 128, 16, 16]	147,456
BatchNorm2d-21	[-1, 128, 16, 16]	256
Conv2d-22	[-1, 128, 16, 16]	147,456
BatchNorm2d-23	[-1, 128, 16, 16]	256
BasicBlock-24	[-1, 128, 16, 16]	0
Conv2d-25	[-1, 256, 8, 8]	294,912
BatchNorm2d-26	[-1, 256, 8, 8]	512
Conv2d-27	[-1, 256, 8, 8]	589,824
BatchNorm2d-28	[-1, 256, 8, 8]	512
Conv2d-29	[-1, 256, 8, 8]	32,768
BatchNorm2d-30	[-1, 256, 8, 8]	512
BasicBlock-31	[-1, 256, 8, 8]	0
Conv2d-32	[-1, 256, 8, 8]	589,824
BatchNorm2d-33	[-1, 256, 8, 8]	512
Conv2d-34	[-1, 256, 8, 8]	589,824
BatchNorm2d-35	[-1, 256, 8, 8]	512
BasicBlock-36	[-1, 256, 8, 8]	0
Conv2d-37	[-1, 512, 4, 4]	1,179,648
BatchNorm2d-38	[-1, 512, 4, 4]	1,024
Conv2d-39	[-1, 512, 4, 4]	2,359,296
BatchNorm2d-40	[-1, 512, 4, 4]	1,024
Conv2d-41	[-1, 512, 4, 4]	131,072
BatchNorm2d-42	[-1, 512, 4, 4]	1,024
BasicBlock-43	[-1, 512, 4, 4]	0
Conv2d-44	[-1, 512, 4, 4]	2,359,296
BatchNorm2d-45	[-1, 512, 4, 4]	1,024
Conv2d-46	[-1, 512, 4, 4]	2,359,296
BatchNorm2d-47	[-1, 512, 4, 4]	1,024
BasicBlock-48	[-1, 512, 4, 4]	0
Linear-49	[-1, 10]	5,130


```

=====
Total params: 11,173,962
Trainable params: 11,173,962
Non-trainable params: 0
-----
Input size (MB): 0.01
Forward/backward pass size (MB): 11.25
Params size (MB): 42.63
Estimated Total Size (MB): 53.89
-----
None

```

Problem Q1

There are 18 convolutional layers in ResNet18.

Problem Q2

As we can see from the above summary, there are the input dimension of the last linear layer is $512 \times 4 \times 4$.

Problem Q3

```

from torchsummary import summary

# model is ResNet()
print(summary(model, (3, 32, 32)))

```

Number of parameters: 11,173,962. The above mentioned generates the above mentioned view of our model, and lists the models parameters at the bottom. The number of trainable parameters is equal to the number of gradients in case of SGD.

Problem Q4

In case of Adam, the learning rate is also a learnable parameter. For each pair of weights and biases, there exists an adaptive learning rate. Therefore, the amount of parameters of the network with Adam will be $(\text{\#weights} + \text{\#biases}) / 1.5 + \text{\#other parameters mentioned in the table}$.

