C2

- I used one GPU to train the model for 5 epochs
- Data-loading time for each epoch

Epochs	Time (seconds)
1	0.194432
2	0.189121
3	0.189422
4	0.194292
5	0.189678

- Training (i.e., mini-batch calculation) time for each epoch

Epochs	Time (seconds)
1	13.974
2	13.949
3	12.964
4	13.015
5	12.957

- Total running time for each epoch Run 5 epochs.
    - 66.814367 seconds

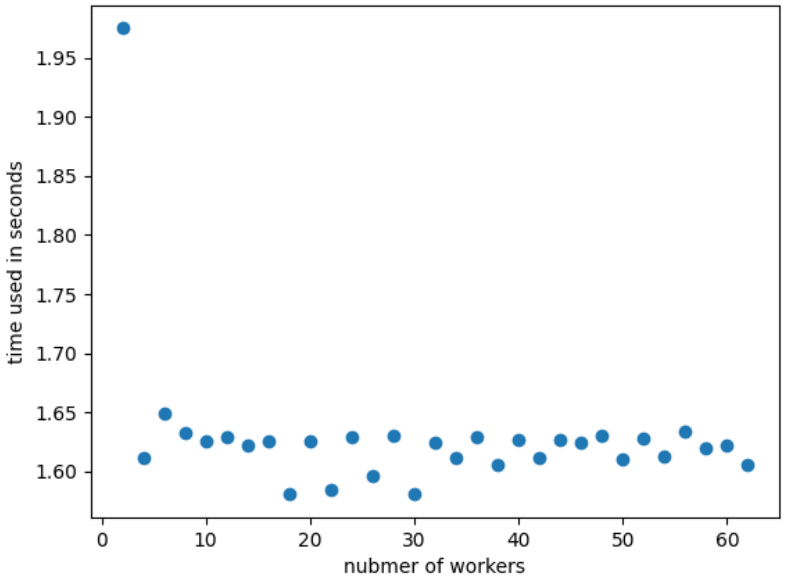
Raw output

```
==> Preparing data..
Files already downloaded and verified
Files already downloaded and verified
total number of workers: 2;
 total time used for loading data 2.167426216416061

Epoch: 0
epoch number: 0, training time used: 13.97391712013632s
epoch number: 0, data-loading time used: 0.19443207886070013s

Epoch: 1
epoch number: 1, training time used: 12.948571659624577s
epoch number: 1, data-loading time used: 0.18912078067660332s

Epoch: 2
epoch number: 2, training time used: 12.963752321898937s
epoch number: 2, data-loading time used: 0.18942245189100504s

Epoch: 3
epoch number: 3, training time used: 13.014581978321075s
epoch number: 3, data-loading time used: 0.19429199863225222s

Epoch: 4
epoch number: 4, training time used: 12.956598524935544s
epoch number: 4, data-loading time used: 0.18967782985419035s
total running time used for 5 epochs: 66.8143667448312s
```

C3

I tried number of workers from 2 to 64 and created a graph to indicate what is the ideal number of workers.

- 18 workers are needed for best runtime performance



Raw output

Number of workers	Total time	Number of workers	Total time
2	1.9748013447970152	16	1.6261683721095324
4	1.6119763320311904	18	1.580678628757596
6	1.6494588823989034	20	1.6262038461863995
8	1.6333424467593431	22	1.585087320767343
10	1.6259271977469325	24	1.6290130829438567
12	1.6290582856163383	26	1.5967026548460126
14	1.622746579349041	28	1.6309454776346684

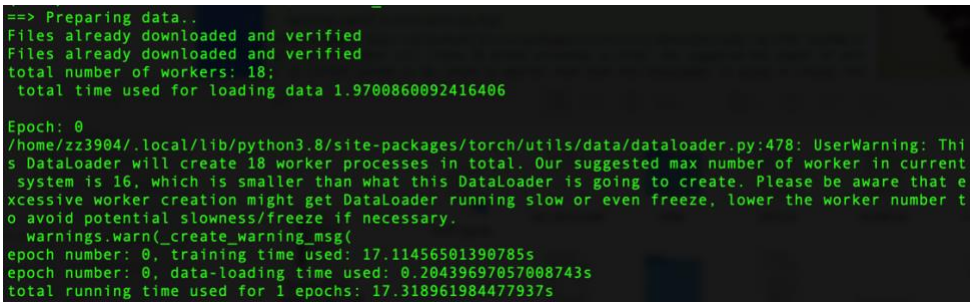
Number of workers	Total time	Number of workers	Total time
30	1.5814029155299067	36	1.6294998843222857
32	1.6242609722539783	38	1.6057686852291226
34	1.6115982653573155	40	1.6266595842316747

C4

- I used 1 GPU with 1 worker and 18 workers to run for 1 epoch.
- Use 1 worker

```
==> Preparing data..
Files already downloaded and verified
Files already downloaded and verified
total number of workers: 1;
 total time used for loading data 2.421840745024383

Epoch: 0
epoch number: 0, training time used: 17.645460643805563s
epoch number: 0, data-loading time used: 0.18510321527719498s
total running time used for 1 epochs: 17.83056385908276s
```

- Use 16 workers

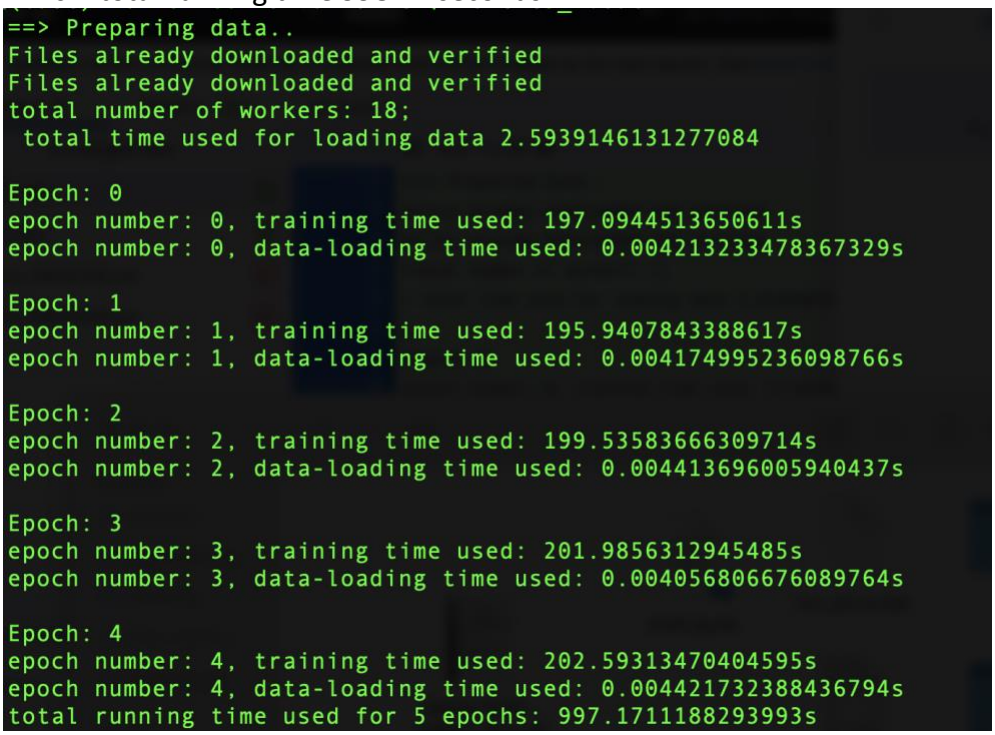


Explain the difference

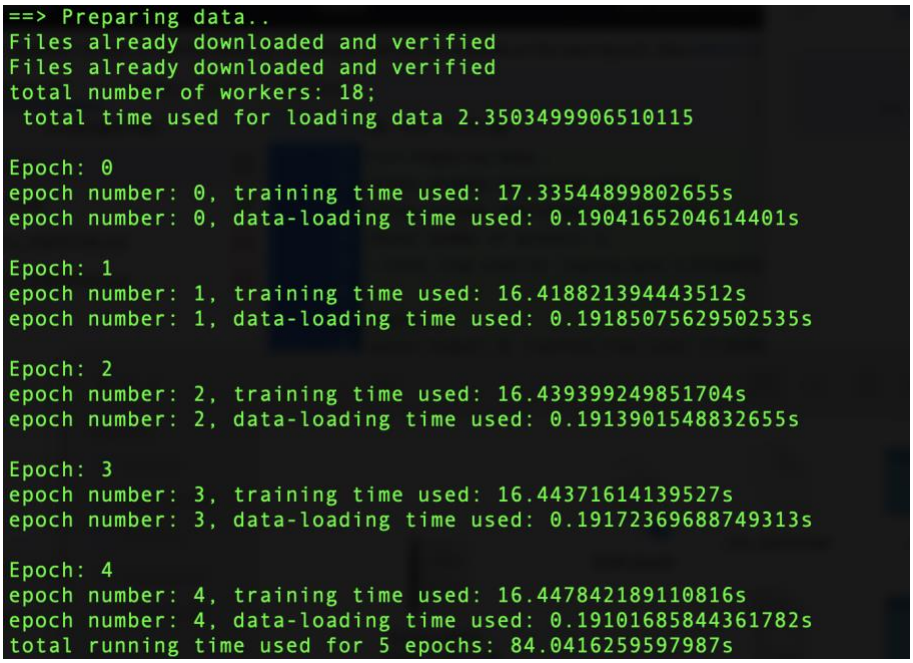
- Data loading time increased.
    - By having more processes simultaneously doing random access IO, the program would start overloading whatever IO device it's using. When number of workers are higher than number of processes, it's likely to have a lot of processes blocked on IO.
- Total running time decreased.
    - when num_workers=18, There are at most 18 workers simultaneously putting data into RAM. It uses parallelism to reduce training time.

C5

- use CPU with 18 workers
    - total running time 998.17 seconds



- used 1 GPU with 18 workers
    - total running time: 84.04 seconds



C6

SGD

Epoch: 0

Step: 855ms | Tot: 17s565ms | Loss: 1.962 | Acc: 29.888% (14944/50000)

Epoch: 1

Step: 30ms | Tot: 16s773ms | Loss: 1.463 | Acc: 45.954% (22977/50000)

Epoch: 2

Step: 30ms | Tot: 16s851ms | Loss: 1.195 | Acc: 56.976% (28488/50000)

Epoch: 3

Step: 29ms | Tot: 16s848ms | Loss: 0.981 | Acc: 65.382% (32691/50000)

Epoch: 4

Step: 30ms | Tot: 16s894ms | Loss: 0.804 | Acc: 71.928% (35964/50000)

best training accuracy: 0.72314453125

nesterov

Epoch: 0

Step: 264ms | Tot: 17s145ms | Loss: 2.022 | Acc: 28.620% (14310/50000)

Epoch: 1

Step: 30ms | Tot: 16s975ms | Loss: 1.440 | Acc: 47.030% (23515/50000)

Epoch: 2

Step: 30ms | Tot: 16s954ms | Loss: 1.148 | Acc: 58.740% (29370/50000)

Epoch: 3

Step: 29ms | Tot: 16s941ms | Loss: 0.949 | Acc: 66.244% (33122/50000)

Epoch: 4

Step: 30ms | Tot: 16s847ms | Loss: 0.815 | Acc: 71.186% (35593/50000)

best training accuracy: 0.71186

adadelta

Epoch: 0

Step: 270ms | Tot: 40s948ms | Loss: 1.368 | Acc: 49.722% (24861/50000)

Epoch: 1

Step: 32ms | Tot: 59s273ms | Loss: 0.888 | Acc: 68.358% (34179/50000)

Epoch: 2

Step: 31ms | Tot: 1m17s | Loss: 0.687 | Acc: 75.760% (37880/50000)

Epoch: 3

Step: 31ms | Tot: 1m35s | Loss: 0.580 | Acc: 79.982% (39991/50000)

Epoch: 4

Step: 31ms | Tot: 1m54s | Loss: 0.508 | Acc: 82.318% (41159/50000)

best training accuracy: 0.8359375

adagrad

Epoch: 0

Step: 267ms | Tot: 17s37ms | Loss: 2.097 | Acc: 27.074% (13537/50000)

Epoch: 1

Step: 30ms | Tot: 16s766ms | Loss: 1.593 | Acc: 40.742% (20371/50000)

Epoch: 2

Step: 30ms | Tot: 16s753ms | Loss: 1.351 | Acc: 50.928% (25464/50000)

Epoch: 3

Step: 30ms | Tot: 16s747ms | Loss: 1.130 | Acc: 59.186% (29593/50000)

Epoch: 4

Step: 30ms | Tot: 16s699ms | Loss: 0.983 | Acc: 65.104% (32552/50000)

best training accuracy: 0.65104

adam

Epoch: 0

Step: 1s87ms | Tot: 29s869ms | Loss: 2.261 | Acc: 22.014% (11007/50000)

Epoch: 1

Step: 31ms | Tot: 47s908ms | Loss: 1.848 | Acc: 29.894% (14947/50000)

Epoch: 2

Step: 30ms | Tot: 1m5s | Loss: 1.811 | Acc: 30.972% (15486/50000)

Epoch: 3

Step: 31ms | Tot: 1m24s | Loss: 1.811 | Acc: 30.200% (15100/50000)

Epoch: 4

Step: 31ms | Tot: 1m42s | Loss: 1.796 | Acc: 31.086% (15543/50000)

best training accuracy: 0.3671875

C7

Epoch: 0

Step: 202ms | Tot: 34s890ms | Loss: 2.159 | Acc: 17.300% (8650/50000)

Epoch: 1

Step: 20ms | Tot: 47s174ms | Loss: 1.740 | Acc: 34.462% (17231/50000)

Epoch: 2

Step: 20ms | Tot: 59s436ms | Loss: 1.486 | Acc: 45.346% (22673/50000)

Epoch: 3

Step: 20ms | Tot: 1m11s | Loss: 1.269 | Acc: 54.592% (27296/50000)

Epoch: 4

Step: 20ms | Tot: 1m23s | Loss: 1.100 | Acc: 61.282% (30641/50000)

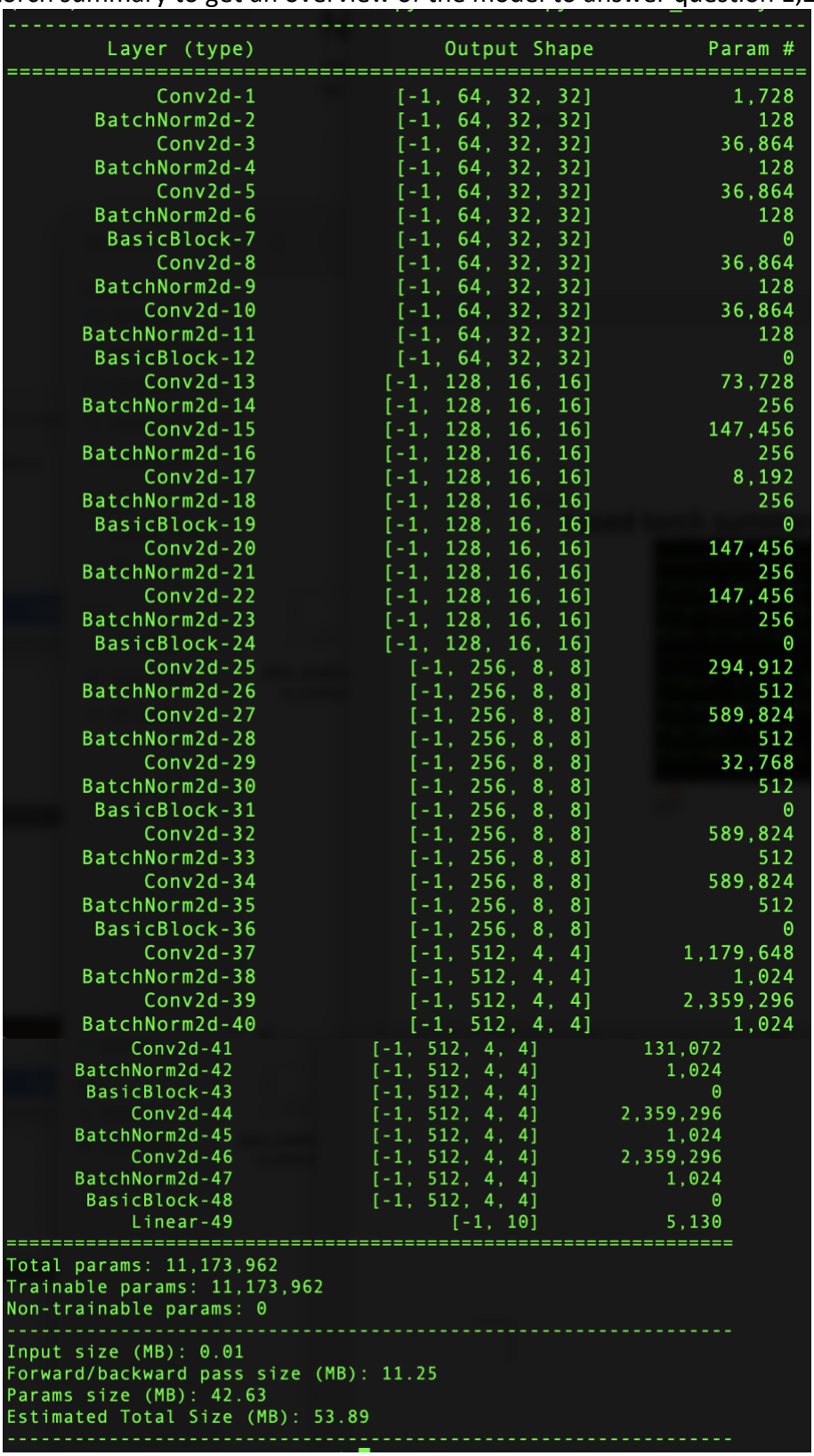
best training accuracy: 0.640625

Model summary

- Total number of parameters: 11, 164, 362

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 64, 32, 32]           1,728
            Conv2d-2          [-1, 64, 32, 32]          36,864
            Conv2d-3          [-1, 64, 32, 32]          36,864
        BasicBlock-4          [-1, 64, 32, 32]               0
            Conv2d-5          [-1, 64, 32, 32]          36,864
            Conv2d-6          [-1, 64, 32, 32]          36,864
        BasicBlock-7          [-1, 64, 32, 32]               0
            Conv2d-8         [-1, 128, 16, 16]          73,728
            Conv2d-9         [-1, 128, 16, 16]         147,456
           Conv2d-10         [-1, 128, 16, 16]           8,192
       BasicBlock-11         [-1, 128, 16, 16]               0
           Conv2d-12         [-1, 128, 16, 16]         147,456
           Conv2d-13         [-1, 128, 16, 16]         147,456
       BasicBlock-14         [-1, 128, 16, 16]               0
           Conv2d-15           [-1, 256, 8, 8]         294,912
           Conv2d-16           [-1, 256, 8, 8]         589,824
           Conv2d-17           [-1, 256, 8, 8]          32,768
       BasicBlock-18           [-1, 256, 8, 8]               0
           Conv2d-19           [-1, 256, 8, 8]         589,824
           Conv2d-20           [-1, 256, 8, 8]         589,824
       BasicBlock-21           [-1, 256, 8, 8]               0
           Conv2d-22           [-1, 512, 4, 4]       1,179,648
           Conv2d-23           [-1, 512, 4, 4]       2,359,296
           Conv2d-24           [-1, 512, 4, 4]         131,072
       BasicBlock-25           [-1, 512, 4, 4]               0
           Conv2d-26           [-1, 512, 4, 4]       2,359,296
           Conv2d-27           [-1, 512, 4, 4]       2,359,296
       BasicBlock-28           [-1, 512, 4, 4]               0
           Linear-29                 [-1, 10]           5,130
================================================================
Total params: 11,164,362
Trainable params: 11,164,362
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 6.56
Params size (MB): 42.59
Estimated Total Size (MB): 49.16
----------------------------------------------------------------
```

I used torch summary to get an overview of the model to answer question 1,2 and 3



```
----------------------------------------------------------------
        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
----------------------------------------------------------------
```

Q1

- Total number of convolutional layers: 46

Q2

- Input dimension of last layer is (512,4,4)

Q3

- Total number of trainable parameters: 11,173,962
- Total number of gradients: 11,173,962
- I used torchsummary library to count the parameters

```
from torchsummary import summary

def get_model_summary(self):
    summary(self.net, input_size=(3, 32, 32))
```

Q4

- Total number of trainable parameters: 11,173,962
- Total number of gradients: 11,173,962