C2

Note: CUDA is enabled as per Ed post #74

(C2.1) Data loading time

- Epoch 1: 0.7697923679943415 sec
- Epoch 2: 0.7602724030025456 sec
- Epoch 3: 0.779841631002455 sec
- Epoch 4: 0.7662150970049879 sec
- Epoch 5: 0.7567201880015091 sec

(C2.2) Training time

- Epoch 1: 16.1071728919992 sec
- Epoch 2: 16.153877125006602 sec
- Epoch 3: 16.202515732003576 sec
- Epoch 4: 16.2590576030002 sec
- Epoch 5: 16.294209903995124 sec

(C2.3) Total running time

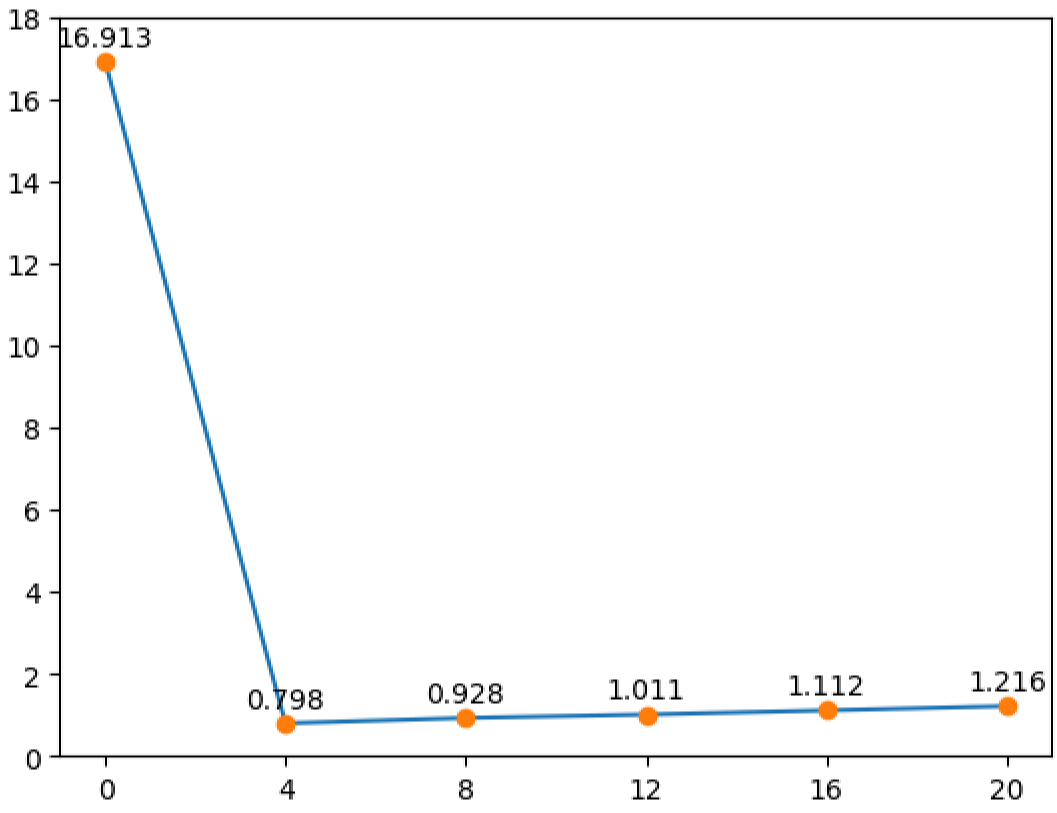
- Epoch 1: 17.483852927000044 sec
- Epoch 2: 17.523637065000003 sec
- Epoch 3: 17.59785359899979 sec
- Epoch 4: 17.63568424999994 sec
- Epoch 5: 17.664012863000153 sec

C3

Note: CUDA is enabled as per Ed post #74

(C3.1) Total data loading time per epoch

- No workers: 16.913351496999894 sec
- 4 workers: 0.7979606640019483 sec
- 8 workers: 0.9282079270064969 sec
- 12 workers: 1.0108642480067829 sec
- 16 workers: 1.1118243409996467 sec
- 20 workers: 1.2158078610027587 sec



(C3.2)

A total of 4 workers are needed for the best runtime performance.

C4

Note: CUDA is enabled as per Ed post #74

Average running times with 1 worker

- Dataloading time: 2.03401839879898 sec
- Training time: 16.06387521659999 sec
- Epoch time: 18.725226706000104 sec

Average running times with 4 workers

- Dataloading time: 0.8470184418014469 sec
- Training time: 16.481447054999535 sec
- Epoch time: 17.969459787600137 sec

Explanation

Using 4 workers makes the data loading time faster because it allows for parallelization, which enables splitting the data loading process across multiple cores and threads. Each worker can independently load and preprocess a portion of the data simultaneously, which results in a significant decrease in overall data loading time compared to using a single worker that loads and preprocesses data sequentially.

C5

Note: 4 workers are used

Average running times over 5 epochs on GPU

- Dataloading time: 0.8414340840028671 sec
- Training time: 16.187658810596805 sec
- Epoch time: 17.655478401399886 sec

Average running times over 5 epochs on CPU

- Dataloading time: 0.9559728062058639 sec
- Training time: 568.1058389383977 sec
- Epoch time: 569.7203961809997 sec

C6

SGD

- Epoch 1:
    - Training time: 16.075835018016733 sec
    - Training loss: 1.4367764410765276
    - Top-1 training accuracy: 0.47062
- Epoch 2:
    - Training time: 16.129305656018914 sec
    - Training loss: 1.175429884734971
    - Top-1 training accuracy: 0.5788
- Epoch 3:
    - Training time: 16.166585693010347 sec
    - Training loss: 0.9933257920052999
    - Top-1 training accuracy: 0.64804
- Epoch 4:
    - Training time: 16.229201563044626 sec

    - Training loss: 0.8558873660729059
    - Top-1 training accuracy: 0.6956
- Epoch 5:
    - Training time: 16.273045752021062 sec
    - Training loss: 0.7333825519475181
    - Top-1 training accuracy: 0.74266
- Average training time: 16.174794736422335 sec

SGD with nesterov

- Epoch 1:
    - Training time: 16.307665036978506 sec
    - Training loss: 1.3387060137965796
    - Top-1 training accuracy: 0.51074
- Epoch 2:
    - Training time: 16.36929449901254 sec
    - Training loss: 1.070419619455362
    - Top-1 training accuracy: 0.6167
- Epoch 3:
    - Training time: 16.420381668009213 sec
    - Training loss: 0.9196580410613429
    - Top-1 training accuracy: 0.67376
- Epoch 4:
    - Training time: 16.476820116015006 sec
    - Training loss: 0.8075268893595546
    - Top-1 training accuracy: 0.71546
- Epoch 5:
    - Training time: 16.52917695103315 sec
    - Training loss: 0.7074232554954031
    - Top-1 training accuracy: 0.7524
- Average training time: 16.420667654209684 sec

Adagrad

- Epoch 1:
    - Training time: 16.623913410996465 sec
    - Training loss: 1.6554738371573445
    - Top-1 training accuracy: 0.38234
- Epoch 2:
    - Training time: 16.627333019006983 sec
    - Training loss: 1.4209891493668032
    - Top-1 training accuracy: 0.4782
- Epoch 3:
    - Training time: 16.608916811024756 sec

    - Training loss: 1.2121688754052458
    - Top-1 training accuracy: 0.5592
- Epoch 4:
    - Training time: 16.596610782991775 sec
    - Training loss: 1.0551333807008652
    - Top-1 training accuracy: 0.62128
- Epoch 5:
    - Training time: 16.598212804010473 sec
    - Training loss: 0.9473572821568346
    - Top-1 training accuracy: 0.6626
- Average training time: 16.61099736560609 sec

Adadelta

- Epoch 1:
    - Training time: 17.782298907044606 sec
    - Training loss: 0.8826482599348668
    - Top-1 training accuracy: 0.68924
- Epoch 2:
    - Training time: 17.82743576601024 sec
    - Training loss: 0.6852396269283636
    - Top-1 training accuracy: 0.75964
- Epoch 3:
    - Training time: 17.82888523098336 sec
    - Training loss: 0.5770558550229767
    - Top-1 training accuracy: 0.79916
- Epoch 4:
    - Training time: 17.838106042008803 sec
    - Training loss: 0.5083957332784258
    - Top-1 training accuracy: 0.82408
- Epoch 5:
    - Training time: 17.846446872998058 sec
    - Training loss: 0.45148394033884454
    - Top-1 training accuracy: 0.84262
- Average training time: 17.824634563809013 sec

Adam

- Epoch 1:
    - Training time: 16.84951057599028 sec
    - Training loss: 1.8591171228672232
    - Top-1 training accuracy: 0.2864
- Epoch 2:
    - Training time: 16.726857498992104 sec

    - Training loss: 1.8433763959523661
    - Top-1 training accuracy: 0.29212
- Epoch 3:
    - Training time: 16.62185980600043 sec
    - Training loss: 1.8406581223163458
    - Top-1 training accuracy: 0.2926
- Epoch 4:
    - Training time: 16.541029599002286 sec
    - Training loss: 1.8211790564115091
    - Top-1 training accuracy: 0.29566
- Epoch 5:
    - Training time: 16.485243415994773 sec
    - Training loss: 1.8228349999698532
    - Top-1 training accuracy: 0.29372
- Average training time: 16.644900179195975 sec

C7

- Epoch 1: Training loss: 1.5602200470312173; Top-1 training accuracy: 0.4217
- Epoch 2: Training loss: 1.3194459097464677; Top-1 training accuracy: 0.52168
- Epoch 3: Training loss: 1.139399854728328; Top-1 training accuracy: 0.59438
- Epoch 4: Training loss: 1.0071969184729144; Top-1 training accuracy: 0.64618
- Epoch 5: Training loss: 0.8966167668247467; Top-1 training accuracy: 0.68952

Q1

There are 17 convolutional layers in ResNet-18.

Q2

The input dimension of the last linear layer is 512.

Q3

There are 11173962 trainable parameters and 11173962 gradients in the model when using SGD optimizer. See below for the calculation code.

```python
import torch, torchvision
import resnet18
```

```
device = 'cuda' if torch.cuda.is_available() else 'cpu'

# Data
transform_train = torchvision.transforms.Compose([
  torchvision.transforms.RandomCrop(size=32, padding=4),
  torchvision.transforms.RandomHorizontalFlip(p=0.5),
  torchvision.transforms.ToTensor(),
  torchvision.transforms.Normalize((0.4914, 0.4822, 0.4465), (0.2023, 0.1994,
0.2010)),
])
trainset = torchvision.datasets.CIFAR10(root='./data', train=True, download=True,
transform=transform_train)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=128, shuffle=True,
num_workers=4)

# Model
net = resnet18.ResNet18().to(device)
criterion = torch.nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(net.parameters(), lr=0.1, momentum=0.9,
weight_decay=5e-4)

# Train
net.train()
inputs, targets = next(iter(trainloader))
inputs, targets = inputs.to(device), targets.to(device)
optimizer.zero_grad()
outputs = net(inputs)
loss = criterion(outputs, targets)
loss.backward()
optimizer.step()

# Calculate Number of Trainable Parameters and Gradients
num_params = sum(p.numel() for p in net.parameters() if p.requires_grad)
num_grad = sum(p.grad.numel() for p in net.parameters() if p.requires_grad)
print('Number of Trainable Parameters:', num_params)
print('Number of Gradients:', num_grad)
```

Q4

When using Adam, we see the same number of trainable parameters and gradients.
Number of Trainable Parameters: 11173962
Number of Gradients: 11173962