EE21S055_Tutorial6_Bonus

September 18, 2022

0.1 Imports

[1]:
```
import torch
import torchvision
import torchvision.transforms as transforms
from torch.utils.data import Dataset, DataLoader
import torch.nn as nn
import torch.nn.functional as F
import sys
import numpy as np
import os
```

0.2 Utilising GPU using Pytorch

[2]:
```
# cpu-gpu
a = torch.randn((3, 4))
print(a.device)

device = torch.device("cuda")
a = a.to(device)
print(a.device)

# a more generic code
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

```
cpu
cuda:0
```

[3]:
```
torch.cuda.is_available()
```

[3]:
```
True
```

[4]:
```
!nvidia-smi
```

```
Sun Sep 18 11:55:15 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
```

```
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   45C    P0    26W /  70W |    612MiB / 15109MiB |      1%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
+-----------------------------------------------------------------------------+
```

0.3 Dataset and Transforms

[5]:
```
train_transform = transforms.Compose([
  transforms.RandomCrop(32, padding=4),
  transforms.RandomHorizontalFlip(),
  transforms.ToTensor(),
  transforms.Normalize((0.4914, 0.4822, 0.4465), (0.2023, 0.1994, 0.2010)),
])
test_transform = transforms.Compose([
  transforms.ToTensor(),
  transforms.Normalize((0.4914, 0.4822, 0.4465), (0.2023, 0.1994, 0.2010)),
])

train_dset = torchvision.datasets.CIFAR10(root="data/", train=True,␣
  ↪transform=train_transform, download=True)
test_dset = torchvision.datasets.CIFAR10(root="data/", train=False,␣
  ↪transform=test_transform, download=True)
```

```
Downloading https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz to
data/cifar-10-python.tar.gz

  0%|          | 0/170498071 [00:00<?, ?it/s]

Extracting data/cifar-10-python.tar.gz to data/
Files already downloaded and verified
```

[6]:
```
print(f"# of train samples: {len(train_dset)}")
print(f"# of test samples: {len(test_dset)}")
```

```
# of train samples: 50000
# of test samples: 10000
```

[7]:
```
train_loader = DataLoader(train_dset, batch_size=100, shuffle=True,␣
 ↪num_workers=2)
test_loader = DataLoader(test_dset, batch_size=100, shuffle=False,␣
 ↪num_workers=2)
```

[8]:
```
print(f"# of train batches: {len(train_loader)}")
print(f"# of test batches: {len(test_loader)}")
```

```
# of train batches: 500
# of test batches: 100
```

[9]:
```
print("sample i/o sizes")
data = next(iter(train_loader))
img, target = data
print(f"input size: {img.shape}")
print(f"output size: {target.shape}")
```

```
sample i/o sizes
input size: torch.Size([100, 3, 32, 32])
output size: torch.Size([100])
```

0.4 LeNet

[43]:
```
class LeNet(nn.Module):
  def __init__(self):
    super(LeNet, self).__init__()
    self.conv1 = nn.Conv2d(3, 6, kernel_size=5)
    self.conv2 = nn.Conv2d(6, 32, kernel_size=5)
    self.conv3 = nn.Conv2d(32, 64, kernel_size=5)
    # TODO: missing input feature size
    self.fc1   = nn.Linear(64*5*5, 120)
    self.fc2   = nn.Linear(120, 84)
    # TODO: missing output feature size
    self.fc3   = nn.Linear(84, 10)
    self.activ = nn.ReLU()

  # TODO: add maxpool operation of given kernel size
  # https://pytorch.org/docs/stable/nn.functional.html
  def pool(self, x, kernel_size=2):
    out = F.max_pool2d(x,kernel_size)
    return out

  def forward(self, x):
    out = self.activ(self.conv1(x))
    out = self.pool(out)
    out = self.activ(self.conv2(out))
    out = self.pool(out)
```

```
        out = self.activ(self.conv3(out))
        out = self.pool(out)


        # TODO: flatten
        out = out.view(out.size(0),-1)
        out = self.activ(self.fc1(out))
        out = self.activ(self.fc2(out))
        out = self.fc3(out)
        return out
```

0.5 Utility functions (can ignore)

[20]:
```
def pbar(p=0, msg="", bar_len=20):
    sys.stdout.write("\033[K")
    sys.stdout.write("\x1b[2K" + "\r")
    block = int(round(bar_len * p))
    text = "Progress: [{}] {}% {}".format(
        "\x1b[32m" + "=" * (block - 1) + ">" + "\033[0m" + "-" * (bar_len -␣
 ↪block),
        round(p * 100, 2),
        msg,
    )
    print(text, end="\r")
    if p == 1:
        print()


class AvgMeter:
    def __init__(self):
        self.reset()

    def reset(self):
        self.metrics = {}

    def add(self, batch_metrics):
        if self.metrics == {}:
            for key, value in batch_metrics.items():
                self.metrics[key] = [value]
        else:
            for key, value in batch_metrics.items():
                self.metrics[key].append(value)

    def get(self):
        return {key: np.mean(value) for key, value in self.metrics.items()}
```

```
    def msg(self):
        avg_metrics = {key: np.mean(value) for key, value in self.metrics.
↪items()}
        return "".join(["[{}] {:.5f} ".format(key, value) for key, value in␣
↪avg_metrics.items()])
```

0.6 Training

[21]:
```
def train(model, optim, lr_sched=None, epochs=20, device=torch.device("cuda" if␣
↪torch.cuda.is_available() else "cpu"), criterion=None, metric_meter=None,␣
↪out_dir="out/"):
 model.to(device)
 best_acc = 0
 for epoch in range(epochs):
   model.train()
   metric_meter.reset()
   for indx, (img, target) in enumerate(train_loader):
     # TODO: send to device (cpu or gpu)
     img = img.to(device)
     target = target.to(device)

     # TODO: missing forward pass
     out = model(img)
     loss = criterion(out, target)
     # TODO: missing backward, parameter update
     optim.zero_grad()
     loss.backward()
     optim.step()
     metric_meter.add({"train loss": loss.item()})
     pbar(indx / len(train_loader), msg=metric_meter.msg())
   pbar(1, msg=metric_meter.msg())

   model.eval()
   metric_meter.reset()
   for indx, (img, target) in enumerate(test_loader):
     # TODO: send to device (cpu or gpu)
     img = img.to(device)
     target = target.to(device)

     # TODO: missing forward pass
     out = model(img)
     loss = criterion(out, target)
     # TODO: compute accuracy
     acc = (out.argmax(1) == target).type(torch.float).sum().item()

     metric_meter.add({"test loss": loss.item(), "test acc": acc})
```

```
      pbar(indx / len(test_loader), msg=metric_meter.msg())
    pbar(1, msg=metric_meter.msg())

    test_metrics = metric_meter.get()
    if test_metrics["test acc"] > best_acc:
      print(
          "\x1b[33m"
          + f"test acc improved from {round(best_acc, 5)} to␣
↪{round(test_metrics['test acc'], 5)}"
          + "\033[0m"
      )
      best_acc = test_metrics['test acc']
      torch.save(model.state_dict(), os.path.join(out_dir, "best.ckpt"))
    lr_sched.step()
```

0.7 Run Experiments

[22]:
```
def run_experiment(model_name="lenet", model_cfg=None, epochs=20):
  if model_name == "lenet":
    model = LeNet()
  optim = torch.optim.SGD(model.parameters(), lr=1e-1, momentum=0.9,␣
↪weight_decay=5e-4)
  lr_sched = torch.optim.lr_scheduler.CosineAnnealingLR(optim, T_max=epochs)
  criterion = nn.CrossEntropyLoss()
  metric_meter = AvgMeter()
  out_dir = f"{model_name}_{model_cfg}"
  os.makedirs(out_dir, exist_ok=True)
  train(model, optim, lr_sched, epochs=epochs, criterion=criterion,␣
↪metric_meter=metric_meter, out_dir=out_dir)
```

[23]:
```
run_experiment(model_name="lenet")
```

```
Progress: [====================>] 100% [train loss] 2.09773
Progress: [====================>] 100% [test loss] 2.00331 [test acc]
25.95000
test acc improved from 0 to 25.95
Progress: [====================>] 100% [train loss] 1.90805
Progress: [====================>] 100% [test loss] 1.88315 [test acc]
29.02000
test acc improved from 25.95 to 29.02
Progress: [====================>] 100% [train loss] 1.87238
Progress: [====================>] 100% [test loss] 1.82579 [test acc]
33.49000
test acc improved from 29.02 to 33.49
Progress: [====================>] 100% [train loss] 1.84685
Progress: [====================>] 100% [test loss] 1.84448 [test acc]
31.75000
```

```
Progress: [====================>] 100% [train loss] 1.81004
Progress: [====================>] 100% [test loss] 1.71023 [test acc]
36.34000
test acc improved from 33.49 to 36.34
Progress: [====================>] 100% [train loss] 1.76901
Progress: [====================>] 100% [test loss] 1.74466 [test acc]
36.95000
test acc improved from 36.34 to 36.95
Progress: [====================>] 100% [train loss] 1.72513
Progress: [====================>] 100% [test loss] 1.65723 [test acc]
37.38000
test acc improved from 36.95 to 37.38
Progress: [====================>] 100% [train loss] 1.68298
Progress: [====================>] 100% [test loss] 1.55102 [test acc]
43.42000
test acc improved from 37.38 to 43.42
Progress: [====================>] 100% [train loss] 1.62786
Progress: [====================>] 100% [test loss] 1.54186 [test acc]
44.17000
test acc improved from 43.42 to 44.17
Progress: [====================>] 100% [train loss] 1.57178
Progress: [====================>] 100% [test loss] 1.54262 [test acc]
44.66000
test acc improved from 44.17 to 44.66
Progress: [====================>] 100% [train loss] 1.52041
Progress: [====================>] 100% [test loss] 1.47860 [test acc]
46.27000
test acc improved from 44.66 to 46.27
Progress: [====================>] 100% [train loss] 1.46437
Progress: [====================>] 100% [test loss] 1.41450 [test acc]
48.96000
test acc improved from 46.27 to 48.96
Progress: [====================>] 100% [train loss] 1.39063
Progress: [====================>] 100% [test loss] 1.31499 [test acc]
53.62000
test acc improved from 48.96 to 53.62
Progress: [====================>] 100% [train loss] 1.31890
Progress: [====================>] 100% [test loss] 1.21614 [test acc]
56.95000
test acc improved from 53.62 to 56.95
Progress: [====================>] 100% [train loss] 1.27173
Progress: [====================>] 100% [test loss] 1.18751 [test acc]
58.10000
test acc improved from 56.95 to 58.1
Progress: [====================>] 100% [train loss] 1.20838
Progress: [====================>] 100% [test loss] 1.14827 [test acc]
59.24000
test acc improved from 58.1 to 59.24
```

```
Progress: [==================>] 100% [train loss] 1.15214
Progress: [==================>] 100% [test loss] 1.10608 [test acc]
61.27000
test acc improved from 59.24 to 61.27
Progress: [==================>] 100% [train loss] 1.11630
Progress: [==================>] 100% [test loss] 1.04973 [test acc]
62.81000
test acc improved from 61.27 to 62.81
Progress: [==================>] 100% [train loss] 1.07916
Progress: [==================>] 100% [test loss] 1.03096 [test acc]
63.75000
test acc improved from 62.81 to 63.75
Progress: [==================>] 100% [train loss] 1.06248
Progress: [==================>] 100% [test loss] 1.02110 [test acc]
64.48000
test acc improved from 63.75 to 64.48
```

16->32 Accuracy Moved from 62.57% to 64.48%

[37]:
```
def run_experiment(model_name="lenet", model_cfg=None, epochs=20):
  if model_name == "lenet":
    model = LeNet()
  optim = torch.optim.SGD(model.parameters(), lr=1e-1, momentum=0.9,␣
 ↪weight_decay=5e-4)
  lr_sched = torch.optim.lr_scheduler.CosineAnnealingLR(optim, T_max=epochs)
  criterion = nn.CrossEntropyLoss()
  metric_meter = AvgMeter()
  out_dir = f"{model_name}_{model_cfg}"
  os.makedirs(out_dir, exist_ok=True)
  train(model, optim, lr_sched, epochs=epochs, criterion=criterion,␣
 ↪metric_meter=metric_meter, out_dir=out_dir)
```

[42]:
```
run_experiment(model_name="lenet")
```

```
Progress: [==================>] 100% [train loss] 2.00005
Progress: [==================>] 100% [test loss] 1.84241 [test acc]
27.50000
test acc improved from 0 to 27.5
Progress: [==================>] 100% [train loss] 1.88387
Progress: [==================>] 100% [test loss] 1.87203 [test acc]
28.54000
test acc improved from 27.5 to 28.54
Progress: [==================>] 100% [train loss] 1.83851
Progress: [==================>] 100% [test loss] 1.86013 [test acc]
32.68000
test acc improved from 28.54 to 32.68
Progress: [==================>] 100% [train loss] 1.75704
Progress: [==================>] 100% [test loss] 1.65365 [test acc]
```

```
40.93000
test acc improved from 32.68 to 40.93
Progress: [==================>] 100% [train loss] 1.69457
Progress: [==================>] 100% [test loss] 1.64754 [test acc]
41.67000
test acc improved from 40.93 to 41.67
Progress: [==================>] 100% [train loss] 1.65990
Progress: [==================>] 100% [test loss] 1.60931 [test acc]
43.41000
test acc improved from 41.67 to 43.41
Progress: [==================>] 100% [train loss] 1.59530
Progress: [==================>] 100% [test loss] 1.54697 [test acc]
44.27000
test acc improved from 43.41 to 44.27
Progress: [==================>] 100% [train loss] 1.54157
Progress: [==================>] 100% [test loss] 1.45000 [test acc]
46.65000
test acc improved from 44.27 to 46.65
Progress: [==================>] 100% [train loss] 1.48376
Progress: [==================>] 100% [test loss] 1.45143 [test acc]
49.88000
test acc improved from 46.65 to 49.88
Progress: [==================>] 100% [train loss] 1.43155
Progress: [==================>] 100% [test loss] 1.33434 [test acc]
52.40000
test acc improved from 49.88 to 52.4
Progress: [==================>] 100% [train loss] 1.36831
Progress: [==================>] 100% [test loss] 1.26346 [test acc]
56.50000
test acc improved from 52.4 to 56.5
Progress: [==================>] 100% [train loss] 1.29312
Progress: [==================>] 100% [test loss] 1.16919 [test acc]
59.38000
test acc improved from 56.5 to 59.38
Progress: [==================>] 100% [train loss] 1.21105
Progress: [==================>] 100% [test loss] 1.08434 [test acc]
63.02000
test acc improved from 59.38 to 63.02
Progress: [==================>] 100% [train loss] 1.14628
Progress: [==================>] 100% [test loss] 1.07678 [test acc]
62.69000
Progress: [==================>] 100% [train loss] 1.07332
Progress: [==================>] 100% [test loss] 1.00292 [test acc]
65.20000
test acc improved from 63.02 to 65.2
Progress: [==================>] 100% [train loss] 1.01084
Progress: [==================>] 100% [test loss] 0.93889 [test acc]
67.80000
```

```
test acc improved from 65.2 to 67.8
Progress: [==================>] 100% [train loss] 0.95182
Progress: [==================>] 100% [test loss] 0.88403 [test acc]
69.45000
test acc improved from 67.8 to 69.45
Progress: [==================>] 100% [train loss] 0.91050
Progress: [==================>] 100% [test loss] 0.85413 [test acc]
70.31000
test acc improved from 69.45 to 70.31
Progress: [==================>] 100% [train loss] 0.86950
Progress: [==================>] 100% [test loss] 0.83311 [test acc]
70.96000
test acc improved from 70.31 to 70.96
Progress: [==================>] 100% [train loss] 0.85926
Progress: [==================>] 100% [test loss] 0.82271 [test acc]
71.33000
test acc improved from 70.96 to 71.33
```

32->64 Accuracy Moved from 64.48% to 71.33

[ ]:

[ ]: