(Optional) Colab Setup

If you aren't using Colab, you can delete the following code cell. This is just to help students with mounting to Google Drive to access the other .py files and downloading the data, which is a little trickier on Colab than on your local machine using Jupyter.

In [47]:

```
# you will be prompted with a window asking to grant permissions
from google.colab import drive
drive.mount("/content/drive")
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
/tmp/ipykernel_3178947/3103978175.py in <module>
      1 # you will be prompted with a window asking to grant permissions
----> 2 from google.colab import drive
      3 drive.mount("/content/drive")

ModuleNotFoundError: No module named 'google.colab'
```

In [ ]:

```
# fill in the path in your Google Drive in the string below. Note: do not escape slash
import os
datadir = "/content/assignment3"
if not os.path.exists(datadir):
  !ln -s "/content/drive/My Drive/Your/A3/path/" $datadir # TODO: Fill your A3 path
os.chdir(datadir)
!pwd
```

Data Setup

The first thing to do is implement a dataset class to load rotated CIFAR10 images with matching labels. Since there is already a CIFAR10 dataset class implemented in `torchvision`, we will extend this class and modify the `__get_item__` method appropriately to load rotated images.

Each rotation label should be an integer in the set {0, 1, 2, 3} which correspond to rotations of 0, 90, 180, or 270 degrees respectively.

In [3]:

```
import torch
import torchvision
import torchvision.transforms as transforms
import numpy as np
import random


def rotate_img(img, rot):
    if rot == 0: # 0 degrees rotation
        return img
    # TODO: Implement rotate_img() - return the rotated img
    elif rot == 1:
        return transforms.functional.rotate(img, 90)
    elif rot == 2:
        return transforms.functional.rotate(img, 180)
    elif rot == 3:
        return transforms.functional.rotate(img, 270)
    else:
        raise ValueError('rotation should be 0, 90, 180, or 270 degrees')


class CIFAR10Rotation(torchvision.datasets.CIFAR10):

    def __init__(self, root, train, download, transform) -> None:
        super().__init__(root=root, train=train, download=download, transform=transfor

    def __len__(self):
        return len(self.data)

    def __getitem__(self, index: int):
        image, cls_label = super().__getitem__(index)

        # randomly select image rotation
        rotation_label = random.choice([0, 1, 2, 3])
        image_rotated = rotate_img(image, rotation_label)

        rotation_label = torch.tensor(rotation_label).long()
        return image, image_rotated, rotation_label, torch.tensor(cls_label).long()
```

In [4]:

```
transform_train = 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)),
])

transform_test = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.4914, 0.4822, 0.4465), (0.2023, 0.1994, 0.2010)),
])

batch_size = 128

trainset = CIFAR10Rotation(root='./data', train=True,
                                        download=True, transform=transform_train)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=batch_size,
                                          shuffle=True, num_workers=2)

testset = CIFAR10Rotation(root='./data', train=False,
                                       download=True, transform=transform_test)
testloader = torch.utils.data.DataLoader(testset, batch_size=batch_size,
                                         shuffle=False, num_workers=2)
```

```
Files already downloaded and verified
Files already downloaded and verified
```

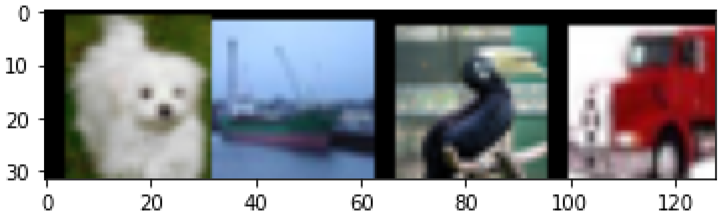
Show some example images and rotated images with labels:

In [6]:

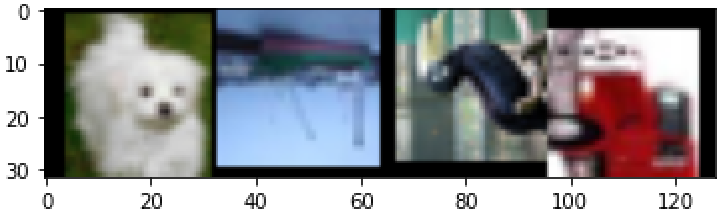
```
import matplotlib.pyplot as plt

classes = ('plane', 'car', 'bird', 'cat',
           'deer', 'dog', 'frog', 'horse', 'ship', 'truck')
```

```
rot_classes = ('0', '90', '180', '270')


def imshow(img):
    # unnormalize
    img = transforms.Normalize((0, 0, 0), (1/0.2023, 1/0.1994, 1/0.2010))(img)
    img = transforms.Normalize((-0.4914, -0.4822, -0.4465), (1, 1, 1))(img)
    npimg = img.numpy()
    plt.imshow(np.transpose(npimg, (1, 2, 0)))
    plt.show()


dataiter = iter(trainloader)
images, rot_images, rot_labels, labels = next(dataiter)

# print images and rotated images
img_grid = imshow(torchvision.utils.make_grid(images[:4], padding=0))
print('Class labels: ', ' '.join(f'{classes[labels[j]]:5s}' for j in range(4)))
img_grid = imshow(torchvision.utils.make_grid(rot_images[:4], padding=0))
print('Rotation labels: ', ' '.join(f'{rot_classes[rot_labels[j]]:5s}' for j in range(
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or
[0..255] for integers).
```



```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or
[0..255] for integers).
Class labels:  dog   ship  bird  truck
```



```
Rotation labels:  0     180   90    270
```

Evaluation code

In [7]:

```
import time

def run_test(net, testloader, criterion, task):
    correct = 0
    total = 0
    avg_test_loss = 0.0
    # since we're not training, we don't need to calculate the gradients for our outpu
    with torch.no_grad():
        for images, images_rotated, labels, cls_labels in testloader:
            if task == 'rotation':
              images, labels = images_rotated.to(device), labels.to(device)
            elif task == 'classification':
              images, labels = images.to(device), cls_labels.to(device)
            # TODO: Calculate outputs by running images through the network
            # The class with the highest energy is what we choose as prediction
            outputs = net(images)

            # TODO: unsure which dim should use
            _, predicted = torch.max(outputs, 1)
            # update total & correct
            total += labels.size(0)
            correct += (predicted == labels).sum().item()

            # loss
            avg_test_loss += criterion(outputs, labels)  / len(testloader)
    print('TESTING:')
    print(f'Accuracy of the network on the 10000 test images: {100 * correct / total:.
    print(f'Average loss on the 10000 test images: {avg_test_loss:.3f}')
```

In [8]:

```
def adjust_learning_rate(optimizer, epoch, init_lr, decay_epochs=30):
    """Sets the learning rate to the initial LR decayed by 10 every 30 epochs"""
    lr = init_lr * (0.1 ** (epoch // decay_epochs))
    for param_group in optimizer.param_groups:
        param_group['lr'] = lr
```

Train a ResNet18 on the rotation task

In this section, we will train a ResNet18 model on the rotation task. The input is a rotated image and the model predicts the rotation label. See the Data Setup section for details.

In [9]:

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

device = torch.device("cuda:1") if torch.cuda.is_available() else 'cpu'
device
```

Out[9]:

```
device(type='cuda', index=1)
```

In [18]:

```
import torch.nn as nn
import torch.nn.functional as F

from torchvision.models import resnet18

net = resnet18(num_classes=4)
net = net.to(device)
```

In [19]:

```
import torch.optim as optim

# TODO: Define criterion and optimizer
lr = 0.01
momentum = 0.9
criterion = nn.CrossEntropyLoss() # criterion -> loss function
# optimizer = optim.SGD(net.parameters(), lr=lr, momentum=momentum)
optimizer = optim.Adam(net.parameters(), lr=lr)
```

In [13]:

```
# Both the self-supervised rotation task and supervised CIFAR10 classification are
# trained with the CrossEntropyLoss, so we can use the training loop code.
```

```
def train(net, criterion, optimizer, num_epochs, decay_epochs, init_lr, task):

    for epoch in range(num_epochs):  # loop over the dataset multiple times

        running_loss = 0.0
        running_correct = 0.0
        running_total = 0.0
        start_time = time.time()

        net.train()

        for i, (imgs, imgs_rotated, rotation_label, cls_label) in enumerate(trainloade
            adjust_learning_rate(optimizer, epoch, init_lr, decay_epochs)

            # TODO: Set the data to the correct device; Different task will use differ

            if task == 'rotation':
              images, labels = imgs_rotated.to(device), rotation_label.to(device)
            elif task == 'classification':
              images, labels = imgs.to(device), cls_label.to(device)

            # TODO: Zero the parameter gradients
            optimizer.zero_grad()

            # TODO: forward + backward + optimize

            # forward
            net = net.to(device)
            outputs = net(images)
            loss = criterion(outputs, labels)
            # backward
            loss.backward()
            # optimize
            optimizer.step()

            # TODO: Get predicted results
            _, predicted = torch.max(outputs, 1)

            # print statistics
            print_freq = 100
            running_loss += loss.item()

            # calc acc
            running_total += labels.size(0)
            running_correct += (predicted == labels).sum().item()

            if i % print_freq == (print_freq - 1):    # print every 2000 mini-batches
                print(f'[{epoch + 1}, {i + 1:5d}] loss: {running_loss / print_freq:.3
                running_loss, running_correct, running_total = 0.0, 0.0, 0.0
                start_time = time.time()

        # TODO: Run the run_test() function after each epoch; Set the model to the eva
        net.eval()
        run_test(net, testloader, criterion, task)

    print('Finished Training')
```

In [20]:

```
train(net, criterion, optimizer, num_epochs=45, decay_epochs=15, init_lr=0.01, task='

# TODO: Save the model
# torch.save(net.state_dict(), 'ResNet18_model.pth')
```

```
[1,   100] loss: 1.493 acc: 32.02 time: 2.07
[1,   200] loss: 1.249 acc: 43.95 time: 1.90
[1,   300] loss: 1.179 acc: 47.96 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 53.02 %
Average loss on the 10000 test images: 1.095
[2,   100] loss: 1.137 acc: 51.09 time: 2.09
[2,   200] loss: 1.120 acc: 51.66 time: 1.90
[2,   300] loss: 1.105 acc: 52.24 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 55.88 %
Average loss on the 10000 test images: 1.044
[3,   100] loss: 1.069 acc: 54.50 time: 2.08
[3,   200] loss: 1.042 acc: 55.88 time: 1.89
[3,   300] loss: 1.037 acc: 55.30 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 56.94 %
Average loss on the 10000 test images: 1.028
[4,   100] loss: 1.016 acc: 56.91 time: 2.06
[4,   200] loss: 1.002 acc: 57.56 time: 1.87
[4,   300] loss: 0.978 acc: 58.66 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 60.20 %
Average loss on the 10000 test images: 0.946
[5,   100] loss: 0.961 acc: 59.16 time: 2.06
[5,   200] loss: 0.948 acc: 60.73 time: 1.90
[5,   300] loss: 0.944 acc: 59.98 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 61.35 %
Average loss on the 10000 test images: 0.915
[6,   100] loss: 0.918 acc: 61.98 time: 2.10
[6,   200] loss: 0.940 acc: 60.55 time: 1.91
[6,   300] loss: 0.923 acc: 61.59 time: 1.92
TESTING:
Accuracy of the network on the 10000 test images: 61.57 %
Average loss on the 10000 test images: 0.919
[7,   100] loss: 0.911 acc: 62.29 time: 2.09
[7,   200] loss: 0.904 acc: 62.74 time: 1.89
[7,   300] loss: 0.884 acc: 62.96 time: 1.94
TESTING:
Accuracy of the network on the 10000 test images: 64.40 %
Average loss on the 10000 test images: 0.845
[8,   100] loss: 0.871 acc: 63.55 time: 2.10
[8,   200] loss: 0.875 acc: 63.45 time: 1.90
[8,   300] loss: 0.857 acc: 64.45 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 64.51 %
Average loss on the 10000 test images: 0.903
[9,   100] loss: 0.858 acc: 64.79 time: 2.08
[9,   200] loss: 0.854 acc: 64.76 time: 1.90
[9,   300] loss: 0.834 acc: 65.39 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 65.95 %
Average loss on the 10000 test images: 0.821
[10,   100] loss: 0.833 acc: 66.18 time: 2.08
[10,   200] loss: 0.838 acc: 65.33 time: 1.90
[10,   300] loss: 0.835 acc: 65.35 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 67.26 %
Average loss on the 10000 test images: 0.806
```

```
[11,   100] loss: 0.815 acc: 66.74 time: 2.08
[11,   200] loss: 0.810 acc: 66.48 time: 1.92
[11,   300] loss: 0.803 acc: 67.16 time: 1.92
TESTING:
Accuracy of the network on the 10000 test images: 68.64 %
Average loss on the 10000 test images: 0.774
[12,   100] loss: 0.805 acc: 67.14 time: 2.08
[12,   200] loss: 0.815 acc: 66.99 time: 1.90
[12,   300] loss: 0.797 acc: 66.95 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 69.35 %
Average loss on the 10000 test images: 0.759
[13,   100] loss: 0.776 acc: 68.44 time: 2.07
[13,   200] loss: 0.793 acc: 68.09 time: 1.89
[13,   300] loss: 0.780 acc: 68.62 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 68.58 %
Average loss on the 10000 test images: 0.774
[14,   100] loss: 0.778 acc: 69.06 time: 2.07
[14,   200] loss: 0.771 acc: 68.90 time: 1.91
[14,   300] loss: 0.770 acc: 68.91 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 71.11 %
Average loss on the 10000 test images: 0.721
[15,   100] loss: 0.770 acc: 69.00 time: 2.07
[15,   200] loss: 0.749 acc: 69.99 time: 1.90
[15,   300] loss: 0.747 acc: 69.71 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 71.63 %
Average loss on the 10000 test images: 0.715
[16,   100] loss: 0.705 acc: 71.93 time: 2.08
[16,   200] loss: 0.686 acc: 72.87 time: 1.94
[16,   300] loss: 0.686 acc: 72.73 time: 1.93
TESTING:
Accuracy of the network on the 10000 test images: 74.08 %
Average loss on the 10000 test images: 0.648
[17,   100] loss: 0.672 acc: 73.55 time: 2.10
[17,   200] loss: 0.666 acc: 73.70 time: 1.89
[17,   300] loss: 0.663 acc: 73.66 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 74.77 %
Average loss on the 10000 test images: 0.630
[18,   100] loss: 0.660 acc: 73.88 time: 2.07
[18,   200] loss: 0.664 acc: 73.48 time: 1.90
[18,   300] loss: 0.664 acc: 73.41 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 75.21 %
Average loss on the 10000 test images: 0.626
[19,   100] loss: 0.643 acc: 74.36 time: 2.05
[19,   200] loss: 0.640 acc: 74.38 time: 1.89
[19,   300] loss: 0.657 acc: 74.24 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 75.16 %
Average loss on the 10000 test images: 0.619
[20,   100] loss: 0.652 acc: 73.96 time: 2.08
[20,   200] loss: 0.640 acc: 74.52 time: 1.92
[20,   300] loss: 0.641 acc: 74.89 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 75.44 %
Average loss on the 10000 test images: 0.619
```

```
[21,   100] loss: 0.630 acc: 75.03 time: 2.11
[21,   200] loss: 0.638 acc: 74.73 time: 1.91
[21,   300] loss: 0.638 acc: 74.73 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 75.70 %
Average loss on the 10000 test images: 0.609
[22,   100] loss: 0.629 acc: 75.02 time: 2.08
[22,   200] loss: 0.633 acc: 74.81 time: 1.88
[22,   300] loss: 0.623 acc: 75.38 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 75.90 %
Average loss on the 10000 test images: 0.602
[23,   100] loss: 0.635 acc: 74.68 time: 2.07
[23,   200] loss: 0.620 acc: 75.30 time: 1.89
[23,   300] loss: 0.624 acc: 75.21 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 75.98 %
Average loss on the 10000 test images: 0.603
[24,   100] loss: 0.610 acc: 76.07 time: 2.07
[24,   200] loss: 0.614 acc: 75.76 time: 1.91
[24,   300] loss: 0.625 acc: 75.52 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 75.70 %
Average loss on the 10000 test images: 0.606
[25,   100] loss: 0.608 acc: 75.80 time: 2.08
[25,   200] loss: 0.626 acc: 75.05 time: 1.88
[25,   300] loss: 0.617 acc: 75.84 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 76.48 %
Average loss on the 10000 test images: 0.595
[26,   100] loss: 0.616 acc: 75.56 time: 2.07
[26,   200] loss: 0.603 acc: 75.98 time: 1.89
[26,   300] loss: 0.612 acc: 75.81 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 76.72 %
Average loss on the 10000 test images: 0.588
[27,   100] loss: 0.599 acc: 76.78 time: 2.08
[27,   200] loss: 0.624 acc: 75.06 time: 1.91
[27,   300] loss: 0.602 acc: 76.05 time: 1.92
TESTING:
Accuracy of the network on the 10000 test images: 77.04 %
Average loss on the 10000 test images: 0.586
[28,   100] loss: 0.608 acc: 75.98 time: 2.07
[28,   200] loss: 0.606 acc: 75.95 time: 1.90
[28,   300] loss: 0.595 acc: 76.57 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 76.34 %
Average loss on the 10000 test images: 0.603
[29,   100] loss: 0.595 acc: 76.48 time: 2.08
[29,   200] loss: 0.607 acc: 76.20 time: 1.91
[29,   300] loss: 0.593 acc: 76.77 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 77.22 %
Average loss on the 10000 test images: 0.580
[30,   100] loss: 0.583 acc: 77.04 time: 2.07
[30,   200] loss: 0.598 acc: 76.36 time: 1.89
[30,   300] loss: 0.593 acc: 76.57 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 77.37 %
Average loss on the 10000 test images: 0.579
```

```
[31,   100] loss: 0.582 acc: 77.38 time: 2.07
[31,   200] loss: 0.581 acc: 77.09 time: 1.90
[31,   300] loss: 0.583 acc: 77.06 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 77.63 %
Average loss on the 10000 test images: 0.572
[32,   100] loss: 0.583 acc: 76.95 time: 2.06
[32,   200] loss: 0.580 acc: 76.74 time: 1.90
[32,   300] loss: 0.580 acc: 77.01 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 77.44 %
Average loss on the 10000 test images: 0.565
[33,   100] loss: 0.595 acc: 76.14 time: 2.08
[33,   200] loss: 0.571 acc: 77.41 time: 1.91
[33,   300] loss: 0.578 acc: 77.46 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 77.41 %
Average loss on the 10000 test images: 0.569
[34,   100] loss: 0.579 acc: 76.99 time: 2.08
[34,   200] loss: 0.587 acc: 77.04 time: 1.90
[34,   300] loss: 0.572 acc: 77.76 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 77.73 %
Average loss on the 10000 test images: 0.573
[35,   100] loss: 0.580 acc: 77.19 time: 2.08
[35,   200] loss: 0.574 acc: 76.96 time: 1.90
[35,   300] loss: 0.580 acc: 77.53 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 77.35 %
Average loss on the 10000 test images: 0.570
[36,   100] loss: 0.584 acc: 76.80 time: 2.08
[36,   200] loss: 0.574 acc: 77.36 time: 1.90
[36,   300] loss: 0.574 acc: 77.41 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 77.89 %
Average loss on the 10000 test images: 0.561
[37,   100] loss: 0.577 acc: 77.59 time: 2.07
[37,   200] loss: 0.583 acc: 77.00 time: 1.89
[37,   300] loss: 0.580 acc: 77.45 time: 1.92
TESTING:
Accuracy of the network on the 10000 test images: 77.84 %
Average loss on the 10000 test images: 0.567
[38,   100] loss: 0.580 acc: 77.34 time: 2.11
[38,   200] loss: 0.561 acc: 77.88 time: 1.90
[38,   300] loss: 0.567 acc: 77.84 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 77.62 %
Average loss on the 10000 test images: 0.567
[39,   100] loss: 0.572 acc: 77.11 time: 2.07
[39,   200] loss: 0.572 acc: 77.94 time: 1.90
[39,   300] loss: 0.571 acc: 77.94 time: 1.93
TESTING:
Accuracy of the network on the 10000 test images: 77.95 %
Average loss on the 10000 test images: 0.565
[40,   100] loss: 0.577 acc: 77.45 time: 2.07
[40,   200] loss: 0.574 acc: 77.54 time: 1.90
[40,   300] loss: 0.571 acc: 77.66 time: 1.92
TESTING:
Accuracy of the network on the 10000 test images: 77.56 %
Average loss on the 10000 test images: 0.567
```

```
[41,   100] loss: 0.569 acc: 77.49 time: 2.07
[41,   200] loss: 0.577 acc: 77.66 time: 1.92
[41,   300] loss: 0.579 acc: 77.52 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 77.80 %
Average loss on the 10000 test images: 0.566
[42,   100] loss: 0.586 acc: 76.59 time: 2.07
[42,   200] loss: 0.562 acc: 78.31 time: 1.89
[42,   300] loss: 0.583 acc: 77.25 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 77.57 %
Average loss on the 10000 test images: 0.571
[43,   100] loss: 0.569 acc: 77.86 time: 2.07
[43,   200] loss: 0.569 acc: 77.62 time: 1.92
[43,   300] loss: 0.579 acc: 77.32 time: 1.93
TESTING:
Accuracy of the network on the 10000 test images: 77.64 %
Average loss on the 10000 test images: 0.565
[44,   100] loss: 0.573 acc: 77.21 time: 2.07
[44,   200] loss: 0.568 acc: 77.51 time: 1.90
[44,   300] loss: 0.579 acc: 76.98 time: 1.92
TESTING:
Accuracy of the network on the 10000 test images: 77.87 %
Average loss on the 10000 test images: 0.568
[45,   100] loss: 0.563 acc: 77.87 time: 2.08
[45,   200] loss: 0.572 acc: 77.20 time: 1.89
[45,   300] loss: 0.565 acc: 77.84 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 77.92 %
Average loss on the 10000 test images: 0.565
Finished Training
```

Fine-tuning on the pre-trained model

In this section, we will load the pre-trained ResNet18 model and fine-tune on the classification task. We will freeze all previous layers except for the 'layer4' block and 'fc' layer.

In [45]:

```python
import torch.nn as nn
import torch.nn.functional as F

from torchvision.models import resnet18

# TODO: Load the pre-trained ResNet18 model

pre_trained_model_path = 'ResNet18_model.pth'
net = resnet18(num_classes=4, pretrained=False)
net = net.to(device)
state_dict = torch.load(pre_trained_model_path)
net.load_state_dict(state_dict)
net.eval()

num_ftrs = net.fc.in_features
net.fc = nn.Linear(num_ftrs, len(classes))
```

In [47]:

```python
# TODO: Freeze all previous layers; only keep the 'layer4' block and 'fc' layer traina

# Freeze all previous layers
```

```
for name, param in net.named_parameters():
    if 'layer4' in name or 'fc' in name:
        param.requires_grad = True
    else:
        param.requires_grad = False
```

In [48]:

```
# Print all the trainable parameters
params_to_update = net.parameters()
print("Params to learn:")
params_to_update = []
for name,param in net.named_parameters():
    if param.requires_grad == True:
        params_to_update.append(param)
        print("\t",name)
```

```
Params to learn:
         layer4.0.conv1.weight
         layer4.0.bn1.weight
         layer4.0.bn1.bias
         layer4.0.conv2.weight
         layer4.0.bn2.weight
         layer4.0.bn2.bias
         layer4.0.downsample.0.weight
         layer4.0.downsample.1.weight
         layer4.0.downsample.1.bias
         layer4.1.conv1.weight
         layer4.1.bn1.weight
         layer4.1.bn1.bias
         layer4.1.conv2.weight
         layer4.1.bn2.weight
         layer4.1.bn2.bias
         fc.weight
         fc.bias
```

In [49]:

```
# TODO: Define criterion and optimizer
# Note that your optimizer only needs to update the parameters that are trainable.
lr = 0.0001
momentum = 0.9
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(params_to_update, lr=lr, momentum=momentum)
```

In [50]:

```
train(net, criterion, optimizer, num_epochs=20, decay_epochs=5, init_lr=0.01, task='cl
[1,   100] loss: 1.522 acc: 46.86 time: 2.00
[1,   200] loss: 1.302 acc: 54.20 time: 1.86
[1,   300] loss: 1.240 acc: 56.32 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 61.01 %
Average loss on the 10000 test images: 1.105
[2,   100] loss: 1.157 acc: 59.09 time: 2.08
[2,   200] loss: 1.166 acc: 58.95 time: 1.87
[2,   300] loss: 1.127 acc: 59.50 time: 1.85
TESTING:
Accuracy of the network on the 10000 test images: 63.95 %
Average loss on the 10000 test images: 1.017
[3,   100] loss: 1.080 acc: 61.07 time: 2.01
[3,   200] loss: 1.123 acc: 60.73 time: 1.84
[3,   300] loss: 1.084 acc: 61.77 time: 1.85
TESTING:
Accuracy of the network on the 10000 test images: 63.51 %
Average loss on the 10000 test images: 1.035
[4,   100] loss: 1.032 acc: 63.07 time: 2.03
[4,   200] loss: 1.048 acc: 62.91 time: 1.88
[4,   300] loss: 1.042 acc: 62.57 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 65.07 %
Average loss on the 10000 test images: 0.993
[5,   100] loss: 0.992 acc: 64.88 time: 2.03
[5,   200] loss: 0.998 acc: 64.58 time: 1.87
[5,   300] loss: 1.008 acc: 64.45 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 63.04 %
Average loss on the 10000 test images: 1.037
[6,   100] loss: 0.940 acc: 66.37 time: 2.04
[6,   200] loss: 0.901 acc: 67.58 time: 1.87
[6,   300] loss: 0.931 acc: 66.70 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 67.97 %
Average loss on the 10000 test images: 0.900
[7,   100] loss: 0.905 acc: 67.98 time: 2.04
[7,   200] loss: 0.917 acc: 67.10 time: 1.88
[7,   300] loss: 0.916 acc: 67.33 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 67.95 %
Average loss on the 10000 test images: 0.896
[8,   100] loss: 0.897 acc: 68.06 time: 2.02
[8,   200] loss: 0.899 acc: 68.21 time: 1.88
[8,   300] loss: 0.901 acc: 67.55 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 68.61 %
Average loss on the 10000 test images: 0.887
[9,   100] loss: 0.883 acc: 68.17 time: 2.03
[9,   200] loss: 0.888 acc: 68.01 time: 1.87
[9,   300] loss: 0.897 acc: 67.71 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 68.42 %
Average loss on the 10000 test images: 0.892
[10,   100] loss: 0.893 acc: 68.26 time: 2.03
[10,   200] loss: 0.886 acc: 68.66 time: 1.89
[10,   300] loss: 0.884 acc: 68.09 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 68.70 %
Average loss on the 10000 test images: 0.880
```

```
[11,   100] loss: 0.879 acc: 68.69 time: 2.03
[11,   200] loss: 0.877 acc: 68.70 time: 1.87
[11,   300] loss: 0.864 acc: 69.40 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 68.73 %
Average loss on the 10000 test images: 0.879
[12,   100] loss: 0.882 acc: 68.11 time: 2.02
[12,   200] loss: 0.871 acc: 68.69 time: 1.87
[12,   300] loss: 0.860 acc: 68.77 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 68.79 %
Average loss on the 10000 test images: 0.877
[13,   100] loss: 0.891 acc: 67.92 time: 2.04
[13,   200] loss: 0.862 acc: 69.22 time: 1.87
[13,   300] loss: 0.859 acc: 68.94 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 68.67 %
Average loss on the 10000 test images: 0.879
[14,   100] loss: 0.869 acc: 69.05 time: 2.03
[14,   200] loss: 0.873 acc: 68.73 time: 1.88
[14,   300] loss: 0.876 acc: 68.69 time: 1.85
TESTING:
Accuracy of the network on the 10000 test images: 69.12 %
Average loss on the 10000 test images: 0.880
[15,   100] loss: 0.862 acc: 69.23 time: 2.03
[15,   200] loss: 0.880 acc: 68.44 time: 1.88
[15,   300] loss: 0.873 acc: 68.91 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 68.80 %
Average loss on the 10000 test images: 0.875
[16,   100] loss: 0.857 acc: 69.41 time: 2.04
[16,   200] loss: 0.878 acc: 68.44 time: 1.89
[16,   300] loss: 0.858 acc: 68.74 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 68.88 %
Average loss on the 10000 test images: 0.876
[17,   100] loss: 0.867 acc: 69.27 time: 2.04
[17,   200] loss: 0.873 acc: 68.56 time: 1.89
[17,   300] loss: 0.867 acc: 68.87 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 68.92 %
Average loss on the 10000 test images: 0.876
[18,   100] loss: 0.888 acc: 67.91 time: 2.04
[18,   200] loss: 0.866 acc: 68.94 time: 1.87
[18,   300] loss: 0.863 acc: 68.93 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 69.00 %
Average loss on the 10000 test images: 0.877
[19,   100] loss: 0.870 acc: 68.73 time: 2.04
[19,   200] loss: 0.879 acc: 68.59 time: 1.90
[19,   300] loss: 0.875 acc: 68.51 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 68.93 %
Average loss on the 10000 test images: 0.876
[20,   100] loss: 0.868 acc: 68.98 time: 2.02
[20,   200] loss: 0.866 acc: 68.92 time: 1.88
[20,   300] loss: 0.866 acc: 69.06 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 68.96 %
```

```
Average loss on the 10000 test images: 0.876
Finished Training
```

Fine-tuning on the randomly initialized model

In this section, we will randomly initialize a ResNet18 model and fine-tune on the classification task. We will freeze all previous layers except for the 'layer4' block and 'fc' layer.

In [51]:

```
import torch.nn as nn
import torch.nn.functional as F

from torchvision.models import resnet18

# TODO: Randomly initialize a ResNet18 model

net = resnet18(num_classes=10, pretrained=False)
net = net.to(device)
```

In [52]:

```
# TODO: Freeze all previous layers; only keep the 'layer4' block and 'fc' layer traina
# To do this, you should set requires_grad=False for the frozen layers.

# Freeze all previous layers
for name, param in net.named_parameters():
    if 'layer4' in name or 'fc' in name:
        param.requires_grad = True
    else:
        param.requires_grad = False
```

In [53]:

```
# Print all the trainable parameters
params_to_update = net.parameters()
print("Params to learn:")
params_to_update = []
for name,param in net.named_parameters():
    if param.requires_grad == True:
        params_to_update.append(param)
        print("\t",name)
```

```
Params to learn:
         layer4.0.conv1.weight
         layer4.0.bn1.weight
         layer4.0.bn1.bias
         layer4.0.conv2.weight
         layer4.0.bn2.weight
         layer4.0.bn2.bias
         layer4.0.downsample.0.weight
         layer4.0.downsample.1.weight
         layer4.0.downsample.1.bias
         layer4.1.conv1.weight
         layer4.1.bn1.weight
         layer4.1.bn1.bias
         layer4.1.conv2.weight
         layer4.1.bn2.weight
         layer4.1.bn2.bias
         fc.weight
         fc.bias
```

In [54]:

```
# TODO: Define criterion and optimizer
# Note that your optimizer only needs to update the parameters that are trainable.

lr = 0.01
momentum = 0.9
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(params_to_update, lr=lr, momentum=momentum)
```

In [41]:

```
train(net, criterion, optimizer, num_epochs=40, decay_epochs=10, init_lr=0.01, task='
[1,   100] loss: 2.154 acc: 23.48 time: 2.04
[1,   200] loss: 1.966 acc: 28.63 time: 1.89
[1,   300] loss: 1.930 acc: 30.09 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 34.80 %
Average loss on the 10000 test images: 1.816
[2,   100] loss: 1.867 acc: 32.33 time: 2.05
[2,   200] loss: 1.848 acc: 32.58 time: 1.87
[2,   300] loss: 1.822 acc: 34.61 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 37.14 %
Average loss on the 10000 test images: 1.736
[3,   100] loss: 1.805 acc: 34.88 time: 2.03
[3,   200] loss: 1.789 acc: 35.70 time: 1.86
[3,   300] loss: 1.812 acc: 34.80 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 39.36 %
Average loss on the 10000 test images: 1.680
[4,   100] loss: 1.761 acc: 36.51 time: 2.04
[4,   200] loss: 1.779 acc: 35.72 time: 1.90
[4,   300] loss: 1.766 acc: 36.39 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 39.61 %
Average loss on the 10000 test images: 1.665
[5,   100] loss: 1.756 acc: 36.80 time: 2.01
[5,   200] loss: 1.732 acc: 37.87 time: 1.85
[5,   300] loss: 1.725 acc: 37.23 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 40.80 %
Average loss on the 10000 test images: 1.636
[6,   100] loss: 1.734 acc: 36.97 time: 2.05
[6,   200] loss: 1.728 acc: 38.09 time: 1.87
[6,   300] loss: 1.712 acc: 37.79 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 41.48 %
Average loss on the 10000 test images: 1.633
[7,   100] loss: 1.698 acc: 38.93 time: 2.02
[7,   200] loss: 1.717 acc: 38.07 time: 1.88
[7,   300] loss: 1.707 acc: 38.38 time: 2.04
TESTING:
Accuracy of the network on the 10000 test images: 40.46 %
Average loss on the 10000 test images: 1.671
[8,   100] loss: 1.689 acc: 39.30 time: 2.03
[8,   200] loss: 1.700 acc: 39.05 time: 1.88
[8,   300] loss: 1.680 acc: 38.95 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 41.96 %
Average loss on the 10000 test images: 1.613
[9,   100] loss: 1.670 acc: 39.31 time: 2.03
[9,   200] loss: 1.669 acc: 40.12 time: 1.87
[9,   300] loss: 1.677 acc: 40.23 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 40.20 %
Average loss on the 10000 test images: 1.691
[10,   100] loss: 1.681 acc: 39.91 time: 2.06
[10,   200] loss: 1.677 acc: 40.22 time: 1.89
[10,   300] loss: 1.653 acc: 40.27 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 42.20 %
Average loss on the 10000 test images: 1.599
```

```
[11,   100] loss: 1.607 acc: 42.59 time: 2.05
[11,   200] loss: 1.607 acc: 42.52 time: 1.88
[11,   300] loss: 1.596 acc: 42.55 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 44.18 %
Average loss on the 10000 test images: 1.551
[12,   100] loss: 1.590 acc: 42.86 time: 2.03
[12,   200] loss: 1.592 acc: 42.67 time: 1.88
[12,   300] loss: 1.586 acc: 43.02 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 44.19 %
Average loss on the 10000 test images: 1.544
[13,   100] loss: 1.588 acc: 43.24 time: 2.02
[13,   200] loss: 1.582 acc: 42.66 time: 1.86
[13,   300] loss: 1.576 acc: 43.36 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 44.22 %
Average loss on the 10000 test images: 1.543
[14,   100] loss: 1.586 acc: 43.12 time: 2.03
[14,   200] loss: 1.575 acc: 43.44 time: 1.87
[14,   300] loss: 1.588 acc: 42.91 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 44.63 %
Average loss on the 10000 test images: 1.536
[15,   100] loss: 1.584 acc: 43.20 time: 2.06
[15,   200] loss: 1.585 acc: 43.16 time: 1.87
[15,   300] loss: 1.577 acc: 43.42 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 44.35 %
Average loss on the 10000 test images: 1.537
[16,   100] loss: 1.568 acc: 43.93 time: 2.03
[16,   200] loss: 1.583 acc: 42.95 time: 1.89
[16,   300] loss: 1.588 acc: 43.09 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 44.57 %
Average loss on the 10000 test images: 1.534
[17,   100] loss: 1.576 acc: 43.32 time: 2.02
[17,   200] loss: 1.566 acc: 43.90 time: 1.87
[17,   300] loss: 1.573 acc: 43.34 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 44.40 %
Average loss on the 10000 test images: 1.534
[18,   100] loss: 1.561 acc: 44.16 time: 2.03
[18,   200] loss: 1.577 acc: 43.59 time: 1.87
[18,   300] loss: 1.571 acc: 44.00 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 44.91 %
Average loss on the 10000 test images: 1.530
[19,   100] loss: 1.569 acc: 43.59 time: 2.05
[19,   200] loss: 1.552 acc: 44.05 time: 1.88
[19,   300] loss: 1.576 acc: 43.63 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 44.93 %
Average loss on the 10000 test images: 1.523
[20,   100] loss: 1.561 acc: 43.63 time: 2.02
[20,   200] loss: 1.566 acc: 43.73 time: 1.86
[20,   300] loss: 1.564 acc: 43.66 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 44.58 %
Average loss on the 10000 test images: 1.526
```

```
[21,   100] loss: 1.564 acc: 43.88 time: 2.03
[21,   200] loss: 1.561 acc: 44.16 time: 1.88
[21,   300] loss: 1.553 acc: 44.79 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 44.86 %
Average loss on the 10000 test images: 1.523
[22,   100] loss: 1.570 acc: 43.29 time: 2.02
[22,   200] loss: 1.564 acc: 43.81 time: 1.86
[22,   300] loss: 1.551 acc: 44.76 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 44.82 %
Average loss on the 10000 test images: 1.523
[23,   100] loss: 1.552 acc: 44.51 time: 2.03
[23,   200] loss: 1.554 acc: 44.38 time: 1.86
[23,   300] loss: 1.557 acc: 44.61 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 44.89 %
Average loss on the 10000 test images: 1.522
[24,   100] loss: 1.565 acc: 44.16 time: 2.03
[24,   200] loss: 1.547 acc: 44.77 time: 1.87
[24,   300] loss: 1.563 acc: 44.20 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 44.88 %
Average loss on the 10000 test images: 1.522
[25,   100] loss: 1.556 acc: 44.26 time: 2.03
[25,   200] loss: 1.560 acc: 44.45 time: 1.86
[25,   300] loss: 1.565 acc: 43.84 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 45.03 %
Average loss on the 10000 test images: 1.522
[26,   100] loss: 1.548 acc: 44.73 time: 2.02
[26,   200] loss: 1.558 acc: 44.62 time: 1.87
[26,   300] loss: 1.566 acc: 43.86 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 44.98 %
Average loss on the 10000 test images: 1.522
[27,   100] loss: 1.562 acc: 44.29 time: 2.05
[27,   200] loss: 1.550 acc: 44.28 time: 1.89
[27,   300] loss: 1.552 acc: 44.66 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 44.73 %
Average loss on the 10000 test images: 1.521
[28,   100] loss: 1.558 acc: 44.42 time: 2.04
[28,   200] loss: 1.562 acc: 43.80 time: 1.86
[28,   300] loss: 1.549 acc: 44.55 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 45.12 %
Average loss on the 10000 test images: 1.518
[29,   100] loss: 1.545 acc: 44.82 time: 2.02
[29,   200] loss: 1.557 acc: 44.58 time: 1.88
[29,   300] loss: 1.571 acc: 43.50 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 45.05 %
Average loss on the 10000 test images: 1.522
[30,   100] loss: 1.569 acc: 43.60 time: 2.03
[30,   200] loss: 1.551 acc: 44.27 time: 1.86
[30,   300] loss: 1.559 acc: 43.93 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 45.32 %
Average loss on the 10000 test images: 1.518
```

```
[31,   100] loss: 1.568 acc: 44.26 time: 2.03
[31,   200] loss: 1.563 acc: 43.93 time: 1.86
[31,   300] loss: 1.560 acc: 44.30 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 45.14 %
Average loss on the 10000 test images: 1.520
[32,   100] loss: 1.544 acc: 44.39 time: 2.01
[32,   200] loss: 1.563 acc: 43.34 time: 1.87
[32,   300] loss: 1.547 acc: 44.46 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 45.10 %
Average loss on the 10000 test images: 1.519
[33,   100] loss: 1.549 acc: 44.76 time: 2.03
[33,   200] loss: 1.550 acc: 44.66 time: 1.89
[33,   300] loss: 1.549 acc: 44.42 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 45.11 %
Average loss on the 10000 test images: 1.522
[34,   100] loss: 1.545 acc: 45.06 time: 2.04
[34,   200] loss: 1.551 acc: 44.54 time: 1.86
[34,   300] loss: 1.563 acc: 43.60 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 45.05 %
Average loss on the 10000 test images: 1.519
[35,   100] loss: 1.552 acc: 44.09 time: 2.03
[35,   200] loss: 1.550 acc: 44.18 time: 1.88
[35,   300] loss: 1.550 acc: 44.44 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 44.98 %
Average loss on the 10000 test images: 1.521
[36,   100] loss: 1.549 acc: 44.55 time: 2.05
[36,   200] loss: 1.549 acc: 44.48 time: 1.88
[36,   300] loss: 1.558 acc: 43.88 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 45.16 %
Average loss on the 10000 test images: 1.520
[37,   100] loss: 1.546 acc: 44.26 time: 2.00
[37,   200] loss: 1.563 acc: 44.78 time: 1.85
[37,   300] loss: 1.550 acc: 44.54 time: 1.85
TESTING:
Accuracy of the network on the 10000 test images: 45.04 %
Average loss on the 10000 test images: 1.519
[38,   100] loss: 1.550 acc: 44.13 time: 2.02
[38,   200] loss: 1.556 acc: 44.45 time: 1.87
[38,   300] loss: 1.565 acc: 43.69 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 44.95 %
Average loss on the 10000 test images: 1.520
[39,   100] loss: 1.564 acc: 44.52 time: 2.03
[39,   200] loss: 1.551 acc: 44.37 time: 1.88
[39,   300] loss: 1.551 acc: 44.82 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 45.14 %
Average loss on the 10000 test images: 1.518
[40,   100] loss: 1.545 acc: 44.29 time: 2.03
[40,   200] loss: 1.544 acc: 44.52 time: 1.86
[40,   300] loss: 1.560 acc: 44.23 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 45.12 %
```

```
Average loss on the 10000 test images: 1.521
Finished Training
```

Supervised training on the pre-trained model

In this section, we will load the pre-trained ResNet18 model and re-train the whole model on the classification task.

In [15]:

```
import torch.nn as nn
import torch.nn.functional as F

from torchvision.models import resnet18

# TODO: Load the pre-trained ResNet18 model

pre_trained_model_path = 'ResNet18_model.pth'
net = resnet18(num_classes=4, pretrained=False)
net = net.to(device)
state_dict = torch.load(pre_trained_model_path)
net.load_state_dict(state_dict)

num_ftrs = net.fc.in_features
net.fc = nn.Linear(num_ftrs, len(classes))
```

In [16]:

```
# TODO: Define criterion and optimizer

lr = 0.01
momentum = 0.9
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(net.parameters(), lr=lr, momentum=momentum)
```

In [17]:

```
train(net, criterion, optimizer, num_epochs=60, decay_epochs=20, init_lr=0.01, task='
[1,   100] loss: 1.419 acc: 50.20 time: 2.08
[1,   200] loss: 1.189 acc: 58.62 time: 1.89
[1,   300] loss: 1.079 acc: 62.54 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 65.65 %
Average loss on the 10000 test images: 0.985
[2,   100] loss: 0.969 acc: 65.82 time: 2.06
[2,   200] loss: 0.948 acc: 66.51 time: 1.90
[2,   300] loss: 0.933 acc: 67.12 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 69.88 %
Average loss on the 10000 test images: 0.876
[3,   100] loss: 0.853 acc: 69.84 time: 2.09
[3,   200] loss: 0.858 acc: 69.96 time: 1.91
[3,   300] loss: 0.841 acc: 70.23 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 72.50 %
Average loss on the 10000 test images: 0.812
[4,   100] loss: 0.801 acc: 71.64 time: 2.08
[4,   200] loss: 0.777 acc: 72.48 time: 1.94
[4,   300] loss: 0.774 acc: 72.73 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 73.82 %
Average loss on the 10000 test images: 0.776
[5,   100] loss: 0.730 acc: 74.79 time: 2.11
[5,   200] loss: 0.740 acc: 73.81 time: 1.90
[5,   300] loss: 0.733 acc: 74.35 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 74.48 %
Average loss on the 10000 test images: 0.735
[6,   100] loss: 0.700 acc: 75.61 time: 2.08
[6,   200] loss: 0.711 acc: 75.16 time: 1.90
[6,   300] loss: 0.681 acc: 76.16 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 75.95 %
Average loss on the 10000 test images: 0.690
[7,   100] loss: 0.682 acc: 76.17 time: 2.08
[7,   200] loss: 0.673 acc: 76.21 time: 1.91
[7,   300] loss: 0.656 acc: 76.84 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 77.11 %
Average loss on the 10000 test images: 0.682
[8,   100] loss: 0.633 acc: 77.52 time: 2.08
[8,   200] loss: 0.651 acc: 77.21 time: 1.91
[8,   300] loss: 0.627 acc: 77.78 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 77.74 %
Average loss on the 10000 test images: 0.651
[9,   100] loss: 0.600 acc: 79.21 time: 2.12
[9,   200] loss: 0.597 acc: 79.21 time: 1.92
[9,   300] loss: 0.615 acc: 78.39 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 77.61 %
Average loss on the 10000 test images: 0.666
[10,   100] loss: 0.579 acc: 79.33 time: 2.07
[10,   200] loss: 0.582 acc: 79.62 time: 1.90
[10,   300] loss: 0.589 acc: 79.62 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 78.46 %
Average loss on the 10000 test images: 0.621
```

```
[11,   100] loss: 0.561 acc: 80.09 time: 2.08
[11,   200] loss: 0.570 acc: 80.02 time: 1.91
[11,   300] loss: 0.556 acc: 80.56 time: 1.94
TESTING:
Accuracy of the network on the 10000 test images: 79.08 %
Average loss on the 10000 test images: 0.632
[12,   100] loss: 0.557 acc: 79.95 time: 2.08
[12,   200] loss: 0.554 acc: 80.27 time: 1.90
[12,   300] loss: 0.549 acc: 80.73 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 79.32 %
Average loss on the 10000 test images: 0.606
[13,   100] loss: 0.530 acc: 81.41 time: 2.08
[13,   200] loss: 0.529 acc: 81.57 time: 1.92
[13,   300] loss: 0.535 acc: 81.22 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 80.06 %
Average loss on the 10000 test images: 0.593
[14,   100] loss: 0.516 acc: 81.62 time: 2.06
[14,   200] loss: 0.520 acc: 81.66 time: 1.92
[14,   300] loss: 0.503 acc: 82.38 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 79.09 %
Average loss on the 10000 test images: 0.629
[15,   100] loss: 0.500 acc: 82.42 time: 2.10
[15,   200] loss: 0.494 acc: 82.55 time: 1.91
[15,   300] loss: 0.497 acc: 82.22 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 80.22 %
Average loss on the 10000 test images: 0.593
[16,   100] loss: 0.473 acc: 83.59 time: 2.09
[16,   200] loss: 0.484 acc: 82.81 time: 1.91
[16,   300] loss: 0.495 acc: 82.77 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 79.77 %
Average loss on the 10000 test images: 0.596
[17,   100] loss: 0.462 acc: 83.69 time: 2.09
[17,   200] loss: 0.465 acc: 83.21 time: 1.90
[17,   300] loss: 0.474 acc: 83.09 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 79.95 %
Average loss on the 10000 test images: 0.602
[18,   100] loss: 0.465 acc: 83.45 time: 2.09
[18,   200] loss: 0.456 acc: 83.88 time: 1.90
[18,   300] loss: 0.448 acc: 84.15 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 80.29 %
Average loss on the 10000 test images: 0.589
[19,   100] loss: 0.437 acc: 84.34 time: 2.09
[19,   200] loss: 0.430 acc: 85.03 time: 1.88
[19,   300] loss: 0.448 acc: 84.38 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 80.15 %
Average loss on the 10000 test images: 0.597
[20,   100] loss: 0.428 acc: 85.14 time: 2.07
[20,   200] loss: 0.444 acc: 83.95 time: 1.90
[20,   300] loss: 0.430 acc: 84.85 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 81.33 %
Average loss on the 10000 test images: 0.569
```

```
[21,   100] loss: 0.379 acc: 86.76 time: 2.08
[21,   200] loss: 0.369 acc: 87.00 time: 1.91
[21,   300] loss: 0.360 acc: 87.38 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 83.29 %
Average loss on the 10000 test images: 0.503
[22,   100] loss: 0.344 acc: 87.65 time: 2.08
[22,   200] loss: 0.342 acc: 88.19 time: 1.90
[22,   300] loss: 0.326 acc: 88.64 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 83.24 %
Average loss on the 10000 test images: 0.509
[23,   100] loss: 0.324 acc: 88.91 time: 2.06
[23,   200] loss: 0.327 acc: 88.64 time: 1.91
[23,   300] loss: 0.334 acc: 88.19 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 83.51 %
Average loss on the 10000 test images: 0.509
[24,   100] loss: 0.319 acc: 88.75 time: 2.09
[24,   200] loss: 0.331 acc: 88.31 time: 1.90
[24,   300] loss: 0.314 acc: 88.77 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.53 %
Average loss on the 10000 test images: 0.511
[25,   100] loss: 0.320 acc: 88.63 time: 2.07
[25,   200] loss: 0.304 acc: 89.29 time: 1.89
[25,   300] loss: 0.322 acc: 88.58 time: 1.93
TESTING:
Accuracy of the network on the 10000 test images: 83.62 %
Average loss on the 10000 test images: 0.516
[26,   100] loss: 0.300 acc: 89.30 time: 2.08
[26,   200] loss: 0.310 acc: 88.75 time: 1.92
[26,   300] loss: 0.304 acc: 89.36 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 83.56 %
Average loss on the 10000 test images: 0.515
[27,   100] loss: 0.301 acc: 88.98 time: 2.07
[27,   200] loss: 0.314 acc: 89.05 time: 1.90
[27,   300] loss: 0.300 acc: 89.21 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.74 %
Average loss on the 10000 test images: 0.521
[28,   100] loss: 0.296 acc: 89.68 time: 2.06
[28,   200] loss: 0.310 acc: 89.16 time: 1.89
[28,   300] loss: 0.298 acc: 89.41 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.90 %
Average loss on the 10000 test images: 0.518
[29,   100] loss: 0.288 acc: 89.88 time: 2.08
[29,   200] loss: 0.301 acc: 89.53 time: 1.90
[29,   300] loss: 0.298 acc: 89.45 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 83.69 %
Average loss on the 10000 test images: 0.517
[30,   100] loss: 0.291 acc: 89.77 time: 2.13
[30,   200] loss: 0.288 acc: 89.47 time: 1.89
[30,   300] loss: 0.300 acc: 89.50 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.46 %
Average loss on the 10000 test images: 0.520
```

```
[31,   100] loss: 0.280 acc: 89.85 time: 2.07
[31,   200] loss: 0.295 acc: 89.28 time: 1.90
[31,   300] loss: 0.287 acc: 89.98 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.80 %
Average loss on the 10000 test images: 0.522
[32,   100] loss: 0.288 acc: 89.70 time: 2.07
[32,   200] loss: 0.291 acc: 89.80 time: 1.90
[32,   300] loss: 0.283 acc: 89.89 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.63 %
Average loss on the 10000 test images: 0.523
[33,   100] loss: 0.279 acc: 90.19 time: 2.06
[33,   200] loss: 0.284 acc: 89.90 time: 1.91
[33,   300] loss: 0.280 acc: 90.23 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 83.86 %
Average loss on the 10000 test images: 0.520
[34,   100] loss: 0.285 acc: 90.09 time: 2.08
[34,   200] loss: 0.280 acc: 90.00 time: 1.93
[34,   300] loss: 0.288 acc: 89.84 time: 1.93
TESTING:
Accuracy of the network on the 10000 test images: 83.57 %
Average loss on the 10000 test images: 0.523
[35,   100] loss: 0.277 acc: 90.27 time: 2.07
[35,   200] loss: 0.279 acc: 90.14 time: 1.90
[35,   300] loss: 0.274 acc: 90.33 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.63 %
Average loss on the 10000 test images: 0.529
[36,   100] loss: 0.262 acc: 90.91 time: 2.09
[36,   200] loss: 0.271 acc: 90.33 time: 1.89
[36,   300] loss: 0.277 acc: 89.97 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 83.86 %
Average loss on the 10000 test images: 0.525
[37,   100] loss: 0.262 acc: 90.80 time: 2.08
[37,   200] loss: 0.274 acc: 90.27 time: 1.90
[37,   300] loss: 0.282 acc: 89.91 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 83.67 %
Average loss on the 10000 test images: 0.527
[38,   100] loss: 0.261 acc: 90.71 time: 2.08
[38,   200] loss: 0.267 acc: 90.55 time: 1.90
[38,   300] loss: 0.263 acc: 90.53 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.99 %
Average loss on the 10000 test images: 0.522
[39,   100] loss: 0.263 acc: 90.35 time: 2.07
[39,   200] loss: 0.271 acc: 90.16 time: 1.88
[39,   300] loss: 0.278 acc: 90.29 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.83 %
Average loss on the 10000 test images: 0.533
[40,   100] loss: 0.254 acc: 91.03 time: 2.07
[40,   200] loss: 0.259 acc: 90.59 time: 1.90
[40,   300] loss: 0.262 acc: 90.37 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.86 %
Average loss on the 10000 test images: 0.531
```

```
[41,   100] loss: 0.255 acc: 91.15 time: 2.07
[41,   200] loss: 0.251 acc: 90.91 time: 1.90
[41,   300] loss: 0.255 acc: 90.91 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 83.90 %
Average loss on the 10000 test images: 0.528
[42,   100] loss: 0.246 acc: 91.16 time: 2.07
[42,   200] loss: 0.260 acc: 90.73 time: 1.91
[42,   300] loss: 0.253 acc: 91.08 time: 1.92
TESTING:
Accuracy of the network on the 10000 test images: 83.93 %
Average loss on the 10000 test images: 0.524
[43,   100] loss: 0.256 acc: 90.97 time: 2.04
[43,   200] loss: 0.247 acc: 91.16 time: 1.88
[43,   300] loss: 0.250 acc: 91.19 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 84.05 %
Average loss on the 10000 test images: 0.527
[44,   100] loss: 0.241 acc: 91.45 time: 2.07
[44,   200] loss: 0.242 acc: 91.59 time: 1.89
[44,   300] loss: 0.256 acc: 91.01 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 84.04 %
Average loss on the 10000 test images: 0.527
[45,   100] loss: 0.236 acc: 91.52 time: 2.09
[45,   200] loss: 0.248 acc: 91.12 time: 1.92
[45,   300] loss: 0.252 acc: 91.09 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 84.02 %
Average loss on the 10000 test images: 0.527
[46,   100] loss: 0.233 acc: 92.05 time: 2.07
[46,   200] loss: 0.248 acc: 91.48 time: 1.89
[46,   300] loss: 0.255 acc: 90.86 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 84.07 %
Average loss on the 10000 test images: 0.529
[47,   100] loss: 0.244 acc: 91.62 time: 2.05
[47,   200] loss: 0.252 acc: 90.99 time: 1.88
[47,   300] loss: 0.246 acc: 91.28 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 84.00 %
Average loss on the 10000 test images: 0.530
[48,   100] loss: 0.237 acc: 91.47 time: 2.08
[48,   200] loss: 0.242 acc: 91.54 time: 1.90
[48,   300] loss: 0.250 acc: 91.20 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 84.03 %
Average loss on the 10000 test images: 0.531
[49,   100] loss: 0.253 acc: 90.95 time: 2.05
[49,   200] loss: 0.245 acc: 91.06 time: 1.87
[49,   300] loss: 0.248 acc: 91.38 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 83.96 %
Average loss on the 10000 test images: 0.529
[50,   100] loss: 0.245 acc: 91.19 time: 2.06
[50,   200] loss: 0.249 acc: 91.45 time: 1.91
[50,   300] loss: 0.239 acc: 91.64 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 83.82 %
Average loss on the 10000 test images: 0.530
```

```
[51,   100] loss: 0.246 acc: 91.58 time: 2.07
[51,   200] loss: 0.249 acc: 91.12 time: 1.91
[51,   300] loss: 0.248 acc: 91.31 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 84.00 %
Average loss on the 10000 test images: 0.532
[52,   100] loss: 0.247 acc: 91.29 time: 2.07
[52,   200] loss: 0.247 acc: 90.95 time: 1.91
[52,   300] loss: 0.244 acc: 91.27 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 83.95 %
Average loss on the 10000 test images: 0.531
[53,   100] loss: 0.239 acc: 91.66 time: 2.14
[53,   200] loss: 0.249 acc: 91.28 time: 1.91
[53,   300] loss: 0.246 acc: 91.30 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.90 %
Average loss on the 10000 test images: 0.533
[54,   100] loss: 0.235 acc: 91.85 time: 2.08
[54,   200] loss: 0.244 acc: 91.45 time: 1.89
[54,   300] loss: 0.242 acc: 91.23 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 84.06 %
Average loss on the 10000 test images: 0.529
[55,   100] loss: 0.244 acc: 91.27 time: 2.08
[55,   200] loss: 0.239 acc: 91.84 time: 1.92
[55,   300] loss: 0.249 acc: 91.28 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 84.09 %
Average loss on the 10000 test images: 0.530
[56,   100] loss: 0.242 acc: 91.31 time: 2.07
[56,   200] loss: 0.239 acc: 91.36 time: 1.90
[56,   300] loss: 0.243 acc: 91.66 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.86 %
Average loss on the 10000 test images: 0.530
[57,   100] loss: 0.243 acc: 91.48 time: 2.08
[57,   200] loss: 0.248 acc: 91.13 time: 1.89
[57,   300] loss: 0.232 acc: 92.05 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 83.77 %
Average loss on the 10000 test images: 0.533
[58,   100] loss: 0.242 acc: 91.57 time: 2.10
[58,   200] loss: 0.242 acc: 91.45 time: 1.92
[58,   300] loss: 0.243 acc: 91.45 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 83.93 %
Average loss on the 10000 test images: 0.533
[59,   100] loss: 0.246 acc: 91.11 time: 2.08
[59,   200] loss: 0.240 acc: 91.53 time: 1.90
[59,   300] loss: 0.240 acc: 91.38 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 84.05 %
Average loss on the 10000 test images: 0.533
[60,   100] loss: 0.254 acc: 91.06 time: 2.07
[60,   200] loss: 0.251 acc: 91.12 time: 1.90
[60,   300] loss: 0.236 acc: 91.47 time: 1.92
TESTING:
Accuracy of the network on the 10000 test images: 83.94 %
```

```
Average loss on the 10000 test images: 0.530
Finished Training
```

Supervised training on the randomly initialized model

In this section, we will randomly initialize a ResNet18 model and re-train the whole model on the classification task.

In [64]:

```
import torch.nn as nn
import torch.nn.functional as F

from torchvision.models import resnet18

# TODO: Randomly initialize a ResNet18 model

net = resnet18(num_classes=10, pretrained=False)
net = net.to(device)
```

In [65]:

```
# TODO: Define criterion and optimizer

lr = 0.001
momentum = 0.9
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(net.parameters(), lr=lr, momentum=momentum)
```

In [66]:

```
train(net, criterion, optimizer, num_epochs=20, decay_epochs=10, init_lr=0.01, task='
[1,   100] loss: 2.053 acc: 26.84 time: 2.05
[1,   200] loss: 1.699 acc: 37.98 time: 1.88
[1,   300] loss: 1.591 acc: 41.96 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 47.75 %
Average loss on the 10000 test images: 1.444
[2,   100] loss: 1.430 acc: 47.80 time: 2.05
[2,   200] loss: 1.384 acc: 49.88 time: 1.88
[2,   300] loss: 1.332 acc: 52.10 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 55.49 %
Average loss on the 10000 test images: 1.226
[3,   100] loss: 1.237 acc: 55.75 time: 2.05
[3,   200] loss: 1.181 acc: 57.93 time: 1.87
[3,   300] loss: 1.157 acc: 59.18 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 62.08 %
Average loss on the 10000 test images: 1.068
[4,   100] loss: 1.077 acc: 61.70 time: 2.05
[4,   200] loss: 1.054 acc: 62.62 time: 1.87
[4,   300] loss: 1.057 acc: 62.28 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 65.64 %
Average loss on the 10000 test images: 0.963
[5,   100] loss: 0.999 acc: 64.55 time: 2.06
[5,   200] loss: 0.960 acc: 65.87 time: 1.88
[5,   300] loss: 0.987 acc: 64.98 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 68.57 %
Average loss on the 10000 test images: 0.921
[6,   100] loss: 0.922 acc: 67.55 time: 2.06
[6,   200] loss: 0.926 acc: 66.88 time: 1.91
[6,   300] loss: 0.905 acc: 68.28 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 65.82 %
Average loss on the 10000 test images: 0.976
[7,   100] loss: 0.860 acc: 70.02 time: 2.09
[7,   200] loss: 0.864 acc: 69.59 time: 1.87
[7,   300] loss: 0.857 acc: 69.91 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 70.73 %
Average loss on the 10000 test images: 0.827
[8,   100] loss: 0.815 acc: 71.29 time: 2.05
[8,   200] loss: 0.813 acc: 70.88 time: 1.87
[8,   300] loss: 0.818 acc: 71.52 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 72.11 %
Average loss on the 10000 test images: 0.806
[9,   100] loss: 0.777 acc: 72.59 time: 2.04
[9,   200] loss: 0.763 acc: 73.13 time: 1.86
[9,   300] loss: 0.781 acc: 72.27 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 73.34 %
Average loss on the 10000 test images: 0.780
[10,   100] loss: 0.740 acc: 73.62 time: 2.07
[10,   200] loss: 0.745 acc: 73.48 time: 1.88
[10,   300] loss: 0.746 acc: 73.37 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 73.99 %
Average loss on the 10000 test images: 0.768
```

```
[11,   100] loss: 0.659 acc: 77.00 time: 2.07
[11,   200] loss: 0.631 acc: 77.93 time: 1.89
[11,   300] loss: 0.635 acc: 77.13 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 77.07 %
Average loss on the 10000 test images: 0.657
[12,   100] loss: 0.610 acc: 78.55 time: 2.05
[12,   200] loss: 0.613 acc: 77.99 time: 1.88
[12,   300] loss: 0.595 acc: 78.84 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 77.32 %
Average loss on the 10000 test images: 0.648
[13,   100] loss: 0.605 acc: 78.58 time: 2.05
[13,   200] loss: 0.597 acc: 79.23 time: 1.87
[13,   300] loss: 0.603 acc: 78.77 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 77.50 %
Average loss on the 10000 test images: 0.644
[14,   100] loss: 0.589 acc: 79.38 time: 2.06
[14,   200] loss: 0.589 acc: 79.14 time: 1.88
[14,   300] loss: 0.583 acc: 79.74 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 77.60 %
Average loss on the 10000 test images: 0.640
[15,   100] loss: 0.573 acc: 79.60 time: 2.04
[15,   200] loss: 0.580 acc: 79.34 time: 1.87
[15,   300] loss: 0.580 acc: 79.47 time: 1.86
TESTING:
Accuracy of the network on the 10000 test images: 77.76 %
Average loss on the 10000 test images: 0.639
[16,   100] loss: 0.559 acc: 80.38 time: 2.04
[16,   200] loss: 0.573 acc: 79.67 time: 1.87
[16,   300] loss: 0.579 acc: 79.80 time: 1.87
TESTING:
Accuracy of the network on the 10000 test images: 77.86 %
Average loss on the 10000 test images: 0.636
[17,   100] loss: 0.566 acc: 79.84 time: 2.06
[17,   200] loss: 0.564 acc: 79.76 time: 1.88
[17,   300] loss: 0.562 acc: 80.20 time: 1.89
TESTING:
Accuracy of the network on the 10000 test images: 78.11 %
Average loss on the 10000 test images: 0.628
[18,   100] loss: 0.550 acc: 80.88 time: 2.08
[18,   200] loss: 0.554 acc: 80.65 time: 1.90
[18,   300] loss: 0.566 acc: 79.70 time: 1.90
TESTING:
Accuracy of the network on the 10000 test images: 78.43 %
Average loss on the 10000 test images: 0.625
[19,   100] loss: 0.563 acc: 80.16 time: 2.05
[19,   200] loss: 0.558 acc: 80.05 time: 1.88
[19,   300] loss: 0.544 acc: 81.01 time: 1.91
TESTING:
Accuracy of the network on the 10000 test images: 78.39 %
Average loss on the 10000 test images: 0.621
[20,   100] loss: 0.554 acc: 80.46 time: 2.04
[20,   200] loss: 0.559 acc: 80.22 time: 1.88
[20,   300] loss: 0.531 acc: 81.64 time: 1.88
TESTING:
Accuracy of the network on the 10000 test images: 78.57 %
```

```
Average loss on the 10000 test images: 0.616
Finished Training
```