main

May 21, 2021

[1]:
```
import torch
import torchvision
import torch.nn as nn
import torchvision.transforms as transforms
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
torch.manual_seed(4)
np.random.seed(4)
```

[2]:
```
BATCH_SIZE = 128
NUM_ITERS = int(2e4)
CRITERION = nn.CrossEntropyLoss()
```

[37]:
```
transform = transforms.Compose(
    [transforms.ToTensor(),
     transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])


trainset = torchvision.datasets.CIFAR10(root='./data', train=True,
                                        download=True, transform=transform)


testset = torchvision.datasets.CIFAR10(root='./data', train=False,
                                       download=True,  transform=transform)


train_loader = torch.utils.data.DataLoader(trainset, batch_size=BATCH_SIZE,
                                        shuffle=True, num_workers=2)

test_loader = torch.utils.data.DataLoader(testset, batch_size=BATCH_SIZE,
                                        shuffle=False, num_workers=2)

EPOCHS = int(NUM_ITERS / (len(trainset) / BATCH_SIZE))

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

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

print(DEVICE)
```

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

[3]:

[4]:
```
# Block

class ConvBlock(nn.Module):

    def __init__(self, stride=1, padding=1, batch_norm=False):
        super().__init__()
        self.batch_norm = batch_norm

        self.conv1 = nn.Conv2d(in_channels=8, out_channels=8, kernel_size=3,␣
 ↪stride=stride, padding=padding)

        if batch_norm:
            self.bn1 = nn.BatchNorm2d(8)

        self.relu1 = nn.ReLU()

        self.conv2 = nn.Conv2d(in_channels=8, out_channels=8, kernel_size=3,␣
 ↪stride=stride, padding=padding)

        if batch_norm:
            self.bn2 = nn.BatchNorm2d(8)

        self.relu2 = nn.ReLU()




    def forward(self, x):

        out = self.conv1(x)

        if self.batch_norm:
            out = self.bn1(out)

        out = self.relu1(out)
```

```
        out = self.conv2(x)

        if self.batch_norm:
            out = self.bn2(out)

        out = self.relu2(out)

        return out

# Model
class CNNModel(nn.Module):
    def __init__(self, batch_norm=False, N=10):

        super().__init__()

        self.conv1 = nn.Conv2d(in_channels=3, out_channels=8, kernel_size=5,␣
↪stride=1, padding=0)
        # Size: 28 x 28

        self.conv2 = nn.Conv2d(in_channels=8, out_channels=8, kernel_size=5,␣
↪stride=2, padding=0)
        # Size: 12 x 12

        # Dynamic block num
        self.blocks = nn.Sequential(*[ConvBlock(batch_norm=batch_norm) for _ in␣
↪range(N)])
        # Size: 12 x 12

        self.conv3 = nn.Conv2d(in_channels=8, out_channels=4, kernel_size=3,␣
↪stride=2, padding=0)
        # Size: 5 X 5

        self.fc1 = nn.Linear(5 * 5 * 4, 100, bias=True)
        self.relu1 = nn.ReLU()
        self.fc2 = nn.Linear(100, 10, bias=False)


    def forward(self, x):

        out = self.conv1(x)
        out = self.conv2(out)

        out = self.blocks(out)          # How is the shape the same after this???
        out = self.conv3(out)
        out = out.view(out.size(0), -1)
        out = self.fc1(out)
        out = self.relu1(out)
```

```
        out = self.fc2(out)

        return out
```

[5]:
```
def calc_accuracy(model, train=False): # add train param to calculate accuracy␣
↪on both train and test
    # Calculate Accuracy
    correct = 0
    total = 0

    d_loader = train_loader if train else test_loader
    # Iterate through test dataset
    for images, labels in d_loader:
        # Load images
        images, labels = images.to(DEVICE), labels.to(DEVICE)

        # Forward pass only to get logits/output
        outputs = model(images)

        # Get predictions from the maximum value
        _, predicted = torch.max(outputs.data, 1)

        # Total number of labels
        total += labels.size(0)

        # Total correct predictions
        correct += (predicted == labels).sum()

    return 100 * correct / total

def train(model):
    print(f'Training for {EPOCHS} epochs')
    optimizer = torch.optim.SGD(model.parameters(), lr=.01)
    accuracy = {'train': [], 'test': []}

    for epoch in range(EPOCHS):
        for i, (images, labels) in enumerate(train_loader):
            # This will load batch_size amount of samples
            images, labels = images.requires_grad_().to(DEVICE), labels.
↪to(DEVICE)

            # Clear gradients w.r.t. parameters
            optimizer.zero_grad()

            # Forward pass to get output/logits
            outputs = model(images)
```

```
            # Calculate Loss: softmax --> cross entropy loss
            loss = CRITERION(outputs, labels)

            # Getting gradients w.r.t. parameters
            loss.backward()

            # Updating parameters
            optimizer.step()

        train_accuracy = calc_accuracy(model, train=True) # abstract accuracy␣
↪function away
        test_accuracy = calc_accuracy(model) # abstract accuracy function away
        # Print Loss
        print('Epoch: {} Loss: {}. Train Accuracy: {}, Test Accuracy: {}'.
↪format(epoch, loss.item(), train_accuracy, test_accuracy))

        accuracy['train'].append(train_accuracy.item())
        accuracy['test'].append(test_accuracy.item())

    return pd.DataFrame(accuracy)
```

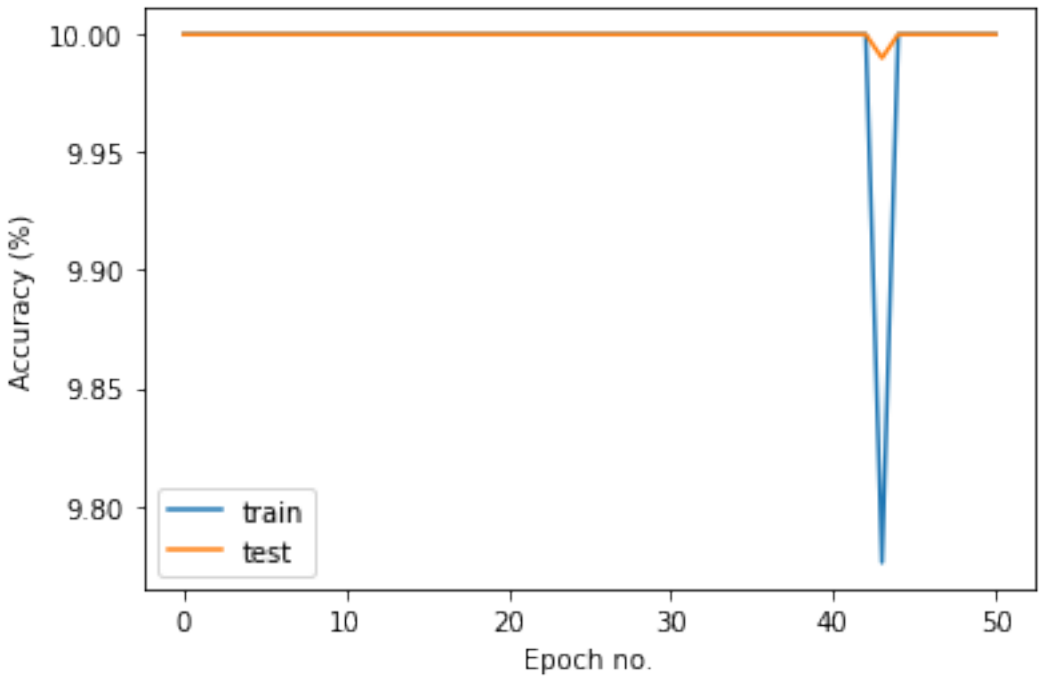
0.1 a) Plot Train and Test accuracy

[6]:
```
no_batch_norm = CNNModel().to(DEVICE)
nbn_history = train(no_batch_norm)
nbn_history.plot(y=['train', 'test'], use_index=True, xlabel='Epoch no.',␣
↪ylabel='Accuracy (%)')
```

```
Training for 51 epochs
Epoch: 0 Loss: 2.3027138710021973. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 1 Loss: 2.3021628856658936. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 2 Loss: 2.302372455596924. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 3 Loss: 2.3020453453063965. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 4 Loss: 2.302574634552002. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 5 Loss: 2.30250883102417. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 6 Loss: 2.3025784492492676. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 7 Loss: 2.3028244972229004. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 8 Loss: 2.3024027347564697. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 9 Loss: 2.3030056953430176. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 10 Loss: 2.3030290603637695. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 11 Loss: 2.3028054237365723. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 12 Loss: 2.302643299102783. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 13 Loss: 2.3026976585388184. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 14 Loss: 2.3025741577148438. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 15 Loss: 2.3025989532470703. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 16 Loss: 2.302600383758545. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 17 Loss: 2.302710771560669. Train Accuracy: 10.0, Test Accuracy: 10.0
```

```
Epoch: 18 Loss: 2.302713394165039. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 19 Loss: 2.3025708198547363. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 20 Loss: 2.302690029144287. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 21 Loss: 2.3026790618896484. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 22 Loss: 2.3026633262634277. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 23 Loss: 2.302595615386963. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 24 Loss: 2.3026137351989746. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 25 Loss: 2.302618980407715. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 26 Loss: 2.3026578426361084. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 27 Loss: 2.3026373386383057. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 28 Loss: 2.302682876586914. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 29 Loss: 2.302621364593506. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 30 Loss: 2.3026227951049805. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 31 Loss: 2.302603006362915. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 32 Loss: 2.3025662899017334. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 33 Loss: 2.3026013374328613. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 34 Loss: 2.302619218826294. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 35 Loss: 2.3025660514831543. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 36 Loss: 2.3026490211486816. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 37 Loss: 2.302577257156372. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 38 Loss: 2.302602529525757. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 39 Loss: 2.302590847015381. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 40 Loss: 2.3026604652404785. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 41 Loss: 2.302598714828491. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 42 Loss: 2.3026556968688965. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 43 Loss: 2.3026108741760254. Train Accuracy: 9.776000022888184, Test
Accuracy: 9.989999771118164
Epoch: 44 Loss: 2.3026251792907715. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 45 Loss: 2.3025920391082764. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 46 Loss: 2.3026022911071777. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 47 Loss: 2.302647113800049. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 48 Loss: 2.3026561737060547. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 49 Loss: 2.302588939666748. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 50 Loss: 2.302631378173828. Train Accuracy: 10.0, Test Accuracy: 10.0
```

[6]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f24680b0c90>`



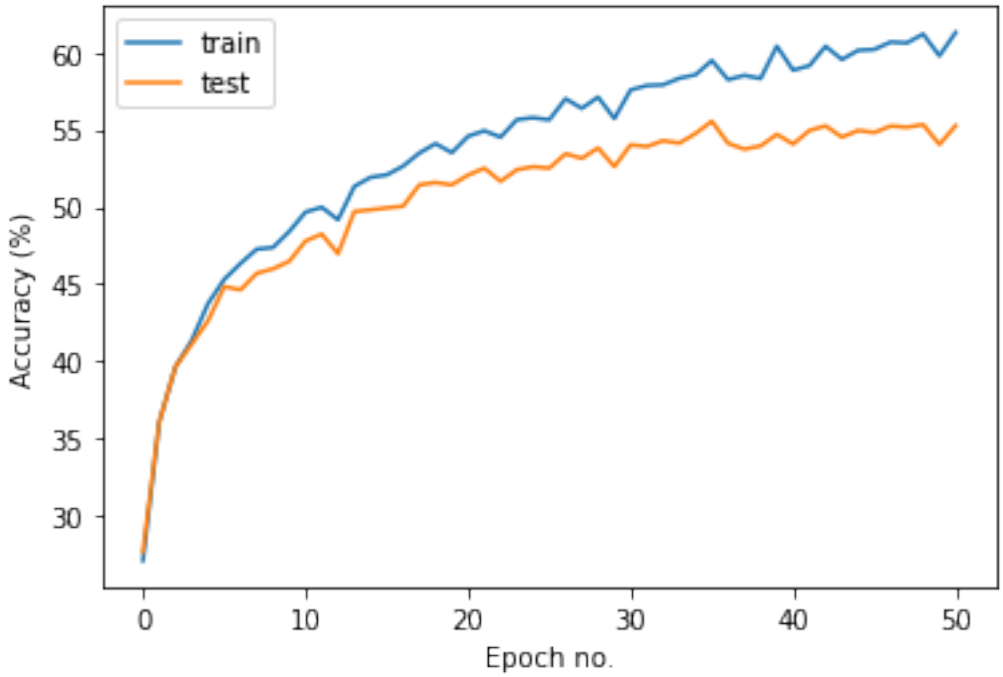
[7]:
```
batch_norm = CNNModel(batch_norm=True).to(DEVICE)
bn_history = train(batch_norm)
bn_history.plot(y=['train', 'test'], use_index=True, xlabel='Epoch no.',␣
↪ylabel='Accuracy (%)')
```

```
Training for 51 epochs
Epoch: 0 Loss: 1.940789818763733. Train Accuracy: 27.08599853515625, Test
Accuracy: 27.649999618530273
Epoch: 1 Loss: 1.7568286657333374. Train Accuracy: 36.04199981689453, Test
Accuracy: 36.05999755859375
Epoch: 2 Loss: 1.5260580778121948. Train Accuracy: 39.6359977722168, Test
Accuracy: 39.63999938964844
Epoch: 3 Loss: 1.596287488937378. Train Accuracy: 41.343997955322266, Test
Accuracy: 41.11000061035156
Epoch: 4 Loss: 1.6181385517120361. Train Accuracy: 43.72599792480469, Test
Accuracy: 42.61000061035156
Epoch: 5 Loss: 1.3493764400482178. Train Accuracy: 45.301998138427734, Test
Accuracy: 44.82999801635742
Epoch: 6 Loss: 1.4732204675674438. Train Accuracy: 46.32600021362305, Test
Accuracy: 44.619998931884766
Epoch: 7 Loss: 1.6574041843414307. Train Accuracy: 47.25600051879883, Test
Accuracy: 45.689998626708984
Epoch: 8 Loss: 1.3595701456069946. Train Accuracy: 47.36000061035156, Test
Accuracy: 45.97999954223633
Epoch: 9 Loss: 1.2848687171936035. Train Accuracy: 48.39799880981445, Test
```

```
Accuracy: 46.46999740600586
Epoch: 10 Loss: 1.3703778982162476. Train Accuracy: 49.65599822998047, Test
Accuracy: 47.779998779296875
Epoch: 11 Loss: 1.521883249282837. Train Accuracy: 49.96999740600586, Test
Accuracy: 48.22999954223633
Epoch: 12 Loss: 1.2616653442382812. Train Accuracy: 49.157997131347656, Test
Accuracy: 46.94999694824219
Epoch: 13 Loss: 1.5325905084609985. Train Accuracy: 51.31399917602539, Test
Accuracy: 49.689998626708984
Epoch: 14 Loss: 1.4466838836669922. Train Accuracy: 51.90599822998047, Test
Accuracy: 49.79999923706055
Epoch: 15 Loss: 1.315706729888916. Train Accuracy: 52.06399917602539, Test
Accuracy: 49.93000030517578
Epoch: 16 Loss: 1.27448570728302. Train Accuracy: 52.62999725341797, Test
Accuracy: 50.029998779296875
Epoch: 17 Loss: 1.0878854990005493. Train Accuracy: 53.47599792480469, Test
Accuracy: 51.41999816894531
Epoch: 18 Loss: 1.3482965230941772. Train Accuracy: 54.09000015258789, Test
Accuracy: 51.56999969482422
Epoch: 19 Loss: 1.2141932249069214. Train Accuracy: 53.503997802734375, Test
Accuracy: 51.41999816894531
Epoch: 20 Loss: 1.277710199356079. Train Accuracy: 54.56599807739258, Test
Accuracy: 52.04999923706055
Epoch: 21 Loss: 0.9568778276443481. Train Accuracy: 54.9219970703125, Test
Accuracy: 52.5
Epoch: 22 Loss: 1.4196288585662842. Train Accuracy: 54.50600051879883, Test
Accuracy: 51.64999771118164
Epoch: 23 Loss: 1.3153622150421143. Train Accuracy: 55.641998291015625, Test
Accuracy: 52.39999771118164
Epoch: 24 Loss: 1.203505039215088. Train Accuracy: 55.76799774169922, Test
Accuracy: 52.59000015258789
Epoch: 25 Loss: 1.3626664876937866. Train Accuracy: 55.63399887084961, Test
Accuracy: 52.5
Epoch: 26 Loss: 1.3062270879745483. Train Accuracy: 56.99799728393555, Test
Accuracy: 53.43000030517578
Epoch: 27 Loss: 1.0552632808685303. Train Accuracy: 56.37199783325195, Test
Accuracy: 53.13999938964844
Epoch: 28 Loss: 1.0693984031677246. Train Accuracy: 57.104000091552734, Test
Accuracy: 53.80999755859375
Epoch: 29 Loss: 1.367674469947815. Train Accuracy: 55.71799850463867, Test
Accuracy: 52.599998474121094
Epoch: 30 Loss: 0.9478651285171509. Train Accuracy: 57.56599807739258, Test
Accuracy: 54.0099983215332
Epoch: 31 Loss: 1.0071887969970703. Train Accuracy: 57.854000091552734, Test
Accuracy: 53.89999771118164
Epoch: 32 Loss: 1.2435133457183838. Train Accuracy: 57.89799880981445, Test
Accuracy: 54.279998779296875
Epoch: 33 Loss: 1.1926630735397339. Train Accuracy: 58.32600021362305, Test
```

```
Accuracy: 54.119998931884766
Epoch: 34 Loss: 1.0727357864379883. Train Accuracy: 58.551998138427734, Test
Accuracy: 54.75
Epoch: 35 Loss: 1.0114580392837524. Train Accuracy: 59.46999740600586, Test
Accuracy: 55.53999710083008
Epoch: 36 Loss: 1.2681186199188232. Train Accuracy: 58.231998443603516, Test
Accuracy: 54.1099967956543
Epoch: 37 Loss: 1.0123722553253174. Train Accuracy: 58.5099983215332, Test
Accuracy: 53.72999954223633
Epoch: 38 Loss: 1.2464014291763306. Train Accuracy: 58.305999755859375, Test
Accuracy: 53.939998626708984
Epoch: 39 Loss: 1.048458218574524. Train Accuracy: 60.40399932861328, Test
Accuracy: 54.689998626708984
Epoch: 40 Loss: 1.1024326086044312. Train Accuracy: 58.849998474121094, Test
Accuracy: 54.05999755859375
Epoch: 41 Loss: 1.221086025238037. Train Accuracy: 59.13399887084961, Test
Accuracy: 54.939998626708984
Epoch: 42 Loss: 1.1473438739776611. Train Accuracy: 60.39799880981445, Test
Accuracy: 55.23999786376953
Epoch: 43 Loss: 1.1321758031845093. Train Accuracy: 59.5359992980957, Test
Accuracy: 54.5099983215332
Epoch: 44 Loss: 1.1598156690597534. Train Accuracy: 60.13199996948242, Test
Accuracy: 54.939998626708984
Epoch: 45 Loss: 1.1739332675933838. Train Accuracy: 60.19199752807617, Test
Accuracy: 54.80999755859375
Epoch: 46 Loss: 1.084607481956482. Train Accuracy: 60.66999816894531, Test
Accuracy: 55.22999954223633
Epoch: 47 Loss: 1.1161339282989502. Train Accuracy: 60.61199951171875, Test
Accuracy: 55.14999771118164
Epoch: 48 Loss: 0.9278648495674133. Train Accuracy: 61.183998107910156, Test
Accuracy: 55.31999969482422
Epoch: 49 Loss: 1.2202932834625244. Train Accuracy: 59.77799987792969, Test
Accuracy: 54.03999710083008
Epoch: 50 Loss: 0.9955610036849976. Train Accuracy: 61.29399871826172, Test
Accuracy: 55.2599983215332
```

[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f24691326d0>



0.2 b) Optimal value of N w/o batch normalization

[8]:
```
nbn_models = []
nbn_histories = []
for test_n in range(2, 10):
    if test_n % 2 == 0: # if it's even
        model = CNNModel(N=test_n).to(DEVICE)
        nbn_models.append(model)
        history = train(model)
        nbn_histories.append(history)
        history.plot(y=['train', 'test'], use_index=True, xlabel='Epoch no.',␣
↪ylabel='Accuracy (%)')
        plt.title(f'Accuracy plot with {test_n} blocks')
```

```
Training for 51 epochs
Epoch: 0 Loss: 2.3076510429382324. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 1 Loss: 2.3001952171325684. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 2 Loss: 2.2752082347869873. Train Accuracy: 16.219999313354492, Test
Accuracy: 16.10999870300293
Epoch: 3 Loss: 2.0460093021392822. Train Accuracy: 22.920000076293945, Test
Accuracy: 23.76999855041504
Epoch: 4 Loss: 1.9569168090820312. Train Accuracy: 27.529998779296875, Test
Accuracy: 28.389999389648438
Epoch: 5 Loss: 2.050244092941284. Train Accuracy: 29.104000091552734, Test
```

```
Accuracy: 29.5
Epoch: 6 Loss: 1.9138145446777344. Train Accuracy: 31.229999542236328, Test
Accuracy: 32.279998779296875
Epoch: 7 Loss: 1.9316638708114624. Train Accuracy: 30.807998657226562, Test
Accuracy: 31.459999084472656
Epoch: 8 Loss: 1.7460505962371826. Train Accuracy: 33.7599983215332, Test
Accuracy: 34.040000915527344
Epoch: 9 Loss: 1.7481815814971924. Train Accuracy: 35.06599807739258, Test
Accuracy: 35.25
Epoch: 10 Loss: 1.60017991065979. Train Accuracy: 37.27399826049805, Test
Accuracy: 37.40999984741211
Epoch: 11 Loss: 1.6154870986938477. Train Accuracy: 39.051998138427734, Test
Accuracy: 39.32999801635742
Epoch: 12 Loss: 1.619463562965393. Train Accuracy: 39.88399887084961, Test
Accuracy: 39.939998626708984
Epoch: 13 Loss: 1.831345796585083. Train Accuracy: 40.959999084472656, Test
Accuracy: 41.029998779296875
Epoch: 14 Loss: 1.9257045984268188. Train Accuracy: 41.94199752807617, Test
Accuracy: 41.41999816894531
Epoch: 15 Loss: 1.6907148361206055. Train Accuracy: 43.21799850463867, Test
Accuracy: 42.619998931884766
Epoch: 16 Loss: 1.5912286043167114. Train Accuracy: 44.40599822998047, Test
Accuracy: 43.73999786376953
Epoch: 17 Loss: 1.452712059020996. Train Accuracy: 44.604000091552734, Test
Accuracy: 44.38999938964844
Epoch: 18 Loss: 1.3509966135025024. Train Accuracy: 45.736000061035156, Test
Accuracy: 45.30999755859375
Epoch: 19 Loss: 1.5109295845031738. Train Accuracy: 46.795997619628906, Test
Accuracy: 46.5
Epoch: 20 Loss: 1.5012121200561523. Train Accuracy: 46.534000396728516, Test
Accuracy: 46.27000045776367
Epoch: 21 Loss: 1.3291124105453491. Train Accuracy: 47.54399871826172, Test
Accuracy: 47.189998626708984
Epoch: 22 Loss: 1.5488126277923584. Train Accuracy: 47.53999710083008, Test
Accuracy: 46.68000030517578
Epoch: 23 Loss: 1.3671048879623413. Train Accuracy: 48.89399719238281, Test
Accuracy: 47.91999816894531
Epoch: 24 Loss: 1.4384206533432007. Train Accuracy: 50.33799743652344, Test
Accuracy: 49.369998931884766
Epoch: 25 Loss: 1.372834324836731. Train Accuracy: 50.14399719238281, Test
Accuracy: 49.13999938964844
Epoch: 26 Loss: 1.4650028944015503. Train Accuracy: 47.685997009277344, Test
Accuracy: 46.53999710083008
Epoch: 27 Loss: 1.5329041481018066. Train Accuracy: 49.641998291015625, Test
Accuracy: 48.38999938964844
Epoch: 28 Loss: 1.2675299644470215. Train Accuracy: 51.02199935913086, Test
Accuracy: 49.57999801635742
Epoch: 29 Loss: 1.3762714862823486. Train Accuracy: 51.80999755859375, Test
```

```
Accuracy: 50.16999816894531
Epoch: 30 Loss: 1.2536674737930298. Train Accuracy: 52.433998107910156, Test
Accuracy: 50.88999938964844
Epoch: 31 Loss: 1.5217714309692383. Train Accuracy: 52.29999923706055, Test
Accuracy: 50.62999725341797
Epoch: 32 Loss: 1.562849760055542. Train Accuracy: 53.3599967956543, Test
Accuracy: 51.5099983215332
Epoch: 33 Loss: 1.4181386232376099. Train Accuracy: 53.2239990234375, Test
Accuracy: 51.47999954223633
Epoch: 34 Loss: 1.1894325017929077. Train Accuracy: 52.69199752807617, Test
Accuracy: 50.689998626708984
Epoch: 35 Loss: 1.206437349319458. Train Accuracy: 53.27000045776367, Test
Accuracy: 51.3599967956543
Epoch: 36 Loss: 1.42647385597229. Train Accuracy: 54.3859977722168, Test
Accuracy: 51.94999694824219
Epoch: 37 Loss: 1.0540165901184082. Train Accuracy: 54.44199752807617, Test
Accuracy: 51.96999740600586
Epoch: 38 Loss: 1.2340093851089478. Train Accuracy: 54.39999771118164, Test
Accuracy: 52.0
Epoch: 39 Loss: 1.0536246299743652. Train Accuracy: 55.125999450683594, Test
Accuracy: 52.31999969482422
Epoch: 40 Loss: 1.2768125534057617. Train Accuracy: 54.987998962402344, Test
Accuracy: 52.18000030517578
Epoch: 41 Loss: 1.1377394199371338. Train Accuracy: 55.2239990234375, Test
Accuracy: 53.0099983215332
Epoch: 42 Loss: 1.3240002393722534. Train Accuracy: 55.76199722290039, Test
Accuracy: 53.09000015258789
Epoch: 43 Loss: 1.2034603357315063. Train Accuracy: 54.645999908447266, Test
Accuracy: 51.54999923706055
Epoch: 44 Loss: 1.2034717798233032. Train Accuracy: 55.727996826171875, Test
Accuracy: 52.98999786376953
Epoch: 45 Loss: 1.2251152992248535. Train Accuracy: 55.981998443603516, Test
Accuracy: 53.15999984741211
Epoch: 46 Loss: 1.2580918073654175. Train Accuracy: 56.402000427246094, Test
Accuracy: 53.75
Epoch: 47 Loss: 1.3445061445236206. Train Accuracy: 56.47200012207031, Test
Accuracy: 53.09000015258789
Epoch: 48 Loss: 1.277292251586914. Train Accuracy: 56.63399887084961, Test
Accuracy: 53.65999984741211
Epoch: 49 Loss: 1.3492717742919922. Train Accuracy: 56.301998138427734, Test
Accuracy: 53.37999725341797
Epoch: 50 Loss: 1.158198595046997. Train Accuracy: 57.36399841308594, Test
Accuracy: 53.77000045776367
Training for 51 epochs
Epoch: 0 Loss: 2.298962354660034. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 1 Loss: 2.303330659866333. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 2 Loss: 2.3033831119537354. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 3 Loss: 2.3001532554626465. Train Accuracy: 10.0, Test Accuracy: 10.0
```

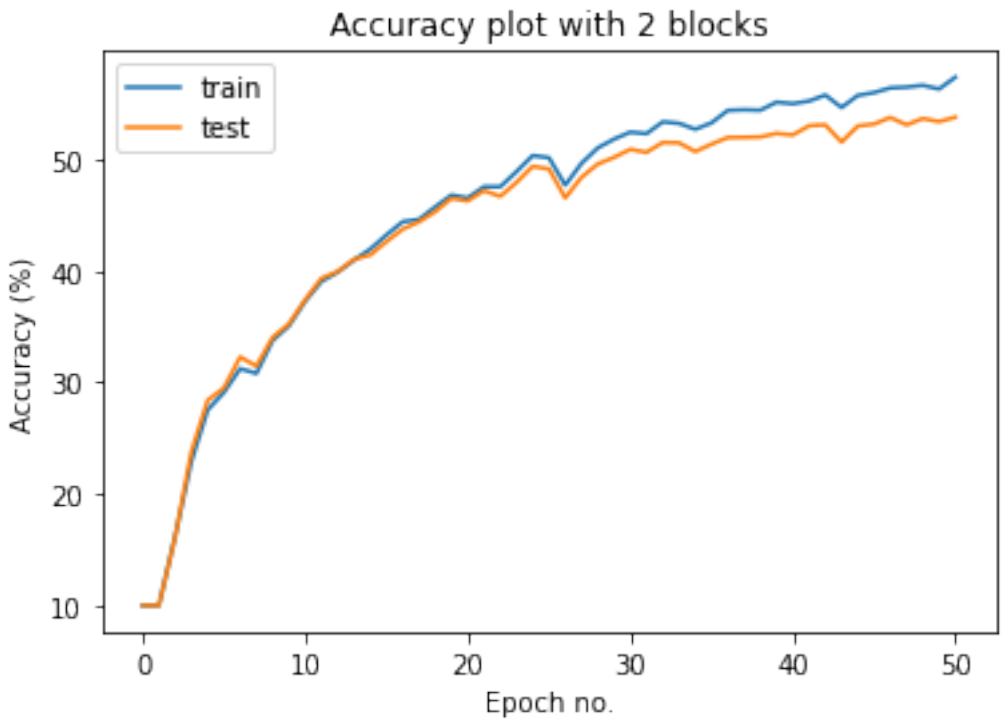
```
Epoch: 4 Loss: 2.3028221130371094. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 5 Loss: 2.3013291358947754. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 6 Loss: 2.302985668182373. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 7 Loss: 2.303194522857666. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 8 Loss: 2.3031764030456543. Train Accuracy: 10.001999855041504, Test
Accuracy: 10.0
Epoch: 9 Loss: 2.302279233932495. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 10 Loss: 2.3025498390197754. Train Accuracy: 10.053999900817871, Test
Accuracy: 10.050000190734863
Epoch: 11 Loss: 2.302375316619873. Train Accuracy: 10.01200008392334, Test
Accuracy: 10.00999927520752
Epoch: 12 Loss: 2.302955150604248. Train Accuracy: 10.52400016784668, Test
Accuracy: 10.5
Epoch: 13 Loss: 2.3020808696746826. Train Accuracy: 14.401999473571777, Test
Accuracy: 15.029999732971191
Epoch: 14 Loss: 2.302793264389038. Train Accuracy: 12.545999526977539, Test
Accuracy: 12.819999694824219
Epoch: 15 Loss: 2.3026092052459717. Train Accuracy: 12.907999992370605, Test
Accuracy: 13.069999694824219
Epoch: 16 Loss: 2.3022732734680176. Train Accuracy: 13.01200008392334, Test
Accuracy: 13.25999927520752
Epoch: 17 Loss: 2.3024356365203857. Train Accuracy: 12.187999725341797, Test
Accuracy: 11.899999618530273
Epoch: 18 Loss: 2.301940679550171. Train Accuracy: 12.541999816894531, Test
Accuracy: 12.329999923706055
Epoch: 19 Loss: 2.3017659187316895. Train Accuracy: 12.483999252319336, Test
Accuracy: 12.329999923706055
Epoch: 20 Loss: 2.3013737201690674. Train Accuracy: 12.719999313354492, Test
Accuracy: 12.59999942779541
Epoch: 21 Loss: 2.2995972633361816. Train Accuracy: 14.595999717712402, Test
Accuracy: 14.710000038146973
Epoch: 22 Loss: 2.2969956398010254. Train Accuracy: 16.095998764038086, Test
Accuracy: 16.31999969482422
Epoch: 23 Loss: 2.282621145248413. Train Accuracy: 17.187999725341797, Test
Accuracy: 17.329999923706055
Epoch: 24 Loss: 2.183666706085205. Train Accuracy: 20.195999145507812, Test
Accuracy: 20.119998931884766
Epoch: 25 Loss: 2.095083236694336. Train Accuracy: 23.137998580932617, Test
Accuracy: 23.670000076293945
Epoch: 26 Loss: 2.080845832824707. Train Accuracy: 26.197999954223633, Test
Accuracy: 26.469999313354492
Epoch: 27 Loss: 1.786321997642517. Train Accuracy: 28.191999435424805, Test
Accuracy: 28.69999885559082
Epoch: 28 Loss: 2.0489554405212402. Train Accuracy: 28.13599967956543, Test
Accuracy: 28.42999839782715
Epoch: 29 Loss: 1.9047462940216064. Train Accuracy: 29.27199935913086, Test
Accuracy: 28.809999465942383
Epoch: 30 Loss: 1.6748828887939453. Train Accuracy: 32.082000732421875, Test
```

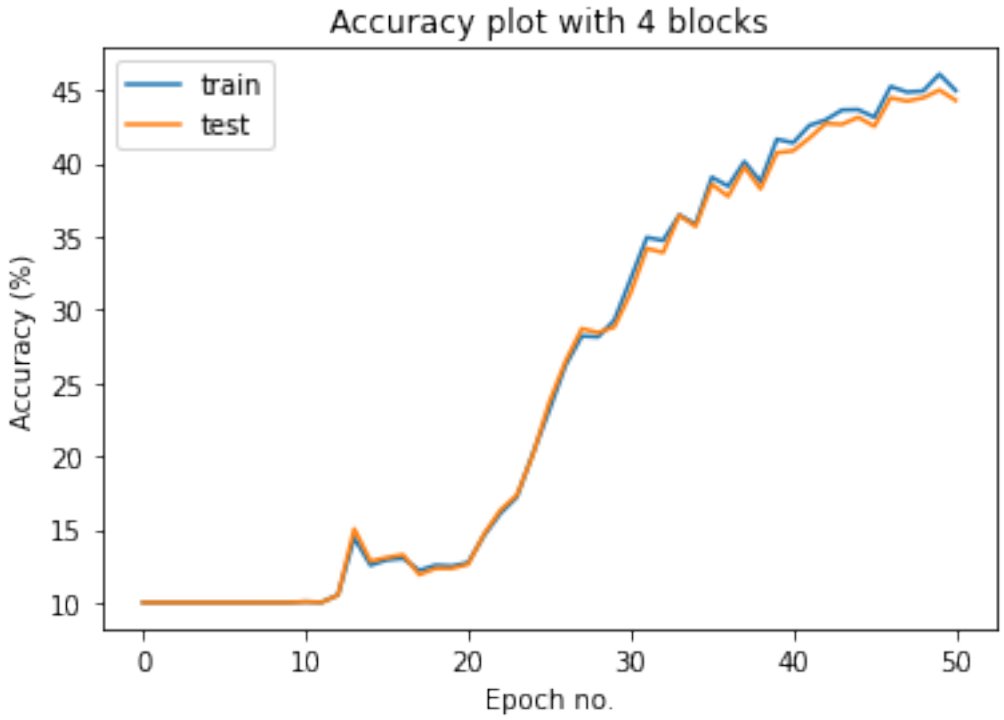
```
Accuracy: 31.139999389648438
Epoch: 31 Loss: 1.872369408607483. Train Accuracy: 34.92799758911133, Test
Accuracy: 34.16999816894531
Epoch: 32 Loss: 1.8651447296142578. Train Accuracy: 34.71999740600586, Test
Accuracy: 33.89999771118164
Epoch: 33 Loss: 1.6419813632965088. Train Accuracy: 36.461997985839844, Test
Accuracy: 36.43000030517578
Epoch: 34 Loss: 1.574902057647705. Train Accuracy: 35.827999114990234, Test
Accuracy: 35.68000030517578
Epoch: 35 Loss: 1.6901792287826538. Train Accuracy: 39.04199981689453, Test
Accuracy: 38.55999755859375
Epoch: 36 Loss: 1.7024803161621094. Train Accuracy: 38.43000030517578, Test
Accuracy: 37.72999954223633
Epoch: 37 Loss: 1.438204050064087. Train Accuracy: 40.10599899291992, Test
Accuracy: 39.73999786376953
Epoch: 38 Loss: 1.5515315532684326. Train Accuracy: 38.742000579833984, Test
Accuracy: 38.23999786376953
Epoch: 39 Loss: 1.5755096673965454. Train Accuracy: 41.63399887084961, Test
Accuracy: 40.709999084472656
Epoch: 40 Loss: 1.5728132724761963. Train Accuracy: 41.391998291015625, Test
Accuracy: 40.84000015258789
Epoch: 41 Loss: 1.6659173965454102. Train Accuracy: 42.577999114990234, Test
Accuracy: 41.709999084472656
Epoch: 42 Loss: 1.5644294023513794. Train Accuracy: 42.947998046875, Test
Accuracy: 42.709999084472656
Epoch: 43 Loss: 1.3784466981887817. Train Accuracy: 43.62799835205078, Test
Accuracy: 42.64999771118164
Epoch: 44 Loss: 1.635140061378479. Train Accuracy: 43.65599822998047, Test
Accuracy: 43.12999725341797
Epoch: 45 Loss: 1.4886810779571533. Train Accuracy: 43.141998291015625, Test
Accuracy: 42.52000045776367
Epoch: 46 Loss: 1.2476471662521362. Train Accuracy: 45.231998443603516, Test
Accuracy: 44.459999084472656
Epoch: 47 Loss: 1.5722951889038086. Train Accuracy: 44.854000091552734, Test
Accuracy: 44.23999786376953
Epoch: 48 Loss: 1.5661360025405884. Train Accuracy: 44.92399978637695, Test
Accuracy: 44.459999084472656
Epoch: 49 Loss: 1.4472699165344238. Train Accuracy: 46.07600021362305, Test
Accuracy: 44.98999786376953
Epoch: 50 Loss: 1.494956612586975. Train Accuracy: 44.955997467041016, Test
Accuracy: 44.28999710083008
Training for 51 epochs
Epoch: 0 Loss: 2.299844741821289. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 1 Loss: 2.3037679195404053. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 2 Loss: 2.302757740020752. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 3 Loss: 2.303771734237671. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 4 Loss: 2.3037095069885254. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 5 Loss: 2.3022170066833496. Train Accuracy: 10.0, Test Accuracy: 10.0
```

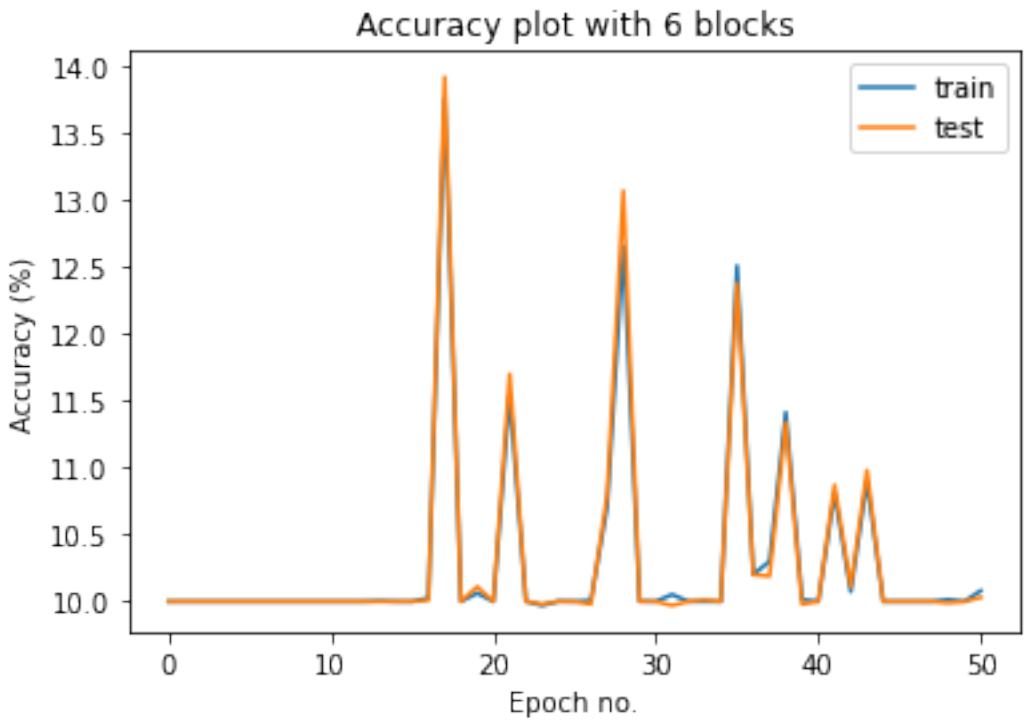
```
Epoch: 6 Loss: 2.302462100982666. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 7 Loss: 2.3021225929260254. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 8 Loss: 2.302424192428589. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 9 Loss: 2.3024823665618896. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 10 Loss: 2.3026750087738037. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 11 Loss: 2.302889585494995. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 12 Loss: 2.302616596221924. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 13 Loss: 2.302645206451416. Train Accuracy: 10.003999710083008, Test
Accuracy: 10.0
Epoch: 14 Loss: 2.3026115894317627. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 15 Loss: 2.3026998043060303. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 16 Loss: 2.3027005195617676. Train Accuracy: 10.026000022888184, Test
Accuracy: 10.00999927520752
Epoch: 17 Loss: 2.3025832176208496. Train Accuracy: 13.753999710083008, Test
Accuracy: 13.920000076293945
Epoch: 18 Loss: 2.302766799926758. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 19 Loss: 2.302690029144287. Train Accuracy: 10.061999320983887, Test
Accuracy: 10.109999656677246
Epoch: 20 Loss: 2.3027126789093018. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 21 Loss: 2.3026602268218994. Train Accuracy: 11.543999671936035, Test
Accuracy: 11.699999809265137
Epoch: 22 Loss: 2.3026392459869385. Train Accuracy: 10.001999855041504, Test
Accuracy: 10.0
Epoch: 23 Loss: 2.3025903701782227. Train Accuracy: 9.969999313354492, Test
Accuracy: 9.979999542236328
Epoch: 24 Loss: 2.302612781524658. Train Accuracy: 10.001999855041504, Test
Accuracy: 10.0
Epoch: 25 Loss: 2.302633762359619. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 26 Loss: 2.3026657104492188. Train Accuracy: 10.007999420166016, Test
Accuracy: 9.979999542236328
Epoch: 27 Loss: 2.302597999572754. Train Accuracy: 10.703999519348145, Test
Accuracy: 10.789999961853027
Epoch: 28 Loss: 2.302598476409912. Train Accuracy: 12.653999328613281, Test
Accuracy: 13.069999694824219
Epoch: 29 Loss: 2.3026890754699707. Train Accuracy: 10.001999855041504, Test
Accuracy: 10.0
Epoch: 30 Loss: 2.3025920391082764. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 31 Loss: 2.302579402923584. Train Accuracy: 10.052000045776367, Test
Accuracy: 9.969999313354492
Epoch: 32 Loss: 2.3026022911071777. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 33 Loss: 2.3026201725006104. Train Accuracy: 10.0, Test Accuracy:
10.00999927520752
Epoch: 34 Loss: 2.302673101425171. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 35 Loss: 2.3025355339050293. Train Accuracy: 12.50999927520752, Test
Accuracy: 12.380000114440918
Epoch: 36 Loss: 2.302638530731201. Train Accuracy: 10.20199966430664, Test
Accuracy: 10.199999809265137
Epoch: 37 Loss: 2.3026490211486816. Train Accuracy: 10.300000190734863, Test
```

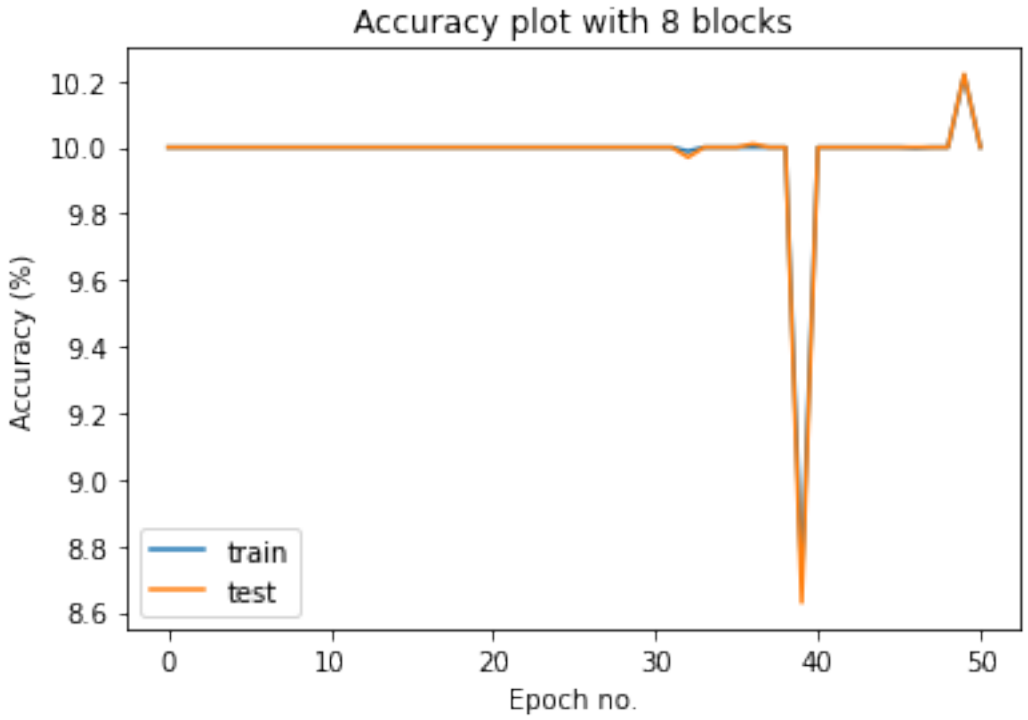
```
Accuracy: 10.1899995803833
Epoch: 38 Loss: 2.302614212036133. Train Accuracy: 11.413999557495117, Test
Accuracy: 11.34000015258789
Epoch: 39 Loss: 2.3026068210601807. Train Accuracy: 10.013999938964844, Test
Accuracy: 9.979999542236328
Epoch: 40 Loss: 2.302586078643799. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 41 Loss: 2.302700996398926. Train Accuracy: 10.809999465942383, Test
Accuracy: 10.869999885559082
Epoch: 42 Loss: 2.3026516437530518. Train Accuracy: 10.073999404907227, Test
Accuracy: 10.109999656677246
Epoch: 43 Loss: 2.302595376968384. Train Accuracy: 10.899999618530273, Test
Accuracy: 10.979999542236328
Epoch: 44 Loss: 2.3026630878448486. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 45 Loss: 2.3026247024536133. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 46 Loss: 2.3025124073028564. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 47 Loss: 2.3025689125061035. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 48 Loss: 2.3025548458099365. Train Accuracy: 10.013999938964844, Test
Accuracy: 9.989999771118164
Epoch: 49 Loss: 2.302584171295166. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 50 Loss: 2.3025832176208496. Train Accuracy: 10.079999923706055, Test
Accuracy: 10.029999732971191
Training for 51 epochs
Epoch: 0 Loss: 2.301138401031494. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 1 Loss: 2.303593635559082. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 2 Loss: 2.302870035171509. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 3 Loss: 2.301819086074829. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 4 Loss: 2.302201271057129. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 5 Loss: 2.3018105030059814. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 6 Loss: 2.301304340362549. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 7 Loss: 2.3031117916107178. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 8 Loss: 2.3022847175598145. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 9 Loss: 2.303086757659912. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 10 Loss: 2.3025550842285156. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 11 Loss: 2.303415060043335. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 12 Loss: 2.3025882244110107. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 13 Loss: 2.302704334259033. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 14 Loss: 2.3027191162109375. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 15 Loss: 2.3024866580963135. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 16 Loss: 2.302493095397949. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 17 Loss: 2.302490234375. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 18 Loss: 2.30265474319458. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 19 Loss: 2.3027162551879883. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 20 Loss: 2.302614688873291. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 21 Loss: 2.302518606185913. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 22 Loss: 2.302665948867798. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 23 Loss: 2.302588939666748. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 24 Loss: 2.3025755882263184. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 25 Loss: 2.3025476932525635. Train Accuracy: 10.0, Test Accuracy: 10.0
```

```
Epoch: 26 Loss: 2.302725315093994. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 27 Loss: 2.30252742767334. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 28 Loss: 2.3026480674743652. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 29 Loss: 2.302598476409912. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 30 Loss: 2.302670478820801. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 31 Loss: 2.302578926086426. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 32 Loss: 2.3026089668273926. Train Accuracy: 9.98799991607666, Test
Accuracy: 9.969999313354492
Epoch: 33 Loss: 2.3026528358459473. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 34 Loss: 2.302597761154175. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 35 Loss: 2.302584171295166. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 36 Loss: 2.30265212059021. Train Accuracy: 10.0, Test Accuracy:
10.00999927520752
Epoch: 37 Loss: 2.302633762359619. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 38 Loss: 2.3026022911071777. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 39 Loss: 2.3026671409606934. Train Accuracy: 8.708000183105469, Test
Accuracy: 8.630000114440918
Epoch: 40 Loss: 2.302643060684204. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 41 Loss: 2.3025832176208496. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 42 Loss: 2.302579164505005. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 43 Loss: 2.3027091026306152. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 44 Loss: 2.3025898933410645. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 45 Loss: 2.3025736808776855. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 46 Loss: 2.302579164505005. Train Accuracy: 9.998000144958496, Test
Accuracy: 10.0
Epoch: 47 Loss: 2.3026740550994873. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 48 Loss: 2.3026082515716553. Train Accuracy: 10.0, Test Accuracy: 10.0
Epoch: 49 Loss: 2.3026442527770996. Train Accuracy: 10.215999603271484, Test
Accuracy: 10.219999313354492
Epoch: 50 Loss: 2.3025708198547363. Train Accuracy: 10.0, Test Accuracy: 10.0
```


Accuracy plot with 2 blocks
train
test
Accuracy (%)
Epoch no.
10
20
30
40
50
0
10
20
30
40
50


Accuracy plot with 4 blocks
train
test
Accuracy (%)
Epoch no.
10
15
20
25
30
35
40
45
0
10
20
30
40
50


Accuracy plot with 6 blocks
train
test
Accuracy (%)
Epoch no.
14.0
13.5
13.0
12.5
12.0
11.5
11.0
10.5
10.0
0
10
20
30
40
50


Accuracy plot with 8 blocks
train
test
Accuracy (%)
Epoch no.
10.2
10.0
9.8
9.6
9.4
9.2
9.0
8.8
8.6
0
10
20
30
40
50

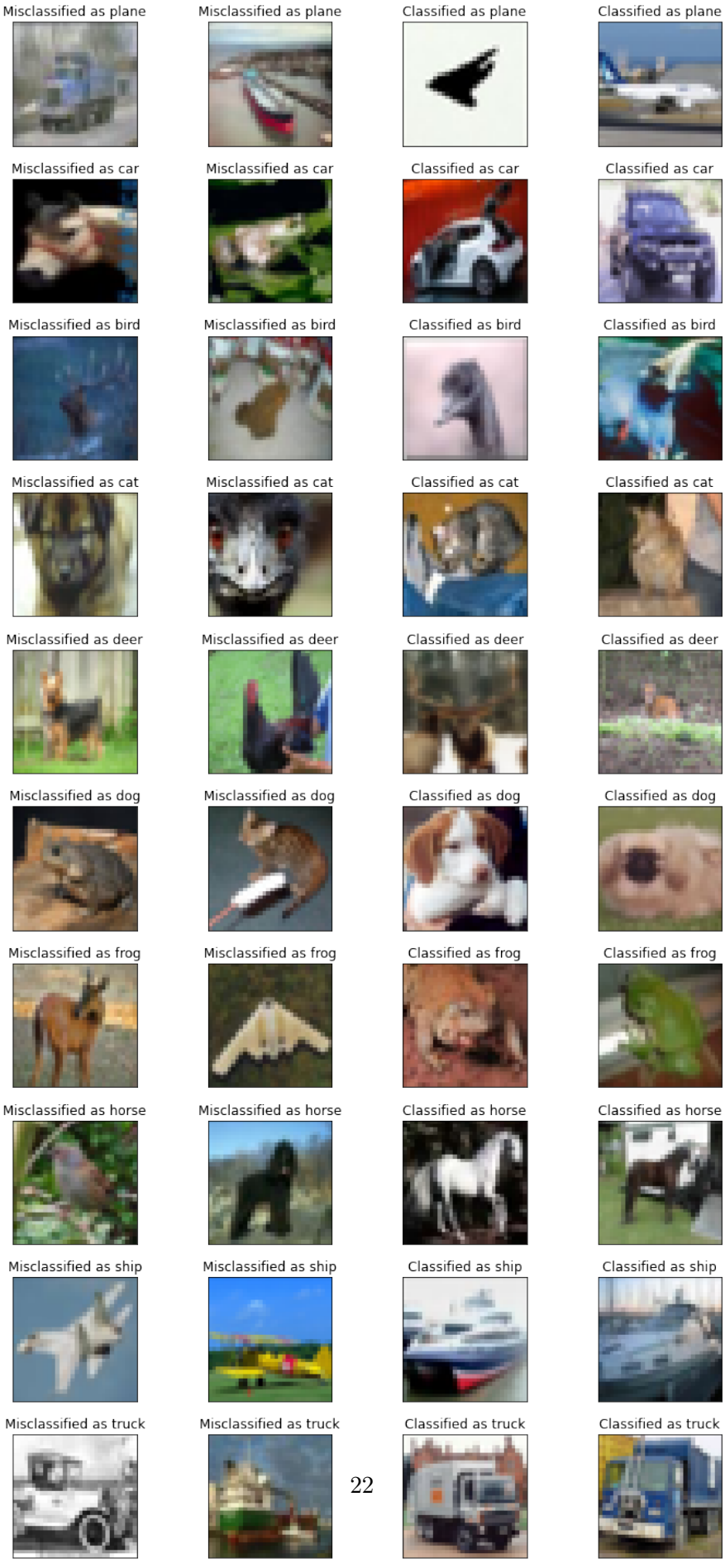
0.2.1 Analysis

From these experiments, we can see that 2 is the optimal number of blocks with our CNN model without batch normalization. This was a little surprising to me, as I thought that increasing the depth of the model would directly correlate to better performance. However, we see that this really isn't the case because without batch normalization it will take many many more epocsh to converge.

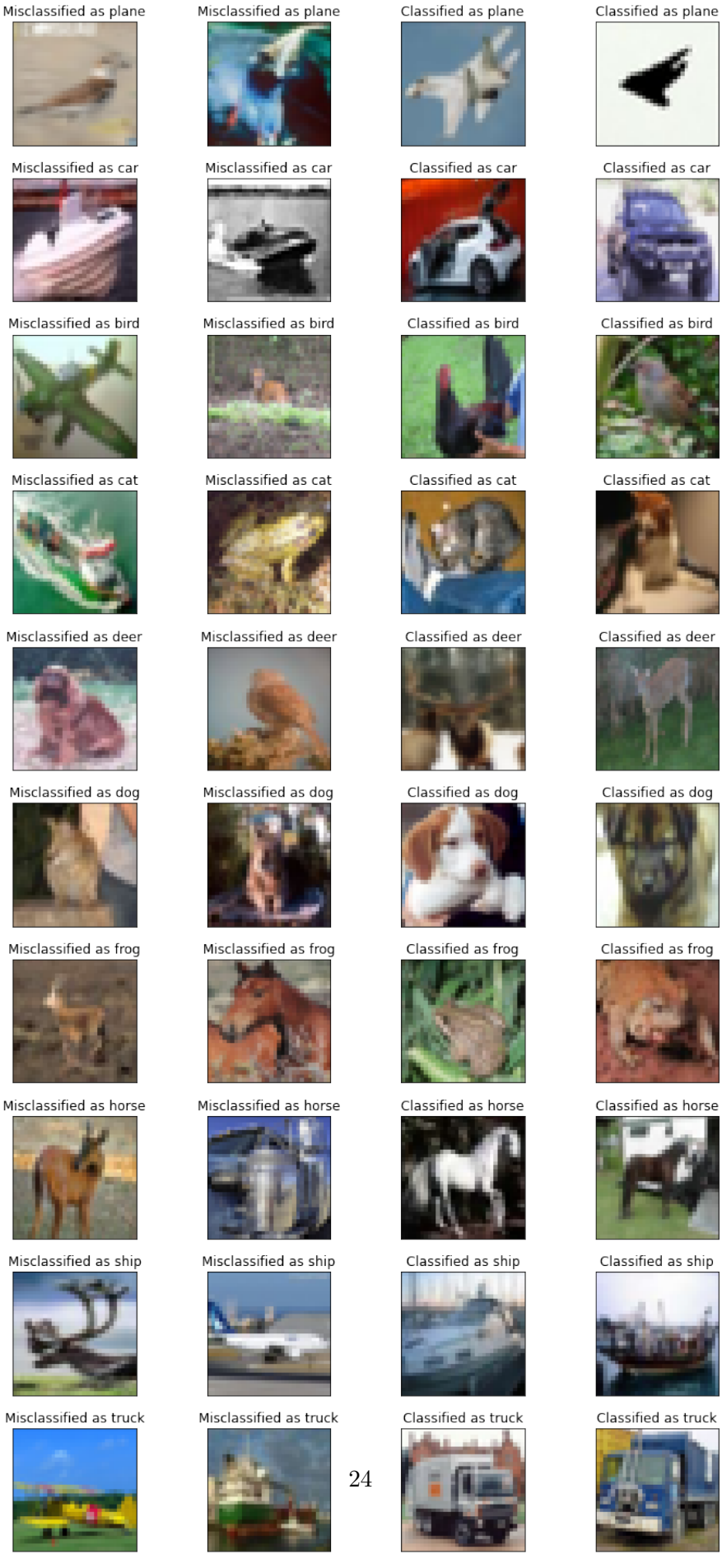
0.3 c) Plot 2 misclassified and 2 correctly classified samples for each class

[38]:
```
inv_normalize = transforms.Normalize(
  (-1, -1, -1),
  (2, 2, 2)
)

def plot_per_class(model):
  # Find 2 misclassified and 2 correclty classified test samples for each␣
 ↪output class
  misclassified = {k: [] for k in range(10)}
  correctly_classified = {k: [] for k in range(10)}

  for images, labels in test_loader:
      images, labels = images.to(DEVICE), labels.to(DEVICE)
      outputs = model(images)
      _, predicted = torch.max(outputs.data, 1)

      for pred, act, image in zip(predicted, labels, images):
          if pred != act:
              misclassified[pred.item()].append(image) # save misclassified␣
 ↪images
          else:
              correctly_classified[pred.item()].append(image) # save correctly␣
 ↪classified

  fig=plt.figure(figsize=(10, 40))

  # Plot misclasssified and classified correctly
  for clas in range(10):
      for i in range(1, 3):
          img = inv_normalize(misclassified[clas][i].cpu()).numpy()
          img = np.dstack((img[0], img[1], img[2]))
          ax = fig.add_subplot(20, 4, clas * 4 + i)
          ax.set_xticks([])
          ax.set_yticks([])
          ax.set_title(f"Misclassified as {classes[clas]}")
          plt.imshow(img)
      for j in range(3, 5):
          img = inv_normalize(correctly_classified[clas][j-3].cpu()).numpy()
          img = np.dstack((img[0], img[1], img[2]))
```

```
            ax = fig.add_subplot(20, 4, clas * 4 + j)
            ax.set_xticks([])
            ax.set_yticks([])
            ax.set_title(f"Classified as {classes[clas]}")
            plt.imshow(img)


    plt.tight_layout()
    plt.show()


best_nbn = nbn_models[0]

plot_per_class(best_nbn)
```


Misclassified as plane
Misclassified as plane
Classified as plane
Classified as plane
Misclassified as car
Misclassified as car
Classified as car
Classified as car
Misclassified as bird
Misclassified as bird
Classified as bird
Classified as bird
Misclassified as cat
Misclassified as cat
Classified as cat
Classified as cat
Misclassified as deer
Misclassified as deer
Classified as deer
Classified as deer
Misclassified as dog
Misclassified as dog
Classified as dog
Classified as dog
Misclassified as frog
Misclassified as frog
Classified as frog
Classified as frog
Misclassified as horse
Misclassified as horse
Classified as horse
Classified as horse
Misclassified as ship
Misclassified as ship
Classified as ship
Classified as ship
Misclassified as truck
Misclassified as truck
Classified as truck
Classified as truck

[39]:
```
plot_per_class(batch_norm)
```


Misclassified as plane
Misclassified as plane
Classified as plane
Classified as plane
Misclassified as car
Misclassified as car
Classified as car
Classified as car
Misclassified as bird
Misclassified as bird
Classified as bird
Classified as bird
Misclassified as cat
Misclassified as cat
Classified as cat
Classified as cat
Misclassified as deer
Misclassified as deer
Classified as deer
Classified as deer
Misclassified as dog
Misclassified as dog
Classified as dog
Classified as dog
Misclassified as frog
Misclassified as frog
Classified as frog
Classified as frog
Misclassified as horse
Misclassified as horse
Classified as horse
Classified as horse
Misclassified as ship
Misclassified as ship
Classified as ship
Classified as ship
Misclassified as truck
Misclassified as truck
Classified as truck
Classified as truck

0.4 d) Comparison with previous homework

On the last homework we used a similar CNN model on the CIFAR 10 dataset as well, but the accuracy received was tended to be much higher than the ones on this homework, which is not what I would expected. With the best model last homework, which only had two convolutional layers and one fully connected layer, we were able to achieve an accuracy of around 65%. In comparison, the best model we trained in this homework, the batch normalized with 10 blocks, was only able to achieve an accuracy of around 55%. Initially, I was tempted to draw this difference up to the fact that sense the batch size was much lower and the model was simpler, so it would be able to converge more quickly and this is where the accuracy difference was experienced. However, I realized that we only trained for 3 epochs in the last hoemwork whilst we trained for 51 epochs with these experiments. I'm not quite sure why such a stark difference in performance has been observed. I was expecting that a deeper model would be able to perform better. Maybe we need to train for many more epochs.

[ ]: