

# Homework Assignment 2

Name: Linqi Xiao

NetID: lx130

```
1 import torch
2 from torch import nn
3 from torchvision import datasets, transforms
4 from torch.utils.data import DataLoader
5
6 from PIL import Image
7 import time
8 import matplotlib.pyplot as plt

1 BATCH_SIZE = 64
2 EPOCHS = 10
3 DEVICE = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
4 pipeline = transforms.Compose([
5     transforms.ToTensor(),
6     transforms.Normalize(mean=(0.1307,), std=(0.3081,))
7 ])
8
9 train_dataset = datasets.MNIST(root='./dataset', train=True, download=True, transform=pipeline)
10 test_dataset = datasets.MNIST(root='./dataset', train=False, download=True, transform=pipeline)
11 train_loader = DataLoader(dataset=train_dataset, batch_size=BATCH_SIZE, shuffle=True)
12 test_loader = DataLoader(dataset=test_dataset, batch_size=BATCH_SIZE, shuffle=True)

1 # image, target = train_dataset[0]
2 # print(image.shape)
3 #
4 # plt.imshow(image.reshape(28, 28), cmap='gray')
5 # print(target)
```

60000

## Build Model

```
1 class MnistFC(nn.Module):
2     def __init__(self):
3         super(MnistFC, self).__init__()
4         self.model = nn.Sequential(
5             nn.Flatten(),
6             nn.Linear(in_features=784, out_features=200),
7             nn.ReLU(),
8             nn.Linear(in_features=200, out_features=50),
9             nn.ReLU(),
10            nn.Linear(in_features=50, out_features=10),
11        )
12
13    def forward(self, x):
14        x = self.model(x)
15        return x

1 model = MnistFC().to(device=DEVICE)
2 optimizer = torch.optim.SGD(model.parameters(), lr=0.01, momentum=0.9)
```

```

1 model = MnistFC().to(device=DEVICE)
2 #optimizer = torch.optim.SGD(model.parameters(), lr=0.01, momentum=0.9)
3 optimizer = torch.optim.Adam(params=model.parameters(), lr=0.0005)
4 loss_func = nn.CrossEntropyLoss()

```

## Training

```

1 def train_model(model, device, train_loader, optimizer, loss_function, epoch):
2     # print('Start Training...')
3     model.train()
4     for batch_idx, (image, target) in enumerate(train_loader):
5         image, target = image.to(device=device), target.to(device=device)
6         optimizer.zero_grad()
7         output = model(image)
8         train_loss = loss_function(output, target)
9         train_loss.backward()
10        optimizer.step()
11
12        if batch_idx % 3000 == 0:
13            print("Train Round :{}\tLoss :{:0.6f}".format(epoch, train_loss.item()))

```

## Testing

```

1 def test_model(model, device, test_loader, loss_function):
2     # print('Start Testing...')
3     model.eval()
4     accuracy = 0.0
5     test_loss = 0.0
6
7     with torch.no_grad():
8         for image, target in test_loader:
9             image, target = image.to(device=device), target.to(device=device)
10            output = model(image)
11            test_loss += loss_function(output, target)
12            pred = output.argmax(dim=1)
13            accuracy += pred.eq(target.view_as(pred)).sum().item()
14        test_loss /= len(test_loader.dataset)
15        accuracy /= len(test_loader.dataset)
16        print("Average Test Loss :{:0.4f}, Accuracy :{:0.2f}%\n".format(test_loss.item(), accuracy * 100.0))

```

## Run

```
1 start_time = time.time()
2 for epoch in range(1, EPOCHS + 1):
3     train_model(model=model, device=DEVICE, train_loader=train_loader, optimizer=optimizer, loss_function=loss_func, epoch=epoch)
4     test_model(model=model, device=DEVICE, test_loader=test_loader, loss_function=loss_func)
5 end_time = time.time()
6 print('>> Total time:', end_time - start_time)
```

✓ Train Round :1 Loss :3.910919  
Average Test Loss :0.0488, Accuracy :84.29%

Train Round :2 Loss :3.105080  
Average Test Loss :0.0486, Accuracy :85.62%

Train Round :3 Loss :3.006492  
Average Test Loss :0.0485, Accuracy :85.88%

Train Round :4 Loss :3.105409  
Average Test Loss :0.0484, Accuracy :86.58%

Train Round :5 Loss :3.044711  
Average Test Loss :0.0469, Accuracy :96.40%

Train Round :6 Loss :2.960890  
Average Test Loss :0.0468, Accuracy :96.82%

Train Round :7 Loss :2.978237  
Average Test Loss :0.0468, Accuracy :97.02%

Train Round :8 Loss :2.989521  
Average Test Loss :0.0467, Accuracy :97.42%

Train Round :9 Loss :2.977625  
Average Test Loss :0.0467, Accuracy :97.32%

Train Round :10 Loss :2.957033  
Average Test Loss :0.0467, Accuracy :97.44%

>> Total time: 63.01949715614319