PyTorch Model Notebook

This notebook will allow you to get practice in building and working with PyTorch models. Code excersises denoted by a problem number (i.e. Problem #1) will include a task and a code block that asks for your solution. These blocks will be denoted by comments of the form '# YOUR CODE HERE #'. The code immediately following include assertions that are used to check completeness of the response. They will raise an exception if the previous solution is not complete or not correct.

Datasets and DataLoaders

Reference: The Linux Foundation, "Datasets & DataLoaders - PyTorch Tutorials 2.6.0 +cu124 documentation," pytorch.org https://pytorch.org/tutorials/beginner/basics/data_tutorial.html (accessed Mar. 20, 2025).

In [1]:

```
import torch
from torch.utils.data import Dataset
from torch.utils.data import DataLoader
```

Problem #1: Finish implementing the "RandDataset" Dataset by 1) setting "self.mapping" to a random tensor of dimension (output_dims, input_dims), 2) implementing the '__len__' method by returning the length of the dataset, and 3) setting the **output_tensor** = **Mx** in the '__getitem__' method, where **M** is the "self.mapping" tensor and **x** is the "input_tensor". Also remember to implement the "self.target_transform" (if not None) on the "output_tensor", analagous to the "self.transform" already implemented.

In [2]:

```
class RandDataset(Dataset):
    def __init__(self, input_dims, output_dims, length, transform=None, target_tran
        self.input_dims = input_dims
        self.output_dims = output_dims
        self.transform = transform
        self.target_transform = target_transform
        ### BEGIN SOLUTION
        self.mapping = torch.rand(output_dims, input_dims)  #Hint this is the rando
        ### END SOLUTION
        self.length = length
    ### BEGIN SOLUTION
    def __len__(self):
        return self.length
    ### END SOLUTION

    def __getitem__(self, idx):
        input_tensor = torch.rand(self.input_dims)
        if self.transform:
            input_tensor = self.transform(input_tensor)
        ### BEGIN SOLUTION
        output_tensor = self.mapping.matmul(input_tensor)
        if self.target_transform:
            output_tensor = self.target_transform(output_tensor)
        ### END SOLUTION
        return input_tensor, output_tensor

assert len(RandDataset(5,10,1000)) == 1000
assert (RandDataset(5, 10, 1000)).mapping.shape == (10,5)
assert (RandDataset(5, 10, 1000))[1][1].shape[0] == 10
assert ((RandDataset(5, 10, 1000, target_transform=lambda x: x + 20))[1][1] > 20).a
```

Problem #2: Instatiate the RandDataset class with a length=32000 and variables "input_dims" and "output_dims". Set variables named "input_dims" and "output_dims" to apropriate values and use in the RandDataset instantiation call. Also, instantiate a DataLoader using this dataset object using a "batch_size" of 32, already implemented. Name this dataloader object "rand_dataset".

In [3]:

```
batch_size = 32
### BEGIN SOLUTION
input_dims, output_dims = 5, 10
rand_dataset = RandDataset(input_dims, output_dims, 32000)
rand_dataloader = DataLoader(rand_dataset, batch_size=batch_size)
### END SOLUTION

assert input_dims > 0 and output_dims > 0
assert len(rand_dataloader.dataset) == 32000
assert rand_dataloader.batch_size == 32
assert len(rand_dataloader) == 1000
```

Setting the train and test dataloaders from above

In [4]:

```
import copy

train_dataloader = rand_dataloader

test_dataset = copy.deepcopy(train_dataloader.dataset)
test_dataset.length = 1000
test_dataloader = DataLoader(test_dataset, batch_size=batch_size)

assert len(test_dataloader.dataset) == 1000
assert (train_dataloader.dataset.mapping == test_dataloader.dataset.mapping).all()
```

Building the PyTorch Model

Reference: The Linux Foundation, "Build the Neural Network - PyTorch Tutorials 2.6.0 +cu124 documentation," pytorch.org https://pytorch.org/tutorials/beginner/basics/buildmodel_tutorial.html (accessed Mar. 13, 2025).

Problem #3: Implement a Pytorch model class named "NNModel". Fill in the instantiation of the model's layers, which should include a nn.Linear, nn.ReLU, nn.Linear, nn.ReLU, and nn.Linear layers. There should be **n** input neurons, **h** hidden neurons, and **m** output neurons. Hint: Both hidden linear layers (first two) should have **h** neurons. Implement the forward computation of the model using the **input_tensor** as input and return the result.

In [5]:

```python
from torch import nn

class NNModel(nn.Module):
    def __init__(self,n,m,h):
        super().__init__()

        ### BEGIN SOLUTION
        self.stack = nn.Sequential(
             nn.Linear(n,h),
             nn.ReLU(),
             nn.Linear(h,h),
             nn.ReLU(),
             nn.Linear(h, m),
        )
        ### END SOLUTION

    def forward(self, input_tensor):
        ### BEGIN SOLUTION
        y = self.stack(input_tensor)
        return y
        ### END SOLUTION

import re
assert ((NNModel(input_dims, output_dims, 50))(torch.rand(5,input_dims))).shape ==
assert len(re.findall('Linear', str(NNModel(input_dims, output_dims, 50)))) == 3
assert len(re.findall('ReLU()', str(NNModel(input_dims, output_dims, 50)))) == 2
```

Optimizing the PyTorch Model

Reference: The Linux Foundation, "Optimizing Model Parameters - PyTorch Tutorials 2.6.0 +cu124 documentation," pytorch.org https://pytorch.org/tutorials/beginner/basics/optimization_tutorial.html (accessed Mar. 24, 2025).

Problem #4: Instantiate the nn.MSELoss function with reduction='sum' and name the object, "my_loss_fn". Instantiate the NNModel using "input_dims", "output_dims", and any number of hidden neurons and name the object, "my_model". Instantiate the optim.SGD optimizer with the model parameters and the learning_rate defined above and name the object, "my_optimizer".

In [6]:

```python
from torch import optim

learning_rate = 1e-2
epochs = 50
tolerance = 1e-2

### BEGIN SOLUTION
```

```
my_loss_fn = nn.MSELoss(reduction='sum')
my_model = NNModel(input_dims, output_dims, 20)
my_optimizer = optim.SGD(my_model.parameters(), lr=learning_rate)
### END SOLUTION

assert isinstance(my_loss_fn, nn.MSELoss)
assert my_loss_fn.reduction == 'sum'
assert isinstance(my_model, NNModel)
assert isinstance(my_optimizer, optim.SGD)
```

Problem #5: Implement the training loop by including the 1) model predictions from the batch inputs, **X**, and calculating the loss via the loss_fn using the predictions and batch outputs, **Y**. Divide the loss by the number of predictions to get the **avg_loss**.

In [7]:

```
def train_loop(dataloader, model, loss_fn, optimizer, device=None):
    size = len(dataloader.dataset)
    model.train()
    for batch, (X,Y) in enumerate(dataloader):
        if device:
            X = X.to(device)
            Y = Y.to(device)
        ### BEGIN SOLUTION
        pred = model(X)
        loss = loss_fn(pred, Y)
        avg_loss = loss / len(pred)
        ### END SOLUTION

        avg_loss.backward()
        optimizer.step()
        optimizer.zero_grad()

        if (batch+1) %100 == 0:
            avg_loss, current = avg_loss.item(), batch * batch_size + len(pred)
            print(f"Avg. loss: {avg_loss:>7f}, [current:{current:>5d}/{size:>5d}]")

    return True

assert train_loop(DataLoader(RandDataset(3,5,batch_size*100),batch_size=batch_size)
```

```
Avg. loss: 4.258340, [current: 3200/ 3200]
```

Problem #6: Implement the test loop by including the 1) model predictions from the batch inputs, **X**, and calculating the test loss via the loss_fn using the predictions and batch outputs, **Y**. Divide the test_loss by the number of predictions and remember to use the .item() method to extract the scalar value.

In [8]:

```
def test_loop(dataloader, model, loss_fn, tolerance, device=None):
    size = len(dataloader.dataset)
    num_batches = len(dataloader)
    model.eval()

    test_loss = 0
    correct = 0
    with torch.no_grad():
        for (X,Y) in dataloader:
            if device:
                X = X.to(device)
                Y = Y.to(device)
            ### BEGIN SOLUTION
            pred = model(X)
            test_loss += (loss_fn(pred, Y) / len(pred)).item()
            ### END SOLUTION
            correct += ((pred - Y).abs() < tolerance).all(dim=1).type(torch.float).

    test_loss /= num_batches
    correct /= size

    print(f"Test Error: \n Accuracy: {(100*correct):>0.1f}%, Avg. loss: {test_loss:
    return True

assert test_loop(DataLoader(RandDataset(3,5,batch_size),batch_size=batch_size), NNM
```

```
Test Error: 
 Accuracy: 0.0%, Avg. loss: 3.864772
```

Implementing the epoch loop and running the training loop

Problem #7: Implement the epoch loop by using the train_loop and test_loop functions defined above.

In [9]:

```
def epoch_loop(epochs, train_dataloader, test_dataloader, model, loss_fn, optimizer
    for t in range(epochs):
        print(f"Epoch {t+1}\n-------------------------------")
        ### BEGIN SOLUTION ###
        train_loop(train_dataloader, model, loss_fn, optimizer, device)
        test_loop(test_dataloader, model, loss_fn, tolerance, device)
        ### END SOLUTION ###
    print("Done")
    return True

epochs_cpu = 2
assert epoch_loop(epochs_cpu, train_dataloader, test_dataloader, my_model, my_loss_
Epoch 1
------------------------------
Avg. loss: 0.466994, [current: 3200/32000]
Avg. loss: 0.275905, [current: 6400/32000]
Avg. loss: 0.173477, [current: 9600/32000]
Avg. loss: 0.155771, [current:12800/32000]
Avg. loss: 0.138271, [current:16000/32000]
Avg. loss: 0.105612, [current:19200/32000]
Avg. loss: 0.106437, [current:22400/32000]
Avg. loss: 0.107267, [current:25600/32000]
Avg. loss: 0.081672, [current:28800/32000]
Avg. loss: 0.061309, [current:32000/32000]
Test Error:
 Accuracy: 0.0%, Avg. loss: 0.069360

Epoch 2
------------------------------
Avg. loss: 0.052960, [current: 3200/32000]
Avg. loss: 0.045673, [current: 6400/32000]
Avg. loss: 0.034378, [current: 9600/32000]
Avg. loss: 0.030830, [current:12800/32000]
Avg. loss: 0.018746, [current:16000/32000]
Avg. loss: 0.015056, [current:19200/32000]
Avg. loss: 0.011399, [current:22400/32000]
Avg. loss: 0.010450, [current:25600/32000]
Avg. loss: 0.007051, [current:28800/32000]
Avg. loss: 0.007651, [current:32000/32000]
Test Error:
 Accuracy: 2.0%, Avg. loss: 0.005618

Done
```

Utilizing the GPU for Training

In [10]:

```python
device = torch.device('cpu')

if torch.cuda.is_available():
    device = torch.device(torch.cuda.current_device())

print(f"Using device - {device}")
```

```
Using device - cuda:0
```

Problem #8: If a GPU device is available, move "my_model" to the device and save as "my_model_gpu". Reinitialize the SGD optimizer using "my_model_gpu.parameters()".

In [11]:

```python
if torch.cuda.is_available():
    ### BEGIN SOUTION
    my_model_gpu = my_model.to(device)
    my_optimizer_gpu = optim.SGD(my_model_gpu.parameters(), lr=learning_rate)
    ### END SOLUTION

    assert epoch_loop(epochs, train_dataloader, test_dataloader, my_model_gpu, my_l
Epoch 1
-------------------------------
Avg. loss: 0.004681, [current: 3200/32000]
Avg. loss: 0.002996, [current: 6400/32000]
Avg. loss: 0.003277, [current: 9600/32000]
Avg. loss: 0.001482, [current:12800/32000]
Avg. loss: 0.001766, [current:16000/32000]
Avg. loss: 0.001653, [current:19200/32000]
Avg. loss: 0.000958, [current:22400/32000]
Avg. loss: 0.002503, [current:25600/32000]
Avg. loss: 0.000641, [current:28800/32000]
Avg. loss: 0.000622, [current:32000/32000]
Test Error: 
 Accuracy: 42.7%, Avg. loss: 0.001426 

Epoch 2
-------------------------------
Avg. loss: 0.001284, [current: 3200/32000]
Avg. loss: 0.001154, [current: 6400/32000]
Avg. loss: 0.001246, [current: 9600/32000]
Avg. loss: 0.000883, [current:12800/32000]
Avg. loss: 0.000802, [current:16000/32000]
Avg. loss: 0.000592, [current:19200/32000]
Avg. loss: 0.000378, [current:22400/32000]
Avg. loss: 0.000213, [current:25600/32000]
Avg. loss: 0.000165, [current:28800/32000]
Avg. loss: 0.000207, [current:32000/32000]
Test Error: 
 Accuracy: 88.6%, Avg. loss: 0.000410 

Epoch 3
-------------------------------
Avg. loss: 0.000061, [current: 3200/32000]
Avg. loss: 0.000440, [current: 6400/32000]
Avg. loss: 0.001429, [current: 9600/32000]
Avg. loss: 0.000525, [current:12800/32000]
Avg. loss: 0.001630, [current:16000/32000]
Avg. loss: 0.000237, [current:19200/32000]
Avg. loss: 0.000529, [current:22400/32000]
Avg. loss: 0.000137, [current:25600/32000]
Avg. loss: 0.000900, [current:28800/32000]
Avg. loss: 0.000709, [current:32000/32000]
Test Error: 
 Accuracy: 91.4%, Avg. loss: 0.000313 

Epoch 4
-------------------------------
Avg. loss: 0.000120, [current: 3200/32000]
Avg. loss: 0.000384, [current: 6400/32000]
Avg. loss: 0.000415, [current: 9600/32000]
Avg. loss: 0.000222, [current:12800/32000]
Avg. loss: 0.000769, [current:16000/32000]
Avg. loss: 0.000100, [current:19200/32000]
Avg. loss: 0.000102, [current:22400/32000]
Avg. loss: 0.000282, [current:25600/32000]
Avg. loss: 0.000081, [current:28800/32000]
Avg. loss: 0.000813, [current:32000/32000]
Test Error:
 Accuracy: 91.8%, Avg. loss: 0.000299

Epoch 5
------------------------------
Avg. loss: 0.000043, [current: 3200/32000]
Avg. loss: 0.000546, [current: 6400/32000]
Avg. loss: 0.000055, [current: 9600/32000]
Avg. loss: 0.000019, [current:12800/32000]
Avg. loss: 0.000171, [current:16000/32000]
Avg. loss: 0.000797, [current:19200/32000]
Avg. loss: 0.000291, [current:22400/32000]
Avg. loss: 0.000553, [current:25600/32000]
Avg. loss: 0.000078, [current:28800/32000]
Avg. loss: 0.000557, [current:32000/32000]
Test Error:
 Accuracy: 93.5%, Avg. loss: 0.000359

Epoch 6
------------------------------
Avg. loss: 0.000433, [current: 3200/32000]
Avg. loss: 0.000076, [current: 6400/32000]
Avg. loss: 0.000012, [current: 9600/32000]
Avg. loss: 0.000438, [current:12800/32000]
Avg. loss: 0.000049, [current:16000/32000]
Avg. loss: 0.000052, [current:19200/32000]
Avg. loss: 0.000014, [current:22400/32000]
Avg. loss: 0.000221, [current:25600/32000]
Avg. loss: 0.000218, [current:28800/32000]
Avg. loss: 0.000054, [current:32000/32000]
Test Error:
 Accuracy: 93.3%, Avg. loss: 0.000195

Epoch 7
------------------------------
Avg. loss: 0.000037, [current: 3200/32000]
Avg. loss: 0.000066, [current: 6400/32000]
Avg. loss: 0.000009, [current: 9600/32000]
Avg. loss: 0.000102, [current:12800/32000]
Avg. loss: 0.000370, [current:16000/32000]
Avg. loss: 0.000098, [current:19200/32000]
Avg. loss: 0.000080, [current:22400/32000]
Avg. loss: 0.000241, [current:25600/32000]
Avg. loss: 0.000181, [current:28800/32000]
Avg. loss: 0.000103, [current:32000/32000]
Test Error:
 Accuracy: 94.2%, Avg. loss: 0.000248

Epoch 8
------------------------------
Avg. loss: 0.000006, [current: 3200/32000]
Avg. loss: 0.000244, [current: 6400/32000]
Avg. loss: 0.000396, [current: 9600/32000]
Avg. loss: 0.000219, [current:12800/32000]
Avg. loss: 0.000019, [current:16000/32000]
Avg. loss: 0.001043, [current:19200/32000]
Avg. loss: 0.000520, [current:22400/32000]
Avg. loss: 0.000077, [current:25600/32000]
Avg. loss: 0.000381, [current:28800/32000]
Avg. loss: 0.000129, [current:32000/32000]
Test Error:
 Accuracy: 94.8%, Avg. loss: 0.000154

Epoch 9
------------------------------
Avg. loss: 0.000013, [current: 3200/32000]
Avg. loss: 0.001963, [current: 6400/32000]
Avg. loss: 0.000024, [current: 9600/32000]
Avg. loss: 0.000028, [current:12800/32000]
Avg. loss: 0.000081, [current:16000/32000]
Avg. loss: 0.000048, [current:19200/32000]
Avg. loss: 0.000036, [current:22400/32000]
Avg. loss: 0.000061, [current:25600/32000]
Avg. loss: 0.000077, [current:28800/32000]
Avg. loss: 0.000005, [current:32000/32000]
Test Error:
 Accuracy: 95.4%, Avg. loss: 0.000167

Epoch 10
------------------------------
Avg. loss: 0.000110, [current: 3200/32000]
Avg. loss: 0.000228, [current: 6400/32000]
Avg. loss: 0.000058, [current: 9600/32000]
Avg. loss: 0.000270, [current:12800/32000]
Avg. loss: 0.000033, [current:16000/32000]
Avg. loss: 0.000069, [current:19200/32000]
Avg. loss: 0.000025, [current:22400/32000]
Avg. loss: 0.000121, [current:25600/32000]
Avg. loss: 0.000276, [current:28800/32000]
Avg. loss: 0.000127, [current:32000/32000]
Test Error:
 Accuracy: 96.4%, Avg. loss: 0.000110

Epoch 11
------------------------------
Avg. loss: 0.000259, [current: 3200/32000]
Avg. loss: 0.000020, [current: 6400/32000]
Avg. loss: 0.000005, [current: 9600/32000]
Avg. loss: 0.000018, [current:12800/32000]
Avg. loss: 0.000049, [current:16000/32000]
Avg. loss: 0.000024, [current:19200/32000]
Avg. loss: 0.000082, [current:22400/32000]
Avg. loss: 0.000004, [current:25600/32000]
Avg. loss: 0.000195, [current:28800/32000]
Avg. loss: 0.000026, [current:32000/32000]
Test Error:
 Accuracy: 96.1%, Avg. loss: 0.000113

Epoch 12
------------------------------
Avg. loss: 0.000016, [current: 3200/32000]
Avg. loss: 0.000401, [current: 6400/32000]
Avg. loss: 0.000209, [current: 9600/32000]
Avg. loss: 0.000060, [current:12800/32000]
Avg. loss: 0.000194, [current:16000/32000]
Avg. loss: 0.000346, [current:19200/32000]
Avg. loss: 0.000012, [current:22400/32000]
Avg. loss: 0.000008, [current:25600/32000]
Avg. loss: 0.000110, [current:28800/32000]
Avg. loss: 0.000080, [current:32000/32000]
Test Error:
 Accuracy: 96.1%, Avg. loss: 0.000118

Epoch 13
------------------------------
Avg. loss: 0.000043, [current: 3200/32000]
Avg. loss: 0.000009, [current: 6400/32000]
Avg. loss: 0.000122, [current: 9600/32000]
Avg. loss: 0.000377, [current:12800/32000]
Avg. loss: 0.000020, [current:16000/32000]
Avg. loss: 0.000087, [current:19200/32000]
Avg. loss: 0.000038, [current:22400/32000]
Avg. loss: 0.000003, [current:25600/32000]
Avg. loss: 0.000025, [current:28800/32000]
Avg. loss: 0.000007, [current:32000/32000]
Test Error:
 Accuracy: 97.0%, Avg. loss: 0.000104

Epoch 14
------------------------------
Avg. loss: 0.000005, [current: 3200/32000]
Avg. loss: 0.000046, [current: 6400/32000]
Avg. loss: 0.000002, [current: 9600/32000]
Avg. loss: 0.000072, [current:12800/32000]
Avg. loss: 0.000010, [current:16000/32000]
Avg. loss: 0.000024, [current:19200/32000]
Avg. loss: 0.000008, [current:22400/32000]
Avg. loss: 0.000003, [current:25600/32000]
Avg. loss: 0.000134, [current:28800/32000]
Avg. loss: 0.000019, [current:32000/32000]
Test Error:
 Accuracy: 95.7%, Avg. loss: 0.000128

Epoch 15
------------------------------
Avg. loss: 0.000088, [current: 3200/32000]
Avg. loss: 0.000009, [current: 6400/32000]
Avg. loss: 0.000002, [current: 9600/32000]
Avg. loss: 0.000002, [current:12800/32000]
Avg. loss: 0.000004, [current:16000/32000]
Avg. loss: 0.000003, [current:19200/32000]
Avg. loss: 0.000002, [current:22400/32000]
Avg. loss: 0.000348, [current:25600/32000]
Avg. loss: 0.000108, [current:28800/32000]
Avg. loss: 0.000010, [current:32000/32000]
Test Error:
 Accuracy: 97.5%, Avg. loss: 0.000100
```

```
Epoch 16
------------------------------
Avg. loss: 0.000027, [current: 3200/32000]
Avg. loss: 0.000015, [current: 6400/32000]
Avg. loss: 0.000123, [current: 9600/32000]
Avg. loss: 0.000012, [current:12800/32000]
Avg. loss: 0.000088, [current:16000/32000]
Avg. loss: 0.000002, [current:19200/32000]
Avg. loss: 0.000013, [current:22400/32000]
Avg. loss: 0.000003, [current:25600/32000]
Avg. loss: 0.000155, [current:28800/32000]
Avg. loss: 0.000224, [current:32000/32000]
Test Error:
 Accuracy: 97.2%, Avg. loss: 0.000058

Epoch 17
------------------------------
Avg. loss: 0.000051, [current: 3200/32000]
Avg. loss: 0.000056, [current: 6400/32000]
Avg. loss: 0.000024, [current: 9600/32000]
Avg. loss: 0.000003, [current:12800/32000]
Avg. loss: 0.000042, [current:16000/32000]
Avg. loss: 0.000003, [current:19200/32000]
Avg. loss: 0.000015, [current:22400/32000]
Avg. loss: 0.000200, [current:25600/32000]
Avg. loss: 0.000020, [current:28800/32000]
Avg. loss: 0.000026, [current:32000/32000]
Test Error:
 Accuracy: 97.5%, Avg. loss: 0.000038

Epoch 18
------------------------------
Avg. loss: 0.000005, [current: 3200/32000]
Avg. loss: 0.000002, [current: 6400/32000]
Avg. loss: 0.000029, [current: 9600/32000]
Avg. loss: 0.000612, [current:12800/32000]
Avg. loss: 0.000002, [current:16000/32000]
Avg. loss: 0.000032, [current:19200/32000]
Avg. loss: 0.000006, [current:22400/32000]
Avg. loss: 0.000002, [current:25600/32000]
Avg. loss: 0.000001, [current:28800/32000]
Avg. loss: 0.000011, [current:32000/32000]
Test Error:
 Accuracy: 98.6%, Avg. loss: 0.000088

Epoch 19
------------------------------
Avg. loss: 0.000051, [current: 3200/32000]
Avg. loss: 0.000028, [current: 6400/32000]
Avg. loss: 0.000003, [current: 9600/32000]
Avg. loss: 0.000180, [current:12800/32000]
Avg. loss: 0.000002, [current:16000/32000]
Avg. loss: 0.000002, [current:19200/32000]
Avg. loss: 0.000021, [current:22400/32000]
Avg. loss: 0.000001, [current:25600/32000]
Avg. loss: 0.000005, [current:28800/32000]
Avg. loss: 0.000029, [current:32000/32000]
Test Error:
 Accuracy: 97.7%, Avg. loss: 0.000027

Epoch 20
------------------------------
Avg. loss: 0.000002, [current: 3200/32000]
Avg. loss: 0.000090, [current: 6400/32000]
Avg. loss: 0.000008, [current: 9600/32000]
Avg. loss: 0.000049, [current:12800/32000]
Avg. loss: 0.000003, [current:16000/32000]
Avg. loss: 0.000004, [current:19200/32000]
Avg. loss: 0.000006, [current:22400/32000]
Avg. loss: 0.000011, [current:25600/32000]
Avg. loss: 0.000015, [current:28800/32000]
Avg. loss: 0.000005, [current:32000/32000]
Test Error:
 Accuracy: 97.6%, Avg. loss: 0.000045

Epoch 21
------------------------------
Avg. loss: 0.000012, [current: 3200/32000]
Avg. loss: 0.000170, [current: 6400/32000]
Avg. loss: 0.000064, [current: 9600/32000]
Avg. loss: 0.000009, [current:12800/32000]
Avg. loss: 0.000063, [current:16000/32000]
Avg. loss: 0.000024, [current:19200/32000]
Avg. loss: 0.000021, [current:22400/32000]
Avg. loss: 0.000077, [current:25600/32000]
Avg. loss: 0.000001, [current:28800/32000]
Avg. loss: 0.000012, [current:32000/32000]
Test Error:
 Accuracy: 97.0%, Avg. loss: 0.000087

Epoch 22
------------------------------
Avg. loss: 0.000050, [current: 3200/32000]
Avg. loss: 0.000028, [current: 6400/32000]
Avg. loss: 0.000001, [current: 9600/32000]
Avg. loss: 0.000001, [current:12800/32000]
Avg. loss: 0.000003, [current:16000/32000]
Avg. loss: 0.000001, [current:19200/32000]
Avg. loss: 0.000001, [current:22400/32000]
Avg. loss: 0.000001, [current:25600/32000]
Avg. loss: 0.000001, [current:28800/32000]
Avg. loss: 0.000016, [current:32000/32000]
Test Error:
 Accuracy: 97.5%, Avg. loss: 0.000109

Epoch 23
------------------------------
Avg. loss: 0.000219, [current: 3200/32000]
Avg. loss: 0.000231, [current: 6400/32000]
Avg. loss: 0.000001, [current: 9600/32000]
Avg. loss: 0.000206, [current:12800/32000]
Avg. loss: 0.000001, [current:16000/32000]
Avg. loss: 0.000001, [current:19200/32000]
Avg. loss: 0.000001, [current:22400/32000]
Avg. loss: 0.000035, [current:25600/32000]
Avg. loss: 0.000001, [current:28800/32000]
Avg. loss: 0.000002, [current:32000/32000]
Test Error:
 Accuracy: 97.9%, Avg. loss: 0.000065

Epoch 24
------------------------------
Avg. loss: 0.000010, [current: 3200/32000]
Avg. loss: 0.000007, [current: 6400/32000]
Avg. loss: 0.000001, [current: 9600/32000]
Avg. loss: 0.000006, [current:12800/32000]
Avg. loss: 0.000001, [current:16000/32000]
Avg. loss: 0.000010, [current:19200/32000]
Avg. loss: 0.000001, [current:22400/32000]
Avg. loss: 0.000006, [current:25600/32000]
Avg. loss: 0.000161, [current:28800/32000]
Avg. loss: 0.000009, [current:32000/32000]
Test Error:
 Accuracy: 98.6%, Avg. loss: 0.000026

Epoch 25
------------------------------
Avg. loss: 0.000002, [current: 3200/32000]
Avg. loss: 0.000045, [current: 6400/32000]
Avg. loss: 0.000001, [current: 9600/32000]
Avg. loss: 0.000002, [current:12800/32000]
Avg. loss: 0.000001, [current:16000/32000]
Avg. loss: 0.000771, [current:19200/32000]
Avg. loss: 0.000002, [current:22400/32000]
Avg. loss: 0.000045, [current:25600/32000]
Avg. loss: 0.000008, [current:28800/32000]
Avg. loss: 0.000021, [current:32000/32000]
Test Error:
 Accuracy: 98.7%, Avg. loss: 0.000024

Epoch 26
------------------------------
Avg. loss: 0.000004, [current: 3200/32000]
Avg. loss: 0.000001, [current: 6400/32000]
Avg. loss: 0.000001, [current: 9600/32000]
Avg. loss: 0.000003, [current:12800/32000]
Avg. loss: 0.000008, [current:16000/32000]
Avg. loss: 0.000088, [current:19200/32000]
Avg. loss: 0.000051, [current:22400/32000]
Avg. loss: 0.000018, [current:25600/32000]
Avg. loss: 0.000032, [current:28800/32000]
Avg. loss: 0.000001, [current:32000/32000]
Test Error:
 Accuracy: 99.0%, Avg. loss: 0.000017

Epoch 27
------------------------------
```

```
Avg. loss: 0.000001, [current: 3200/32000]
Avg. loss: 0.000004, [current: 6400/32000]
Avg. loss: 0.000001, [current: 9600/32000]
Avg. loss: 0.000979, [current:12800/32000]
Avg. loss: 0.000001, [current:16000/32000]
Avg. loss: 0.000005, [current:19200/32000]
Avg. loss: 0.000001, [current:22400/32000]
Avg. loss: 0.000001, [current:25600/32000]
Avg. loss: 0.000002, [current:28800/32000]
Avg. loss: 0.000001, [current:32000/32000]
Test Error:
 Accuracy: 98.9%, Avg. loss: 0.000033

Epoch 28
------------------------------
Avg. loss: 0.000020, [current: 3200/32000]
Avg. loss: 0.000004, [current: 6400/32000]
Avg. loss: 0.000052, [current: 9600/32000]
Avg. loss: 0.000018, [current:12800/32000]
Avg. loss: 0.000001, [current:16000/32000]
Avg. loss: 0.000002, [current:19200/32000]
Avg. loss: 0.000001, [current:22400/32000]
Avg. loss: 0.000035, [current:25600/32000]
Avg. loss: 0.000002, [current:28800/32000]
Avg. loss: 0.000102, [current:32000/32000]
Test Error:
 Accuracy: 98.3%, Avg. loss: 0.000026

Epoch 29
------------------------------
Avg. loss: 0.000001, [current: 3200/32000]
Avg. loss: 0.000009, [current: 6400/32000]
Avg. loss: 0.000020, [current: 9600/32000]
Avg. loss: 0.000003, [current:12800/32000]
Avg. loss: 0.000003, [current:16000/32000]
Avg. loss: 0.000099, [current:19200/32000]
Avg. loss: 0.000000, [current:22400/32000]
Avg. loss: 0.000027, [current:25600/32000]
Avg. loss: 0.000012, [current:28800/32000]
Avg. loss: 0.000001, [current:32000/32000]
Test Error:
 Accuracy: 98.9%, Avg. loss: 0.000049

Epoch 30
------------------------------
Avg. loss: 0.000011, [current: 3200/32000]
Avg. loss: 0.000008, [current: 6400/32000]
Avg. loss: 0.000015, [current: 9600/32000]
Avg. loss: 0.000001, [current:12800/32000]
Avg. loss: 0.000000, [current:16000/32000]
Avg. loss: 0.000014, [current:19200/32000]
Avg. loss: 0.000046, [current:22400/32000]
Avg. loss: 0.000014, [current:25600/32000]
Avg. loss: 0.000289, [current:28800/32000]
Avg. loss: 0.000001, [current:32000/32000]
Test Error:
 Accuracy: 98.5%, Avg. loss: 0.000059

Epoch 31
------------------------------
Avg. loss: 0.000019, [current: 3200/32000]
Avg. loss: 0.000012, [current: 6400/32000]
Avg. loss: 0.000000, [current: 9600/32000]
Avg. loss: 0.000337, [current:12800/32000]
Avg. loss: 0.000018, [current:16000/32000]
Avg. loss: 0.000001, [current:19200/32000]
Avg. loss: 0.000000, [current:22400/32000]
Avg. loss: 0.000001, [current:25600/32000]
Avg. loss: 0.000001, [current:28800/32000]
Avg. loss: 0.000017, [current:32000/32000]
Test Error:
 Accuracy: 98.9%, Avg. loss: 0.000024

Epoch 32
------------------------------
Avg. loss: 0.000006, [current: 3200/32000]
Avg. loss: 0.000004, [current: 6400/32000]
Avg. loss: 0.000001, [current: 9600/32000]
Avg. loss: 0.000010, [current:12800/32000]
Avg. loss: 0.000083, [current:16000/32000]
Avg. loss: 0.000106, [current:19200/32000]
Avg. loss: 0.000001, [current:22400/32000]
Avg. loss: 0.000003, [current:25600/32000]
Avg. loss: 0.000001, [current:28800/32000]
Avg. loss: 0.000001, [current:32000/32000]
Test Error:
 Accuracy: 98.8%, Avg. loss: 0.000015

Epoch 33
------------------------------
Avg. loss: 0.000001, [current: 3200/32000]
Avg. loss: 0.000085, [current: 6400/32000]
Avg. loss: 0.000003, [current: 9600/32000]
Avg. loss: 0.000034, [current:12800/32000]
Avg. loss: 0.000001, [current:16000/32000]
Avg. loss: 0.000001, [current:19200/32000]
Avg. loss: 0.000007, [current:22400/32000]
Avg. loss: 0.000008, [current:25600/32000]
Avg. loss: 0.000002, [current:28800/32000]
Avg. loss: 0.000001, [current:32000/32000]
Test Error:
 Accuracy: 98.4%, Avg. loss: 0.000047

Epoch 34
------------------------------
Avg. loss: 0.000000, [current: 3200/32000]
Avg. loss: 0.000003, [current: 6400/32000]
Avg. loss: 0.000007, [current: 9600/32000]
Avg. loss: 0.000015, [current:12800/32000]
Avg. loss: 0.000022, [current:16000/32000]
Avg. loss: 0.000017, [current:19200/32000]
Avg. loss: 0.000000, [current:22400/32000]
Avg. loss: 0.000016, [current:25600/32000]
Avg. loss: 0.000001, [current:28800/32000]
Avg. loss: 0.000002, [current:32000/32000]
Test Error:
 Accuracy: 98.8%, Avg. loss: 0.000018

Epoch 35
------------------------------
Avg. loss: 0.000004, [current: 3200/32000]
Avg. loss: 0.000007, [current: 6400/32000]
Avg. loss: 0.000000, [current: 9600/32000]
Avg. loss: 0.000001, [current:12800/32000]
Avg. loss: 0.000000, [current:16000/32000]
Avg. loss: 0.000000, [current:19200/32000]
Avg. loss: 0.000000, [current:22400/32000]
Avg. loss: 0.000003, [current:25600/32000]
Avg. loss: 0.000015, [current:28800/32000]
Avg. loss: 0.000003, [current:32000/32000]
Test Error:
 Accuracy: 98.9%, Avg. loss: 0.000148

Epoch 36
------------------------------
Avg. loss: 0.000007, [current: 3200/32000]
Avg. loss: 0.000000, [current: 6400/32000]
Avg. loss: 0.000000, [current: 9600/32000]
Avg. loss: 0.000000, [current:12800/32000]
Avg. loss: 0.000007, [current:16000/32000]
Avg. loss: 0.000003, [current:19200/32000]
Avg. loss: 0.000002, [current:22400/32000]
Avg. loss: 0.000001, [current:25600/32000]
Avg. loss: 0.000007, [current:28800/32000]
Avg. loss: 0.000032, [current:32000/32000]
Test Error:
 Accuracy: 99.1%, Avg. loss: 0.000010

Epoch 37
------------------------------
Avg. loss: 0.000014, [current: 3200/32000]
Avg. loss: 0.000023, [current: 6400/32000]
Avg. loss: 0.000003, [current: 9600/32000]
Avg. loss: 0.000030, [current:12800/32000]
Avg. loss: 0.000019, [current:16000/32000]
Avg. loss: 0.000002, [current:19200/32000]
Avg. loss: 0.000007, [current:22400/32000]
Avg. loss: 0.000054, [current:25600/32000]
Avg. loss: 0.000008, [current:28800/32000]
Avg. loss: 0.000001, [current:32000/32000]
Test Error:
 Accuracy: 99.0%, Avg. loss: 0.000065

Epoch 38
------------------------------
Avg. loss: 0.000004, [current: 3200/32000]
Avg. loss: 0.000006, [current: 6400/32000]
Avg. loss: 0.000000, [current: 9600/32000]
Avg. loss: 0.000055, [current:12800/32000]
Avg. loss: 0.000014, [current:16000/32000]
Avg. loss: 0.000000, [current:19200/32000]
Avg. loss: 0.000001, [current:22400/32000]
Avg. loss: 0.000000, [current:25600/32000]
Avg. loss: 0.000002, [current:28800/32000]
Avg. loss: 0.000020, [current:32000/32000]
Test Error:
 Accuracy: 99.0%, Avg. loss: 0.000018

Epoch 39
------------------------------
Avg. loss: 0.000000, [current: 3200/32000]
Avg. loss: 0.000000, [current: 6400/32000]
Avg. loss: 0.000001, [current: 9600/32000]
Avg. loss: 0.000001, [current:12800/32000]
Avg. loss: 0.000003, [current:16000/32000]
Avg. loss: 0.000006, [current:19200/32000]
Avg. loss: 0.000003, [current:22400/32000]
Avg. loss: 0.000045, [current:25600/32000]
Avg. loss: 0.000000, [current:28800/32000]
Avg. loss: 0.000124, [current:32000/32000]
Test Error:
 Accuracy: 98.9%, Avg. loss: 0.000059

Epoch 40
------------------------------
Avg. loss: 0.000026, [current: 3200/32000]
Avg. loss: 0.000001, [current: 6400/32000]
Avg. loss: 0.000047, [current: 9600/32000]
Avg. loss: 0.000025, [current:12800/32000]
Avg. loss: 0.000045, [current:16000/32000]
Avg. loss: 0.000089, [current:19200/32000]
Avg. loss: 0.000000, [current:22400/32000]
Avg. loss: 0.000000, [current:25600/32000]
Avg. loss: 0.000000, [current:28800/32000]
Avg. loss: 0.000002, [current:32000/32000]
Test Error:
 Accuracy: 98.3%, Avg. loss: 0.000033

Epoch 41
------------------------------
Avg. loss: 0.000003, [current: 3200/32000]
Avg. loss: 0.000069, [current: 6400/32000]
Avg. loss: 0.000299, [current: 9600/32000]
Avg. loss: 0.000000, [current:12800/32000]
Avg. loss: 0.000001, [current:16000/32000]
Avg. loss: 0.000032, [current:19200/32000]
Avg. loss: 0.000002, [current:22400/32000]
Avg. loss: 0.000141, [current:25600/32000]
Avg. loss: 0.000000, [current:28800/32000]
Avg. loss: 0.000000, [current:32000/32000]
Test Error:
 Accuracy: 98.6%, Avg. loss: 0.000013

Epoch 42
```

```
------------------------------
Avg. loss: 0.000000, [current: 3200/32000]
Avg. loss: 0.000001, [current: 6400/32000]
Avg. loss: 0.000001, [current: 9600/32000]
Avg. loss: 0.000010, [current:12800/32000]
Avg. loss: 0.000002, [current:16000/32000]
Avg. loss: 0.000000, [current:19200/32000]
Avg. loss: 0.000012, [current:22400/32000]
Avg. loss: 0.000017, [current:25600/32000]
Avg. loss: 0.000000, [current:28800/32000]
Avg. loss: 0.000000, [current:32000/32000]
Test Error:
 Accuracy: 99.4%, Avg. loss: 0.000005

Epoch 43
------------------------------
Avg. loss: 0.000000, [current: 3200/32000]
Avg. loss: 0.000000, [current: 6400/32000]
Avg. loss: 0.000000, [current: 9600/32000]
Avg. loss: 0.000008, [current:12800/32000]
Avg. loss: 0.000000, [current:16000/32000]
Avg. loss: 0.000001, [current:19200/32000]
Avg. loss: 0.000011, [current:22400/32000]
Avg. loss: 0.000024, [current:25600/32000]
Avg. loss: 0.000075, [current:28800/32000]
Avg. loss: 0.000001, [current:32000/32000]
Test Error:
 Accuracy: 98.8%, Avg. loss: 0.000024

Epoch 44
------------------------------
Avg. loss: 0.000000, [current: 3200/32000]
Avg. loss: 0.000005, [current: 6400/32000]
Avg. loss: 0.000000, [current: 9600/32000]
Avg. loss: 0.000012, [current:12800/32000]
Avg. loss: 0.000000, [current:16000/32000]
Avg. loss: 0.000041, [current:19200/32000]
Avg. loss: 0.000027, [current:22400/32000]
Avg. loss: 0.000018, [current:25600/32000]
Avg. loss: 0.000007, [current:28800/32000]
Avg. loss: 0.000002, [current:32000/32000]
Test Error:
 Accuracy: 99.1%, Avg. loss: 0.000007

Epoch 45
------------------------------
Avg. loss: 0.000020, [current: 3200/32000]
Avg. loss: 0.000000, [current: 6400/32000]
Avg. loss: 0.000000, [current: 9600/32000]
Avg. loss: 0.000000, [current:12800/32000]
Avg. loss: 0.000000, [current:16000/32000]
Avg. loss: 0.000099, [current:19200/32000]
Avg. loss: 0.000000, [current:22400/32000]
Avg. loss: 0.000000, [current:25600/32000]
Avg. loss: 0.000000, [current:28800/32000]
Avg. loss: 0.000014, [current:32000/32000]
```

```
Test Error:
 Accuracy: 99.2%, Avg. loss: 0.000017

Epoch 46
-------------------------------
Avg. loss: 0.000000, [current: 3200/32000]
Avg. loss: 0.000000, [current: 6400/32000]
Avg. loss: 0.000000, [current: 9600/32000]
Avg. loss: 0.000000, [current:12800/32000]
Avg. loss: 0.000032, [current:16000/32000]
Avg. loss: 0.000036, [current:19200/32000]
Avg. loss: 0.000182, [current:22400/32000]
Avg. loss: 0.000007, [current:25600/32000]
Avg. loss: 0.000000, [current:28800/32000]
Avg. loss: 0.000000, [current:32000/32000]
Test Error:
 Accuracy: 99.0%, Avg. loss: 0.000015

Epoch 47
-------------------------------
Avg. loss: 0.000051, [current: 3200/32000]
Avg. loss: 0.000023, [current: 6400/32000]
Avg. loss: 0.000003, [current: 9600/32000]
Avg. loss: 0.000007, [current:12800/32000]
Avg. loss: 0.000006, [current:16000/32000]
Avg. loss: 0.000000, [current:19200/32000]
Avg. loss: 0.000000, [current:22400/32000]
Avg. loss: 0.000000, [current:25600/32000]
Avg. loss: 0.000003, [current:28800/32000]
Avg. loss: 0.000013, [current:32000/32000]
Test Error:
 Accuracy: 99.0%, Avg. loss: 0.000014

Epoch 48
-------------------------------
Avg. loss: 0.000074, [current: 3200/32000]
Avg. loss: 0.000000, [current: 6400/32000]
Avg. loss: 0.000205, [current: 9600/32000]
Avg. loss: 0.000000, [current:12800/32000]
Avg. loss: 0.000000, [current:16000/32000]
Avg. loss: 0.000000, [current:19200/32000]
Avg. loss: 0.000000, [current:22400/32000]
Avg. loss: 0.000000, [current:25600/32000]
Avg. loss: 0.000002, [current:28800/32000]
Avg. loss: 0.000002, [current:32000/32000]
Test Error:
 Accuracy: 99.2%, Avg. loss: 0.000015

Epoch 49
-------------------------------
Avg. loss: 0.000005, [current: 3200/32000]
Avg. loss: 0.000000, [current: 6400/32000]
Avg. loss: 0.000002, [current: 9600/32000]
Avg. loss: 0.000009, [current:12800/32000]
Avg. loss: 0.000001, [current:16000/32000]
Avg. loss: 0.000029, [current:19200/32000]
Avg. loss: 0.000000, [current:22400/32000]
Avg. loss: 0.000002, [current:25600/32000]
Avg. loss: 0.000002, [current:28800/32000]
Avg. loss: 0.000000, [current:32000/32000]
Test Error:
 Accuracy: 98.8%, Avg. loss: 0.000014

Epoch 50
------------------------------
Avg. loss: 0.000000, [current: 3200/32000]
Avg. loss: 0.000000, [current: 6400/32000]
Avg. loss: 0.000040, [current: 9600/32000]
Avg. loss: 0.000023, [current:12800/32000]
Avg. loss: 0.000026, [current:16000/32000]
Avg. loss: 0.000000, [current:19200/32000]
Avg. loss: 0.000000, [current:22400/32000]
Avg. loss: 0.000000, [current:25600/32000]
Avg. loss: 0.000083, [current:28800/32000]
Avg. loss: 0.000000, [current:32000/32000]
Test Error:
 Accuracy: 99.0%, Avg. loss: 0.000055

Done
```

Saving and Loading PyTorch Models

Reference: The Linux Foundation, "Save and Load the Model - PyTorch Tutorials 2.6.0 +cu124 documentation," pytorch.org https://pytorch.org/tutorials/beginner/basics/saveloadrun_tutorial.html (accessed Mar. 24, 2025).

Reference: The Linux Foundation, "Saving and Loading Models - PyTorch Tutorials 2.6.0 +cu124 documentation," pytorch.org https://pytorch.org/tutorials/beginner/saving_loading_models.html (accessed Mar. 24, 2025).

Problem #9: Finish the implementation of the save_model_checkpoint funciton. Add the elements to the save_dict dictionary corresponding to keys, "model_state_dict", "optimizer_state_dict", and "epoch". The values for these should be the .state_dict() for the model and optimizer and the epoch. Secondly, add the function to save the save_dict as a file given in file_path. Hint: Use torch.save.

In [12]:

```
def save_model_checkpoint(model, optimizer, dataloader, epoch, file_path):
    dataloader_mapping = dataloader.dataset.mapping
    save_dict = dict(
        ### BEGIN SOLUTION ###
        model_state_dict = model.state_dict(),
        optimizer_state_dict = optimizer.state_dict(),
        epoch = epoch,
        ### END SOLUTION ###
        dataloader_mapping = dataloader_mapping,
    )
    ### BEGIN SOLUTION ###
    torch.save(save_dict,file_path)
    ### END SOLUTION ###
    return True

from pathlib import Path
assert save_model_checkpoint(NNModel(3,5,20), optim.SGD((NNModel(3,5,20)).parameter
assert not set(['model_state_dict','optimizer_state_dict','epoch','dataloader_mappi
for key, val in (torch.load(Path() / "test_checkpoint.pth", weights_only=True)).ite
    if 'dict' in key:
        assert isinstance(val, dict)
    elif key == 'epoch':
        assert val == 20
```

Problem #9: Finish the implementation of the "restore_model_checkpoint" function. Load the checkpoint file defined at "file_path" using torch.load(...). Update the "model" and "optimizer" state_dicts from the checkpoint. Update the "epoch" variable from the checkpoint.

In [13]:

```
def restore_model_checkpoint(model, optimizer, train_dataloader, test_dataloader, f
    epoch = -1
    if file_path.exists():
        print(f"Restarting from checkpoint: {str(file_path)}")
        ### BEGIN SOLUTION ###
        checkpoint = torch.load(file_path, weights_only=True)
        model.load_state_dict(checkpoint['model_state_dict'])
        optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
        epoch = checkpoint['epoch']
        ### END SOLUTION
        train_dataloader.dataset.mapping = checkpoint['dataloader_mapping']
        test_dataloader.dataset.mapping = checkpoint['dataloader_mapping']
    return epoch

test_model = NNModel(3,5,20)
test_optim = optim.SGD(test_model.parameters(), lr=learning_rate)
assert restore_model_checkpoint(test_model,test_optim,DataLoader(RandDataset(3,5,ba
(Path() / "test_checkpoint.pth").unlink()
```

```
Restarting from checkpoint: test_checkpoint.pth
```

Problem #10: Reimplement the "epoch_loop" with the restore_model_checkpoint and save_model_checkpoint functions. The epoch returned by restore_model_checkpoint should be saved to the "epoch_last" variable.

In [14]:

```
def epoch_loop(epochs, train_dataloader, test_dataloader, model, loss_fn, optimizer
    if file_path:
        ### BEGIN SOLUTION ###
        epoch_last = restore_model_checkpoint(model, optimizer, train_dataloader, t
        ### END SOLUTION ###
    for t in range(epoch_last+1,epochs):
        print(f"Epoch {t+1}\n-------------------------------")
        train_loop(train_dataloader, model, loss_fn, optimizer, device)
        test_loop(test_dataloader, model, loss_fn, tolerance, device)
        if file_path:
            ### BEGIN SOLUTION ###
            save_model_checkpoint(model, optimizer, train_dataloader, t, file_path)
            ### END SOLUTION ###
    print("Done")
    return True

use_model = NNModel(input_dims, output_dims, 20).to(device)
use_optimizer = optim.SGD(use_model.parameters(), lr=learning_rate)
my_epochs = 3
assert epoch_loop(my_epochs, train_dataloader, test_dataloader, use_model, my_loss_
if torch.cuda.is_available():
    my_epochs = epochs
else:
    my_epochs = 6
assert epoch_loop(my_epochs, train_dataloader, test_dataloader, use_model, my_loss_
(Path()/"my_checkpoint_file.pth").unlink()
```

```
Epoch 1
-------------------------------
Avg. loss: 0.688889, [current: 3200/32000]
Avg. loss: 0.300688, [current: 6400/32000]
Avg. loss: 0.207974, [current: 9600/32000]
Avg. loss: 0.161903, [current:12800/32000]
Avg. loss: 0.165507, [current:16000/32000]
Avg. loss: 0.160524, [current:19200/32000]
Avg. loss: 0.169589, [current:22400/32000]
Avg. loss: 0.153118, [current:25600/32000]
Avg. loss: 0.108491, [current:28800/32000]
Avg. loss: 0.100971, [current:32000/32000]
Test Error:
 Accuracy: 0.0%, Avg. loss: 0.111498

Epoch 2
-------------------------------
Avg. loss: 0.103993, [current: 3200/32000]
Avg. loss: 0.093950, [current: 6400/32000]
Avg. loss: 0.069131, [current: 9600/32000]
Avg. loss: 0.050902, [current:12800/32000]
Avg. loss: 0.056272, [current:16000/32000]
Avg. loss: 0.042546, [current:19200/32000]
Avg. loss: 0.035998, [current:22400/32000]
Avg. loss: 0.028343, [current:25600/32000]
Avg. loss: 0.022143, [current:28800/32000]
Avg. loss: 0.023384, [current:32000/32000]
Test Error:
 Accuracy: 0.2%, Avg. loss: 0.017768

Epoch 3
-------------------------------
Avg. loss: 0.016162, [current: 3200/32000]
Avg. loss: 0.010472, [current: 6400/32000]
Avg. loss: 0.007944, [current: 9600/32000]
Avg. loss: 0.006801, [current:12800/32000]
Avg. loss: 0.004342, [current:16000/32000]
Avg. loss: 0.003429, [current:19200/32000]
Avg. loss: 0.002728, [current:22400/32000]
Avg. loss: 0.002350, [current:25600/32000]
Avg. loss: 0.002087, [current:28800/32000]
Avg. loss: 0.001348, [current:32000/32000]
Test Error:
 Accuracy: 7.0%, Avg. loss: 0.002230

Done
Restarting from checkpoint: my_checkpoint_file.pth
Epoch 4
-------------------------------
Avg. loss: 0.001103, [current: 3200/32000]
Avg. loss: 0.000698, [current: 6400/32000]
Avg. loss: 0.001993, [current: 9600/32000]
Avg. loss: 0.001170, [current:12800/32000]
Avg. loss: 0.001497, [current:16000/32000]
Avg. loss: 0.001427, [current:19200/32000]
Avg. loss: 0.001197, [current:22400/32000]
Avg. loss: 0.001315, [current:25600/32000]
Avg. loss: 0.000369, [current:28800/32000]
Avg. loss: 0.001260, [current:32000/32000]
Test Error:
 Accuracy: 41.3%, Avg. loss: 0.000775

Epoch 5
------------------------------
Avg. loss: 0.000614, [current: 3200/32000]
Avg. loss: 0.000647, [current: 6400/32000]
Avg. loss: 0.000366, [current: 9600/32000]
Avg. loss: 0.000267, [current:12800/32000]
Avg. loss: 0.001092, [current:16000/32000]
Avg. loss: 0.001104, [current:19200/32000]
Avg. loss: 0.000758, [current:22400/32000]
Avg. loss: 0.000551, [current:25600/32000]
Avg. loss: 0.000522, [current:28800/32000]
Avg. loss: 0.000787, [current:32000/32000]
Test Error:
 Accuracy: 51.3%, Avg. loss: 0.000695

Epoch 6
------------------------------
Avg. loss: 0.001064, [current: 3200/32000]
Avg. loss: 0.000580, [current: 6400/32000]
Avg. loss: 0.000513, [current: 9600/32000]
Avg. loss: 0.000629, [current:12800/32000]
Avg. loss: 0.000368, [current:16000/32000]
Avg. loss: 0.000305, [current:19200/32000]
Avg. loss: 0.000490, [current:22400/32000]
Avg. loss: 0.000828, [current:25600/32000]
Avg. loss: 0.000267, [current:28800/32000]
Avg. loss: 0.000248, [current:32000/32000]
Test Error:
 Accuracy: 58.8%, Avg. loss: 0.000614

Epoch 7
------------------------------
Avg. loss: 0.000453, [current: 3200/32000]
Avg. loss: 0.000231, [current: 6400/32000]
Avg. loss: 0.000764, [current: 9600/32000]
Avg. loss: 0.000282, [current:12800/32000]
Avg. loss: 0.000759, [current:16000/32000]
Avg. loss: 0.000249, [current:19200/32000]
Avg. loss: 0.000639, [current:22400/32000]
Avg. loss: 0.000373, [current:25600/32000]
Avg. loss: 0.001084, [current:28800/32000]
Avg. loss: 0.000612, [current:32000/32000]
Test Error:
 Accuracy: 61.3%, Avg. loss: 0.000571

Epoch 8
------------------------------
Avg. loss: 0.000411, [current: 3200/32000]
Avg. loss: 0.000386, [current: 6400/32000]
Avg. loss: 0.000562, [current: 9600/32000]
Avg. loss: 0.001299, [current:12800/32000]
Avg. loss: 0.001211, [current:16000/32000]
Avg. loss: 0.000445, [current:19200/32000]
Avg. loss: 0.000417, [current:22400/32000]
Avg. loss: 0.000440, [current:25600/32000]
Avg. loss: 0.000215, [current:28800/32000]
Avg. loss: 0.000498, [current:32000/32000]
Test Error:
 Accuracy: 70.5%, Avg. loss: 0.000352

Epoch 9
------------------------------
Avg. loss: 0.000425, [current: 3200/32000]
Avg. loss: 0.000368, [current: 6400/32000]
Avg. loss: 0.000341, [current: 9600/32000]
Avg. loss: 0.000341, [current:12800/32000]
Avg. loss: 0.000418, [current:16000/32000]
Avg. loss: 0.000193, [current:19200/32000]
Avg. loss: 0.000600, [current:22400/32000]
Avg. loss: 0.000365, [current:25600/32000]
Avg. loss: 0.000354, [current:28800/32000]
Avg. loss: 0.000223, [current:32000/32000]
Test Error:
 Accuracy: 74.5%, Avg. loss: 0.000296

Epoch 10
------------------------------
Avg. loss: 0.000286, [current: 3200/32000]
Avg. loss: 0.000290, [current: 6400/32000]
Avg. loss: 0.000442, [current: 9600/32000]
Avg. loss: 0.000131, [current:12800/32000]
Avg. loss: 0.000512, [current:16000/32000]
Avg. loss: 0.000396, [current:19200/32000]
Avg. loss: 0.000604, [current:22400/32000]
Avg. loss: 0.000377, [current:25600/32000]
Avg. loss: 0.000351, [current:28800/32000]
Avg. loss: 0.000199, [current:32000/32000]
Test Error:
 Accuracy: 79.5%, Avg. loss: 0.000327

Epoch 11
------------------------------
Avg. loss: 0.000262, [current: 3200/32000]
Avg. loss: 0.000345, [current: 6400/32000]
Avg. loss: 0.000134, [current: 9600/32000]
Avg. loss: 0.000246, [current:12800/32000]
Avg. loss: 0.000667, [current:16000/32000]
Avg. loss: 0.000101, [current:19200/32000]
Avg. loss: 0.000759, [current:22400/32000]
Avg. loss: 0.000409, [current:25600/32000]
Avg. loss: 0.000193, [current:28800/32000]
Avg. loss: 0.000315, [current:32000/32000]
Test Error:
 Accuracy: 83.4%, Avg. loss: 0.000249

Epoch 12
```

```
------------------------------
Avg. loss: 0.000098, [current: 3200/32000]
Avg. loss: 0.000411, [current: 6400/32000]
Avg. loss: 0.000194, [current: 9600/32000]
Avg. loss: 0.000175, [current:12800/32000]
Avg. loss: 0.000206, [current:16000/32000]
Avg. loss: 0.000085, [current:19200/32000]
Avg. loss: 0.000438, [current:22400/32000]
Avg. loss: 0.000202, [current:25600/32000]
Avg. loss: 0.000264, [current:28800/32000]
Avg. loss: 0.000139, [current:32000/32000]
Test Error:
 Accuracy: 85.9%, Avg. loss: 0.000415

Epoch 13
------------------------------
Avg. loss: 0.000274, [current: 3200/32000]
Avg. loss: 0.000151, [current: 6400/32000]
Avg. loss: 0.000145, [current: 9600/32000]
Avg. loss: 0.000250, [current:12800/32000]
Avg. loss: 0.000179, [current:16000/32000]
Avg. loss: 0.000321, [current:19200/32000]
Avg. loss: 0.000519, [current:22400/32000]
Avg. loss: 0.000214, [current:25600/32000]
Avg. loss: 0.000126, [current:28800/32000]
Avg. loss: 0.000238, [current:32000/32000]
Test Error:
 Accuracy: 87.4%, Avg. loss: 0.000265

Epoch 14
------------------------------
Avg. loss: 0.000262, [current: 3200/32000]
Avg. loss: 0.000360, [current: 6400/32000]
Avg. loss: 0.000122, [current: 9600/32000]
Avg. loss: 0.000190, [current:12800/32000]
Avg. loss: 0.000102, [current:16000/32000]
Avg. loss: 0.000173, [current:19200/32000]
Avg. loss: 0.000123, [current:22400/32000]
Avg. loss: 0.000346, [current:25600/32000]
Avg. loss: 0.000213, [current:28800/32000]
Avg. loss: 0.000107, [current:32000/32000]
Test Error:
 Accuracy: 87.8%, Avg. loss: 0.000146

Epoch 15
------------------------------
Avg. loss: 0.000117, [current: 3200/32000]
Avg. loss: 0.000110, [current: 6400/32000]
Avg. loss: 0.000413, [current: 9600/32000]
Avg. loss: 0.000077, [current:12800/32000]
Avg. loss: 0.000173, [current:16000/32000]
Avg. loss: 0.000214, [current:19200/32000]
Avg. loss: 0.000116, [current:22400/32000]
Avg. loss: 0.000146, [current:25600/32000]
Avg. loss: 0.000072, [current:28800/32000]
Avg. loss: 0.000098, [current:32000/32000]
```

```
Test Error:
 Accuracy: 91.6%, Avg. loss: 0.000190

Epoch 16
------------------------------
Avg. loss: 0.000152, [current: 3200/32000]
Avg. loss: 0.000082, [current: 6400/32000]
Avg. loss: 0.000068, [current: 9600/32000]
Avg. loss: 0.000213, [current:12800/32000]
Avg. loss: 0.000144, [current:16000/32000]
Avg. loss: 0.000145, [current:19200/32000]
Avg. loss: 0.000093, [current:22400/32000]
Avg. loss: 0.000115, [current:25600/32000]
Avg. loss: 0.000148, [current:28800/32000]
Avg. loss: 0.000108, [current:32000/32000]
Test Error:
 Accuracy: 91.0%, Avg. loss: 0.000186

Epoch 17
------------------------------
Avg. loss: 0.000113, [current: 3200/32000]
Avg. loss: 0.000217, [current: 6400/32000]
Avg. loss: 0.000107, [current: 9600/32000]
Avg. loss: 0.000087, [current:12800/32000]
Avg. loss: 0.000161, [current:16000/32000]
Avg. loss: 0.000078, [current:19200/32000]
Avg. loss: 0.000047, [current:22400/32000]
Avg. loss: 0.000108, [current:25600/32000]
Avg. loss: 0.000062, [current:28800/32000]
Avg. loss: 0.000084, [current:32000/32000]
Test Error:
 Accuracy: 90.5%, Avg. loss: 0.000122

Epoch 18
------------------------------
Avg. loss: 0.000146, [current: 3200/32000]
Avg. loss: 0.000246, [current: 6400/32000]
Avg. loss: 0.000100, [current: 9600/32000]
Avg. loss: 0.000070, [current:12800/32000]
Avg. loss: 0.000128, [current:16000/32000]
Avg. loss: 0.000151, [current:19200/32000]
Avg. loss: 0.000059, [current:22400/32000]
Avg. loss: 0.000121, [current:25600/32000]
Avg. loss: 0.000465, [current:28800/32000]
Avg. loss: 0.000081, [current:32000/32000]
Test Error:
 Accuracy: 92.4%, Avg. loss: 0.000134

Epoch 19
------------------------------
Avg. loss: 0.000072, [current: 3200/32000]
Avg. loss: 0.000046, [current: 6400/32000]
Avg. loss: 0.000076, [current: 9600/32000]
Avg. loss: 0.000101, [current:12800/32000]
Avg. loss: 0.000067, [current:16000/32000]
Avg. loss: 0.000077, [current:19200/32000]
Avg. loss: 0.000042, [current:22400/32000]
Avg. loss: 0.000161, [current:25600/32000]
Avg. loss: 0.000077, [current:28800/32000]
Avg. loss: 0.000071, [current:32000/32000]
Test Error:
 Accuracy: 93.4%, Avg. loss: 0.000122

Epoch 20
------------------------------
Avg. loss: 0.000037, [current: 3200/32000]
Avg. loss: 0.000352, [current: 6400/32000]
Avg. loss: 0.000108, [current: 9600/32000]
Avg. loss: 0.000381, [current:12800/32000]
Avg. loss: 0.000115, [current:16000/32000]
Avg. loss: 0.000084, [current:19200/32000]
Avg. loss: 0.000121, [current:22400/32000]
Avg. loss: 0.000061, [current:25600/32000]
Avg. loss: 0.000070, [current:28800/32000]
Avg. loss: 0.000189, [current:32000/32000]
Test Error:
 Accuracy: 94.7%, Avg. loss: 0.000092

Epoch 21
------------------------------
Avg. loss: 0.000062, [current: 3200/32000]
Avg. loss: 0.000199, [current: 6400/32000]
Avg. loss: 0.000138, [current: 9600/32000]
Avg. loss: 0.000123, [current:12800/32000]
Avg. loss: 0.000047, [current:16000/32000]
Avg. loss: 0.000047, [current:19200/32000]
Avg. loss: 0.000044, [current:22400/32000]
Avg. loss: 0.000043, [current:25600/32000]
Avg. loss: 0.000051, [current:28800/32000]
Avg. loss: 0.000050, [current:32000/32000]
Test Error:
 Accuracy: 93.6%, Avg. loss: 0.000114

Epoch 22
------------------------------
Avg. loss: 0.000170, [current: 3200/32000]
Avg. loss: 0.000031, [current: 6400/32000]
Avg. loss: 0.000084, [current: 9600/32000]
Avg. loss: 0.000074, [current:12800/32000]
Avg. loss: 0.000113, [current:16000/32000]
Avg. loss: 0.000048, [current:19200/32000]
Avg. loss: 0.000026, [current:22400/32000]
Avg. loss: 0.000027, [current:25600/32000]
Avg. loss: 0.000119, [current:28800/32000]
Avg. loss: 0.000039, [current:32000/32000]
Test Error:
 Accuracy: 95.8%, Avg. loss: 0.000068

Epoch 23
------------------------------
Avg. loss: 0.000026, [current: 3200/32000]
Avg. loss: 0.000286, [current: 6400/32000]
Avg. loss: 0.000027, [current: 9600/32000]
Avg. loss: 0.000054, [current:12800/32000]
Avg. loss: 0.000045, [current:16000/32000]
Avg. loss: 0.000039, [current:19200/32000]
Avg. loss: 0.000054, [current:22400/32000]
Avg. loss: 0.000045, [current:25600/32000]
Avg. loss: 0.000064, [current:28800/32000]
Avg. loss: 0.000030, [current:32000/32000]
Test Error: 
 Accuracy: 95.3%, Avg. loss: 0.000163 

Epoch 24
------------------------------
Avg. loss: 0.000036, [current: 3200/32000]
Avg. loss: 0.000029, [current: 6400/32000]
Avg. loss: 0.000030, [current: 9600/32000]
Avg. loss: 0.000037, [current:12800/32000]
Avg. loss: 0.000032, [current:16000/32000]
Avg. loss: 0.000031, [current:19200/32000]
Avg. loss: 0.000039, [current:22400/32000]
Avg. loss: 0.000128, [current:25600/32000]
Avg. loss: 0.000054, [current:28800/32000]
Avg. loss: 0.000047, [current:32000/32000]
Test Error: 
 Accuracy: 95.7%, Avg. loss: 0.000128 

Epoch 25
------------------------------
Avg. loss: 0.000035, [current: 3200/32000]
Avg. loss: 0.000058, [current: 6400/32000]
Avg. loss: 0.000035, [current: 9600/32000]
Avg. loss: 0.000018, [current:12800/32000]
Avg. loss: 0.000045, [current:16000/32000]
Avg. loss: 0.000028, [current:19200/32000]
Avg. loss: 0.000029, [current:22400/32000]
Avg. loss: 0.000030, [current:25600/32000]
Avg. loss: 0.000043, [current:28800/32000]
Avg. loss: 0.000036, [current:32000/32000]
Test Error: 
 Accuracy: 95.9%, Avg. loss: 0.000065 

Epoch 26
------------------------------
Avg. loss: 0.000027, [current: 3200/32000]
Avg. loss: 0.000123, [current: 6400/32000]
Avg. loss: 0.000047, [current: 9600/32000]
Avg. loss: 0.000074, [current:12800/32000]
Avg. loss: 0.000070, [current:16000/32000]
Avg. loss: 0.000020, [current:19200/32000]
Avg. loss: 0.000058, [current:22400/32000]
Avg. loss: 0.000029, [current:25600/32000]
Avg. loss: 0.000058, [current:28800/32000]
Avg. loss: 0.000044, [current:32000/32000]
Test Error: 
 Accuracy: 96.0%, Avg. loss: 0.000073 
```

```
Epoch 27
-------------------------------
Avg. loss: 0.000050, [current: 3200/32000]
Avg. loss: 0.000015, [current: 6400/32000]
Avg. loss: 0.000032, [current: 9600/32000]
Avg. loss: 0.000096, [current:12800/32000]
Avg. loss: 0.000095, [current:16000/32000]
Avg. loss: 0.000026, [current:19200/32000]
Avg. loss: 0.000081, [current:22400/32000]
Avg. loss: 0.000043, [current:25600/32000]
Avg. loss: 0.000451, [current:28800/32000]
Avg. loss: 0.000019, [current:32000/32000]
Test Error:
 Accuracy: 96.6%, Avg. loss: 0.000071

Epoch 28
-------------------------------
Avg. loss: 0.000152, [current: 3200/32000]
Avg. loss: 0.000023, [current: 6400/32000]
Avg. loss: 0.000089, [current: 9600/32000]
Avg. loss: 0.000022, [current:12800/32000]
Avg. loss: 0.000048, [current:16000/32000]
Avg. loss: 0.000074, [current:19200/32000]
Avg. loss: 0.000030, [current:22400/32000]
Avg. loss: 0.000032, [current:25600/32000]
Avg. loss: 0.000028, [current:28800/32000]
Avg. loss: 0.000033, [current:32000/32000]
Test Error:
 Accuracy: 97.9%, Avg. loss: 0.000078

Epoch 29
-------------------------------
Avg. loss: 0.000045, [current: 3200/32000]
Avg. loss: 0.000024, [current: 6400/32000]
Avg. loss: 0.000014, [current: 9600/32000]
Avg. loss: 0.000043, [current:12800/32000]
Avg. loss: 0.000057, [current:16000/32000]
Avg. loss: 0.000017, [current:19200/32000]
Avg. loss: 0.000052, [current:22400/32000]
Avg. loss: 0.000112, [current:25600/32000]
Avg. loss: 0.000010, [current:28800/32000]
Avg. loss: 0.000113, [current:32000/32000]
Test Error:
 Accuracy: 96.9%, Avg. loss: 0.000135

Epoch 30
-------------------------------
Avg. loss: 0.000023, [current: 3200/32000]
Avg. loss: 0.000030, [current: 6400/32000]
Avg. loss: 0.000277, [current: 9600/32000]
Avg. loss: 0.000015, [current:12800/32000]
Avg. loss: 0.000172, [current:16000/32000]
Avg. loss: 0.000175, [current:19200/32000]
Avg. loss: 0.000025, [current:22400/32000]
Avg. loss: 0.000092, [current:25600/32000]
Avg. loss: 0.000045, [current:28800/32000]
Avg. loss: 0.000058, [current:32000/32000]
Test Error:
 Accuracy: 97.6%, Avg. loss: 0.000030

Epoch 31
------------------------------
Avg. loss: 0.000046, [current: 3200/32000]
Avg. loss: 0.000011, [current: 6400/32000]
Avg. loss: 0.000053, [current: 9600/32000]
Avg. loss: 0.000019, [current:12800/32000]
Avg. loss: 0.000023, [current:16000/32000]
Avg. loss: 0.000032, [current:19200/32000]
Avg. loss: 0.000011, [current:22400/32000]
Avg. loss: 0.000016, [current:25600/32000]
Avg. loss: 0.000044, [current:28800/32000]
Avg. loss: 0.000015, [current:32000/32000]
Test Error:
 Accuracy: 97.6%, Avg. loss: 0.000043

Epoch 32
------------------------------
Avg. loss: 0.000021, [current: 3200/32000]
Avg. loss: 0.000098, [current: 6400/32000]
Avg. loss: 0.000019, [current: 9600/32000]
Avg. loss: 0.000024, [current:12800/32000]
Avg. loss: 0.000033, [current:16000/32000]
Avg. loss: 0.000188, [current:19200/32000]
Avg. loss: 0.000024, [current:22400/32000]
Avg. loss: 0.000073, [current:25600/32000]
Avg. loss: 0.000019, [current:28800/32000]
Avg. loss: 0.000018, [current:32000/32000]
Test Error:
 Accuracy: 95.8%, Avg. loss: 0.000148

Epoch 33
------------------------------
Avg. loss: 0.000011, [current: 3200/32000]
Avg. loss: 0.000099, [current: 6400/32000]
Avg. loss: 0.000124, [current: 9600/32000]
Avg. loss: 0.000010, [current:12800/32000]
Avg. loss: 0.000033, [current:16000/32000]
Avg. loss: 0.000015, [current:19200/32000]
Avg. loss: 0.000021, [current:22400/32000]
Avg. loss: 0.000057, [current:25600/32000]
Avg. loss: 0.000023, [current:28800/32000]
Avg. loss: 0.000054, [current:32000/32000]
Test Error:
 Accuracy: 98.0%, Avg. loss: 0.000032

Epoch 34
------------------------------
Avg. loss: 0.000163, [current: 3200/32000]
Avg. loss: 0.000019, [current: 6400/32000]
Avg. loss: 0.000057, [current: 9600/32000]
Avg. loss: 0.000010, [current:12800/32000]
Avg. loss: 0.000021, [current:16000/32000]
Avg. loss: 0.000010, [current:19200/32000]
Avg. loss: 0.000068, [current:22400/32000]
Avg. loss: 0.000032, [current:25600/32000]
Avg. loss: 0.000021, [current:28800/32000]
Avg. loss: 0.000031, [current:32000/32000]
Test Error:
 Accuracy: 98.0%, Avg. loss: 0.000050

Epoch 35
------------------------------
Avg. loss: 0.000007, [current: 3200/32000]
Avg. loss: 0.000030, [current: 6400/32000]
Avg. loss: 0.000019, [current: 9600/32000]
Avg. loss: 0.000022, [current:12800/32000]
Avg. loss: 0.000025, [current:16000/32000]
Avg. loss: 0.000091, [current:19200/32000]
Avg. loss: 0.000017, [current:22400/32000]
Avg. loss: 0.000019, [current:25600/32000]
Avg. loss: 0.000012, [current:28800/32000]
Avg. loss: 0.000036, [current:32000/32000]
Test Error:
 Accuracy: 98.0%, Avg. loss: 0.000089

Epoch 36
------------------------------
Avg. loss: 0.000008, [current: 3200/32000]
Avg. loss: 0.000011, [current: 6400/32000]
Avg. loss: 0.000011, [current: 9600/32000]
Avg. loss: 0.000483, [current:12800/32000]
Avg. loss: 0.000055, [current:16000/32000]
Avg. loss: 0.000021, [current:19200/32000]
Avg. loss: 0.000021, [current:22400/32000]
Avg. loss: 0.000010, [current:25600/32000]
Avg. loss: 0.000009, [current:28800/32000]
Avg. loss: 0.000025, [current:32000/32000]
Test Error:
 Accuracy: 98.2%, Avg. loss: 0.000026

Epoch 37
------------------------------
Avg. loss: 0.000025, [current: 3200/32000]
Avg. loss: 0.000020, [current: 6400/32000]
Avg. loss: 0.000010, [current: 9600/32000]
Avg. loss: 0.000017, [current:12800/32000]
Avg. loss: 0.000008, [current:16000/32000]
Avg. loss: 0.000426, [current:19200/32000]
Avg. loss: 0.000013, [current:22400/32000]
Avg. loss: 0.000019, [current:25600/32000]
Avg. loss: 0.000011, [current:28800/32000]
Avg. loss: 0.000017, [current:32000/32000]
Test Error:
 Accuracy: 98.3%, Avg. loss: 0.000040

Epoch 38
------------------------------
Avg. loss: 0.000005, [current: 3200/32000]
Avg. loss: 0.000050, [current: 6400/32000]
Avg. loss: 0.000169, [current: 9600/32000]
Avg. loss: 0.000005, [current:12800/32000]
Avg. loss: 0.000052, [current:16000/32000]
Avg. loss: 0.000031, [current:19200/32000]
Avg. loss: 0.000031, [current:22400/32000]
Avg. loss: 0.000006, [current:25600/32000]
Avg. loss: 0.000010, [current:28800/32000]
Avg. loss: 0.000043, [current:32000/32000]
Test Error:
 Accuracy: 98.2%, Avg. loss: 0.000062

Epoch 39
------------------------------
Avg. loss: 0.000008, [current: 3200/32000]
Avg. loss: 0.000075, [current: 6400/32000]
Avg. loss: 0.000011, [current: 9600/32000]
Avg. loss: 0.000007, [current:12800/32000]
Avg. loss: 0.000027, [current:16000/32000]
Avg. loss: 0.000009, [current:19200/32000]
Avg. loss: 0.000019, [current:22400/32000]
Avg. loss: 0.000013, [current:25600/32000]
Avg. loss: 0.000015, [current:28800/32000]
Avg. loss: 0.000005, [current:32000/32000]
Test Error:
 Accuracy: 98.6%, Avg. loss: 0.000026

Epoch 40
------------------------------
Avg. loss: 0.000189, [current: 3200/32000]
Avg. loss: 0.000216, [current: 6400/32000]
Avg. loss: 0.000089, [current: 9600/32000]
Avg. loss: 0.000006, [current:12800/32000]
Avg. loss: 0.000018, [current:16000/32000]
Avg. loss: 0.000075, [current:19200/32000]
Avg. loss: 0.000005, [current:22400/32000]
Avg. loss: 0.000025, [current:25600/32000]
Avg. loss: 0.000014, [current:28800/32000]
Avg. loss: 0.000016, [current:32000/32000]
Test Error:
 Accuracy: 99.0%, Avg. loss: 0.000028

Epoch 41
------------------------------
Avg. loss: 0.000006, [current: 3200/32000]
Avg. loss: 0.000043, [current: 6400/32000]
Avg. loss: 0.000006, [current: 9600/32000]
Avg. loss: 0.000029, [current:12800/32000]
Avg. loss: 0.000025, [current:16000/32000]
Avg. loss: 0.000008, [current:19200/32000]
Avg. loss: 0.000021, [current:22400/32000]
Avg. loss: 0.000020, [current:25600/32000]
Avg. loss: 0.000081, [current:28800/32000]
Avg. loss: 0.000012, [current:32000/32000]
Test Error:
 Accuracy: 98.4%, Avg. loss: 0.000045
```

```
Epoch 42
------------------------------
Avg. loss: 0.000138, [current: 3200/32000]
Avg. loss: 0.000009, [current: 6400/32000]
Avg. loss: 0.000018, [current: 9600/32000]
Avg. loss: 0.000022, [current:12800/32000]
Avg. loss: 0.000036, [current:16000/32000]
Avg. loss: 0.000022, [current:19200/32000]
Avg. loss: 0.000032, [current:22400/32000]
Avg. loss: 0.000025, [current:25600/32000]
Avg. loss: 0.000008, [current:28800/32000]
Avg. loss: 0.000048, [current:32000/32000]
Test Error:
 Accuracy: 99.0%, Avg. loss: 0.000024

Epoch 43
------------------------------
Avg. loss: 0.000007, [current: 3200/32000]
Avg. loss: 0.000021, [current: 6400/32000]
Avg. loss: 0.000009, [current: 9600/32000]
Avg. loss: 0.000016, [current:12800/32000]
Avg. loss: 0.000004, [current:16000/32000]
Avg. loss: 0.000008, [current:19200/32000]
Avg. loss: 0.000014, [current:22400/32000]
Avg. loss: 0.000012, [current:25600/32000]
Avg. loss: 0.000007, [current:28800/32000]
Avg. loss: 0.000015, [current:32000/32000]
Test Error:
 Accuracy: 98.5%, Avg. loss: 0.000110

Epoch 44
------------------------------
Avg. loss: 0.000010, [current: 3200/32000]
Avg. loss: 0.000023, [current: 6400/32000]
Avg. loss: 0.000016, [current: 9600/32000]
Avg. loss: 0.000030, [current:12800/32000]
Avg. loss: 0.000006, [current:16000/32000]
Avg. loss: 0.000016, [current:19200/32000]
Avg. loss: 0.000006, [current:22400/32000]
Avg. loss: 0.000017, [current:25600/32000]
Avg. loss: 0.000006, [current:28800/32000]
Avg. loss: 0.000014, [current:32000/32000]
Test Error:
 Accuracy: 98.5%, Avg. loss: 0.000030

Epoch 45
------------------------------
Avg. loss: 0.000009, [current: 3200/32000]
Avg. loss: 0.000014, [current: 6400/32000]
Avg. loss: 0.000010, [current: 9600/32000]
Avg. loss: 0.000023, [current:12800/32000]
Avg. loss: 0.000018, [current:16000/32000]
Avg. loss: 0.000004, [current:19200/32000]
Avg. loss: 0.000017, [current:22400/32000]
Avg. loss: 0.000005, [current:25600/32000]
Avg. loss: 0.000003, [current:28800/32000]
Avg. loss: 0.000036, [current:32000/32000]
Test Error:
 Accuracy: 99.0%, Avg. loss: 0.000036

Epoch 46
------------------------------
Avg. loss: 0.000009, [current: 3200/32000]
Avg. loss: 0.000014, [current: 6400/32000]
Avg. loss: 0.000006, [current: 9600/32000]
Avg. loss: 0.000021, [current:12800/32000]
Avg. loss: 0.000003, [current:16000/32000]
Avg. loss: 0.000008, [current:19200/32000]
Avg. loss: 0.000011, [current:22400/32000]
Avg. loss: 0.000015, [current:25600/32000]
Avg. loss: 0.000016, [current:28800/32000]
Avg. loss: 0.000013, [current:32000/32000]
Test Error:
 Accuracy: 98.6%, Avg. loss: 0.000036

Epoch 47
------------------------------
Avg. loss: 0.000014, [current: 3200/32000]
Avg. loss: 0.000040, [current: 6400/32000]
Avg. loss: 0.000045, [current: 9600/32000]
Avg. loss: 0.000006, [current:12800/32000]
Avg. loss: 0.000036, [current:16000/32000]
Avg. loss: 0.000026, [current:19200/32000]
Avg. loss: 0.000007, [current:22400/32000]
Avg. loss: 0.000065, [current:25600/32000]
Avg. loss: 0.000007, [current:28800/32000]
Avg. loss: 0.000006, [current:32000/32000]
Test Error:
 Accuracy: 98.6%, Avg. loss: 0.000019

Epoch 48
------------------------------
Avg. loss: 0.000036, [current: 3200/32000]
Avg. loss: 0.000024, [current: 6400/32000]
Avg. loss: 0.000010, [current: 9600/32000]
Avg. loss: 0.000009, [current:12800/32000]
Avg. loss: 0.000009, [current:16000/32000]
Avg. loss: 0.000013, [current:19200/32000]
Avg. loss: 0.000006, [current:22400/32000]
Avg. loss: 0.000012, [current:25600/32000]
Avg. loss: 0.000013, [current:28800/32000]
Avg. loss: 0.000035, [current:32000/32000]
Test Error:
 Accuracy: 98.7%, Avg. loss: 0.000024

Epoch 49
------------------------------
Avg. loss: 0.000027, [current: 3200/32000]
Avg. loss: 0.000074, [current: 6400/32000]
Avg. loss: 0.000102, [current: 9600/32000]
Avg. loss: 0.000008, [current:12800/32000]
Avg. loss: 0.000004, [current:16000/32000]
Avg. loss: 0.000019, [current:19200/32000]
Avg. loss: 0.000013, [current:22400/32000]
Avg. loss: 0.000096, [current:25600/32000]
Avg. loss: 0.000005, [current:28800/32000]
Avg. loss: 0.000018, [current:32000/32000]
Test Error:
 Accuracy: 99.5%, Avg. loss: 0.000011

Epoch 50
------------------------------
Avg. loss: 0.000006, [current: 3200/32000]
Avg. loss: 0.000003, [current: 6400/32000]
Avg. loss: 0.000008, [current: 9600/32000]
Avg. loss: 0.000004, [current:12800/32000]
Avg. loss: 0.000033, [current:16000/32000]
Avg. loss: 0.000003, [current:19200/32000]
Avg. loss: 0.000009, [current:22400/32000]
Avg. loss: 0.000057, [current:25600/32000]
Avg. loss: 0.000005, [current:28800/32000]
Avg. loss: 0.000005, [current:32000/32000]
Test Error:
 Accuracy: 99.3%, Avg. loss: 0.000015

Done
```

In [ ]: