

# Pytorch with the MNIST Dataset - MNIST

```
## import libraries
from __future__ import print_function
import argparse
import torch
import torch.nn as nn
import torch.nn.functional as F
import torchvision

import torch.optim as optim
from torchvision import datasets, transforms
from torch.autograd import Variable

print(torch.__version__)

2.5.0+cu121

args={}
kwargs={}
args['batch_size']=64 #changed from 32 to 64
args['test_batch_size']=64
args['epochs']=1 #The number of Epochs is the number of times you go
through the full dataset.
args['lr']=0.01 #Learning rate is how fast it will descend.
args['momentum']=0.5 #SGD momentum (default: 0.5) Momentum is a moving
average of our gradients (helps to keep direction).

args['seed']=1 #random seed
args['log_interval']=10
args['cuda']=True #if the computer has a GPU, type True, otherwise,
False
```

This code is adopted from the pytorch examples repository. It is licensed under BSD 3-Clause "New" or "Revised" License. Source: <https://github.com/pytorch/examples/> LICENSE: <https://github.com/pytorch/examples/blob/master/LICENSE>

## Load Dataset

The first step before training the model is to import the data. We will use the [MNIST dataset](#).

```
## transformations
transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.1307,), (0.3081,))
])

## download and load training dataset
```

```
trainset = datasets.MNIST(root='../data', train=True, download=True,
transform=transform)
train_loader = torch.utils.data.DataLoader(trainset,
batch_size=args['batch_size'], shuffle=True, **kwargs)

## download and load testing dataset
testset = torchvision.datasets.MNIST(root='../data', train=False,
download=True, transform=transform)
test_loader = torch.utils.data.DataLoader(testset,
batch_size=args['test_batch_size'], shuffle=True, **kwargs)
```

## Exploring the Data

Let's check what the train and test dataset contains. I will use `matplotlib` to print out some of the images from our dataset.

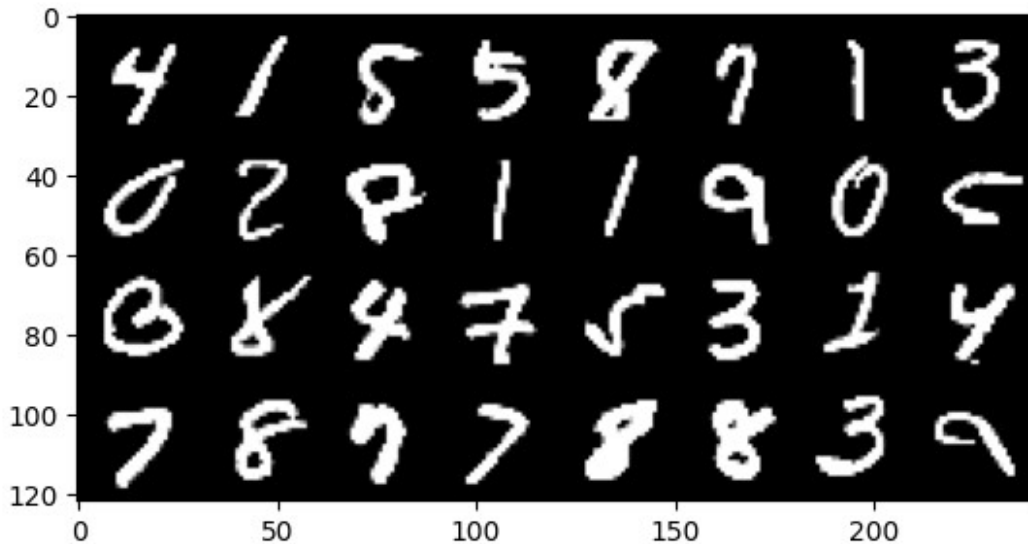
```
import matplotlib.pyplot as plt
import numpy as np

## functions to show an image
def imshow(img):
    #img = img / 2 + 0.5      # unnormalize
    npimg = img.numpy()
    plt.imshow(np.transpose(npimg, (1, 2, 0)))

## get some random training images
dataiter = iter(train_loader)
images, labels = next(dataiter)

## show images
imshow(torchvision.utils.make_grid(images))
```

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



Let's check the dimensions of a batch.

```
for images, labels in train_loader:
    print("Image batch dimensions:", images.shape)
    print("Image label dimensions:", labels.shape)
    break
```

```
Image batch dimensions: torch.Size([32, 1, 28, 28])
Image label dimensions: torch.Size([32])
```

## The Model

---

Function Without CNN Layer []

```
class Net(nn.Module):
    #This defines the structure of the NN.
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(784, 256)
        self.fc2 = nn.Linear(256, 10)

    def forward(self, x):
        x=x.view(-1,784)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)

        #Softmax gets probabilities.
        return F.log_softmax(x, dim=1)
```

---

## Function With 1 CNN Layer []

```
class Net(nn.Module):
    #This defines the structure of the NN.
    def __init__(self):
        super(Net, self).__init__()
        # First convolutional layer: 1 input channel, 32 output
        channels
        self.conv1 = nn.Conv2d(1, 32, kernel_size=5, padding=2)
        self.pool = nn.MaxPool2d(2, 2)
        self.fc1 = nn.Linear(32 * 14 * 14, 256)
        self.fc2 = nn.Linear(256, 10)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = torch.flatten(x, 1) # flatten all dimensions except batch
        #x=x.view(-1,784)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)

        #Softmax gets probabilities.
        return F.log_softmax(x, dim=1)
```

---

## Function With 2 CNN Layer []

```
class Net(nn.Module):
    #This defines the structure of the NN.
    def __init__(self):
        super(Net, self).__init__()
        # First convolutional layer: 1 input channel, 28 output
        channels
        self.conv1 = nn.Conv2d(1, 28, kernel_size=5, padding=2)
        self.pool = nn.MaxPool2d(2, 2)
        # Second convolutional layer: 28 input channels, 32 output
        channels
        self.conv2 = nn.Conv2d(28, 32, kernel_size=5, padding=2)
        self.fc1 = nn.Linear(784, 256)
        self.fc2 = nn.Linear(256, 10)
        # Dimensionality reduction layer to match 784 inputs for fc1
        self.dim_reducer = nn.Linear(1568, 784)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = torch.flatten(x, 1) # flatten all dimensions except batch
        #x=x.view(-1,784)
        # Apply dimensionality reduction to match input size of fc1
        x = self.dim_reducer(x)
```

```

        x = F.relu(self.fc1(x))
        x = self.fc2(x)

        #Softmax gets probabilities.
        return F.log_softmax(x, dim=1)

## test the model with 1 batch
model = Net()
#print(model)
for images, labels in train_loader:
    print("batch size:", args['batch_size'])
    out = model(images)
    print(out.shape)
    break

batch size: 32
torch.Size([32, 10])

```

## Training the Model

Now we are ready to train the model.

```

def train(epoch):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        if args['cuda']:
            data, target = data.cuda(), target.cuda()
            #Variables in Pytorch are differentiable.
            data, target = Variable(data), Variable(target)
            #This will zero out the gradients for this batch.
            optimizer.zero_grad()
            output = model(data)
            # Calculate the loss The negative log likelihood loss. It is
            useful to train a classification problem with C classes.
            loss = F.nll_loss(output, target)
            #dloss/dx for every Variable
            loss.backward()
            #to do a one-step update on our parameter.
            optimizer.step()
            #Print out the loss periodically.
            if batch_idx % args['log_interval'] == 0:
                print('Train Epoch: {} [{}/{}] {:.0f}%)\tLoss:
{: .6f}'.format(
                    epoch, batch_idx * len(data),
                    len(train_loader.dataset),
                    100. * batch_idx / len(train_loader),
                    loss.data.item()))

```

```

def test():
    model.eval()
    test_loss = 0
    correct = 0

    with torch.no_grad():
        for data, target in test_loader:
            if args['cuda']:
                data, target = data.cuda(), target.cuda()
                data, target = Variable(data), Variable(target)
                output = model(data)
                test_loss += F.nll_loss(output, target,
size_average=False).data.item() # sum up batch loss
                pred = output.data.max(1, keepdim=True)[1] # get the index of the max log-probability
                correct +=
pred.eq(target.data.view_as(pred)).long().cpu().sum()

            test_loss /= len(test_loader.dataset)
            print('\nTest set: Average loss: {:.4f}, Accuracy: {}/{} ({:.0f}
%)'\n'.format(
                test_loss, correct, len(test_loader.dataset),
                100. * correct / len(test_loader.dataset)))

```

---

## Results - Without CNN Layer

```

model = Net()
if args['cuda']:
    model.cuda()

optimizer = optim.SGD(model.parameters(), lr=args['lr'],
momentum=args['momentum'])

for epoch in range(1, args['epochs'] + 1):
    train(epoch)
    test()

```

Train Epoch: 1 [0/60000 (0%)]      Loss: 2.320207  
 Train Epoch: 1 [640/60000 (1%)]      Loss: 2.004853  
 Train Epoch: 1 [1280/60000 (2%)]      Loss: 1.742395  
 Train Epoch: 1 [1920/60000 (3%)]      Loss: 1.367994  
 Train Epoch: 1 [2560/60000 (4%)]      Loss: 1.218946  
 Train Epoch: 1 [3200/60000 (5%)]      Loss: 0.966231  
 Train Epoch: 1 [3840/60000 (6%)]      Loss: 0.834849  
 Train Epoch: 1 [4480/60000 (7%)]      Loss: 0.647816  
 Train Epoch: 1 [5120/60000 (9%)]      Loss: 0.663867  
 Train Epoch: 1 [5760/60000 (10%)]      Loss: 0.536120  
 Train Epoch: 1 [6400/60000 (11%)]      Loss: 0.643585

Train Epoch: 1	[7040/60000 (12%)]	Loss: 0.634093
Train Epoch: 1	[7680/60000 (13%)]	Loss: 0.555330
Train Epoch: 1	[8320/60000 (14%)]	Loss: 0.452556
Train Epoch: 1	[8960/60000 (15%)]	Loss: 0.601981
Train Epoch: 1	[9600/60000 (16%)]	Loss: 0.471001
Train Epoch: 1	[10240/60000 (17%)]	Loss: 0.528277
Train Epoch: 1	[10880/60000 (18%)]	Loss: 0.426070
Train Epoch: 1	[11520/60000 (19%)]	Loss: 0.525767
Train Epoch: 1	[12160/60000 (20%)]	Loss: 0.447959
Train Epoch: 1	[12800/60000 (21%)]	Loss: 0.399419
Train Epoch: 1	[13440/60000 (22%)]	Loss: 0.586153
Train Epoch: 1	[14080/60000 (23%)]	Loss: 0.410921
Train Epoch: 1	[14720/60000 (25%)]	Loss: 0.261390
Train Epoch: 1	[15360/60000 (26%)]	Loss: 0.315323
Train Epoch: 1	[16000/60000 (27%)]	Loss: 0.427393
Train Epoch: 1	[16640/60000 (28%)]	Loss: 0.476579
Train Epoch: 1	[17280/60000 (29%)]	Loss: 0.443073
Train Epoch: 1	[17920/60000 (30%)]	Loss: 0.447854
Train Epoch: 1	[18560/60000 (31%)]	Loss: 0.289942
Train Epoch: 1	[19200/60000 (32%)]	Loss: 0.575735
Train Epoch: 1	[19840/60000 (33%)]	Loss: 0.231088
Train Epoch: 1	[20480/60000 (34%)]	Loss: 0.288677
Train Epoch: 1	[21120/60000 (35%)]	Loss: 0.252904
Train Epoch: 1	[21760/60000 (36%)]	Loss: 0.365159
Train Epoch: 1	[22400/60000 (37%)]	Loss: 0.359134
Train Epoch: 1	[23040/60000 (38%)]	Loss: 0.308817
Train Epoch: 1	[23680/60000 (39%)]	Loss: 0.333603
Train Epoch: 1	[24320/60000 (41%)]	Loss: 0.389700
Train Epoch: 1	[24960/60000 (42%)]	Loss: 0.383618
Train Epoch: 1	[25600/60000 (43%)]	Loss: 0.272328
Train Epoch: 1	[26240/60000 (44%)]	Loss: 0.463397
Train Epoch: 1	[26880/60000 (45%)]	Loss: 0.390437
Train Epoch: 1	[27520/60000 (46%)]	Loss: 0.311975
Train Epoch: 1	[28160/60000 (47%)]	Loss: 0.404672
Train Epoch: 1	[28800/60000 (48%)]	Loss: 0.347380
Train Epoch: 1	[29440/60000 (49%)]	Loss: 0.444060
Train Epoch: 1	[30080/60000 (50%)]	Loss: 0.288145
Train Epoch: 1	[30720/60000 (51%)]	Loss: 0.241818
Train Epoch: 1	[31360/60000 (52%)]	Loss: 0.265705
Train Epoch: 1	[32000/60000 (53%)]	Loss: 0.447085
Train Epoch: 1	[32640/60000 (54%)]	Loss: 0.277420
Train Epoch: 1	[33280/60000 (55%)]	Loss: 0.371893
Train Epoch: 1	[33920/60000 (57%)]	Loss: 0.163706
Train Epoch: 1	[34560/60000 (58%)]	Loss: 0.464842
Train Epoch: 1	[35200/60000 (59%)]	Loss: 0.229153
Train Epoch: 1	[35840/60000 (60%)]	Loss: 0.518350
Train Epoch: 1	[36480/60000 (61%)]	Loss: 0.445439
Train Epoch: 1	[37120/60000 (62%)]	Loss: 0.527507
Train Epoch: 1	[37760/60000 (63%)]	Loss: 0.696661

```

Train Epoch: 1 [38400/60000 (64%)] Loss: 0.307871
Train Epoch: 1 [39040/60000 (65%)] Loss: 0.407040
Train Epoch: 1 [39680/60000 (66%)] Loss: 0.230685
Train Epoch: 1 [40320/60000 (67%)] Loss: 0.396762
Train Epoch: 1 [40960/60000 (68%)] Loss: 0.344710
Train Epoch: 1 [41600/60000 (69%)] Loss: 0.286728
Train Epoch: 1 [42240/60000 (70%)] Loss: 0.328665
Train Epoch: 1 [42880/60000 (71%)] Loss: 0.177785
Train Epoch: 1 [43520/60000 (72%)] Loss: 0.249442
Train Epoch: 1 [44160/60000 (74%)] Loss: 0.380182
Train Epoch: 1 [44800/60000 (75%)] Loss: 0.189400
Train Epoch: 1 [45440/60000 (76%)] Loss: 0.239653
Train Epoch: 1 [46080/60000 (77%)] Loss: 0.404898
Train Epoch: 1 [46720/60000 (78%)] Loss: 0.338228
Train Epoch: 1 [47360/60000 (79%)] Loss: 0.295037
Train Epoch: 1 [48000/60000 (80%)] Loss: 0.184089
Train Epoch: 1 [48640/60000 (81%)] Loss: 0.220591
Train Epoch: 1 [49280/60000 (82%)] Loss: 0.232725
Train Epoch: 1 [49920/60000 (83%)] Loss: 0.144263
Train Epoch: 1 [50560/60000 (84%)] Loss: 0.197499
Train Epoch: 1 [51200/60000 (85%)] Loss: 0.275229
Train Epoch: 1 [51840/60000 (86%)] Loss: 0.408600
Train Epoch: 1 [52480/60000 (87%)] Loss: 0.088620
Train Epoch: 1 [53120/60000 (88%)] Loss: 0.204144
Train Epoch: 1 [53760/60000 (90%)] Loss: 0.231934
Train Epoch: 1 [54400/60000 (91%)] Loss: 0.196021
Train Epoch: 1 [55040/60000 (92%)] Loss: 0.390102
Train Epoch: 1 [55680/60000 (93%)] Loss: 0.335723
Train Epoch: 1 [56320/60000 (94%)] Loss: 0.234095
Train Epoch: 1 [56960/60000 (95%)] Loss: 0.178172
Train Epoch: 1 [57600/60000 (96%)] Loss: 0.392131
Train Epoch: 1 [58240/60000 (97%)] Loss: 0.276648
Train Epoch: 1 [58880/60000 (98%)] Loss: 0.290332
Train Epoch: 1 [59520/60000 (99%)] Loss: 0.238533

```

```

/usr/local/lib/python3.10/dist-packages/torch/nn/_reduction.py:51:
UserWarning: size_average and reduce args will be deprecated, please
use reduction='sum' instead.
  warnings.warn(warning.format(ret))

```

Test set: Average loss: 0.2561, Accuracy: 9262/10000 (93%)

## Results - With 1 CNN Layer []

```

model = Net()
if args['cuda']:

```



```

model.cuda()

optimizer = optim.SGD(model.parameters(), lr=args['lr'],
momentum=args['momentum'])

for epoch in range(1, args['epochs'] + 1):
    train(epoch)
    test()

```

```

Train Epoch: 1 [0/60000 (0%)]    Loss: 2.324679
Train Epoch: 1 [640/60000 (1%)]  Loss: 1.801129
Train Epoch: 1 [1280/60000 (2%)] Loss: 1.220307
Train Epoch: 1 [1920/60000 (3%)] Loss: 0.798972
Train Epoch: 1 [2560/60000 (4%)] Loss: 0.553820
Train Epoch: 1 [3200/60000 (5%)] Loss: 0.446185
Train Epoch: 1 [3840/60000 (6%)] Loss: 0.432305
Train Epoch: 1 [4480/60000 (7%)] Loss: 0.603873
Train Epoch: 1 [5120/60000 (9%)] Loss: 0.542304
Train Epoch: 1 [5760/60000 (10%)] Loss: 0.427222
Train Epoch: 1 [6400/60000 (11%)] Loss: 0.399443
Train Epoch: 1 [7040/60000 (12%)] Loss: 0.366460
Train Epoch: 1 [7680/60000 (13%)] Loss: 0.420878
Train Epoch: 1 [8320/60000 (14%)] Loss: 0.332036
Train Epoch: 1 [8960/60000 (15%)] Loss: 0.364819
Train Epoch: 1 [9600/60000 (16%)] Loss: 0.280376
Train Epoch: 1 [10240/60000 (17%)] Loss: 0.412950
Train Epoch: 1 [10880/60000 (18%)] Loss: 0.326366
Train Epoch: 1 [11520/60000 (19%)] Loss: 0.368769
Train Epoch: 1 [12160/60000 (20%)] Loss: 0.204294
Train Epoch: 1 [12800/60000 (21%)] Loss: 0.216059
Train Epoch: 1 [13440/60000 (22%)] Loss: 0.526026
Train Epoch: 1 [14080/60000 (23%)] Loss: 0.381680
Train Epoch: 1 [14720/60000 (25%)] Loss: 0.335499
Train Epoch: 1 [15360/60000 (26%)] Loss: 0.266350
Train Epoch: 1 [16000/60000 (27%)] Loss: 0.452274
Train Epoch: 1 [16640/60000 (28%)] Loss: 0.277955
Train Epoch: 1 [17280/60000 (29%)] Loss: 0.347437
Train Epoch: 1 [17920/60000 (30%)] Loss: 0.223950
Train Epoch: 1 [18560/60000 (31%)] Loss: 0.182515
Train Epoch: 1 [19200/60000 (32%)] Loss: 0.146121
Train Epoch: 1 [19840/60000 (33%)] Loss: 0.180061
Train Epoch: 1 [20480/60000 (34%)] Loss: 0.209676
Train Epoch: 1 [21120/60000 (35%)] Loss: 0.254402
Train Epoch: 1 [21760/60000 (36%)] Loss: 0.200449
Train Epoch: 1 [22400/60000 (37%)] Loss: 0.132338
Train Epoch: 1 [23040/60000 (38%)] Loss: 0.232276
Train Epoch: 1 [23680/60000 (39%)] Loss: 0.368520
Train Epoch: 1 [24320/60000 (41%)] Loss: 0.193264
Train Epoch: 1 [24960/60000 (42%)] Loss: 0.232754
Train Epoch: 1 [25600/60000 (43%)] Loss: 0.274308

```

Train Epoch: 1	[26240/60000 (44%) ]	Loss: 0.242453
Train Epoch: 1	[26880/60000 (45%) ]	Loss: 0.378693
Train Epoch: 1	[27520/60000 (46%) ]	Loss: 0.157882
Train Epoch: 1	[28160/60000 (47%) ]	Loss: 0.210491
Train Epoch: 1	[28800/60000 (48%) ]	Loss: 0.209797
Train Epoch: 1	[29440/60000 (49%) ]	Loss: 0.111751
Train Epoch: 1	[30080/60000 (50%) ]	Loss: 0.329846
Train Epoch: 1	[30720/60000 (51%) ]	Loss: 0.160969
Train Epoch: 1	[31360/60000 (52%) ]	Loss: 0.203546
Train Epoch: 1	[32000/60000 (53%) ]	Loss: 0.175185
Train Epoch: 1	[32640/60000 (54%) ]	Loss: 0.101511
Train Epoch: 1	[33280/60000 (55%) ]	Loss: 0.332586
Train Epoch: 1	[33920/60000 (57%) ]	Loss: 0.215917
Train Epoch: 1	[34560/60000 (58%) ]	Loss: 0.338527
Train Epoch: 1	[35200/60000 (59%) ]	Loss: 0.160138
Train Epoch: 1	[35840/60000 (60%) ]	Loss: 0.213011
Train Epoch: 1	[36480/60000 (61%) ]	Loss: 0.165414
Train Epoch: 1	[37120/60000 (62%) ]	Loss: 0.086370
Train Epoch: 1	[37760/60000 (63%) ]	Loss: 0.078025
Train Epoch: 1	[38400/60000 (64%) ]	Loss: 0.195158
Train Epoch: 1	[39040/60000 (65%) ]	Loss: 0.134542
Train Epoch: 1	[39680/60000 (66%) ]	Loss: 0.145388
Train Epoch: 1	[40320/60000 (67%) ]	Loss: 0.205886
Train Epoch: 1	[40960/60000 (68%) ]	Loss: 0.144490
Train Epoch: 1	[41600/60000 (69%) ]	Loss: 0.304415
Train Epoch: 1	[42240/60000 (70%) ]	Loss: 0.288605
Train Epoch: 1	[42880/60000 (71%) ]	Loss: 0.060211
Train Epoch: 1	[43520/60000 (72%) ]	Loss: 0.095719
Train Epoch: 1	[44160/60000 (74%) ]	Loss: 0.151825
Train Epoch: 1	[44800/60000 (75%) ]	Loss: 0.098358
Train Epoch: 1	[45440/60000 (76%) ]	Loss: 0.265757
Train Epoch: 1	[46080/60000 (77%) ]	Loss: 0.203382
Train Epoch: 1	[46720/60000 (78%) ]	Loss: 0.226423
Train Epoch: 1	[47360/60000 (79%) ]	Loss: 0.325331
Train Epoch: 1	[48000/60000 (80%) ]	Loss: 0.129530
Train Epoch: 1	[48640/60000 (81%) ]	Loss: 0.039108
Train Epoch: 1	[49280/60000 (82%) ]	Loss: 0.231548
Train Epoch: 1	[49920/60000 (83%) ]	Loss: 0.316183
Train Epoch: 1	[50560/60000 (84%) ]	Loss: 0.268469
Train Epoch: 1	[51200/60000 (85%) ]	Loss: 0.126358
Train Epoch: 1	[51840/60000 (86%) ]	Loss: 0.221436
Train Epoch: 1	[52480/60000 (87%) ]	Loss: 0.197918
Train Epoch: 1	[53120/60000 (88%) ]	Loss: 0.050993
Train Epoch: 1	[53760/60000 (90%) ]	Loss: 0.140745
Train Epoch: 1	[54400/60000 (91%) ]	Loss: 0.094131
Train Epoch: 1	[55040/60000 (92%) ]	Loss: 0.077014
Train Epoch: 1	[55680/60000 (93%) ]	Loss: 0.159227
Train Epoch: 1	[56320/60000 (94%) ]	Loss: 0.130148
Train Epoch: 1	[56960/60000 (95%) ]	Loss: 0.103697

```
Train Epoch: 1 [57600/60000 (96%)]    Loss: 0.189535
Train Epoch: 1 [58240/60000 (97%)]    Loss: 0.150114
Train Epoch: 1 [58880/60000 (98%)]    Loss: 0.149976
Train Epoch: 1 [59520/60000 (99%)]    Loss: 0.158036
```

```
/usr/local/lib/python3.10/dist-packages/torch/nn/_reduction.py:51:
UserWarning: size_average and reduce args will be deprecated, please
use reduction='sum' instead.
  warnings.warn(warning.format(ret))
```

```
Test set: Average loss: 0.1375, Accuracy: 9593/10000 (96%)
```

---

## Experimenting with different set of HyperParameters and Network Topologies

### 1) Results - With 2 CNN Layer []

```
model = Net()
if args['cuda']:
    model.cuda()

optimizer = optim.SGD(model.parameters(), lr=args['lr'],
momentum=args['momentum'])

for epoch in range(1, args['epochs'] + 1):
    train(epoch)
    test()
```

```
Train Epoch: 1 [0/60000 (0%)]    Loss: 2.303020
Train Epoch: 1 [640/60000 (1%)]  Loss: 2.281819
Train Epoch: 1 [1280/60000 (2%)] Loss: 2.252346
Train Epoch: 1 [1920/60000 (3%)] Loss: 2.223011
Train Epoch: 1 [2560/60000 (4%)] Loss: 2.154228
Train Epoch: 1 [3200/60000 (5%)] Loss: 2.051044
Train Epoch: 1 [3840/60000 (6%)] Loss: 1.949421
Train Epoch: 1 [4480/60000 (7%)] Loss: 1.595366
Train Epoch: 1 [5120/60000 (9%)] Loss: 1.228488
Train Epoch: 1 [5760/60000 (10%)] Loss: 0.982484
Train Epoch: 1 [6400/60000 (11%)] Loss: 0.724905
Train Epoch: 1 [7040/60000 (12%)] Loss: 0.729990
Train Epoch: 1 [7680/60000 (13%)] Loss: 0.590910
Train Epoch: 1 [8320/60000 (14%)] Loss: 0.889242
Train Epoch: 1 [8960/60000 (15%)] Loss: 0.359823
Train Epoch: 1 [9600/60000 (16%)] Loss: 0.503985
Train Epoch: 1 [10240/60000 (17%)] Loss: 0.554815
```

Train Epoch: 1	[10880/60000 (18%)]	Loss: 0.372829
Train Epoch: 1	[11520/60000 (19%)]	Loss: 0.392720
Train Epoch: 1	[12160/60000 (20%)]	Loss: 0.353639
Train Epoch: 1	[12800/60000 (21%)]	Loss: 0.468223
Train Epoch: 1	[13440/60000 (22%)]	Loss: 0.415563
Train Epoch: 1	[14080/60000 (23%)]	Loss: 0.394513
Train Epoch: 1	[14720/60000 (25%)]	Loss: 0.386249
Train Epoch: 1	[15360/60000 (26%)]	Loss: 0.479020
Train Epoch: 1	[16000/60000 (27%)]	Loss: 0.374048
Train Epoch: 1	[16640/60000 (28%)]	Loss: 0.509387
Train Epoch: 1	[17280/60000 (29%)]	Loss: 0.275199
Train Epoch: 1	[17920/60000 (30%)]	Loss: 0.220288
Train Epoch: 1	[18560/60000 (31%)]	Loss: 0.190482
Train Epoch: 1	[19200/60000 (32%)]	Loss: 0.437952
Train Epoch: 1	[19840/60000 (33%)]	Loss: 0.546517
Train Epoch: 1	[20480/60000 (34%)]	Loss: 0.273954
Train Epoch: 1	[21120/60000 (35%)]	Loss: 0.327001
Train Epoch: 1	[21760/60000 (36%)]	Loss: 0.419312
Train Epoch: 1	[22400/60000 (37%)]	Loss: 0.192281
Train Epoch: 1	[23040/60000 (38%)]	Loss: 0.349590
Train Epoch: 1	[23680/60000 (39%)]	Loss: 0.173149
Train Epoch: 1	[24320/60000 (41%)]	Loss: 0.400282
Train Epoch: 1	[24960/60000 (42%)]	Loss: 0.396371
Train Epoch: 1	[25600/60000 (43%)]	Loss: 0.255573
Train Epoch: 1	[26240/60000 (44%)]	Loss: 0.069998
Train Epoch: 1	[26880/60000 (45%)]	Loss: 0.234842
Train Epoch: 1	[27520/60000 (46%)]	Loss: 0.277792
Train Epoch: 1	[28160/60000 (47%)]	Loss: 0.108131
Train Epoch: 1	[28800/60000 (48%)]	Loss: 0.263422
Train Epoch: 1	[29440/60000 (49%)]	Loss: 0.121483
Train Epoch: 1	[30080/60000 (50%)]	Loss: 0.162130
Train Epoch: 1	[30720/60000 (51%)]	Loss: 0.079969
Train Epoch: 1	[31360/60000 (52%)]	Loss: 0.147537
Train Epoch: 1	[32000/60000 (53%)]	Loss: 0.153209
Train Epoch: 1	[32640/60000 (54%)]	Loss: 0.189218
Train Epoch: 1	[33280/60000 (55%)]	Loss: 0.158108
Train Epoch: 1	[33920/60000 (57%)]	Loss: 0.272247
Train Epoch: 1	[34560/60000 (58%)]	Loss: 0.264606
Train Epoch: 1	[35200/60000 (59%)]	Loss: 0.234035
Train Epoch: 1	[35840/60000 (60%)]	Loss: 0.083643
Train Epoch: 1	[36480/60000 (61%)]	Loss: 0.171657
Train Epoch: 1	[37120/60000 (62%)]	Loss: 0.116725
Train Epoch: 1	[37760/60000 (63%)]	Loss: 0.481733
Train Epoch: 1	[38400/60000 (64%)]	Loss: 0.126468
Train Epoch: 1	[39040/60000 (65%)]	Loss: 0.054077
Train Epoch: 1	[39680/60000 (66%)]	Loss: 0.178630
Train Epoch: 1	[40320/60000 (67%)]	Loss: 0.319627
Train Epoch: 1	[40960/60000 (68%)]	Loss: 0.102561
Train Epoch: 1	[41600/60000 (69%)]	Loss: 0.151872

```

Train Epoch: 1 [42240/60000 (70%)] Loss: 0.220235
Train Epoch: 1 [42880/60000 (71%)] Loss: 0.160888
Train Epoch: 1 [43520/60000 (72%)] Loss: 0.069203
Train Epoch: 1 [44160/60000 (74%)] Loss: 0.137903
Train Epoch: 1 [44800/60000 (75%)] Loss: 0.208397
Train Epoch: 1 [45440/60000 (76%)] Loss: 0.181498
Train Epoch: 1 [46080/60000 (77%)] Loss: 0.095169
Train Epoch: 1 [46720/60000 (78%)] Loss: 0.148019
Train Epoch: 1 [47360/60000 (79%)] Loss: 0.124289
Train Epoch: 1 [48000/60000 (80%)] Loss: 0.168926
Train Epoch: 1 [48640/60000 (81%)] Loss: 0.237950
Train Epoch: 1 [49280/60000 (82%)] Loss: 0.219675
Train Epoch: 1 [49920/60000 (83%)] Loss: 0.188637
Train Epoch: 1 [50560/60000 (84%)] Loss: 0.036207
Train Epoch: 1 [51200/60000 (85%)] Loss: 0.374976
Train Epoch: 1 [51840/60000 (86%)] Loss: 0.096544
Train Epoch: 1 [52480/60000 (87%)] Loss: 0.089064
Train Epoch: 1 [53120/60000 (88%)] Loss: 0.163821
Train Epoch: 1 [53760/60000 (90%)] Loss: 0.055521
Train Epoch: 1 [54400/60000 (91%)] Loss: 0.233503
Train Epoch: 1 [55040/60000 (92%)] Loss: 0.127848
Train Epoch: 1 [55680/60000 (93%)] Loss: 0.283392
Train Epoch: 1 [56320/60000 (94%)] Loss: 0.032731
Train Epoch: 1 [56960/60000 (95%)] Loss: 0.070723
Train Epoch: 1 [57600/60000 (96%)] Loss: 0.055533
Train Epoch: 1 [58240/60000 (97%)] Loss: 0.111102
Train Epoch: 1 [58880/60000 (98%)] Loss: 0.111862
Train Epoch: 1 [59520/60000 (99%)] Loss: 0.241074

```

```

/usr/local/lib/python3.10/dist-packages/torch/nn/_reduction.py:51:
UserWarning: size_average and reduce args will be deprecated, please
use reduction='sum' instead.
  warnings.warn(warning.format(ret))

```

Test set: Average loss: 0.1077, Accuracy: 9660/10000 (97%)

2) Results - lr = 0.05, epochs = 2, batch\_size = 32

```

model = Net()
if args['cuda']:
    model.cuda()

optimizer = optim.SGD(model.parameters(), lr=0.05,
momentum=args['momentum'])

for epoch in range(1, 2 + 1):
    train(epoch)
    test()

```

Train Epoch: 1	[0/60000 (0%)]	Loss: 2.297872
Train Epoch: 1	[320/60000 (1%)]	Loss: 2.198008
Train Epoch: 1	[640/60000 (1%)]	Loss: 1.447390
Train Epoch: 1	[960/60000 (2%)]	Loss: 0.858325
Train Epoch: 1	[1280/60000 (2%)]	Loss: 1.307752
Train Epoch: 1	[1600/60000 (3%)]	Loss: 0.287565
Train Epoch: 1	[1920/60000 (3%)]	Loss: 0.512782
Train Epoch: 1	[2240/60000 (4%)]	Loss: 0.540262
Train Epoch: 1	[2560/60000 (4%)]	Loss: 0.452480
Train Epoch: 1	[2880/60000 (5%)]	Loss: 0.412894
Train Epoch: 1	[3200/60000 (5%)]	Loss: 0.402601
Train Epoch: 1	[3520/60000 (6%)]	Loss: 0.299248
Train Epoch: 1	[3840/60000 (6%)]	Loss: 0.264897
Train Epoch: 1	[4160/60000 (7%)]	Loss: 0.133231
Train Epoch: 1	[4480/60000 (7%)]	Loss: 0.177545
Train Epoch: 1	[4800/60000 (8%)]	Loss: 0.102515
Train Epoch: 1	[5120/60000 (9%)]	Loss: 0.208488
Train Epoch: 1	[5440/60000 (9%)]	Loss: 0.317022
Train Epoch: 1	[5760/60000 (10%)]	Loss: 0.256144
Train Epoch: 1	[6080/60000 (10%)]	Loss: 0.078522
Train Epoch: 1	[6400/60000 (11%)]	Loss: 0.472854
Train Epoch: 1	[6720/60000 (11%)]	Loss: 0.154949
Train Epoch: 1	[7040/60000 (12%)]	Loss: 0.128589
Train Epoch: 1	[7360/60000 (12%)]	Loss: 0.226847
Train Epoch: 1	[7680/60000 (13%)]	Loss: 0.236645
Train Epoch: 1	[8000/60000 (13%)]	Loss: 0.190821
Train Epoch: 1	[8320/60000 (14%)]	Loss: 0.406803
Train Epoch: 1	[8640/60000 (14%)]	Loss: 0.162700
Train Epoch: 1	[8960/60000 (15%)]	Loss: 0.063836
Train Epoch: 1	[9280/60000 (15%)]	Loss: 0.105650
Train Epoch: 1	[9600/60000 (16%)]	Loss: 0.103235
Train Epoch: 1	[9920/60000 (17%)]	Loss: 0.349271
Train Epoch: 1	[10240/60000 (17%)]	Loss: 0.121325
Train Epoch: 1	[10560/60000 (18%)]	Loss: 0.225193
Train Epoch: 1	[10880/60000 (18%)]	Loss: 0.200360
Train Epoch: 1	[11200/60000 (19%)]	Loss: 0.050266
Train Epoch: 1	[11520/60000 (19%)]	Loss: 0.182973
Train Epoch: 1	[11840/60000 (20%)]	Loss: 0.049569
Train Epoch: 1	[12160/60000 (20%)]	Loss: 0.104100
Train Epoch: 1	[12480/60000 (21%)]	Loss: 0.253908
Train Epoch: 1	[12800/60000 (21%)]	Loss: 0.016469
Train Epoch: 1	[13120/60000 (22%)]	Loss: 0.046023
Train Epoch: 1	[13440/60000 (22%)]	Loss: 0.208091
Train Epoch: 1	[13760/60000 (23%)]	Loss: 0.234459
Train Epoch: 1	[14080/60000 (23%)]	Loss: 0.313562
Train Epoch: 1	[14400/60000 (24%)]	Loss: 0.328446
Train Epoch: 1	[14720/60000 (25%)]	Loss: 0.047360
Train Epoch: 1	[15040/60000 (25%)]	Loss: 0.182692
Train Epoch: 1	[15360/60000 (26%)]	Loss: 0.021498
Train Epoch: 1	[15680/60000 (26%)]	Loss: 0.444144

Train Epoch: 1	[16000/60000 (27%) ]	Loss: 0.318842
Train Epoch: 1	[16320/60000 (27%) ]	Loss: 0.069297
Train Epoch: 1	[16640/60000 (28%) ]	Loss: 0.053611
Train Epoch: 1	[16960/60000 (28%) ]	Loss: 0.324071
Train Epoch: 1	[17280/60000 (29%) ]	Loss: 0.095100
Train Epoch: 1	[17600/60000 (29%) ]	Loss: 0.064901
Train Epoch: 1	[17920/60000 (30%) ]	Loss: 0.114505
Train Epoch: 1	[18240/60000 (30%) ]	Loss: 0.081067
Train Epoch: 1	[18560/60000 (31%) ]	Loss: 0.071599
Train Epoch: 1	[18880/60000 (31%) ]	Loss: 0.152778
Train Epoch: 1	[19200/60000 (32%) ]	Loss: 0.132572
Train Epoch: 1	[19520/60000 (33%) ]	Loss: 0.098533
Train Epoch: 1	[19840/60000 (33%) ]	Loss: 0.050271
Train Epoch: 1	[20160/60000 (34%) ]	Loss: 0.301045
Train Epoch: 1	[20480/60000 (34%) ]	Loss: 0.188822
Train Epoch: 1	[20800/60000 (35%) ]	Loss: 0.042791
Train Epoch: 1	[21120/60000 (35%) ]	Loss: 0.074113
Train Epoch: 1	[21440/60000 (36%) ]	Loss: 0.074636
Train Epoch: 1	[21760/60000 (36%) ]	Loss: 0.224908
Train Epoch: 1	[22080/60000 (37%) ]	Loss: 0.158909
Train Epoch: 1	[22400/60000 (37%) ]	Loss: 0.059978
Train Epoch: 1	[22720/60000 (38%) ]	Loss: 0.050103
Train Epoch: 1	[23040/60000 (38%) ]	Loss: 0.115228
Train Epoch: 1	[23360/60000 (39%) ]	Loss: 0.018061
Train Epoch: 1	[23680/60000 (39%) ]	Loss: 0.042953
Train Epoch: 1	[24000/60000 (40%) ]	Loss: 0.007169
Train Epoch: 1	[24320/60000 (41%) ]	Loss: 0.034219
Train Epoch: 1	[24640/60000 (41%) ]	Loss: 0.028249
Train Epoch: 1	[24960/60000 (42%) ]	Loss: 0.114567
Train Epoch: 1	[25280/60000 (42%) ]	Loss: 0.044129
Train Epoch: 1	[25600/60000 (43%) ]	Loss: 0.132320
Train Epoch: 1	[25920/60000 (43%) ]	Loss: 0.034317
Train Epoch: 1	[26240/60000 (44%) ]	Loss: 0.072570
Train Epoch: 1	[26560/60000 (44%) ]	Loss: 0.105058
Train Epoch: 1	[26880/60000 (45%) ]	Loss: 0.050118
Train Epoch: 1	[27200/60000 (45%) ]	Loss: 0.040541
Train Epoch: 1	[27520/60000 (46%) ]	Loss: 0.225276
Train Epoch: 1	[27840/60000 (46%) ]	Loss: 0.071807
Train Epoch: 1	[28160/60000 (47%) ]	Loss: 0.168339
Train Epoch: 1	[28480/60000 (47%) ]	Loss: 0.130358
Train Epoch: 1	[28800/60000 (48%) ]	Loss: 0.033545
Train Epoch: 1	[29120/60000 (49%) ]	Loss: 0.149048
Train Epoch: 1	[29440/60000 (49%) ]	Loss: 0.006199
Train Epoch: 1	[29760/60000 (50%) ]	Loss: 0.113532
Train Epoch: 1	[30080/60000 (50%) ]	Loss: 0.087133
Train Epoch: 1	[30400/60000 (51%) ]	Loss: 0.115689
Train Epoch: 1	[30720/60000 (51%) ]	Loss: 0.044758
Train Epoch: 1	[31040/60000 (52%) ]	Loss: 0.005932
Train Epoch: 1	[31360/60000 (52%) ]	Loss: 0.123455
Train Epoch: 1	[31680/60000 (53%) ]	Loss: 0.387495

Train Epoch: 1	[32000/60000 (53%)]	Loss: 0.150765
Train Epoch: 1	[32320/60000 (54%)]	Loss: 0.038777
Train Epoch: 1	[32640/60000 (54%)]	Loss: 0.018036
Train Epoch: 1	[32960/60000 (55%)]	Loss: 0.017510
Train Epoch: 1	[33280/60000 (55%)]	Loss: 0.022386
Train Epoch: 1	[33600/60000 (56%)]	Loss: 0.142551
Train Epoch: 1	[33920/60000 (57%)]	Loss: 0.005614
Train Epoch: 1	[34240/60000 (57%)]	Loss: 0.170720
Train Epoch: 1	[34560/60000 (58%)]	Loss: 0.097658
Train Epoch: 1	[34880/60000 (58%)]	Loss: 0.007627
Train Epoch: 1	[35200/60000 (59%)]	Loss: 0.086061
Train Epoch: 1	[35520/60000 (59%)]	Loss: 0.020708
Train Epoch: 1	[35840/60000 (60%)]	Loss: 0.094053
Train Epoch: 1	[36160/60000 (60%)]	Loss: 0.129999
Train Epoch: 1	[36480/60000 (61%)]	Loss: 0.218127
Train Epoch: 1	[36800/60000 (61%)]	Loss: 0.080761
Train Epoch: 1	[37120/60000 (62%)]	Loss: 0.030573
Train Epoch: 1	[37440/60000 (62%)]	Loss: 0.094902
Train Epoch: 1	[37760/60000 (63%)]	Loss: 0.039485
Train Epoch: 1	[38080/60000 (63%)]	Loss: 0.009754
Train Epoch: 1	[38400/60000 (64%)]	Loss: 0.032973
Train Epoch: 1	[38720/60000 (65%)]	Loss: 0.095455
Train Epoch: 1	[39040/60000 (65%)]	Loss: 0.134582
Train Epoch: 1	[39360/60000 (66%)]	Loss: 0.017161
Train Epoch: 1	[39680/60000 (66%)]	Loss: 0.097313
Train Epoch: 1	[40000/60000 (67%)]	Loss: 0.182818
Train Epoch: 1	[40320/60000 (67%)]	Loss: 0.014557
Train Epoch: 1	[40640/60000 (68%)]	Loss: 0.080829
Train Epoch: 1	[40960/60000 (68%)]	Loss: 0.188298
Train Epoch: 1	[41280/60000 (69%)]	Loss: 0.021289
Train Epoch: 1	[41600/60000 (69%)]	Loss: 0.120752
Train Epoch: 1	[41920/60000 (70%)]	Loss: 0.029516
Train Epoch: 1	[42240/60000 (70%)]	Loss: 0.035227
Train Epoch: 1	[42560/60000 (71%)]	Loss: 0.034833
Train Epoch: 1	[42880/60000 (71%)]	Loss: 0.009378
Train Epoch: 1	[43200/60000 (72%)]	Loss: 0.016559
Train Epoch: 1	[43520/60000 (73%)]	Loss: 0.007153
Train Epoch: 1	[43840/60000 (73%)]	Loss: 0.121487
Train Epoch: 1	[44160/60000 (74%)]	Loss: 0.045609
Train Epoch: 1	[44480/60000 (74%)]	Loss: 0.031227
Train Epoch: 1	[44800/60000 (75%)]	Loss: 0.009203
Train Epoch: 1	[45120/60000 (75%)]	Loss: 0.003793
Train Epoch: 1	[45440/60000 (76%)]	Loss: 0.000766
Train Epoch: 1	[45760/60000 (76%)]	Loss: 0.002554
Train Epoch: 1	[46080/60000 (77%)]	Loss: 0.234560
Train Epoch: 1	[46400/60000 (77%)]	Loss: 0.074986
Train Epoch: 1	[46720/60000 (78%)]	Loss: 0.109243
Train Epoch: 1	[47040/60000 (78%)]	Loss: 0.081726
Train Epoch: 1	[47360/60000 (79%)]	Loss: 0.017880
Train Epoch: 1	[47680/60000 (79%)]	Loss: 0.045675



Train Epoch: 1	[48000/60000 (80%)]	Loss: 0.090406
Train Epoch: 1	[48320/60000 (81%)]	Loss: 0.012954
Train Epoch: 1	[48640/60000 (81%)]	Loss: 0.033686
Train Epoch: 1	[48960/60000 (82%)]	Loss: 0.186391
Train Epoch: 1	[49280/60000 (82%)]	Loss: 0.016990
Train Epoch: 1	[49600/60000 (83%)]	Loss: 0.001865
Train Epoch: 1	[49920/60000 (83%)]	Loss: 0.034874
Train Epoch: 1	[50240/60000 (84%)]	Loss: 0.022570
Train Epoch: 1	[50560/60000 (84%)]	Loss: 0.016889
Train Epoch: 1	[50880/60000 (85%)]	Loss: 0.046832
Train Epoch: 1	[51200/60000 (85%)]	Loss: 0.013762
Train Epoch: 1	[51520/60000 (86%)]	Loss: 0.075284
Train Epoch: 1	[51840/60000 (86%)]	Loss: 0.004925
Train Epoch: 1	[52160/60000 (87%)]	Loss: 0.067178
Train Epoch: 1	[52480/60000 (87%)]	Loss: 0.094246
Train Epoch: 1	[52800/60000 (88%)]	Loss: 0.005648
Train Epoch: 1	[53120/60000 (89%)]	Loss: 0.033675
Train Epoch: 1	[53440/60000 (89%)]	Loss: 0.036500
Train Epoch: 1	[53760/60000 (90%)]	Loss: 0.014217
Train Epoch: 1	[54080/60000 (90%)]	Loss: 0.015439
Train Epoch: 1	[54400/60000 (91%)]	Loss: 0.092385
Train Epoch: 1	[54720/60000 (91%)]	Loss: 0.021593
Train Epoch: 1	[55040/60000 (92%)]	Loss: 0.107767
Train Epoch: 1	[55360/60000 (92%)]	Loss: 0.028139
Train Epoch: 1	[55680/60000 (93%)]	Loss: 0.003817
Train Epoch: 1	[56000/60000 (93%)]	Loss: 0.002313
Train Epoch: 1	[56320/60000 (94%)]	Loss: 0.254778
Train Epoch: 1	[56640/60000 (94%)]	Loss: 0.128323
Train Epoch: 1	[56960/60000 (95%)]	Loss: 0.296431
Train Epoch: 1	[57280/60000 (95%)]	Loss: 0.011164
Train Epoch: 1	[57600/60000 (96%)]	Loss: 0.004300
Train Epoch: 1	[57920/60000 (97%)]	Loss: 0.002254
Train Epoch: 1	[58240/60000 (97%)]	Loss: 0.075625
Train Epoch: 1	[58560/60000 (98%)]	Loss: 0.021423
Train Epoch: 1	[58880/60000 (98%)]	Loss: 0.120178
Train Epoch: 1	[59200/60000 (99%)]	Loss: 0.012793
Train Epoch: 1	[59520/60000 (99%)]	Loss: 0.004608
Train Epoch: 1	[59840/60000 (100%)]	Loss: 0.102087

Test set: Average loss: 0.0595, Accuracy: 9801/10000 (98%)

Train Epoch: 2	[0/60000 (0%)]	Loss: 0.027598
Train Epoch: 2	[320/60000 (1%)]	Loss: 0.013265
Train Epoch: 2	[640/60000 (1%)]	Loss: 0.116810
Train Epoch: 2	[960/60000 (2%)]	Loss: 0.032350
Train Epoch: 2	[1280/60000 (2%)]	Loss: 0.003220
Train Epoch: 2	[1600/60000 (3%)]	Loss: 0.131106
Train Epoch: 2	[1920/60000 (3%)]	Loss: 0.096831
Train Epoch: 2	[2240/60000 (4%)]	Loss: 0.197314
Train Epoch: 2	[2560/60000 (4%)]	Loss: 0.085863

Train Epoch: 2	[2880/60000 (5%)]	Loss: 0.006777
Train Epoch: 2	[3200/60000 (5%)]	Loss: 0.013499
Train Epoch: 2	[3520/60000 (6%)]	Loss: 0.073700
Train Epoch: 2	[3840/60000 (6%)]	Loss: 0.007383
Train Epoch: 2	[4160/60000 (7%)]	Loss: 0.001416
Train Epoch: 2	[4480/60000 (7%)]	Loss: 0.072672
Train Epoch: 2	[4800/60000 (8%)]	Loss: 0.031931
Train Epoch: 2	[5120/60000 (9%)]	Loss: 0.025991
Train Epoch: 2	[5440/60000 (9%)]	Loss: 0.001769
Train Epoch: 2	[5760/60000 (10%)]	Loss: 0.006585
Train Epoch: 2	[6080/60000 (10%)]	Loss: 0.005234
Train Epoch: 2	[6400/60000 (11%)]	Loss: 0.018970
Train Epoch: 2	[6720/60000 (11%)]	Loss: 0.077367
Train Epoch: 2	[7040/60000 (12%)]	Loss: 0.060200
Train Epoch: 2	[7360/60000 (12%)]	Loss: 0.037338
Train Epoch: 2	[7680/60000 (13%)]	Loss: 0.067538
Train Epoch: 2	[8000/60000 (13%)]	Loss: 0.038583
Train Epoch: 2	[8320/60000 (14%)]	Loss: 0.010117
Train Epoch: 2	[8640/60000 (14%)]	Loss: 0.006974
Train Epoch: 2	[8960/60000 (15%)]	Loss: 0.005288
Train Epoch: 2	[9280/60000 (15%)]	Loss: 0.011427
Train Epoch: 2	[9600/60000 (16%)]	Loss: 0.019216
Train Epoch: 2	[9920/60000 (17%)]	Loss: 0.006359
Train Epoch: 2	[10240/60000 (17%)]	Loss: 0.089099
Train Epoch: 2	[10560/60000 (18%)]	Loss: 0.007622
Train Epoch: 2	[10880/60000 (18%)]	Loss: 0.040056
Train Epoch: 2	[11200/60000 (19%)]	Loss: 0.071797
Train Epoch: 2	[11520/60000 (19%)]	Loss: 0.024527
Train Epoch: 2	[11840/60000 (20%)]	Loss: 0.001512
Train Epoch: 2	[12160/60000 (20%)]	Loss: 0.061668
Train Epoch: 2	[12480/60000 (21%)]	Loss: 0.155943
Train Epoch: 2	[12800/60000 (21%)]	Loss: 0.003285
Train Epoch: 2	[13120/60000 (22%)]	Loss: 0.068065
Train Epoch: 2	[13440/60000 (22%)]	Loss: 0.101387
Train Epoch: 2	[13760/60000 (23%)]	Loss: 0.005211
Train Epoch: 2	[14080/60000 (23%)]	Loss: 0.074164
Train Epoch: 2	[14400/60000 (24%)]	Loss: 0.017029
Train Epoch: 2	[14720/60000 (25%)]	Loss: 0.042840
Train Epoch: 2	[15040/60000 (25%)]	Loss: 0.013853
Train Epoch: 2	[15360/60000 (26%)]	Loss: 0.019392
Train Epoch: 2	[15680/60000 (26%)]	Loss: 0.037165
Train Epoch: 2	[16000/60000 (27%)]	Loss: 0.012781
Train Epoch: 2	[16320/60000 (27%)]	Loss: 0.125172
Train Epoch: 2	[16640/60000 (28%)]	Loss: 0.000166
Train Epoch: 2	[16960/60000 (28%)]	Loss: 0.030852
Train Epoch: 2	[17280/60000 (29%)]	Loss: 0.143978
Train Epoch: 2	[17600/60000 (29%)]	Loss: 0.076106
Train Epoch: 2	[17920/60000 (30%)]	Loss: 0.006497
Train Epoch: 2	[18240/60000 (30%)]	Loss: 0.002177
Train Epoch: 2	[18560/60000 (31%)]	Loss: 0.209025

Train Epoch: 2	[18880/60000 (31%) ]	Loss: 0.048081
Train Epoch: 2	[19200/60000 (32%) ]	Loss: 0.000529
Train Epoch: 2	[19520/60000 (33%) ]	Loss: 0.005182
Train Epoch: 2	[19840/60000 (33%) ]	Loss: 0.004245
Train Epoch: 2	[20160/60000 (34%) ]	Loss: 0.137958
Train Epoch: 2	[20480/60000 (34%) ]	Loss: 0.011953
Train Epoch: 2	[20800/60000 (35%) ]	Loss: 0.135542
Train Epoch: 2	[21120/60000 (35%) ]	Loss: 0.005804
Train Epoch: 2	[21440/60000 (36%) ]	Loss: 0.020207
Train Epoch: 2	[21760/60000 (36%) ]	Loss: 0.030927
Train Epoch: 2	[22080/60000 (37%) ]	Loss: 0.013539
Train Epoch: 2	[22400/60000 (37%) ]	Loss: 0.000750
Train Epoch: 2	[22720/60000 (38%) ]	Loss: 0.009216
Train Epoch: 2	[23040/60000 (38%) ]	Loss: 0.046927
Train Epoch: 2	[23360/60000 (39%) ]	Loss: 0.000872
Train Epoch: 2	[23680/60000 (39%) ]	Loss: 0.000994
Train Epoch: 2	[24000/60000 (40%) ]	Loss: 0.059358
Train Epoch: 2	[24320/60000 (41%) ]	Loss: 0.059416
Train Epoch: 2	[24640/60000 (41%) ]	Loss: 0.076887
Train Epoch: 2	[24960/60000 (42%) ]	Loss: 0.082048
Train Epoch: 2	[25280/60000 (42%) ]	Loss: 0.001909
Train Epoch: 2	[25600/60000 (43%) ]	Loss: 0.012096
Train Epoch: 2	[25920/60000 (43%) ]	Loss: 0.008189
Train Epoch: 2	[26240/60000 (44%) ]	Loss: 0.005143
Train Epoch: 2	[26560/60000 (44%) ]	Loss: 0.017656
Train Epoch: 2	[26880/60000 (45%) ]	Loss: 0.001338
Train Epoch: 2	[27200/60000 (45%) ]	Loss: 0.065360
Train Epoch: 2	[27520/60000 (46%) ]	Loss: 0.002548
Train Epoch: 2	[27840/60000 (46%) ]	Loss: 0.006681
Train Epoch: 2	[28160/60000 (47%) ]	Loss: 0.097544
Train Epoch: 2	[28480/60000 (47%) ]	Loss: 0.015628
Train Epoch: 2	[28800/60000 (48%) ]	Loss: 0.009368
Train Epoch: 2	[29120/60000 (49%) ]	Loss: 0.031152
Train Epoch: 2	[29440/60000 (49%) ]	Loss: 0.133057
Train Epoch: 2	[29760/60000 (50%) ]	Loss: 0.015097
Train Epoch: 2	[30080/60000 (50%) ]	Loss: 0.021248
Train Epoch: 2	[30400/60000 (51%) ]	Loss: 0.012632
Train Epoch: 2	[30720/60000 (51%) ]	Loss: 0.004862
Train Epoch: 2	[31040/60000 (52%) ]	Loss: 0.005654
Train Epoch: 2	[31360/60000 (52%) ]	Loss: 0.017071
Train Epoch: 2	[31680/60000 (53%) ]	Loss: 0.005977
Train Epoch: 2	[32000/60000 (53%) ]	Loss: 0.002037
Train Epoch: 2	[32320/60000 (54%) ]	Loss: 0.001102
Train Epoch: 2	[32640/60000 (54%) ]	Loss: 0.001701
Train Epoch: 2	[32960/60000 (55%) ]	Loss: 0.098245
Train Epoch: 2	[33280/60000 (55%) ]	Loss: 0.275131
Train Epoch: 2	[33600/60000 (56%) ]	Loss: 0.044021
Train Epoch: 2	[33920/60000 (57%) ]	Loss: 0.082431
Train Epoch: 2	[34240/60000 (57%) ]	Loss: 0.011327
Train Epoch: 2	[34560/60000 (58%) ]	Loss: 0.008972

Train Epoch: 2	[34880/60000 (58%) ]	Loss: 0.006956
Train Epoch: 2	[35200/60000 (59%) ]	Loss: 0.025901
Train Epoch: 2	[35520/60000 (59%) ]	Loss: 0.003954
Train Epoch: 2	[35840/60000 (60%) ]	Loss: 0.000959
Train Epoch: 2	[36160/60000 (60%) ]	Loss: 0.006743
Train Epoch: 2	[36480/60000 (61%) ]	Loss: 0.004805
Train Epoch: 2	[36800/60000 (61%) ]	Loss: 0.000256
Train Epoch: 2	[37120/60000 (62%) ]	Loss: 0.111241
Train Epoch: 2	[37440/60000 (62%) ]	Loss: 0.050888
Train Epoch: 2	[37760/60000 (63%) ]	Loss: 0.009277
Train Epoch: 2	[38080/60000 (63%) ]	Loss: 0.005038
Train Epoch: 2	[38400/60000 (64%) ]	Loss: 0.099830
Train Epoch: 2	[38720/60000 (65%) ]	Loss: 0.392858
Train Epoch: 2	[39040/60000 (65%) ]	Loss: 0.007705
Train Epoch: 2	[39360/60000 (66%) ]	Loss: 0.000777
Train Epoch: 2	[39680/60000 (66%) ]	Loss: 0.089281
Train Epoch: 2	[40000/60000 (67%) ]	Loss: 0.007476
Train Epoch: 2	[40320/60000 (67%) ]	Loss: 0.022755
Train Epoch: 2	[40640/60000 (68%) ]	Loss: 0.007490
Train Epoch: 2	[40960/60000 (68%) ]	Loss: 0.001510
Train Epoch: 2	[41280/60000 (69%) ]	Loss: 0.003237
Train Epoch: 2	[41600/60000 (69%) ]	Loss: 0.005331
Train Epoch: 2	[41920/60000 (70%) ]	Loss: 0.024422
Train Epoch: 2	[42240/60000 (70%) ]	Loss: 0.005727
Train Epoch: 2	[42560/60000 (71%) ]	Loss: 0.000456
Train Epoch: 2	[42880/60000 (71%) ]	Loss: 0.107413
Train Epoch: 2	[43200/60000 (72%) ]	Loss: 0.051023
Train Epoch: 2	[43520/60000 (73%) ]	Loss: 0.026257
Train Epoch: 2	[43840/60000 (73%) ]	Loss: 0.095328
Train Epoch: 2	[44160/60000 (74%) ]	Loss: 0.001708
Train Epoch: 2	[44480/60000 (74%) ]	Loss: 0.001896
Train Epoch: 2	[44800/60000 (75%) ]	Loss: 0.014586
Train Epoch: 2	[45120/60000 (75%) ]	Loss: 0.089750
Train Epoch: 2	[45440/60000 (76%) ]	Loss: 0.020695
Train Epoch: 2	[45760/60000 (76%) ]	Loss: 0.004186
Train Epoch: 2	[46080/60000 (77%) ]	Loss: 0.039283
Train Epoch: 2	[46400/60000 (77%) ]	Loss: 0.040797
Train Epoch: 2	[46720/60000 (78%) ]	Loss: 0.026089
Train Epoch: 2	[47040/60000 (78%) ]	Loss: 0.001353
Train Epoch: 2	[47360/60000 (79%) ]	Loss: 0.263497
Train Epoch: 2	[47680/60000 (79%) ]	Loss: 0.008735
Train Epoch: 2	[48000/60000 (80%) ]	Loss: 0.015889
Train Epoch: 2	[48320/60000 (81%) ]	Loss: 0.033036
Train Epoch: 2	[48640/60000 (81%) ]	Loss: 0.033967
Train Epoch: 2	[48960/60000 (82%) ]	Loss: 0.006596
Train Epoch: 2	[49280/60000 (82%) ]	Loss: 0.023370
Train Epoch: 2	[49600/60000 (83%) ]	Loss: 0.008194
Train Epoch: 2	[49920/60000 (83%) ]	Loss: 0.005242
Train Epoch: 2	[50240/60000 (84%) ]	Loss: 0.000618
Train Epoch: 2	[50560/60000 (84%) ]	Loss: 0.164030

```

Train Epoch: 2 [50880/60000 (85%)]    Loss: 0.002200
Train Epoch: 2 [51200/60000 (85%)]    Loss: 0.047767
Train Epoch: 2 [51520/60000 (86%)]    Loss: 0.003815
Train Epoch: 2 [51840/60000 (86%)]    Loss: 0.005997
Train Epoch: 2 [52160/60000 (87%)]    Loss: 0.012298
Train Epoch: 2 [52480/60000 (87%)]    Loss: 0.002426
Train Epoch: 2 [52800/60000 (88%)]    Loss: 0.094435
Train Epoch: 2 [53120/60000 (89%)]    Loss: 0.014694
Train Epoch: 2 [53440/60000 (89%)]    Loss: 0.003037
Train Epoch: 2 [53760/60000 (90%)]    Loss: 0.061238
Train Epoch: 2 [54080/60000 (90%)]    Loss: 0.102848
Train Epoch: 2 [54400/60000 (91%)]    Loss: 0.064298
Train Epoch: 2 [54720/60000 (91%)]    Loss: 0.062178
Train Epoch: 2 [55040/60000 (92%)]    Loss: 0.003232
Train Epoch: 2 [55360/60000 (92%)]    Loss: 0.011045
Train Epoch: 2 [55680/60000 (93%)]    Loss: 0.013708
Train Epoch: 2 [56000/60000 (93%)]    Loss: 0.003573
Train Epoch: 2 [56320/60000 (94%)]    Loss: 0.000815
Train Epoch: 2 [56640/60000 (94%)]    Loss: 0.046185
Train Epoch: 2 [56960/60000 (95%)]    Loss: 0.000721
Train Epoch: 2 [57280/60000 (95%)]    Loss: 0.188234
Train Epoch: 2 [57600/60000 (96%)]    Loss: 0.004570
Train Epoch: 2 [57920/60000 (97%)]    Loss: 0.002158
Train Epoch: 2 [58240/60000 (97%)]    Loss: 0.005151
Train Epoch: 2 [58560/60000 (98%)]    Loss: 0.020486
Train Epoch: 2 [58880/60000 (98%)]    Loss: 0.011041
Train Epoch: 2 [59200/60000 (99%)]    Loss: 0.022238
Train Epoch: 2 [59520/60000 (99%)]    Loss: 0.002136
Train Epoch: 2 [59840/60000 (100%)]  Loss: 0.208707

```

Test set: Average loss: 0.0575, Accuracy: 9813/10000 (98%)

3) Results - lr = 1, epochs = 1, batch\_size = 64

```

model = Net()
if args['cuda']:
    model.cuda()

optimizer = optim.SGD(model.parameters(), lr=1,
momentum=args['momentum'])

for epoch in range(1, args['epochs'] + 1):
    train(epoch)
    test()

Train Epoch: 1 [0/60000 (0%)]    Loss: 2.309644
Train Epoch: 1 [640/60000 (1%)]  Loss: 2.610038
Train Epoch: 1 [1280/60000 (2%)] Loss: 2.293633
Train Epoch: 1 [1920/60000 (3%)] Loss: 2.326873

```

Train Epoch: 1	[2560/60000 (4%)]	Loss: 2.287273
Train Epoch: 1	[3200/60000 (5%)]	Loss: 2.297387
Train Epoch: 1	[3840/60000 (6%)]	Loss: 2.313125
Train Epoch: 1	[4480/60000 (7%)]	Loss: 2.311837
Train Epoch: 1	[5120/60000 (9%)]	Loss: 2.321790
Train Epoch: 1	[5760/60000 (10%)]	Loss: 2.305141
Train Epoch: 1	[6400/60000 (11%)]	Loss: 2.347255
Train Epoch: 1	[7040/60000 (12%)]	Loss: 2.342577
Train Epoch: 1	[7680/60000 (13%)]	Loss: 2.329440
Train Epoch: 1	[8320/60000 (14%)]	Loss: 2.344549
Train Epoch: 1	[8960/60000 (15%)]	Loss: 2.289619
Train Epoch: 1	[9600/60000 (16%)]	Loss: 2.309316
Train Epoch: 1	[10240/60000 (17%)]	Loss: 2.327851
Train Epoch: 1	[10880/60000 (18%)]	Loss: 2.329154
Train Epoch: 1	[11520/60000 (19%)]	Loss: 2.299373
Train Epoch: 1	[12160/60000 (20%)]	Loss: 2.312238
Train Epoch: 1	[12800/60000 (21%)]	Loss: 2.290203
Train Epoch: 1	[13440/60000 (22%)]	Loss: 2.292499
Train Epoch: 1	[14080/60000 (23%)]	Loss: 2.290425
Train Epoch: 1	[14720/60000 (25%)]	Loss: 2.302808
Train Epoch: 1	[15360/60000 (26%)]	Loss: 2.296575
Train Epoch: 1	[16000/60000 (27%)]	Loss: 2.322124
Train Epoch: 1	[16640/60000 (28%)]	Loss: 2.309566
Train Epoch: 1	[17280/60000 (29%)]	Loss: 2.341048
Train Epoch: 1	[17920/60000 (30%)]	Loss: 2.288712
Train Epoch: 1	[18560/60000 (31%)]	Loss: 2.318647
Train Epoch: 1	[19200/60000 (32%)]	Loss: 2.297338
Train Epoch: 1	[19840/60000 (33%)]	Loss: 2.310686
Train Epoch: 1	[20480/60000 (34%)]	Loss: 2.298467
Train Epoch: 1	[21120/60000 (35%)]	Loss: 2.315113
Train Epoch: 1	[21760/60000 (36%)]	Loss: 2.312908
Train Epoch: 1	[22400/60000 (37%)]	Loss: 2.298875
Train Epoch: 1	[23040/60000 (38%)]	Loss: 2.292692
Train Epoch: 1	[23680/60000 (39%)]	Loss: 2.307751
Train Epoch: 1	[24320/60000 (41%)]	Loss: 2.301772
Train Epoch: 1	[24960/60000 (42%)]	Loss: 2.299584
Train Epoch: 1	[25600/60000 (43%)]	Loss: 2.300166
Train Epoch: 1	[26240/60000 (44%)]	Loss: 2.295081
Train Epoch: 1	[26880/60000 (45%)]	Loss: 2.330679
Train Epoch: 1	[27520/60000 (46%)]	Loss: 2.304136
Train Epoch: 1	[28160/60000 (47%)]	Loss: 2.288864
Train Epoch: 1	[28800/60000 (48%)]	Loss: 2.316888
Train Epoch: 1	[29440/60000 (49%)]	Loss: 2.306967
Train Epoch: 1	[30080/60000 (50%)]	Loss: 2.332793
Train Epoch: 1	[30720/60000 (51%)]	Loss: 2.329526
Train Epoch: 1	[31360/60000 (52%)]	Loss: 2.319644
Train Epoch: 1	[32000/60000 (53%)]	Loss: 2.294146
Train Epoch: 1	[32640/60000 (54%)]	Loss: 2.301515
Train Epoch: 1	[33280/60000 (55%)]	Loss: 2.301580

Train Epoch: 1	[33920/60000 (57%)]	Loss: 2.306991
Train Epoch: 1	[34560/60000 (58%)]	Loss: 2.323599
Train Epoch: 1	[35200/60000 (59%)]	Loss: 2.296302
Train Epoch: 1	[35840/60000 (60%)]	Loss: 2.297580
Train Epoch: 1	[36480/60000 (61%)]	Loss: 2.328892
Train Epoch: 1	[37120/60000 (62%)]	Loss: 2.306561
Train Epoch: 1	[37760/60000 (63%)]	Loss: 2.302936
Train Epoch: 1	[38400/60000 (64%)]	Loss: 2.324066
Train Epoch: 1	[39040/60000 (65%)]	Loss: 2.338241
Train Epoch: 1	[39680/60000 (66%)]	Loss: 2.305665
Train Epoch: 1	[40320/60000 (67%)]	Loss: 2.308074
Train Epoch: 1	[40960/60000 (68%)]	Loss: 2.296663
Train Epoch: 1	[41600/60000 (69%)]	Loss: 2.315626
Train Epoch: 1	[42240/60000 (70%)]	Loss: 2.305688
Train Epoch: 1	[42880/60000 (71%)]	Loss: 2.328074
Train Epoch: 1	[43520/60000 (72%)]	Loss: 2.334857
Train Epoch: 1	[44160/60000 (74%)]	Loss: 2.307196
Train Epoch: 1	[44800/60000 (75%)]	Loss: 2.298425
Train Epoch: 1	[45440/60000 (76%)]	Loss: 2.316379
Train Epoch: 1	[46080/60000 (77%)]	Loss: 2.286515
Train Epoch: 1	[46720/60000 (78%)]	Loss: 2.290273
Train Epoch: 1	[47360/60000 (79%)]	Loss: 2.289613
Train Epoch: 1	[48000/60000 (80%)]	Loss: 2.308444
Train Epoch: 1	[48640/60000 (81%)]	Loss: 2.299067
Train Epoch: 1	[49280/60000 (82%)]	Loss: 2.326264
Train Epoch: 1	[49920/60000 (83%)]	Loss: 2.315279
Train Epoch: 1	[50560/60000 (84%)]	Loss: 2.298985
Train Epoch: 1	[51200/60000 (85%)]	Loss: 2.306021
Train Epoch: 1	[51840/60000 (86%)]	Loss: 2.342448
Train Epoch: 1	[52480/60000 (87%)]	Loss: 2.314674
Train Epoch: 1	[53120/60000 (88%)]	Loss: 2.298257
Train Epoch: 1	[53760/60000 (90%)]	Loss: 2.317083
Train Epoch: 1	[54400/60000 (91%)]	Loss: 2.289733
Train Epoch: 1	[55040/60000 (92%)]	Loss: 2.328220
Train Epoch: 1	[55680/60000 (93%)]	Loss: 2.306603
Train Epoch: 1	[56320/60000 (94%)]	Loss: 2.287374
Train Epoch: 1	[56960/60000 (95%)]	Loss: 2.308452
Train Epoch: 1	[57600/60000 (96%)]	Loss: 2.318622
Train Epoch: 1	[58240/60000 (97%)]	Loss: 2.331951
Train Epoch: 1	[58880/60000 (98%)]	Loss: 2.304842
Train Epoch: 1	[59520/60000 (99%)]	Loss: 2.333622

```
/usr/local/lib/python3.10/dist-packages/torch/nn/_reduction.py:51:
UserWarning: size_average and reduce args will be deprecated, please
use reduction='sum' instead.
  warnings.warn(warning.format(ret))
```

Test set: Average loss: 2.3141, Accuracy: 1028/10000 (10%)

# 1. Through the experiment, what is the best configuration?

Through experimentation, the best-performing configuration included two convolutional layers followed by two fully connected layers. This setup was further optimized with the following hyperparameters:

- Learning Rate: 0.05
- Batch Size: 32
- Epochs: 10

This configuration achieved the highest test accuracy at 98% and the lowest test loss.

- **Two Convolutional Layers:** By using two convolutional layers, the network could better capture both low-level features (like edges and textures) and higher-level patterns (like curves and shapes). This made it more capable of distinguishing between visually similar digits. Adding layers improved the model's capacity without significantly increasing the risk of overfitting, as each layer was still focused on extracting different levels of spatial hierarchies.
- **Learning Rate:** A learning rate of 0.05 enabled efficient and stable learning. An excessively high learning rate would risk overshooting optimal parameters, potentially leading to underfitting, while a lower rate could cause slow convergence or even overfitting as it "memorizes" instead of generalizing.
- **Batch Size and Epochs:** The combination of a smaller batch size (32) and more epochs (10) allowed for frequent parameter updates and enough cycles for convergence without excessive training, which could lead to overfitting.

# 2. What prediction accuracy on the test set you got?

Using this optimized configuration, the test set accuracy reached 98%, which was an improvement over the initial configuration with only fully connected layers that achieved 93%. Observing this accuracy on unseen data suggests that the model generalized well, as it avoided common pitfalls like overfitting, where a model performs well on training data but poorly on test data.

- **Overfitting:** In this case, the model showed resilience against overfitting despite the added layers. This may be due to the structured architecture of CNNs, which are less prone to overfitting compared to fully connected networks, especially for image tasks. Convolutional layers apply weight sharing and pooling, making the model learn spatial hierarchies instead of memorizing individual pixels.



- **Underfitting:** Initially, the simpler model (with only fully connected layers) exhibited signs of underfitting. It struggled to capture the complex patterns necessary for digit recognition, resulting in a lower test accuracy. Adding convolutional layers helped mitigate this issue by enabling the model to learn a broader range of features, ultimately improving test performance.

### 3. What did you learn?

This experiment underlined several fundamental machine learning principles such as below:

- **Effectiveness of CNN Layers:** Adding convolutional layers significantly improved the model's ability to capture features in the MNIST images, leading to higher accuracy compared to a model with only fully connected layers.
- **Hyperparameter Tuning:** The optimal combination of learning rate, batch size, and the number of epochs played a crucial role in achieving the best results. For example, increasing the number of epochs and using a smaller batch size allowed the model to learn the data distribution better.
- **Importance of Experimentation:** Testing different network topologies and hyperparameters demonstrated how small adjustments can lead to significant improvements in accuracy, underscoring the importance of experimentation in model optimization.