```
# Import dependencies
import numpy as np
import torch
import torchvision
from torch.utils.data.dataset import Dataset
from torchvision import datasets, transforms
from torch import nn, optim
import matplotlib.pyplot as plt
```

Problem 4.2 Determine appropriate mini-batch size = 32, epoch = 100

```
# ################## Part 1: Load data and create batch
##################
N_total = 600
N_train = 500
x = torch.unsqueeze(torch.linspace(0, 1, N_total), dim=1)
r = torch.randperm(N_total)
x = x[r, :]
y = 0.2 + 0.4 * torch.pow(x, 2) + 0.3 * x * torch.sin(15 * x) + 0.05 *
torch.cos(50 * x)

class CustomDataset(Dataset):
    def __init__(self, x, y):
        self.y = y
        self.x = x

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

    def __getitem__(self, idx):
        y1 = self.y[idx]
        x1 = self.x[idx]
        return (x1, y1)

# Change batch_size here to test different values
batch_size = 32  # Experiment with different batch sizes: 32, 64, 128
trainset = CustomDataset(x[0:N_train, :], y[0:N_train, :])
testset = CustomDataset(x[N_train:N_total, :], y[N_train:N_total, :])
train_loader = torch.utils.data.DataLoader(trainset,
batch_size=batch_size)
test_loader = torch.utils.data.DataLoader(testset,
batch_size=batch_size)

# ################## Part 2: Define Model ##################
model = nn.Sequential(
    nn.Linear(1, 1024, bias=True),
    nn.ReLU(),
    nn.Linear(1024, 1, bias=True)
)
```

```
def init_weights(m):
    if isinstance(m, nn.Linear):
        m.weight.data.uniform_(-1, 1)
        m.bias.data.uniform_(-1, 1)

model.apply(init_weights)
```

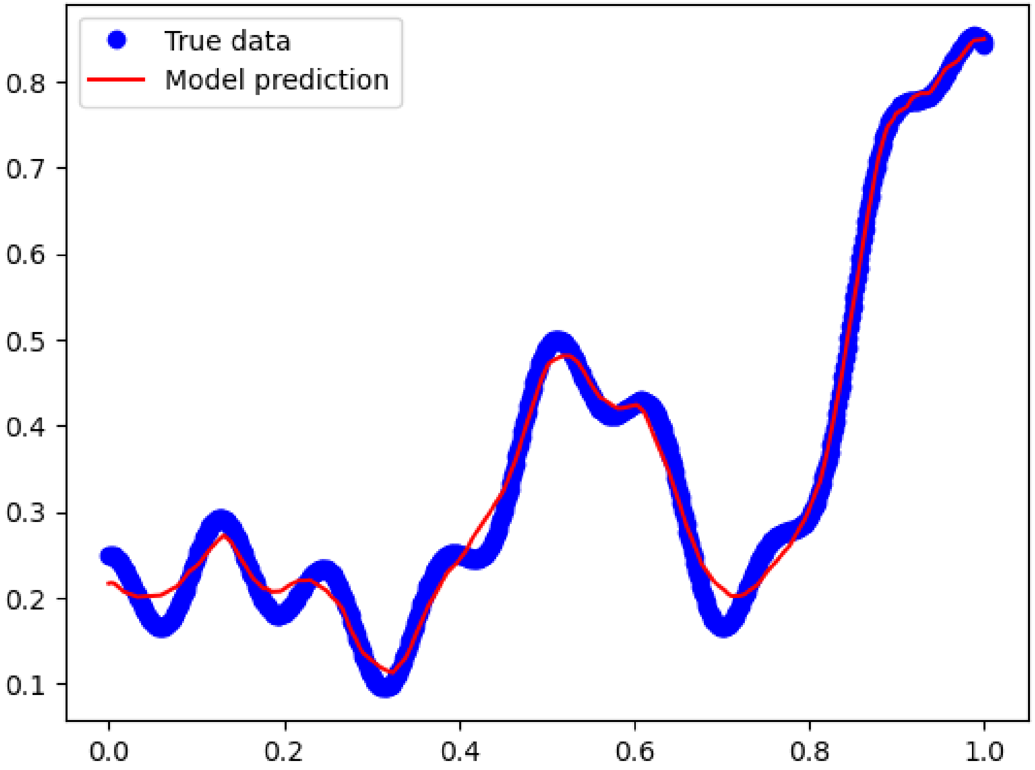
```
Sequential(
  (0): Linear(in_features=1, out_features=1024, bias=True)
  (1): ReLU()
  (2): Linear(in_features=1024, out_features=1, bias=True)
)
```

```
# ################## Part 3: Define Loss and Optimizer
##################
criterion = nn.MSELoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

# ################## Part 4: Training and Testing ##################
def train_NN():
    model.train()
    for images, labels in train_loader:
        out = model(images)
        loss = criterion(out, labels)
        loss.backward()
        optimizer.step()
        optimizer.zero_grad()
    return loss

def test_NN(loader):
    model.eval()
    loss = 0
    with torch.no_grad():
        for images, labels in loader:
            out = model(images)
            loss += criterion(out, labels).item()
    loss = loss / len(loader)
    return loss

# Experiment with different numbers of epochs
N_epoch = 100
train_loss = np.zeros((N_epoch, 1))
test_loss = np.zeros((N_epoch, 1))

for epoch in range(N_epoch):
    train_NN()
    train_loss[epoch, 0] = test_NN(train_loader)
    test_loss[epoch, 0] = test_NN(test_loader)
    print(f'Epoch: {epoch+1:03d}, Train Loss: {train_loss[epoch,
0]:.7f}, Test Loss: {test_loss[epoch, 0]:.7f}')
```

```
Epoch: 001, Train Loss: 3.0408284, Test Loss: 2.6833682
Epoch: 002, Train Loss: 0.1522782, Test Loss: 0.1523975
Epoch: 003, Train Loss: 0.0454572, Test Loss: 0.0447318
Epoch: 004, Train Loss: 0.0322065, Test Loss: 0.0324125
Epoch: 005, Train Loss: 0.0231701, Test Loss: 0.0234418
Epoch: 006, Train Loss: 0.0173245, Test Loss: 0.0179732
Epoch: 007, Train Loss: 0.0132794, Test Loss: 0.0145717
Epoch: 008, Train Loss: 0.0106569, Test Loss: 0.0114137
Epoch: 009, Train Loss: 0.0086298, Test Loss: 0.0096973
Epoch: 010, Train Loss: 0.0073026, Test Loss: 0.0081288
Epoch: 011, Train Loss: 0.0063120, Test Loss: 0.0070580
Epoch: 012, Train Loss: 0.0055604, Test Loss: 0.0062362
Epoch: 013, Train Loss: 0.0049826, Test Loss: 0.0055524
Epoch: 014, Train Loss: 0.0045235, Test Loss: 0.0050087
Epoch: 015, Train Loss: 0.0041535, Test Loss: 0.0045653
Epoch: 016, Train Loss: 0.0038502, Test Loss: 0.0041954
Epoch: 017, Train Loss: 0.0035964, Test Loss: 0.0038838
Epoch: 018, Train Loss: 0.0033807, Test Loss: 0.0036204
Epoch: 019, Train Loss: 0.0031939, Test Loss: 0.0033937
Epoch: 020, Train Loss: 0.0030302, Test Loss: 0.0031965
Epoch: 021, Train Loss: 0.0028844, Test Loss: 0.0030234
Epoch: 022, Train Loss: 0.0027535, Test Loss: 0.0028683
Epoch: 023, Train Loss: 0.0026343, Test Loss: 0.0027294
Epoch: 024, Train Loss: 0.0025246, Test Loss: 0.0026046
Epoch: 025, Train Loss: 0.0024233, Test Loss: 0.0024920
Epoch: 026, Train Loss: 0.0023288, Test Loss: 0.0023890
Epoch: 027, Train Loss: 0.0022407, Test Loss: 0.0022948
Epoch: 028, Train Loss: 0.0021583, Test Loss: 0.0022080
Epoch: 029, Train Loss: 0.0020810, Test Loss: 0.0021280
Epoch: 030, Train Loss: 0.0020086, Test Loss: 0.0020535
Epoch: 031, Train Loss: 0.0019404, Test Loss: 0.0019846
Epoch: 032, Train Loss: 0.0018763, Test Loss: 0.0019204
Epoch: 033, Train Loss: 0.0018161, Test Loss: 0.0018605
Epoch: 034, Train Loss: 0.0017592, Test Loss: 0.0018048
Epoch: 035, Train Loss: 0.0017059, Test Loss: 0.0017534
Epoch: 036, Train Loss: 0.0016557, Test Loss: 0.0017055
Epoch: 037, Train Loss: 0.0016081, Test Loss: 0.0016602
Epoch: 038, Train Loss: 0.0015631, Test Loss: 0.0016174
Epoch: 039, Train Loss: 0.0015205, Test Loss: 0.0015769
Epoch: 040, Train Loss: 0.0014798, Test Loss: 0.0015377
Epoch: 041, Train Loss: 0.0014411, Test Loss: 0.0014996
Epoch: 042, Train Loss: 0.0014041, Test Loss: 0.0014621
Epoch: 043, Train Loss: 0.0013684, Test Loss: 0.0014252
Epoch: 044, Train Loss: 0.0013336, Test Loss: 0.0013883
Epoch: 045, Train Loss: 0.0012997, Test Loss: 0.0013510
Epoch: 046, Train Loss: 0.0012664, Test Loss: 0.0013139
Epoch: 047, Train Loss: 0.0012337, Test Loss: 0.0012764
Epoch: 048, Train Loss: 0.0012013, Test Loss: 0.0012382
Epoch: 049, Train Loss: 0.0011696, Test Loss: 0.0012003
Epoch: 050, Train Loss: 0.0011387, Test Loss: 0.0011627
```

```
Epoch: 051, Train Loss: 0.0011085, Test Loss: 0.0011258
Epoch: 052, Train Loss: 0.0010792, Test Loss: 0.0010898
Epoch: 053, Train Loss: 0.0010507, Test Loss: 0.0010550
Epoch: 054, Train Loss: 0.0010232, Test Loss: 0.0010215
Epoch: 055, Train Loss: 0.0009967, Test Loss: 0.0009893
Epoch: 056, Train Loss: 0.0009712, Test Loss: 0.0009583
Epoch: 057, Train Loss: 0.0009468, Test Loss: 0.0009286
Epoch: 058, Train Loss: 0.0009233, Test Loss: 0.0009000
Epoch: 059, Train Loss: 0.0009008, Test Loss: 0.0008730
Epoch: 060, Train Loss: 0.0008792, Test Loss: 0.0008472
Epoch: 061, Train Loss: 0.0008583, Test Loss: 0.0008225
Epoch: 062, Train Loss: 0.0008382, Test Loss: 0.0007989
Epoch: 063, Train Loss: 0.0008187, Test Loss: 0.0007765
Epoch: 064, Train Loss: 0.0008000, Test Loss: 0.0007552
Epoch: 065, Train Loss: 0.0007819, Test Loss: 0.0007349
Epoch: 066, Train Loss: 0.0007644, Test Loss: 0.0007156
Epoch: 067, Train Loss: 0.0007475, Test Loss: 0.0006971
Epoch: 068, Train Loss: 0.0007312, Test Loss: 0.0006794
Epoch: 069, Train Loss: 0.0007153, Test Loss: 0.0006624
Epoch: 070, Train Loss: 0.0006999, Test Loss: 0.0006460
Epoch: 071, Train Loss: 0.0006849, Test Loss: 0.0006304
Epoch: 072, Train Loss: 0.0006705, Test Loss: 0.0006155
Epoch: 073, Train Loss: 0.0006564, Test Loss: 0.0006010
Epoch: 074, Train Loss: 0.0006427, Test Loss: 0.0005871
Epoch: 075, Train Loss: 0.0006294, Test Loss: 0.0005737
Epoch: 076, Train Loss: 0.0006166, Test Loss: 0.0005608
Epoch: 077, Train Loss: 0.0006040, Test Loss: 0.0005485
Epoch: 078, Train Loss: 0.0005917, Test Loss: 0.0005365
Epoch: 079, Train Loss: 0.0005798, Test Loss: 0.0005250
Epoch: 080, Train Loss: 0.0005683, Test Loss: 0.0005138
Epoch: 081, Train Loss: 0.0005570, Test Loss: 0.0005031
Epoch: 082, Train Loss: 0.0005459, Test Loss: 0.0004926
Epoch: 083, Train Loss: 0.0005352, Test Loss: 0.0004824
Epoch: 084, Train Loss: 0.0005248, Test Loss: 0.0004723
Epoch: 085, Train Loss: 0.0005147, Test Loss: 0.0004624
Epoch: 086, Train Loss: 0.0005050, Test Loss: 0.0004528
Epoch: 087, Train Loss: 0.0004954, Test Loss: 0.0004435
Epoch: 088, Train Loss: 0.0004861, Test Loss: 0.0004343
Epoch: 089, Train Loss: 0.0004769, Test Loss: 0.0004254
Epoch: 090, Train Loss: 0.0004681, Test Loss: 0.0004168
Epoch: 091, Train Loss: 0.0004594, Test Loss: 0.0004084
Epoch: 092, Train Loss: 0.0004510, Test Loss: 0.0004001
Epoch: 093, Train Loss: 0.0004428, Test Loss: 0.0003921
Epoch: 094, Train Loss: 0.0004346, Test Loss: 0.0003844
Epoch: 095, Train Loss: 0.0004266, Test Loss: 0.0003767
Epoch: 096, Train Loss: 0.0004188, Test Loss: 0.0003693
Epoch: 097, Train Loss: 0.0004112, Test Loss: 0.0003621
Epoch: 098, Train Loss: 0.0004038, Test Loss: 0.0003550
```

```
Epoch: 099, Train Loss: 0.0003965, Test Loss: 0.0003481
Epoch: 100, Train Loss: 0.0003893, Test Loss: 0.0003414

# ################## Final Prediction ##################
x_test = torch.unsqueeze(torch.linspace(0, 1, 1999), dim=1)
y_test = model(x_test)

# Plot the results
plt.plot(x[0:N_total], y[0:N_total], 'bo', label='True data')
plt.plot(x_test, y_test.detach().numpy(), 'r', label='Model
prediction')
plt.legend()
plt.show()
```



Problem 4.2 Determine appropriate mini-batch size = 16, epoch = 100

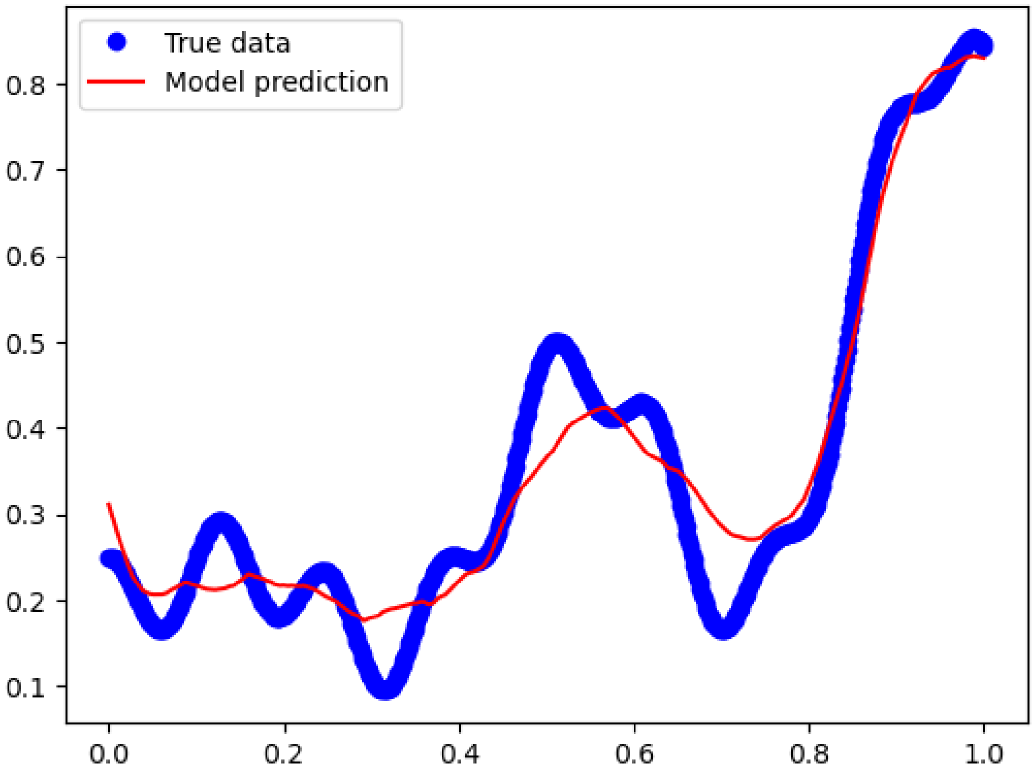
```
# ################## Part 1: Load data and create batch
##################
N_total = 600
N_train = 500
x = torch.unsqueeze(torch.linspace(0, 1, N_total), dim=1)
r = torch.randperm(N_total)
x = x[r, :]
y = 0.2 + 0.4 * torch.pow(x, 2) + 0.3 * x * torch.sin(15 * x) + 0.05 *
torch.cos(50 * x)
```

```
class CustomDataset(Dataset):
    def __init__(self, x, y):
        self.y = y
        self.x = x

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

    def __getitem__(self, idx):
        y1 = self.y[idx]
        x1 = self.x[idx]
        return (x1, y1)

# Change batch_size here to test different values
batch_size = 128  # Experiment with different batch sizes: 32, 64, 128
trainset = CustomDataset(x[0:N_train, :], y[0:N_train, :])
testset = CustomDataset(x[N_train:N_total, :], y[N_train:N_total, :])
train_loader = torch.utils.data.DataLoader(trainset,
batch_size=batch_size)
test_loader = torch.utils.data.DataLoader(testset,
batch_size=batch_size)

# ################## Part 2: Define Model ##################
model = nn.Sequential(
    nn.Linear(1, 1024, bias=True),
    nn.ReLU(),
    nn.Linear(1024, 1, bias=True)
)

def init_weights(m):
    if isinstance(m, nn.Linear):
        m.weight.data.uniform_(-1, 1)
        m.bias.data.uniform_(-1, 1)

model.apply(init_weights)

# ################## Part 3: Define Loss and Optimizer
##################
criterion = nn.MSELoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

# ################## Part 4: Training and Testing ##################
def train_NN():
    model.train()
    for images, labels in train_loader:
        out = model(images)
        loss = criterion(out, labels)
        loss.backward()
        optimizer.step()
        optimizer.zero_grad()
    return loss

def test_NN(loader):
    model.eval()
    loss = 0
    with torch.no_grad():
        for images, labels in loader:
            out = model(images)
            loss += criterion(out, labels).item()
    loss = loss / len(loader)
    return loss

# Experiment with different numbers of epochs
N_epoch = 100
train_loss = np.zeros((N_epoch, 1))
test_loss = np.zeros((N_epoch, 1))

for epoch in range(N_epoch):
    train_NN()
    train_loss[epoch, 0] = test_NN(train_loader)
    test_loss[epoch, 0] = test_NN(test_loader)
    print(f'Epoch: {epoch+1:03d}, Train Loss: {train_loss[epoch,
0]:.7f}, Test Loss: {test_loss[epoch, 0]:.7f}')

# ################## Final Prediction ##################
x_test = torch.unsqueeze(torch.linspace(0, 1, 1999), dim=1)
y_test = model(x_test)

# Plot the results
plt.plot(x[0:N_total], y[0:N_total], 'bo', label='True data')
plt.plot(x_test, y_test.detach().numpy(), 'r', label='Model
prediction')
plt.legend()
plt.show()
```

```
Epoch: 001, Train Loss: 20.4632916, Test Loss: 21.1496086
Epoch: 002, Train Loss: 11.2453420, Test Loss: 12.3400145
Epoch: 003, Train Loss: 9.9087901, Test Loss: 11.2133293
Epoch: 004, Train Loss: 9.3603978, Test Loss: 10.5421286
Epoch: 005, Train Loss: 7.1279855, Test Loss: 8.0017729
Epoch: 006, Train Loss: 4.5086163, Test Loss: 5.0775385
Epoch: 007, Train Loss: 2.8985459, Test Loss: 3.2626579
Epoch: 008, Train Loss: 2.2327487, Test Loss: 2.4915483
Epoch: 009, Train Loss: 1.7291094, Test Loss: 1.9289263
Epoch: 010, Train Loss: 1.1206206, Test Loss: 1.2632594
Epoch: 011, Train Loss: 0.6755160, Test Loss: 0.7543423
Epoch: 012, Train Loss: 0.5096188, Test Loss: 0.5315038
Epoch: 013, Train Loss: 0.4424562, Test Loss: 0.4302731
Epoch: 014, Train Loss: 0.3534248, Test Loss: 0.3353279
```

```
Epoch: 015, Train Loss: 0.2882117, Test Loss: 0.2828371
Epoch: 016, Train Loss: 0.2728086, Test Loss: 0.2817203
Epoch: 017, Train Loss: 0.2624435, Test Loss: 0.2744622
Epoch: 018, Train Loss: 0.2374812, Test Loss: 0.2411609
Epoch: 019, Train Loss: 0.2158103, Test Loss: 0.2082577
Epoch: 020, Train Loss: 0.2009166, Test Loss: 0.1872455
Epoch: 021, Train Loss: 0.1834664, Test Loss: 0.1709518
Epoch: 022, Train Loss: 0.1649070, Test Loss: 0.1576907
Epoch: 023, Train Loss: 0.1499047, Test Loss: 0.1474956
Epoch: 024, Train Loss: 0.1368052, Test Loss: 0.1360662
Epoch: 025, Train Loss: 0.1241553, Test Loss: 0.1224505
Epoch: 026, Train Loss: 0.1130936, Test Loss: 0.1098194
Epoch: 027, Train Loss: 0.1035068, Test Loss: 0.0996372
Epoch: 028, Train Loss: 0.0945654, Test Loss: 0.0913040
Epoch: 029, Train Loss: 0.0863846, Test Loss: 0.0841472
Epoch: 030, Train Loss: 0.0790432, Test Loss: 0.0774125
Epoch: 031, Train Loss: 0.0722813, Test Loss: 0.0705894
Epoch: 032, Train Loss: 0.0660765, Test Loss: 0.0639731
Epoch: 033, Train Loss: 0.0604625, Test Loss: 0.0580483
Epoch: 034, Train Loss: 0.0553455, Test Loss: 0.0529255
Epoch: 035, Train Loss: 0.0506900, Test Loss: 0.0484510
Epoch: 036, Train Loss: 0.0464819, Test Loss: 0.0444016
Epoch: 037, Train Loss: 0.0426712, Test Loss: 0.0406250
Epoch: 038, Train Loss: 0.0392197, Test Loss: 0.0371288
Epoch: 039, Train Loss: 0.0361013, Test Loss: 0.0339912
Epoch: 040, Train Loss: 0.0332790, Test Loss: 0.0312273
Epoch: 041, Train Loss: 0.0307252, Test Loss: 0.0287764
Epoch: 042, Train Loss: 0.0284149, Test Loss: 0.0265639
Epoch: 043, Train Loss: 0.0263250, Test Loss: 0.0245394
Epoch: 044, Train Loss: 0.0244344, Test Loss: 0.0226980
Epoch: 045, Train Loss: 0.0227231, Test Loss: 0.0210416
Epoch: 046, Train Loss: 0.0211722, Test Loss: 0.0195588
Epoch: 047, Train Loss: 0.0197658, Test Loss: 0.0182223
Epoch: 048, Train Loss: 0.0184888, Test Loss: 0.0170080
Epoch: 049, Train Loss: 0.0173279, Test Loss: 0.0159006
Epoch: 050, Train Loss: 0.0162708, Test Loss: 0.0148928
Epoch: 051, Train Loss: 0.0153068, Test Loss: 0.0139772
Epoch: 052, Train Loss: 0.0144259, Test Loss: 0.0131443
Epoch: 053, Train Loss: 0.0136195, Test Loss: 0.0123831
Epoch: 054, Train Loss: 0.0128795, Test Loss: 0.0116848
Epoch: 055, Train Loss: 0.0121990, Test Loss: 0.0110430
Epoch: 056, Train Loss: 0.0115725, Test Loss: 0.0104528
Epoch: 057, Train Loss: 0.0109943, Test Loss: 0.0099096
Epoch: 058, Train Loss: 0.0104592, Test Loss: 0.0094082
Epoch: 059, Train Loss: 0.0099631, Test Loss: 0.0089443
Epoch: 060, Train Loss: 0.0095024, Test Loss: 0.0085143
Epoch: 061, Train Loss: 0.0090736, Test Loss: 0.0081151
Epoch: 062, Train Loss: 0.0086732, Test Loss: 0.0077439
Epoch: 063, Train Loss: 0.0082994, Test Loss: 0.0073980
```

```
Epoch: 064, Train Loss: 0.0079496, Test Loss: 0.0070753
Epoch: 065, Train Loss: 0.0076216, Test Loss: 0.0067739
Epoch: 066, Train Loss: 0.0073139, Test Loss: 0.0064919
Epoch: 067, Train Loss: 0.0070245, Test Loss: 0.0062282
Epoch: 068, Train Loss: 0.0067519, Test Loss: 0.0059808
Epoch: 069, Train Loss: 0.0064954, Test Loss: 0.0057490
Epoch: 070, Train Loss: 0.0062535, Test Loss: 0.0055315
Epoch: 071, Train Loss: 0.0060251, Test Loss: 0.0053267
Epoch: 072, Train Loss: 0.0058093, Test Loss: 0.0051344
Epoch: 073, Train Loss: 0.0056051, Test Loss: 0.0049535
Epoch: 074, Train Loss: 0.0054119, Test Loss: 0.0047829
Epoch: 075, Train Loss: 0.0052291, Test Loss: 0.0046220
Epoch: 076, Train Loss: 0.0050560, Test Loss: 0.0044699
Epoch: 077, Train Loss: 0.0048918, Test Loss: 0.0043266
Epoch: 078, Train Loss: 0.0047361, Test Loss: 0.0041909
Epoch: 079, Train Loss: 0.0045884, Test Loss: 0.0040627
Epoch: 080, Train Loss: 0.0044482, Test Loss: 0.0039412
Epoch: 081, Train Loss: 0.0043151, Test Loss: 0.0038265
Epoch: 082, Train Loss: 0.0041884, Test Loss: 0.0037177
Epoch: 083, Train Loss: 0.0040681, Test Loss: 0.0036145
Epoch: 084, Train Loss: 0.0039535, Test Loss: 0.0035166
Epoch: 085, Train Loss: 0.0038444, Test Loss: 0.0034235
Epoch: 086, Train Loss: 0.0037405, Test Loss: 0.0033353
Epoch: 087, Train Loss: 0.0036414, Test Loss: 0.0032513
Epoch: 088, Train Loss: 0.0035469, Test Loss: 0.0031714
Epoch: 089, Train Loss: 0.0034568, Test Loss: 0.0030951
Epoch: 090, Train Loss: 0.0033708, Test Loss: 0.0030225
Epoch: 091, Train Loss: 0.0032886, Test Loss: 0.0029533
Epoch: 092, Train Loss: 0.0032101, Test Loss: 0.0028868
Epoch: 093, Train Loss: 0.0031351, Test Loss: 0.0028231
Epoch: 094, Train Loss: 0.0030633, Test Loss: 0.0027623
Epoch: 095, Train Loss: 0.0029947, Test Loss: 0.0027041
Epoch: 096, Train Loss: 0.0029288, Test Loss: 0.0026480
Epoch: 097, Train Loss: 0.0028657, Test Loss: 0.0025943
Epoch: 098, Train Loss: 0.0028051, Test Loss: 0.0025427
Epoch: 099, Train Loss: 0.0027470, Test Loss: 0.0024932
Epoch: 100, Train Loss: 0.0026913, Test Loss: 0.0024457
```



Problem 4.2 Determine appropriate mini-batch size = 64, epoch = 100

```
# ################## Part 1: Load data and create batch
##################
N_total = 600
N_train = 500
x = torch.unsqueeze(torch.linspace(0, 1, N_total), dim=1)
r = torch.randperm(N_total)
x = x[r, :]
y = 0.2 + 0.4 * torch.pow(x, 2) + 0.3 * x * torch.sin(15 * x) + 0.05 *
torch.cos(50 * x)

class CustomDataset(Dataset):
    def __init__(self, x, y):
        self.y = y
        self.x = x

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

    def __getitem__(self, idx):
        y1 = self.y[idx]
        x1 = self.x[idx]
        return (x1, y1)

# Change batch_size here to test different values
```

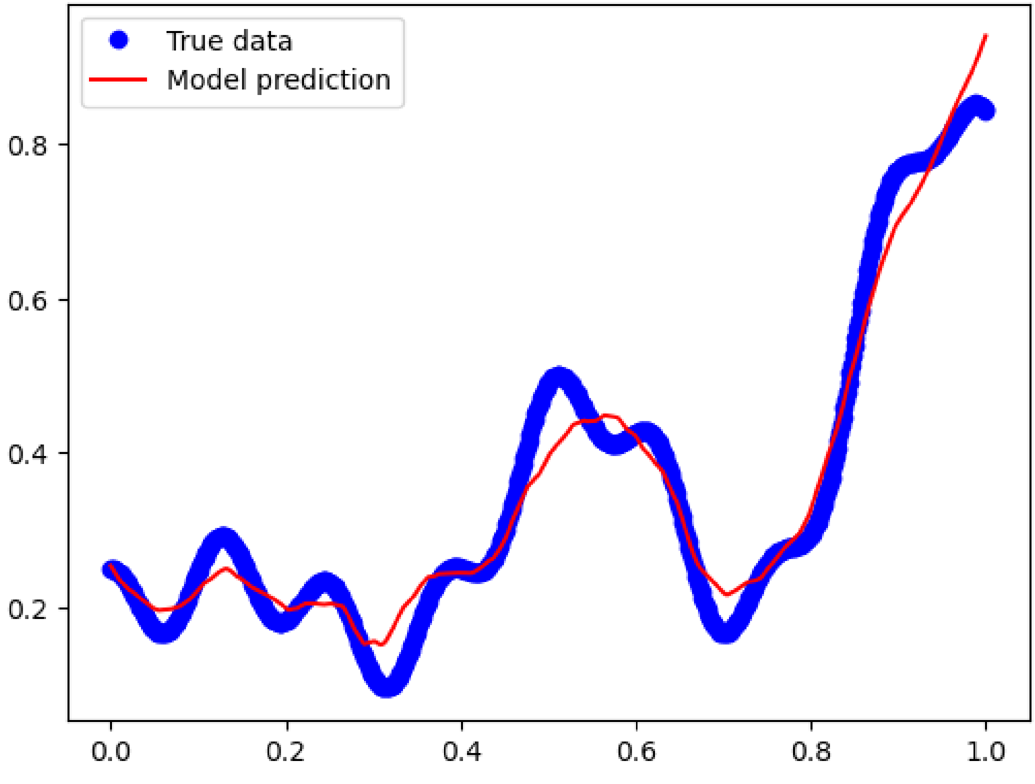
```
batch_size = 64  # Experiment with different batch sizes: 32, 64, 128
trainset = CustomDataset(x[0:N_train, :], y[0:N_train, :])
testset = CustomDataset(x[N_train:N_total, :], y[N_train:N_total, :])
train_loader = torch.utils.data.DataLoader(trainset,
batch_size=batch_size)
test_loader = torch.utils.data.DataLoader(testset,
batch_size=batch_size)

# ################## Part 2: Define Model ##################
model = nn.Sequential(
    nn.Linear(1, 1024, bias=True),
    nn.ReLU(),
    nn.Linear(1024, 1, bias=True)
)

def init_weights(m):
    if isinstance(m, nn.Linear):
        m.weight.data.uniform_(-1, 1)
        m.bias.data.uniform_(-1, 1)

model.apply(init_weights)

# ################## Part 3: Define Loss and Optimizer
##################
criterion = nn.MSELoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

# ################## Part 4: Training and Testing ##################
def train_NN():
    model.train()
    for images, labels in train_loader:
        out = model(images)
        loss = criterion(out, labels)
        loss.backward()
        optimizer.step()
        optimizer.zero_grad()
    return loss

def test_NN(loader):
    model.eval()
    loss = 0
    with torch.no_grad():
        for images, labels in loader:
            out = model(images)
            loss += criterion(out, labels).item()
    loss = loss / len(loader)
    return loss

# Experiment with different numbers of epochs
N_epoch = 100
```

```python
train_loss = np.zeros((N_epoch, 1))
test_loss = np.zeros((N_epoch, 1))

for epoch in range(N_epoch):
    train_NN()
    train_loss[epoch, 0] = test_NN(train_loader)
    test_loss[epoch, 0] = test_NN(test_loader)
    print(f'Epoch: {epoch+1:03d}, Train Loss: {train_loss[epoch,
0]:.7f}, Test Loss: {test_loss[epoch, 0]:.7f}')

# ################## Final Prediction ##################
x_test = torch.unsqueeze(torch.linspace(0, 1, 1999), dim=1)
y_test = model(x_test)

# Plot the results
plt.plot(x[0:N_total], y[0:N_total], 'bo', label='True data')
plt.plot(x_test, y_test.detach().numpy(), 'r', label='Model
prediction')
plt.legend()
plt.show()
```

```
Epoch: 001, Train Loss: 15.6746806, Test Loss: 16.2465343
Epoch: 002, Train Loss: 2.8316783, Test Loss: 2.5567751
Epoch: 003, Train Loss: 5.4225901, Test Loss: 4.9618430
Epoch: 004, Train Loss: 2.1876323, Test Loss: 1.9957126
Epoch: 005, Train Loss: 0.6743857, Test Loss: 0.7711622
Epoch: 006, Train Loss: 0.7423441, Test Loss: 0.9157936
Epoch: 007, Train Loss: 0.2941593, Test Loss: 0.3851405
Epoch: 008, Train Loss: 0.2239520, Test Loss: 0.2369578
Epoch: 009, Train Loss: 0.1893579, Test Loss: 0.1959976
Epoch: 010, Train Loss: 0.1461559, Test Loss: 0.1745133
Epoch: 011, Train Loss: 0.1372006, Test Loss: 0.1686388
Epoch: 012, Train Loss: 0.1171399, Test Loss: 0.1371895
Epoch: 013, Train Loss: 0.1056668, Test Loss: 0.1206233
Epoch: 014, Train Loss: 0.0933011, Test Loss: 0.1111685
Epoch: 015, Train Loss: 0.0835575, Test Loss: 0.1031929
Epoch: 016, Train Loss: 0.0746684, Test Loss: 0.0925782
Epoch: 017, Train Loss: 0.0670494, Test Loss: 0.0831618
Epoch: 018, Train Loss: 0.0603105, Test Loss: 0.0757660
Epoch: 019, Train Loss: 0.0544268, Test Loss: 0.0691215
Epoch: 020, Train Loss: 0.0492410, Test Loss: 0.0627212
Epoch: 021, Train Loss: 0.0446951, Test Loss: 0.0570725
Epoch: 022, Train Loss: 0.0406946, Test Loss: 0.0522399
Epoch: 023, Train Loss: 0.0371691, Test Loss: 0.0479210
Epoch: 024, Train Loss: 0.0340490, Test Loss: 0.0440055
Epoch: 025, Train Loss: 0.0312833, Test Loss: 0.0405369
Epoch: 026, Train Loss: 0.0288218, Test Loss: 0.0374416
Epoch: 027, Train Loss: 0.0266230, Test Loss: 0.0346435
Epoch: 028, Train Loss: 0.0246500, Test Loss: 0.0321161
Epoch: 029, Train Loss: 0.0228736, Test Loss: 0.0298368
```

```
Epoch: 030, Train Loss: 0.0212682, Test Loss: 0.0277690
Epoch: 031, Train Loss: 0.0198125, Test Loss: 0.0258874
Epoch: 032, Train Loss: 0.0184883, Test Loss: 0.0241738
Epoch: 033, Train Loss: 0.0172810, Test Loss: 0.0226092
Epoch: 034, Train Loss: 0.0161777, Test Loss: 0.0211761
Epoch: 035, Train Loss: 0.0151688, Test Loss: 0.0198630
Epoch: 036, Train Loss: 0.0142440, Test Loss: 0.0186585
Epoch: 037, Train Loss: 0.0133935, Test Loss: 0.0175488
Epoch: 038, Train Loss: 0.0126108, Test Loss: 0.0165297
Epoch: 039, Train Loss: 0.0118898, Test Loss: 0.0155915
Epoch: 040, Train Loss: 0.0112241, Test Loss: 0.0147275
Epoch: 041, Train Loss: 0.0106093, Test Loss: 0.0139282
Epoch: 042, Train Loss: 0.0100410, Test Loss: 0.0131867
Epoch: 043, Train Loss: 0.0095152, Test Loss: 0.0124994
Epoch: 044, Train Loss: 0.0090282, Test Loss: 0.0118601
Epoch: 045, Train Loss: 0.0085765, Test Loss: 0.0112658
Epoch: 046, Train Loss: 0.0081566, Test Loss: 0.0107125
Epoch: 047, Train Loss: 0.0077662, Test Loss: 0.0101956
Epoch: 048, Train Loss: 0.0074025, Test Loss: 0.0097150
Epoch: 049, Train Loss: 0.0070629, Test Loss: 0.0092660
Epoch: 050, Train Loss: 0.0067456, Test Loss: 0.0088456
Epoch: 051, Train Loss: 0.0064488, Test Loss: 0.0084519
Epoch: 052, Train Loss: 0.0061710, Test Loss: 0.0080831
Epoch: 053, Train Loss: 0.0059105, Test Loss: 0.0077361
Epoch: 054, Train Loss: 0.0056661, Test Loss: 0.0074098
Epoch: 055, Train Loss: 0.0054362, Test Loss: 0.0071021
Epoch: 056, Train Loss: 0.0052199, Test Loss: 0.0068117
Epoch: 057, Train Loss: 0.0050157, Test Loss: 0.0065380
Epoch: 058, Train Loss: 0.0048233, Test Loss: 0.0062797
Epoch: 059, Train Loss: 0.0046413, Test Loss: 0.0060352
Epoch: 060, Train Loss: 0.0044693, Test Loss: 0.0058043
Epoch: 061, Train Loss: 0.0043063, Test Loss: 0.0055854
Epoch: 062, Train Loss: 0.0041516, Test Loss: 0.0053775
Epoch: 063, Train Loss: 0.0040047, Test Loss: 0.0051804
Epoch: 064, Train Loss: 0.0038651, Test Loss: 0.0049924
Epoch: 065, Train Loss: 0.0037326, Test Loss: 0.0048135
Epoch: 066, Train Loss: 0.0036064, Test Loss: 0.0046423
Epoch: 067, Train Loss: 0.0034859, Test Loss: 0.0044787
Epoch: 068, Train Loss: 0.0033712, Test Loss: 0.0043229
Epoch: 069, Train Loss: 0.0032616, Test Loss: 0.0041740
Epoch: 070, Train Loss: 0.0031568, Test Loss: 0.0040319
Epoch: 071, Train Loss: 0.0030566, Test Loss: 0.0038962
Epoch: 072, Train Loss: 0.0029608, Test Loss: 0.0037660
Epoch: 073, Train Loss: 0.0028691, Test Loss: 0.0036417
Epoch: 074, Train Loss: 0.0027810, Test Loss: 0.0035225
Epoch: 075, Train Loss: 0.0026965, Test Loss: 0.0034083
Epoch: 076, Train Loss: 0.0026155, Test Loss: 0.0032989
Epoch: 077, Train Loss: 0.0025379, Test Loss: 0.0031936
Epoch: 078, Train Loss: 0.0024634, Test Loss: 0.0030926
```

```
Epoch: 079, Train Loss: 0.0023919, Test Loss: 0.0029952
Epoch: 080, Train Loss: 0.0023234, Test Loss: 0.0029022
Epoch: 081, Train Loss: 0.0022573, Test Loss: 0.0028124
Epoch: 082, Train Loss: 0.0021938, Test Loss: 0.0027265
Epoch: 083, Train Loss: 0.0021328, Test Loss: 0.0026439
Epoch: 084, Train Loss: 0.0020741, Test Loss: 0.0025645
Epoch: 085, Train Loss: 0.0020177, Test Loss: 0.0024882
Epoch: 086, Train Loss: 0.0019632, Test Loss: 0.0024149
Epoch: 087, Train Loss: 0.0019108, Test Loss: 0.0023444
Epoch: 088, Train Loss: 0.0018605, Test Loss: 0.0022767
Epoch: 089, Train Loss: 0.0018121, Test Loss: 0.0022114
Epoch: 090, Train Loss: 0.0017654, Test Loss: 0.0021486
Epoch: 091, Train Loss: 0.0017204, Test Loss: 0.0020882
Epoch: 092, Train Loss: 0.0016771, Test Loss: 0.0020299
Epoch: 093, Train Loss: 0.0016355, Test Loss: 0.0019737
Epoch: 094, Train Loss: 0.0015954, Test Loss: 0.0019196
Epoch: 095, Train Loss: 0.0015568, Test Loss: 0.0018675
Epoch: 096, Train Loss: 0.0015194, Test Loss: 0.0018172
Epoch: 097, Train Loss: 0.0014834, Test Loss: 0.0017688
Epoch: 098, Train Loss: 0.0014487, Test Loss: 0.0017224
Epoch: 099, Train Loss: 0.0014151, Test Loss: 0.0016777
Epoch: 100, Train Loss: 0.0013827, Test Loss: 0.0016345
```



Problem 4.2 Determine appropriate mini-batch size = 128, epoch = 100

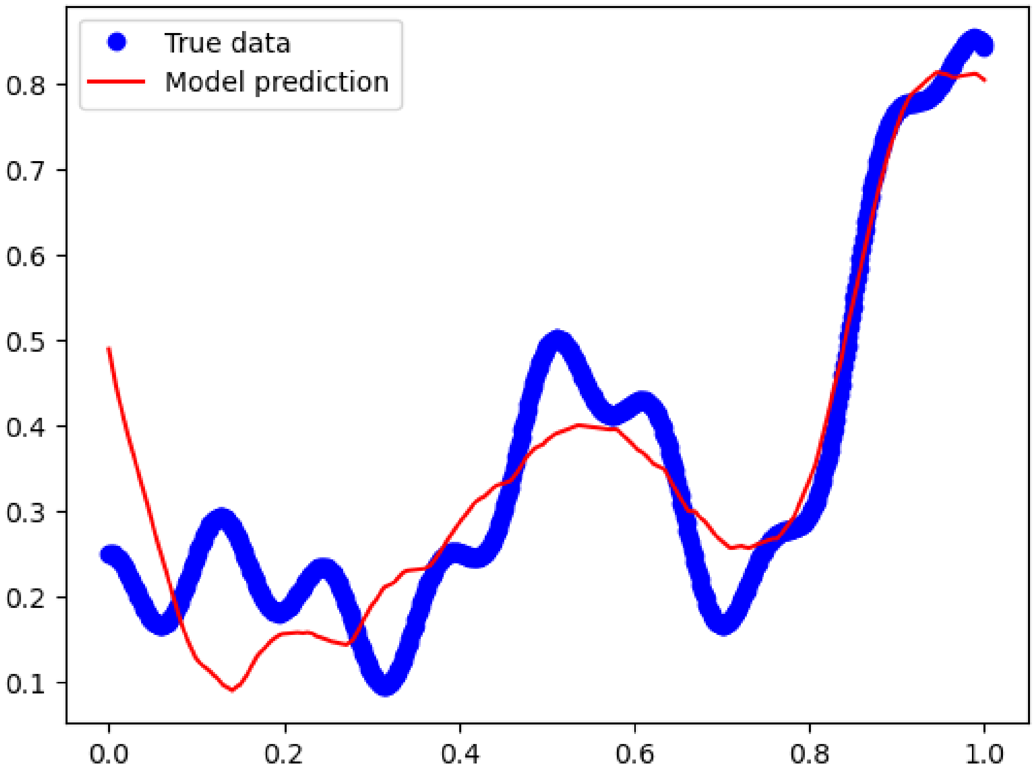
```
# ################## Part 1: Load data and create batch
##################
N_total = 600
N_train = 500
x = torch.unsqueeze(torch.linspace(0, 1, N_total), dim=1)
r = torch.randperm(N_total)
x = x[r, :]
y = 0.2 + 0.4 * torch.pow(x, 2) + 0.3 * x * torch.sin(15 * x) + 0.05 *
torch.cos(50 * x)

class CustomDataset(Dataset):
    def __init__(self, x, y):
        self.y = y
        self.x = x

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

    def __getitem__(self, idx):
        y1 = self.y[idx]
        x1 = self.x[idx]
        return (x1, y1)

# Change batch_size here to test different values
batch_size = 128  # Experiment with different batch sizes: 32, 64, 128
trainset = CustomDataset(x[0:N_train, :], y[0:N_train, :])
testset = CustomDataset(x[N_train:N_total, :], y[N_train:N_total, :])
train_loader = torch.utils.data.DataLoader(trainset,
batch_size=batch_size)
test_loader = torch.utils.data.DataLoader(testset,
batch_size=batch_size)

# ################## Part 2: Define Model ##################
model = nn.Sequential(
    nn.Linear(1, 1024, bias=True),
    nn.ReLU(),
    nn.Linear(1024, 1, bias=True)
)

def init_weights(m):
    if isinstance(m, nn.Linear):
        m.weight.data.uniform_(-1, 1)
        m.bias.data.uniform_(-1, 1)

model.apply(init_weights)

# ################## Part 3: Define Loss and Optimizer
##################
criterion = nn.MSELoss()
```

```
optimizer = optim.Adam(model.parameters(), lr=0.001)

# ################## Part 4: Training and Testing ##################
def train_NN():
    model.train()
    for images, labels in train_loader:
        out = model(images)
        loss = criterion(out, labels)
        loss.backward()
        optimizer.step()
        optimizer.zero_grad()
    return loss

def test_NN(loader):
    model.eval()
    loss = 0
    with torch.no_grad():
        for images, labels in loader:
            out = model(images)
            loss += criterion(out, labels).item()
    loss = loss / len(loader)
    return loss

# Experiment with different numbers of epochs
N_epoch = 100
train_loss = np.zeros((N_epoch, 1))
test_loss = np.zeros((N_epoch, 1))

for epoch in range(N_epoch):
    train_NN()
    train_loss[epoch, 0] = test_NN(train_loader)
    test_loss[epoch, 0] = test_NN(test_loader)
    print(f'Epoch: {epoch+1:03d}, Train Loss: {train_loss[epoch,
0]:.7f}, Test Loss: {test_loss[epoch, 0]:.7f}')

# ################## Final Prediction ##################
x_test = torch.unsqueeze(torch.linspace(0, 1, 1999), dim=1)
y_test = model(x_test)

# Plot the results
plt.plot(x[0:N_total], y[0:N_total], 'bo', label='True data')
plt.plot(x_test, y_test.detach().numpy(), 'r', label='Model
prediction')
plt.legend()
plt.show()
```

```
Epoch: 001, Train Loss: 22.4205217, Test Loss: 19.2805557
Epoch: 002, Train Loss: 5.3662879, Test Loss: 4.1010160
Epoch: 003, Train Loss: 0.8423916, Test Loss: 0.5674591
Epoch: 004, Train Loss: 2.5692085, Test Loss: 2.6380594
```

```
Epoch: 005, Train Loss: 4.0418289, Test Loss: 4.1497045
Epoch: 006, Train Loss: 3.1026638, Test Loss: 3.1714888
Epoch: 007, Train Loss: 1.2745357, Test Loss: 1.2814338
Epoch: 008, Train Loss: 0.2924522, Test Loss: 0.2166031
Epoch: 009, Train Loss: 0.3348801, Test Loss: 0.1885524
Epoch: 010, Train Loss: 0.5696861, Test Loss: 0.4075289
Epoch: 011, Train Loss: 0.4670437, Test Loss: 0.3442137
Epoch: 012, Train Loss: 0.1974116, Test Loss: 0.1294619
Epoch: 013, Train Loss: 0.0737167, Test Loss: 0.0414731
Epoch: 014, Train Loss: 0.1056979, Test Loss: 0.0857564
Epoch: 015, Train Loss: 0.1311933, Test Loss: 0.1138015
Epoch: 016, Train Loss: 0.0954354, Test Loss: 0.0802353
Epoch: 017, Train Loss: 0.0599697, Test Loss: 0.0470680
Epoch: 018, Train Loss: 0.0591872, Test Loss: 0.0475546
Epoch: 019, Train Loss: 0.0664397, Test Loss: 0.0557374
Epoch: 020, Train Loss: 0.0606549, Test Loss: 0.0509934
Epoch: 021, Train Loss: 0.0517262, Test Loss: 0.0426373
Epoch: 022, Train Loss: 0.0496269, Test Loss: 0.0404446
Epoch: 023, Train Loss: 0.0497127, Test Loss: 0.0403316
Epoch: 024, Train Loss: 0.0471728, Test Loss: 0.0378564
Epoch: 025, Train Loss: 0.0441844, Test Loss: 0.0350393
Epoch: 026, Train Loss: 0.0427473, Test Loss: 0.0336998
Epoch: 027, Train Loss: 0.0416015, Test Loss: 0.0326296
Epoch: 028, Train Loss: 0.0398994, Test Loss: 0.0310387
Epoch: 029, Train Loss: 0.0382882, Test Loss: 0.0295686
Epoch: 030, Train Loss: 0.0370320, Test Loss: 0.0284997
Epoch: 031, Train Loss: 0.0357766, Test Loss: 0.0275182
Epoch: 032, Train Loss: 0.0344608, Test Loss: 0.0265612
Epoch: 033, Train Loss: 0.0332472, Test Loss: 0.0257453
Epoch: 034, Train Loss: 0.0321139, Test Loss: 0.0249952
Epoch: 035, Train Loss: 0.0309853, Test Loss: 0.0242101
Epoch: 036, Train Loss: 0.0298918, Test Loss: 0.0234216
Epoch: 037, Train Loss: 0.0288579, Test Loss: 0.0226794
Epoch: 038, Train Loss: 0.0278647, Test Loss: 0.0219856
Epoch: 039, Train Loss: 0.0269045, Test Loss: 0.0213372
Epoch: 040, Train Loss: 0.0259856, Test Loss: 0.0207238
Epoch: 041, Train Loss: 0.0251032, Test Loss: 0.0201202
Epoch: 042, Train Loss: 0.0242530, Test Loss: 0.0195181
Epoch: 043, Train Loss: 0.0234370, Test Loss: 0.0189339
Epoch: 044, Train Loss: 0.0226565, Test Loss: 0.0183753
Epoch: 045, Train Loss: 0.0219085, Test Loss: 0.0178479
Epoch: 046, Train Loss: 0.0211923, Test Loss: 0.0173505
Epoch: 047, Train Loss: 0.0205068, Test Loss: 0.0168726
Epoch: 048, Train Loss: 0.0198505, Test Loss: 0.0164079
Epoch: 049, Train Loss: 0.0192220, Test Loss: 0.0159580
Epoch: 050, Train Loss: 0.0186207, Test Loss: 0.0155293
Epoch: 051, Train Loss: 0.0180448, Test Loss: 0.0151263
Epoch: 052, Train Loss: 0.0174940, Test Loss: 0.0147447
Epoch: 053, Train Loss: 0.0169666, Test Loss: 0.0143809
```

```
Epoch: 054, Train Loss: 0.0164621, Test Loss: 0.0140302
Epoch: 055, Train Loss: 0.0159792, Test Loss: 0.0136921
Epoch: 056, Train Loss: 0.0155167, Test Loss: 0.0133680
Epoch: 057, Train Loss: 0.0150744, Test Loss: 0.0130588
Epoch: 058, Train Loss: 0.0146510, Test Loss: 0.0127638
Epoch: 059, Train Loss: 0.0142451, Test Loss: 0.0124811
Epoch: 060, Train Loss: 0.0138567, Test Loss: 0.0122086
Epoch: 061, Train Loss: 0.0134850, Test Loss: 0.0119459
Epoch: 062, Train Loss: 0.0131284, Test Loss: 0.0116936
Epoch: 063, Train Loss: 0.0127868, Test Loss: 0.0114525
Epoch: 064, Train Loss: 0.0124591, Test Loss: 0.0112216
Epoch: 065, Train Loss: 0.0121444, Test Loss: 0.0110009
Epoch: 066, Train Loss: 0.0118426, Test Loss: 0.0107887
Epoch: 067, Train Loss: 0.0115530, Test Loss: 0.0105838
Epoch: 068, Train Loss: 0.0112750, Test Loss: 0.0103863
Epoch: 069, Train Loss: 0.0110078, Test Loss: 0.0101962
Epoch: 070, Train Loss: 0.0107508, Test Loss: 0.0100143
Epoch: 071, Train Loss: 0.0105033, Test Loss: 0.0098393
Epoch: 072, Train Loss: 0.0102650, Test Loss: 0.0096701
Epoch: 073, Train Loss: 0.0100356, Test Loss: 0.0095059
Epoch: 074, Train Loss: 0.0098149, Test Loss: 0.0093472
Epoch: 075, Train Loss: 0.0096021, Test Loss: 0.0091933
Epoch: 076, Train Loss: 0.0093970, Test Loss: 0.0090446
Epoch: 077, Train Loss: 0.0091991, Test Loss: 0.0089003
Epoch: 078, Train Loss: 0.0090083, Test Loss: 0.0087601
Epoch: 079, Train Loss: 0.0088242, Test Loss: 0.0086234
Epoch: 080, Train Loss: 0.0086462, Test Loss: 0.0084903
Epoch: 081, Train Loss: 0.0084740, Test Loss: 0.0083608
Epoch: 082, Train Loss: 0.0083074, Test Loss: 0.0082342
Epoch: 083, Train Loss: 0.0081463, Test Loss: 0.0081094
Epoch: 084, Train Loss: 0.0079903, Test Loss: 0.0079873
Epoch: 085, Train Loss: 0.0078389, Test Loss: 0.0078680
Epoch: 086, Train Loss: 0.0076922, Test Loss: 0.0077511
Epoch: 087, Train Loss: 0.0075497, Test Loss: 0.0076365
Epoch: 088, Train Loss: 0.0074113, Test Loss: 0.0075242
Epoch: 089, Train Loss: 0.0072772, Test Loss: 0.0074144
Epoch: 090, Train Loss: 0.0071467, Test Loss: 0.0073074
Epoch: 091, Train Loss: 0.0070196, Test Loss: 0.0072017
Epoch: 092, Train Loss: 0.0068958, Test Loss: 0.0070973
Epoch: 093, Train Loss: 0.0067752, Test Loss: 0.0069955
Epoch: 094, Train Loss: 0.0066577, Test Loss: 0.0068957
Epoch: 095, Train Loss: 0.0065431, Test Loss: 0.0067977
Epoch: 096, Train Loss: 0.0064315, Test Loss: 0.0067013
Epoch: 097, Train Loss: 0.0063226, Test Loss: 0.0066064
Epoch: 098, Train Loss: 0.0062163, Test Loss: 0.0065133
Epoch: 099, Train Loss: 0.0061123, Test Loss: 0.0064226
Epoch: 100, Train Loss: 0.0060108, Test Loss: 0.0063334
```



Determine appropriate mini-batch size = 32