CS 7180 Algorithmic and Statistical Aspects of Deep Learning

Mounica Subramani

Problem Set 1

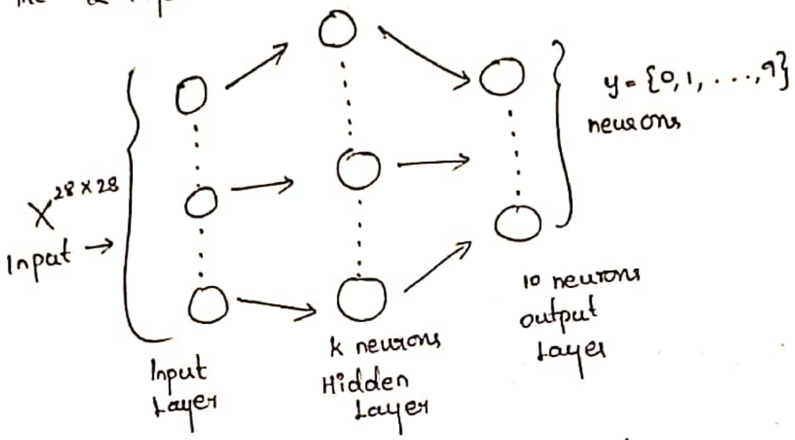
1) a) W_1 and W_2 denote the weight matrices of each layer.

Sample $\rightarrow$ $x \in \mathbb{R}^{28\times 28}$

label $\rightarrow$ $y \in \{0,1,2,\ldots,9\}$

$y_{pred} = ?$

Let the 2 layer neural network architecture be like,



Equation of y^{Pred} as a function of w_1 and w_2, given input x is written as,

output from hidden layer $\Rightarrow$ $\mathrm{ReLU}(w_1 \cdot x)$ $\hookrightarrow$ input of output layer along with weights and activation.

$[W_{input-hidden}] \cdot I/P$

$$\boxed{y^{Pred} = \mathrm{ReLU}\left[w_2(\mathrm{ReLU}(w_1 \cdot x + b_1)) + b_2\right]}$$

x - input matrix of size 784×1

b_1 and b_2 are bias vector.

ReLU - activation function

w_1 - weight matrix of size $k \times 784$

w_2 - weight matrix of size $10 \times K$

Output of hidden layer $\Rightarrow$ $\mathrm{ReLU}(w_1 \cdot x)$

$\Rightarrow \mathrm{ReLU}([k \times 784] \cdot [784 \times 1])$

$X_2 = \mathrm{ReLU}[k \times 1]$

hidden layer to output layer $= \mathrm{ReLU}(w_2 \cdot X_2)$

$\Rightarrow \mathrm{ReLU}([10 \times k] \cdot [k \times 1])$.

$$\boxed{\text{output} = \mathrm{ReLU}[10 \times 1]}$$

1b) Given :

$(x_1, y_1) \dots (x_m, y_m)$ where $m = 50,000$ – training data

$x_i \in \mathbb{R}^{28 \times 28}$

$y_i \in \{0, 1, \dots 9\}$, $i = 1 \dots m$

Loss of predicted label compared to correct label = ?

Here, optimization of the network refers to, reducing the loss of predicted label and bringing close to correct label.

Cross-entropy is widely used loss function for optimizing classification models. So lets minimize that.

Cross-entropy equation,

$$L(w) = -\left[\sum_{i=1}^{m} \sum_{k=1}^{10} I\{y_i = k\} \log \frac{\exp(w_k^T x_i)}{\sum_{j=1}^{10} \exp(w_j^T x_i)}\right]$$

where $I\{y_i = k\}$ – indicator function

W – w_1 and w_2 weight matrices together

k – no. of classes.

$L(w)$ – loss

Prob 2 a)

Given :

w_1, w_2 – weight matrices of the network

ReLU activation and

Quadratic loss function is used.

Training data $= (x_1, y_1) \dots (x_m, y_m)$

$x_i \in \mathbb{R}^{\cancel{28 \times 28}}$

$y_i \in \mathbb{R}$

Training loss,

$$L(w) = \frac{\sum_{i=1}^{m} (y_i - y_{(i)}^{Pred})^2}{m}$$

$y_{(i)}^{Pred}$ can be written as $[\mathrm{ReLU}(w_2(\mathrm{ReLU}(w_1 \cdot x) + b_1)) + b_2)]$

[refered from prob 1 a)]

2b) Backpropagation algorithm.

Back-propagation is the practice of fine-tuning the weights of neural net based on the error rate obtained in the previous cepoch) loss.

Neural networks uses back propagation as a learning algorithm. It is used to calculate derivates quickly. The weights are updated backwards, from output towards input.

Proper tuning of weights results in lower error rates, making the model more reliable by generalizing it.

This algorithm looks for minimum value of the error function in the weight space using a technique called delta rule (or) gradient descent.

[Gradient descent is an iterative optimization algorithm for finding the minimum of a function (error function)]

The error function derivation for w_1 & w_2,

$$\frac{\partial L(w)}{\partial w_2} = \frac{\partial L(w)}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial w_2}.$$

$$\frac{\partial L(w)}{\partial w_2} = \frac{\partial L(w)}{\partial O} \cdot \frac{\partial O}{\partial z} \cdot \frac{\partial z}{\partial w_2} \quad [O_1 \text{ and } O_2].$$

Forward Pass:

$$\begin{cases} O_2 = \mathrm{ReLU}(w_2(\mathrm{ReLU}(w_1 \cdot x + b_1)) + b_2) \\ O_1 = \mathrm{ReLU}(w_1 x + b_1). \\ z = w_1 x + b_1 \end{cases}$$

Backward Pass: computing gradients,

$$\frac{\partial L(w)}{\partial w_i} = \frac{\partial L(w)}{\partial O} \cdot \frac{\partial O}{\partial z} \cdot \frac{\partial z}{\partial w'}.$$

$$\therefore \frac{\partial L(w)}{\partial w_2} \Rightarrow \frac{\partial L(w)}{\partial O_2} \cdot \frac{\partial O_2}{\partial w_2}$$

$$\frac{\partial L(w)}{\partial w_1} = \frac{\partial L(w)}{\partial O_2} \cdot \frac{\partial O_2}{\partial O_1} \cdot \frac{\partial O_1}{\partial z} \cdot \frac{\partial z}{\partial w_1}$$

PS1-MNIST-Handout

October 3, 2020

1 CS7180 Problem Set 1: Implement a two-layer neural network to recognize hand-written digits (40 points)

Welcome to CS7180!

Before you start, make sure to read the problem description in the handout pdf.

[1]:
```
# Uncomment the below line and run to install required packages if you have not␣
↪done so

# !pip install torch torchvision matplotlib tqdm
```

[2]:
```
# pip install torch
```

[17]:
```
# Setup
import torch
import matplotlib.pyplot as plt
from torchvision import datasets, transforms
from tqdm import trange
from torch.autograd import Variable

%matplotlib inline
DEVICE = 'cuda' if torch.cuda.is_available() else 'cpu'

# Set random seed for reproducibility
seed = 1234
# cuDNN uses nondeterministic algorithms, set some options for reproducibility
torch.backends.cudnn.deterministic = True
torch.backends.cudnn.benchmark = False
torch.manual_seed(seed)
```

[17]:
```
<torch._C.Generator at 0x2ddc1192030>
```

1.1 Get MNIST Data

The torchvision package provides a wrapper to download MNIST data. The cell below downloads the training and test datasets and creates dataloaders for each.

[18]:
```
# Initial transform (convert to PyTorch Tensor only)
transform = transforms.Compose([
    transforms.ToTensor(),
])

train_data = datasets.MNIST('data', train=True, download=True,␣
↪transform=transform)
test_data = datasets.MNIST('data', train=False, download=True,␣
↪transform=transform)

# Calculate training data mean and standard deviation to apply normalization to␣
↪data
# train_data.data are of type uint8 (range 0,255) so divide by 255.
# train_mean = train_data.data.double().mean() / 255.
# train_std = train_data.data.double().std() / 255.
# print(f'Train Data: Mean={train_mean}, Std={train_std}')

# Perform normalization of train and test data using calculated training mean␣
↪and standard deviation
# This will convert data to be in the range [-1, 1]
#transform = transforms.Compose([
#    transforms.ToTensor(),
#    transforms.Normalize((train_mean, ), (train_std, ))
#])
train_data.transform = transform
test_data.transform = transform

batch_size = 64
torch.manual_seed(seed)
train_loader = torch.utils.data.DataLoader(train_data, batch_size=batch_size,␣
↪shuffle=True, num_workers=True)
test_loader = torch.utils.data.DataLoader(test_data, batch_size=batch_size,␣
↪shuffle=False, num_workers=True)
```

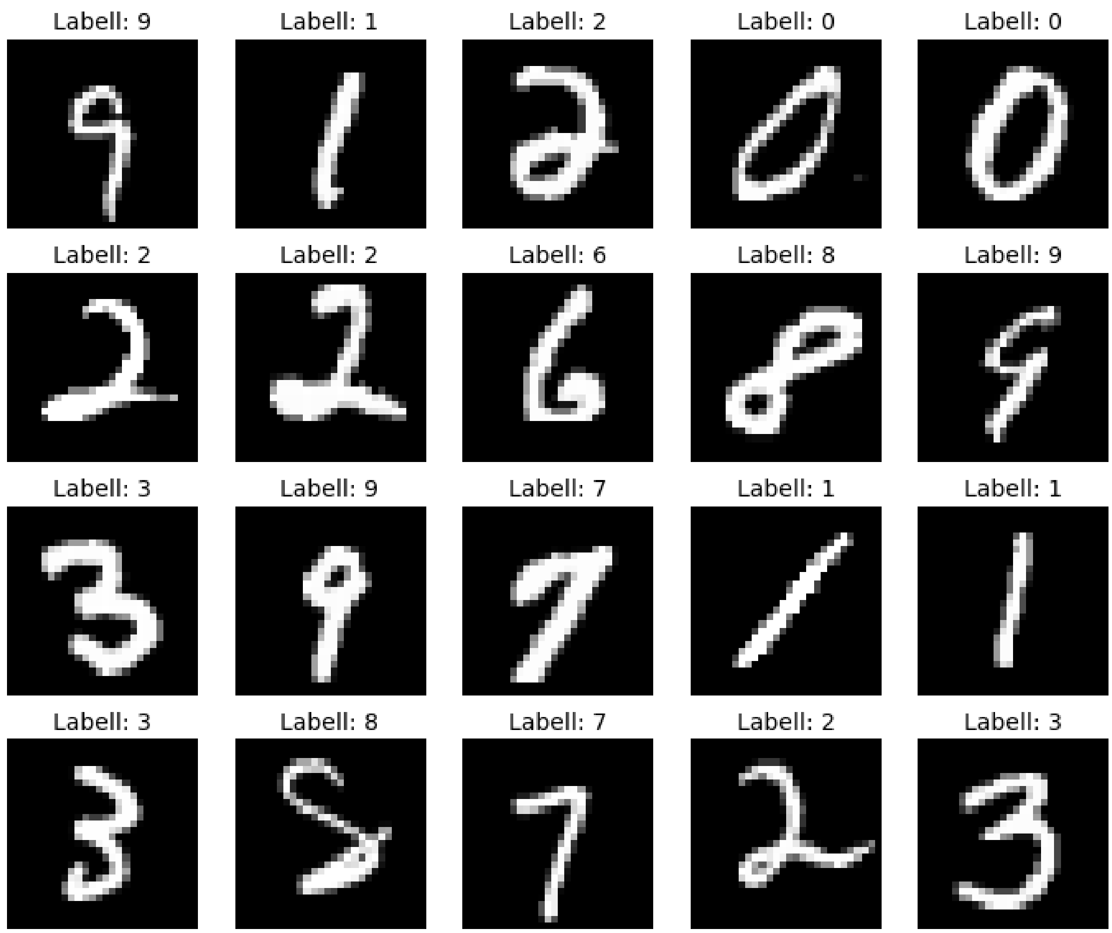
1.2 Part 0: Inspect dataset (0 points)

[19]:
```
test_data
```

[19]:
```
Dataset MNIST
    Number of datapoints: 10000
    Root location: data
    Split: Test
    StandardTransform
Transform: Compose(
               ToTensor()
           )
```

[20]:
```python
# Randomly sample 20 images of the training dataset
# To visualize the i-th sample, use the following code
# > plt.subplot(4, 5, i+1)
# > plt.imshow(images[i].squeeze(), cmap='gray', interpolation='none')
# > plt.title(f'Label: {labels[i]}', fontsize=14)
# > plt.axis('off')

images, labels = iter(train_loader).next()

# Print information and statistics of the first batch of images
print("Images shape: ", images.shape)
print("Labels shape: ", labels.shape)
print(f'Mean={images.mean()}, Std={images.std()}')

fig = plt.figure(figsize=(12, 10))
# ------------------

for i in range(20):
    plt.subplot(4, 5, i+1)
    plt.imshow(images[i].squeeze(), cmap='gray', interpolation='none')
    plt.title(f'Labell: {labels[i]}', fontsize=14)
    plt.axis('off')

# ------------------
```

```
Images shape:  torch.Size([64, 1, 28, 28])
Labels shape:  torch.Size([64])
Mean=0.12825286388397217, Std=0.3058689832687378
```



1.3 Part 1: Implement a two-layer neural network (10 points)

Write a class that constructs a two-layer neural network as specified in the handout. The class consists of two methods, an initialization that sets up the architecture of the model, and a forward pass function given an input feature.

[21]:
```
input_size = 1 * 28 * 28   # input spatial dimension of images
hidden_size = 128          # width of hidden layer
output_size = 10           # number of output neurons

class MNISTClassifierMLP(torch.nn.Module):

    def __init__(self):

        super().__init__()
        self.flatten = torch.nn.Flatten(start_dim=1)

        # ------------------
```

```python
        # Linear layer obtaining input to the hidden layer from the input layer
        self.fc1 = torch.nn.Linear(input_size,hidden_size)

        # Applying activation to the output of hidden layer before feeding into␣
↪final layer
        self.act = torch.nn.ReLU()

        # Linear layer -> narrowing to 10 outputs from hidden layer
        self.fc2 = torch.nn.Linear(hidden_size,output_size)

        # Prevent overfitting
        # self.dropout = torch.nn.Dropout(0.2)
        self.log_softmax = torch.nn.LogSoftmax(dim=1)

        # ------------------

    def forward(self, x):
        # Input image is of shape [batch_size, 1, 28, 28]
        # Need to flatten to [batch_size, 784] before feeding to fc1
        x = self.flatten(x)

        # ------------------

        x = self.act(self.fc1(x))
        x = self.act(x)
        x = self.fc2(x)
        x = self.log_softmax(x)
        y_output = x

        return y_output

        # ------------------

model = MNISTClassifierMLP().to(DEVICE)

# sanity check
print(model)
```

```
MNISTClassifierMLP(
  (flatten): Flatten()
  (fc1): Linear(in_features=784, out_features=128, bias=True)
  (act): ReLU()
  (fc2): Linear(in_features=128, out_features=10, bias=True)
  (log_softmax): LogSoftmax()
)
```

1.4 Part 2: Implement an optimizer to train the neural net model (10 points)

Write a method called `train_one_epoch` that runs one step using the optimizer.

[22]:
```
def train_one_epoch(train_loader, model, device, optimizer, log_interval,␣
↪epoch):
    model.train()
    losses = []
    counter = []

    for i, (img, label) in enumerate(train_loader):
        img, label = img.to(device), label.to(device)

#         img, label = Variable(img), Variable(label)
        # clear the gradients of all optimized variables
        optimizer.zero_grad()
        # forward pass: compute predicted outputs by passing inputs to the model
        output = model(img)
        # calculate the loss
        loss = torch.nn.functional.nll_loss(output,label)
        # backward pass: compute gradient of the loss with respect to model␣
↪parameters
        loss.backward()
        # perform a single optimization step (parameter update)
        optimizer.step()

        # Record training loss every log_interval and keep counter of total␣
↪training images seen
        if (i+1) % log_interval == 0:
            losses.append(loss.item())
            counter.append(
                (i * batch_size) + img.size(0) + epoch * len(train_loader.
↪dataset))

    return losses, counter
```

1.5 Part 3: Run the optimization procedure and test the trained model (10 points)

Write a method called `test_one_epoch` that evalutes the trained model on the test dataset. Return the average test loss and the number of samples that the model predicts correctly.

[23]:
```
def test_one_epoch(test_loader, model, device):
    model.eval()
    test_loss = 0
    num_correct = 0

#     model.eval()
```

```python
    with torch.no_grad():
        for i, (img, label) in enumerate(test_loader):
            img, label = img.to(device), label.to(device)

#              img, label = Variable(img), Variable(label)
            output = model(img)
            pred = output.argmax(dim=1)
            test_loss += torch.nn.functional.
 ↪nll_loss(output,label,reduction='sum').item()
            num_correct += pred.eq(label.view_as(pred)).sum().item()

    test_loss /= len(test_loader.dataset)
#     test_loss /= len(test_loader)
    return test_loss, num_correct
```

Train the model using the cell below. Hyperparameters are given.

[24]:
```python
# Hyperparameters
lr = 0.01
max_epochs=10
gamma = 0.95

# Recording data
log_interval = 100

# Instantiate optimizer (model was created in previous cell)
optimizer = torch.optim.SGD(model.parameters(), lr=lr)

train_losses = []
train_counter = []
test_losses = []
test_correct = []
for epoch in trange(max_epochs, leave=True, desc='Epochs'):
    train_loss, counter = train_one_epoch(train_loader, model, DEVICE,␣
 ↪optimizer, log_interval, epoch)
    test_loss, num_correct = test_one_epoch(test_loader, model, DEVICE)

    # Record results
    train_losses.extend(train_loss)
    train_counter.extend(counter)
    test_losses.append(test_loss)
    test_correct.append(num_correct)

print(f"Test accuracy: {test_correct[-1]/len(test_loader.dataset)}")
```

```
Epochs:
```

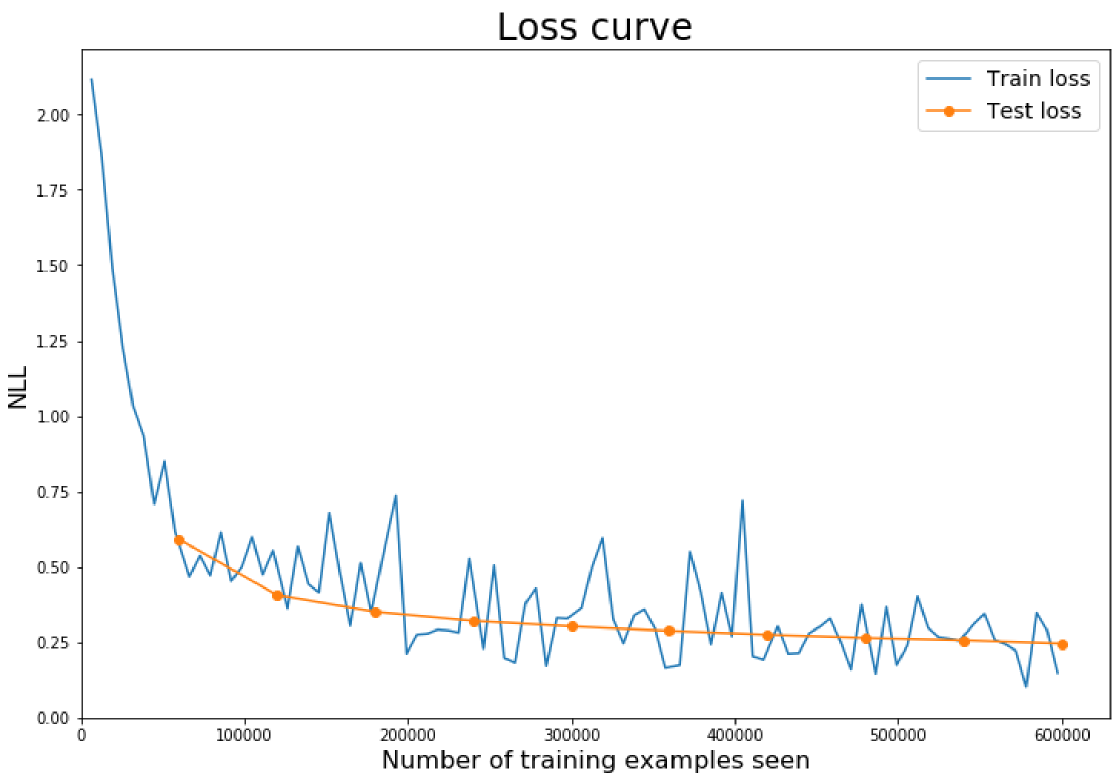
```
100%|                                        |
10/10 [01:44<00:00, 10.44s/it]

Test accuracy: 0.9305
```

1.6 Part 4: Ablation studies (10 points)

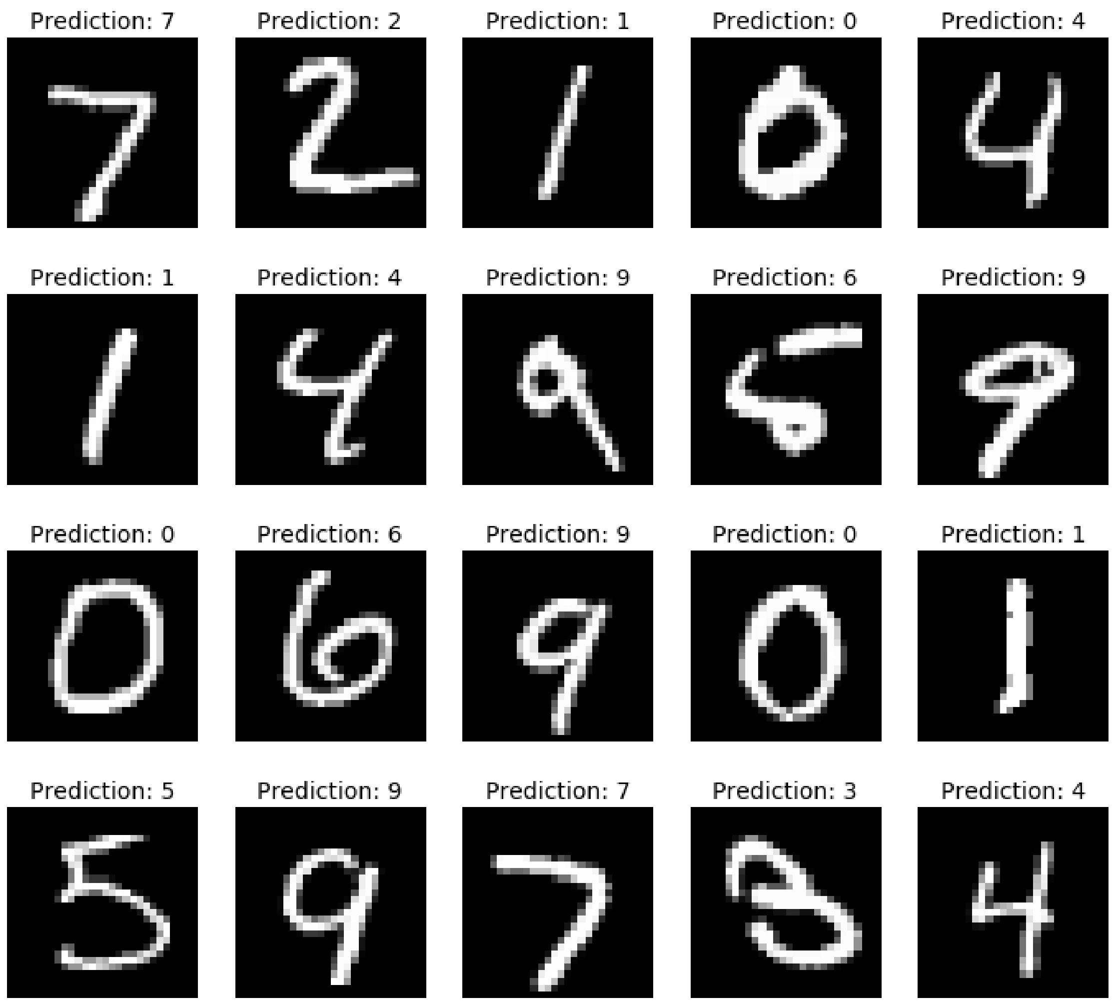
1. Plot the loss curve as the number of epochs increases
2. Show the predictions of the first 20 images of the test set (4 points)
3. Show the first 20 images that the model predicted incorrectly. Discuss about some of the common scenarios that the model predicted incorrectly (4 points)
4. Go back to Part 0, where we created the tranform component to apply on the training and test datasets. Re-run the code after normalizing the training and test dataset to have mean zero and unit variance. Report what you find (2 points)

[25]:
```
# 1. Draw training loss curve
fig = plt.figure(figsize=(12,8))
plt.plot(train_counter, train_losses, label='Train loss')
plt.plot([i * len(train_loader.dataset) for i in range(1, max_epochs + 1)],
         test_losses, label='Test loss', marker='o')
plt.xlim(left=0)
plt.ylim(bottom=0)
plt.title('Loss curve', fontsize=24)
plt.xlabel('Number of training examples seen', fontsize=16)
plt.ylabel('NLL', fontsize=16)
plt.legend(loc='upper right', fontsize=14)
```

[25]:
```
<matplotlib.legend.Legend at 0x2ddc419e608>
```

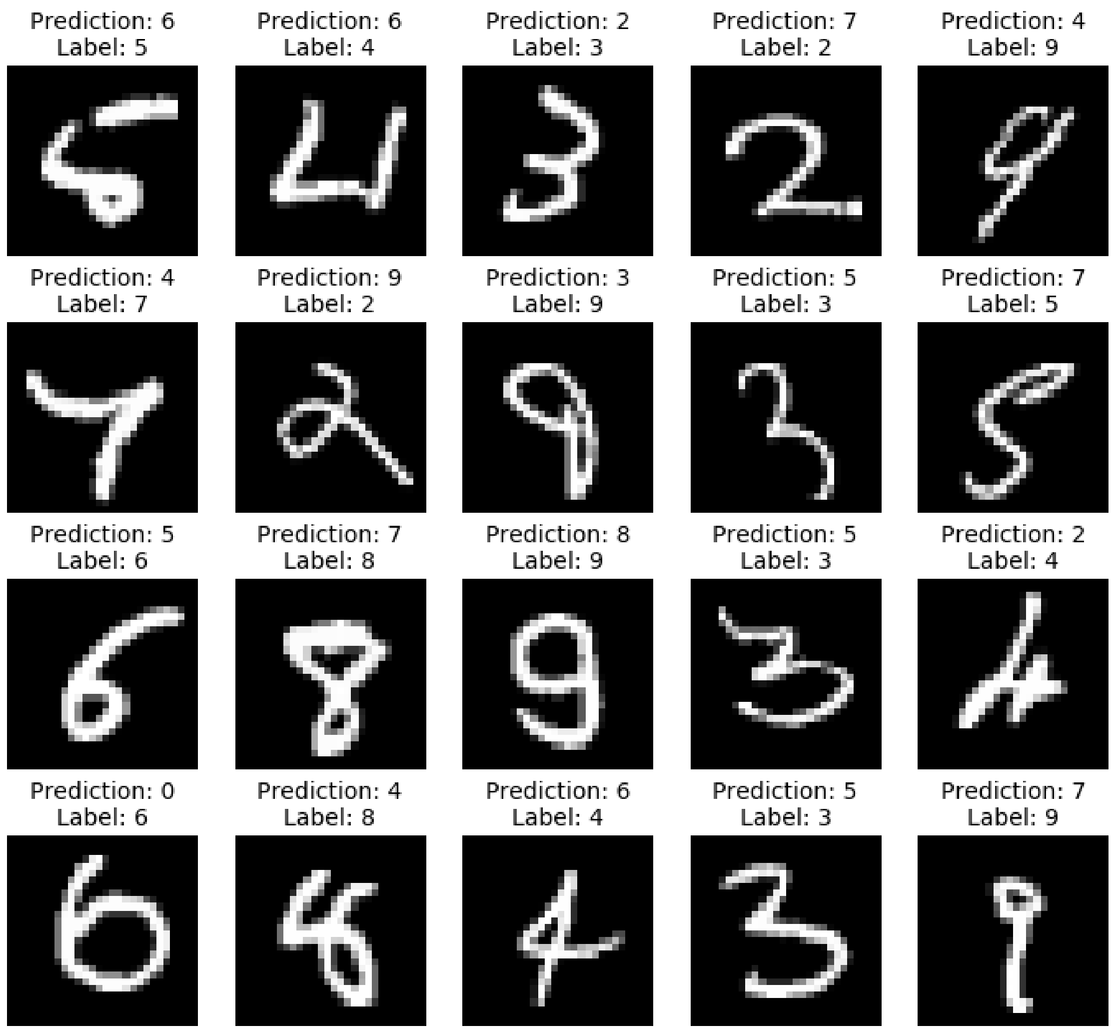


[26]:
```python
# 2. Show the predictions of the first 20 images of the test dataset
images, labels = iter(test_loader).next()
images, labels = images.to(DEVICE), labels.to(DEVICE)

output = model(images)
pred = output.argmax(dim=1)

fig = plt.figure(figsize=(12, 11))

# ------------------
# Write your implementation here. Use the code provided in Part 0 to visualize␣
↪the images.

for i in range(20):
    plt.subplot(4,5,i+1)
    plt.imshow(images[i].squeeze().cpu().numpy(), cmap='gray',␣
↪interpolation='none')
    plt.title(f'Prediction: {pred[i]}',fontsize=14)
    plt.axis('off')

# ------------------
```



[27]:
```
# 3. Get 20 incorrect predictions in test dataset

# Collect the images, predictions, labels for the first 20 incorrect predictions
# Initialize empty tensors and then keep appending to the tensor.
# Make sure that the first dimension of the tensors is the total number of␣
 ↪incorrect
# predictions seen so far
# Ex) incorrect_imgs should be of shape i x C x H x W, where i is the total␣
 ↪number of
# incorrect images so far.

incorrect_imgs = torch.Tensor().to(DEVICE)

# creating lists as we cannot append predictions and labels to tensor objects
incorrect_preds = []
incorrect_labels = []
```

```
with torch.no_grad():
    # Test set iterator
    it = iter(test_loader)
    # Loop over the test set batches until incorrect_imgs.size(0) >= 20
    while incorrect_imgs.size(0) < 20:
        images, labels = it.next()
        images, labels = images.to(DEVICE), labels.to(DEVICE)

        # ------------------
        # Write your implementation here.

        output = model(images)
        pred = output.argmax(dim=1)

        # Compare prediction and true labels and append the incorrect␣
↪predictions
        # using `torch.cat`.
        for index, i in enumerate(output):
            if pred[index] !=labels[index]:
                incorrect_imgs = torch.cat([incorrect_imgs,␣
↪images[index]],dim=0)
                incorrect_preds.append(pred[index])
                incorrect_labels.append(labels[index])

        # ------------------

# Show the first 20 wrong predictions in test set
# incorrect_labels = str(incorrect_labels)
# incorrect_preds = str(incorrect_preds)
fig = plt.figure(figsize=(12, 11))
for i in range(20):
    plt.subplot(4, 5, i+1)
    plt.imshow(incorrect_imgs[i].squeeze().cpu().numpy(), cmap='gray',␣
↪interpolation='none')
    plt.title(f'Prediction: {incorrect_preds[i].item()}\nLabel:␣
↪{incorrect_labels[i].item()}', fontsize=14)
    plt.axis('off')
```



Discuss about some of the common scenarios that the model predicted incorrectly (4 points)

The numbers that are predicted incorrectly might be because of training data being too similar to each other, so when deployed on testing data gives poor accuracy.

Also, few digits are hard to recognize for human eyes.

MNSIT data is already somewhat preprocessed. So model might expect to preprocess our test images as well. (like normalizing the training and testing data in the same way). Can also follow PCA or normalizing by Z-score.

Go back to Part 0, where we created the tranform component to apply on the training and test datasets. Re-run the code after normalizing the training and test dataset to have mean zero and unit variance. Report what you find (2 points)

[28]:
```
# Initial transform (convert to PyTorch Tensor only)
transform = transforms.Compose([
    transforms.ToTensor(),
```

```
])

train_data = datasets.MNIST('data', train=True, download=True,␣
 ↪transform=transform)
test_data = datasets.MNIST('data', train=False, download=True,␣
 ↪transform=transform)

# Calculate training data mean and standard deviation to apply normalization to␣
 ↪data
# train_data.data are of type uint8 (range 0,255) so divide by 255.
train_mean = train_data.data.double().mean() / 255.
train_std = train_data.data.double().std() / 255.
print(f'Train Data: Mean={train_mean}, Std={train_std}')

# Perform normalization of train and test data using calculated training mean␣
 ↪and standard deviation
# This will convert data to be in the range [-1, 1]
transform = transforms.Compose([transforms.ToTensor(),transforms.
 ↪Normalize((train_mean, ), (train_std, ))])
train_data.transform = transform
test_data.transform = transform

batch_size = 64
torch.manual_seed(seed)
train_loader = torch.utils.data.DataLoader(train_data, batch_size=batch_size,␣
 ↪shuffle=True, num_workers=True)
test_loader = torch.utils.data.DataLoader(test_data, batch_size=batch_size,␣
 ↪shuffle=False, num_workers=True)
```

```
Train Data: Mean=0.1306604762738429, Std=0.30810780717887876
```

[29]:

```
# Hyperparameters
lr = 0.01
max_epochs=10
gamma = 0.95

# Recording data
log_interval = 100

# Instantiate optimizer (model was created in previous cell)
optimizer = torch.optim.SGD(model.parameters(), lr=lr)

train_losses = []
train_counter = []
test_losses = []
test_correct = []
for epoch in trange(max_epochs, leave=True, desc='Epochs'):
```

```
    train_loss, counter = train_one_epoch(train_loader, model, DEVICE,␣
↪optimizer, log_interval, epoch)
    test_loss, num_correct = test_one_epoch(test_loader, model, DEVICE)

    # Record results
    train_losses.extend(train_loss)
    train_counter.extend(counter)
    test_losses.append(test_loss)
    test_correct.append(num_correct)

print(f"Test accuracy: {test_correct[-1]/len(test_loader.dataset)}")
```

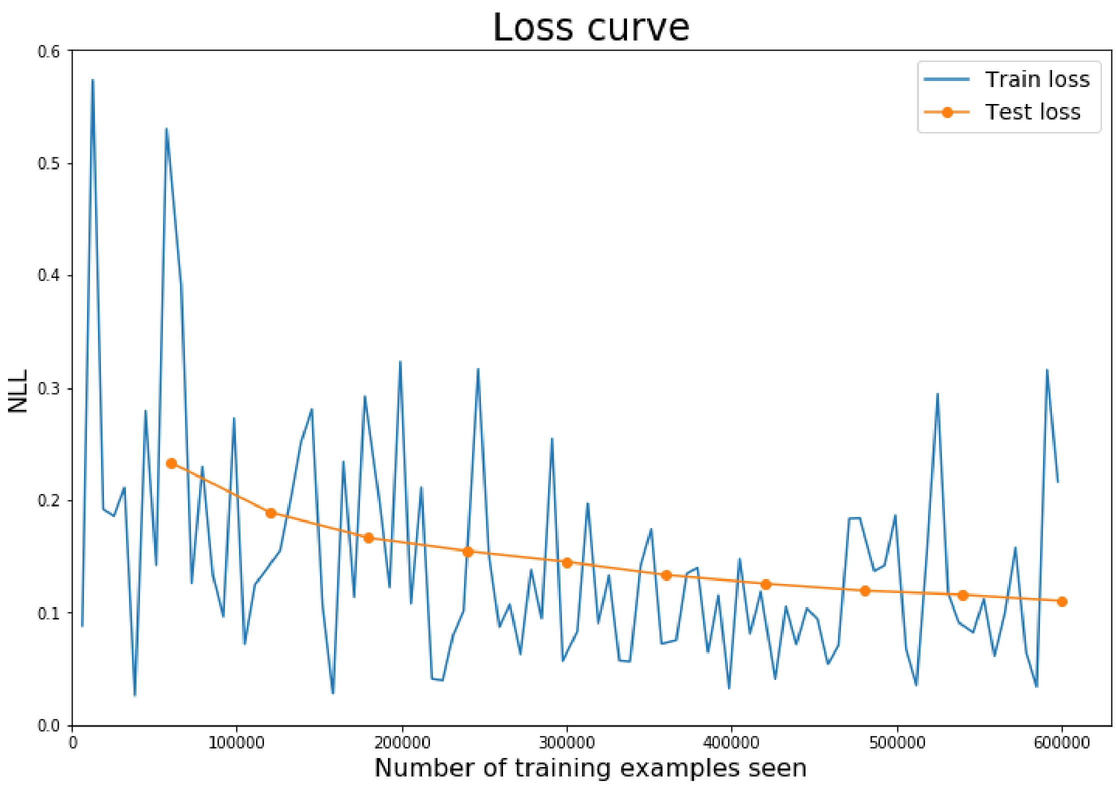
```
Epochs:
100%|                                        |
10/10 [02:38<00:00, 15.80s/it]
```

```
Test accuracy: 0.968
```

The accuracy of the model has however improved from 0.93 to 0.96.

[30]:
```
# 1. Draw training loss curve
fig = plt.figure(figsize=(12,8))
plt.plot(train_counter, train_losses, label='Train loss')
plt.plot([i * len(train_loader.dataset) for i in range(1, max_epochs + 1)],
         test_losses, label='Test loss', marker='o')
plt.xlim(left=0)
plt.ylim(bottom=0)
plt.title('Loss curve', fontsize=24)
plt.xlabel('Number of training examples seen', fontsize=16)
plt.ylabel('NLL', fontsize=16)
plt.legend(loc='upper right', fontsize=14)
```

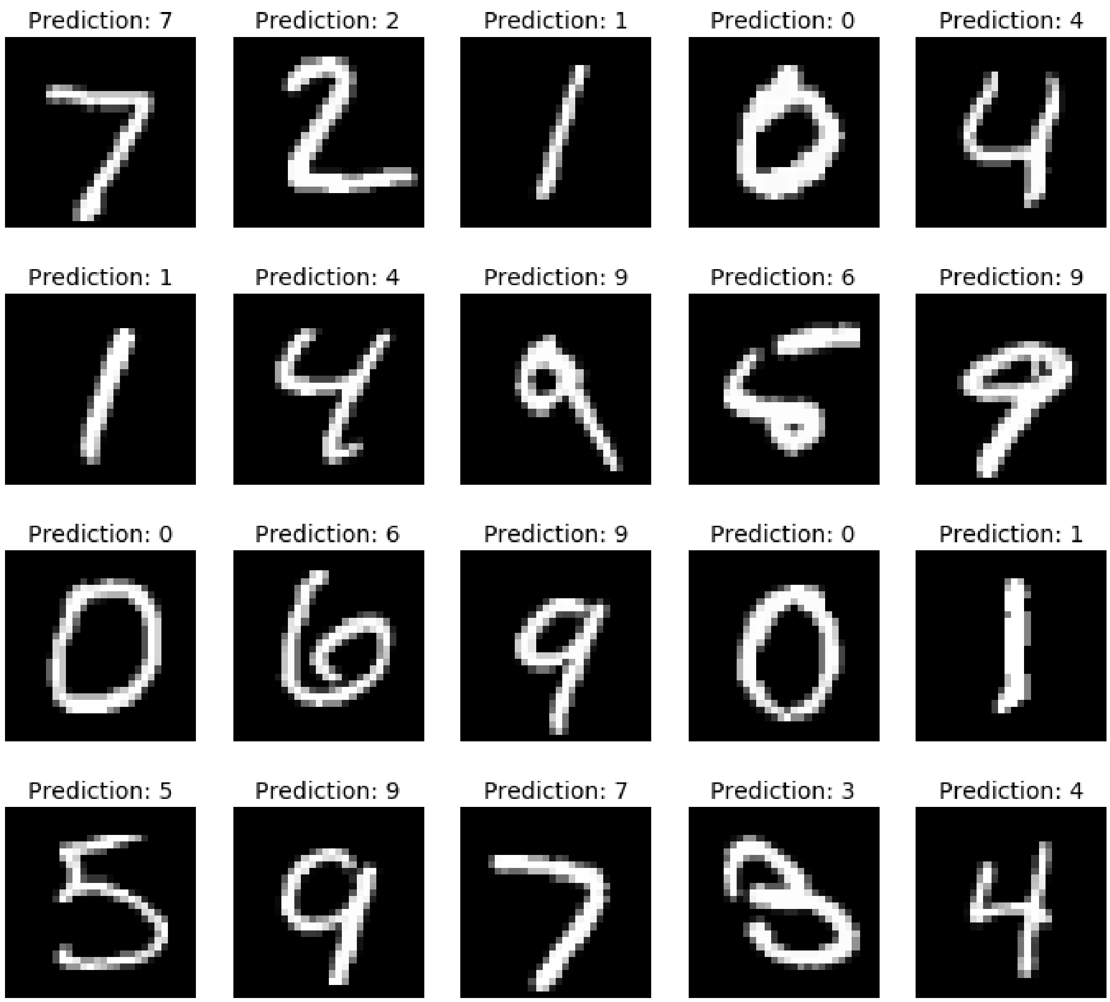
[30]:
```
<matplotlib.legend.Legend at 0x2ddc66a0d08>
```



The curve depicts that the train loss is fluctuating with samples but the test loss is decreasing with increase in samples or examples. The pattern is quite similar to what we obtained before normalizing our data but the numbers and depth of curves vary slightly. Say the test loss is going below *0.2* in the above plot whereas before normalizing it was between *0.3 to 0.5*

[31]:
```
# 2. Show the predictions of the first 20 images of the test dataset
images, labels = iter(test_loader).next()
images, labels = images.to(DEVICE), labels.to(DEVICE)

output = model(images)
pred = output.argmax(dim=1)

fig = plt.figure(figsize=(12, 11))

# -------------------
# Write your implementation here. Use the code provided in Part 0 to visualize␣
↪the images.

for i in range(20):
    plt.subplot(4,5,i+1)
    plt.imshow(images[i].squeeze().cpu().numpy(), cmap='gray',␣
↪interpolation='none')
```

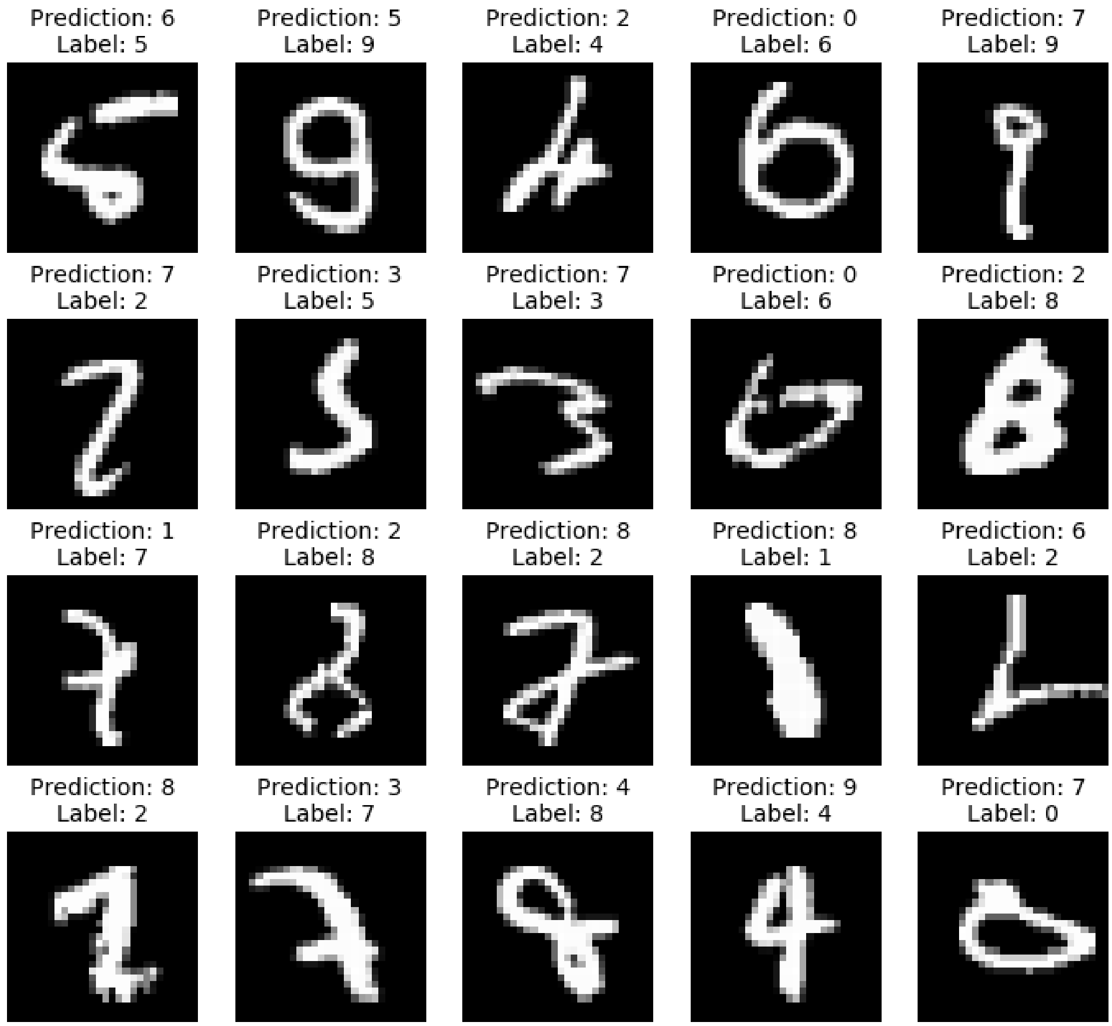
```
    plt.title(f'Prediction: {pred[i]}',fontsize=14)
    plt.axis('off')

# ------------------
```



[32]:

```
# 3. Get 20 incorrect predictions in test dataset

# Collect the images, predictions, labels for the first 20 incorrect predictions
# Initialize empty tensors and then keep appending to the tensor.
# Make sure that the first dimension of the tensors is the total number of␣
↪incorrect
# predictions seen so far
# Ex) incorrect_imgs should be of shape i x C x H x W, where i is the total␣
↪number of
# incorrect images so far.
incorrect_imgs = torch.Tensor().to(DEVICE)
```

```python
incorrect_preds = []
incorrect_labels = []

with torch.no_grad():
    # Test set iterator
    it = iter(test_loader)
    # Loop over the test set batches until incorrect_imgs.size(0) >= 20
    while incorrect_imgs.size(0) < 20:
        images, labels = it.next()
        images, labels = images.to(DEVICE), labels.to(DEVICE)

        # ------------------
        # Write your implementation here.

        output = model(images)
        pred = output.argmax(dim=1)

        # Compare prediction and true labels and append the incorrect␣
↪predictions
        # using `torch.cat`.
        for index, i in enumerate(output):
            if pred[index] !=labels[index]:
                incorrect_imgs = torch.cat([incorrect_imgs,␣
↪images[index]],dim=0)
                incorrect_preds.append(pred[index])
                incorrect_labels.append(labels[index])

        # ------------------

# Show the first 20 wrong predictions in test set
# incorrect_labels = str(incorrect_labels)
# incorrect_preds = str(incorrect_preds)
fig = plt.figure(figsize=(12, 11))
for i in range(20):
    plt.subplot(4, 5, i+1)
    plt.imshow(incorrect_imgs[i].squeeze().cpu().numpy(), cmap='gray',␣
↪interpolation='none')
    plt.title(f'Prediction: {incorrect_preds[i].item()}\nLabel:␣
↪{incorrect_labels[i].item()}', fontsize=14)
    plt.axis('off')
```



In terms of Predictions, the model has improved after normalizing the data. The incorrect predictions above are genuinely hard for humans too recognize at times. *Prediction: 6, Label:2* is hard to predict and there are chances for predicting it as *4* or *6*. Similarly all the above incorrect predictions are genuine and the model performance ahs improved considerally after data normalization.

PS1-Synthetic-Handout

October 3, 2020

1 CS7180 Problem Set 1: Implement a teacher-student network setting for Gaussian inputs (20 points)

Welcome to CS7180!

Before you start, make sure to read the problem description in the handout pdf.

Collaborators: Apoorva Durai, Manaswini, Sinjini Bose. Discussed concepts.

[24]:
```
# Dependencies
import argparse
import torch
import torch.nn.functional as F
import torch.optim as optim
from torchvision import datasets, transforms
import numpy as np


# hyper parameters
batch_size = 100
width = 5
d_input = 100
```

2 Part 1: Implement a two-layer neural network with ReLU activation (5 points)

[2]:
```
class Net(torch.nn.Module):

    def __init__(self, d_input, width):

        super(Net, self).__init__()
        # -------------------
        # Write your implementation here.

        # Linear layer obtaining input to the hidden layer from the input layer
        self.fc1 = torch.nn.Linear(d_input,width)
```

```
        # Applying activation to the output of hidden layer before feeding into␣
↪final layer
        self.act = torch.nn.ReLU()

        # Linear layer -> narrowing to 1 output from hidden layer
        self.fc2 = torch.nn.Linear(width,1)

        # ------------------


    def forward(self, x):

        x = x.view(-1, d_input)
        # ------------------
        # Write your implementation here.

        x = self.act(self.fc1(x))
        x = self.fc2(x)
        return x

        # ------------------
```

2.0.1 Generating the data

[21]:
```
# sample size
N = 5 * width * d_input

# random data from standard normal distribution
x_train = torch.randn(N, d_input)
x_test = torch.randn(N, d_input)

# teacher network with random weights
teacher = Net(d_input, width)

# generate labels using the teacher network
y_train = torch.FloatTensor([teacher.forward(x) for x in x_train])
y_test = torch.FloatTensor([teacher.forward(x) for x in x_test])

# combine the data and labels into pytorch friendly format
train_data = torch.utils.data.TensorDataset(x_train, y_train)
test_data = torch.utils.data.TensorDataset(x_test, y_test)

# prepare data loaders
train_loader = torch.utils.data.DataLoader(train_data, batch_size=batch_size)
test_loader = torch.utils.data.DataLoader(test_data, batch_size=batch_size)
```

3 Part 2: Set up the quadratic loss function and an SGD optimizer (10 points)

[6]:

```
n_epochs = 2000 # the number of epochs can be tuned for better performance

criterion = torch.nn.MSELoss()
optimizer = torch.optim.SGD(teacher.parameters(), lr=0.01)
teacher.train() # prep model for training

for epoch in range(n_epochs):
    train_loss = 0.0

    # train the model
    for idx, (data, labels) in enumerate(train_loader):
        # ------------------
        # Write your implementation here.

        # clear the gradients of all optimized variables
        optimizer.zero_grad()
        # forward pass: compute predicted outputs by passing inputs to the model
        output = teacher(data)
        # calculate the loss
        loss = criterion(output,labels)
        # backward pass: compute gradient of the loss with respect to model␣
↪parameters
        loss.backward()
        # perform a single optimization step (parameter update)
        optimizer.step()
        # update running training loss
        train_loss += loss

        # ------------------

    # print the mean squared loss of the training dataset normalized by the␣
↪mean square of the training dataset labels
    print('Epoch: {} \tTraining Loss: {:.6f}'.format(
        epoch+1,
        train_loss / torch.mean(torch.pow(y_train, 2))))
```

```
C:\Users\mouni\Anaconda3\lib\site-packages\torch\nn\modules\loss.py:431:
UserWarning: Using a target size (torch.Size([100])) that is different to the
input size (torch.Size([100, 1])). This will likely lead to incorrect results
due to broadcasting. Please ensure they have the same size.
  return F.mse_loss(input, target, reduction=self.reduction)
```

```
Epoch: 1        Training Loss: 21.061852
Epoch: 2        Training Loss: 19.397266
```

```
Epoch: 3        Training Loss: 18.120869
Epoch: 4        Training Loss: 17.117247
Epoch: 5        Training Loss: 16.312916
Epoch: 6        Training Loss: 15.657936
Epoch: 7        Training Loss: 15.116551
Epoch: 8        Training Loss: 14.664048
Epoch: 9        Training Loss: 14.281998
Epoch: 10       Training Loss: 13.956030
Epoch: 11       Training Loss: 13.675710
Epoch: 12       Training Loss: 13.432755
Epoch: 13       Training Loss: 13.220798
Epoch: 14       Training Loss: 13.034711
Epoch: 15       Training Loss: 12.870484
Epoch: 16       Training Loss: 12.724731
Epoch: 17       Training Loss: 12.594853
Epoch: 18       Training Loss: 12.478600
Epoch: 19       Training Loss: 12.374182
Epoch: 20       Training Loss: 12.280092
Epoch: 21       Training Loss: 12.194963
Epoch: 22       Training Loss: 12.117729
Epoch: 23       Training Loss: 12.047459
Epoch: 24       Training Loss: 11.983365
Epoch: 25       Training Loss: 11.924751
Epoch: 26       Training Loss: 11.871025
Epoch: 27       Training Loss: 11.821671
Epoch: 28       Training Loss: 11.776239
Epoch: 29       Training Loss: 11.734344
Epoch: 30       Training Loss: 11.695641
Epoch: 31       Training Loss: 11.659824
Epoch: 32       Training Loss: 11.626631
Epoch: 33       Training Loss: 11.595822
Epoch: 34       Training Loss: 11.567187
Epoch: 35       Training Loss: 11.540544
Epoch: 36       Training Loss: 11.515714
Epoch: 37       Training Loss: 11.492553
Epoch: 38       Training Loss: 11.470922
Epoch: 39       Training Loss: 11.450697
Epoch: 40       Training Loss: 11.431772
Epoch: 41       Training Loss: 11.414046
Epoch: 42       Training Loss: 11.397429
Epoch: 43       Training Loss: 11.381837
Epoch: 44       Training Loss: 11.367195
Epoch: 45       Training Loss: 11.353436
Epoch: 46       Training Loss: 11.340497
Epoch: 47       Training Loss: 11.328319
Epoch: 48       Training Loss: 11.316852
Epoch: 49       Training Loss: 11.306046
Epoch: 50       Training Loss: 11.295859
```

```
Epoch: 51       Training Loss: 11.286248
Epoch: 52       Training Loss: 11.277178
Epoch: 53       Training Loss: 11.268612
Epoch: 54       Training Loss: 11.260519
Epoch: 55       Training Loss: 11.252872
Epoch: 56       Training Loss: 11.245639
Epoch: 57       Training Loss: 11.238796
Epoch: 58       Training Loss: 11.232322
Epoch: 59       Training Loss: 11.226194
Epoch: 60       Training Loss: 11.220389
Epoch: 61       Training Loss: 11.214890
Epoch: 62       Training Loss: 11.209678
Epoch: 63       Training Loss: 11.204739
Epoch: 64       Training Loss: 11.200054
Epoch: 65       Training Loss: 11.195611
Epoch: 66       Training Loss: 11.191396
Epoch: 67       Training Loss: 11.187393
Epoch: 68       Training Loss: 11.183597
Epoch: 69       Training Loss: 11.179991
Epoch: 70       Training Loss: 11.176566
Epoch: 71       Training Loss: 11.173314
Epoch: 72       Training Loss: 11.170223
Epoch: 73       Training Loss: 11.167287
Epoch: 74       Training Loss: 11.164497
Epoch: 75       Training Loss: 11.161841
Epoch: 76       Training Loss: 11.159319
Epoch: 77       Training Loss: 11.156919
Epoch: 78       Training Loss: 11.154637
Epoch: 79       Training Loss: 11.152466
Epoch: 80       Training Loss: 11.150402
Epoch: 81       Training Loss: 11.148436
Epoch: 82       Training Loss: 11.146566
Epoch: 83       Training Loss: 11.144786
Epoch: 84       Training Loss: 11.143093
Epoch: 85       Training Loss: 11.141479
Epoch: 86       Training Loss: 11.139944
Epoch: 87       Training Loss: 11.138483
Epoch: 88       Training Loss: 11.137090
Epoch: 89       Training Loss: 11.135763
Epoch: 90       Training Loss: 11.134501
Epoch: 91       Training Loss: 11.133296
Epoch: 92       Training Loss: 11.132152
Epoch: 93       Training Loss: 11.131060
Epoch: 94       Training Loss: 11.130019
Epoch: 95       Training Loss: 11.129027
Epoch: 96       Training Loss: 11.128081
Epoch: 97       Training Loss: 11.127181
Epoch: 98       Training Loss: 11.126324
```

```
Epoch: 99       Training Loss: 11.125506
Epoch: 100      Training Loss: 11.124727
Epoch: 101      Training Loss: 11.123982
Epoch: 102      Training Loss: 11.123277
Epoch: 103      Training Loss: 11.122602
Epoch: 104      Training Loss: 11.121957
Epoch: 105      Training Loss: 11.121343
Epoch: 106      Training Loss: 11.120756
Epoch: 107      Training Loss: 11.120200
Epoch: 108      Training Loss: 11.119667
Epoch: 109      Training Loss: 11.119160
Epoch: 110      Training Loss: 11.118675
Epoch: 111      Training Loss: 11.118217
Epoch: 112      Training Loss: 11.117776
Epoch: 113      Training Loss: 11.117354
Epoch: 114      Training Loss: 11.116956
Epoch: 115      Training Loss: 11.116573
Epoch: 116      Training Loss: 11.116208
Epoch: 117      Training Loss: 11.115862
Epoch: 118      Training Loss: 11.115529
Epoch: 119      Training Loss: 11.115214
Epoch: 120      Training Loss: 11.114911
Epoch: 121      Training Loss: 11.114623
Epoch: 122      Training Loss: 11.114350
Epoch: 123      Training Loss: 11.114089
Epoch: 124      Training Loss: 11.113838
Epoch: 125      Training Loss: 11.113601
Epoch: 126      Training Loss: 11.113374
Epoch: 127      Training Loss: 11.113155
Epoch: 128      Training Loss: 11.112948
Epoch: 129      Training Loss: 11.112752
Epoch: 130      Training Loss: 11.112562
Epoch: 131      Training Loss: 11.112383
Epoch: 132      Training Loss: 11.112211
Epoch: 133      Training Loss: 11.112048
Epoch: 134      Training Loss: 11.111891
Epoch: 135      Training Loss: 11.111743
Epoch: 136      Training Loss: 11.111601
Epoch: 137      Training Loss: 11.111465
Epoch: 138      Training Loss: 11.111335
Epoch: 139      Training Loss: 11.111211
Epoch: 140      Training Loss: 11.111093
Epoch: 141      Training Loss: 11.110980
Epoch: 142      Training Loss: 11.110873
Epoch: 143      Training Loss: 11.110770
Epoch: 144      Training Loss: 11.110674
Epoch: 145      Training Loss: 11.110578
Epoch: 146      Training Loss: 11.110491
```

```
Epoch: 147      Training Loss: 11.110405
Epoch: 148      Training Loss: 11.110324
Epoch: 149      Training Loss: 11.110247
Epoch: 150      Training Loss: 11.110172
Epoch: 151      Training Loss: 11.110103
Epoch: 152      Training Loss: 11.110033
Epoch: 153      Training Loss: 11.109971
Epoch: 154      Training Loss: 11.109910
Epoch: 155      Training Loss: 11.109852
Epoch: 156      Training Loss: 11.109796
Epoch: 157      Training Loss: 11.109740
Epoch: 158      Training Loss: 11.109690
Epoch: 159      Training Loss: 11.109643
Epoch: 160      Training Loss: 11.109597
Epoch: 161      Training Loss: 11.109551
Epoch: 162      Training Loss: 11.109509
Epoch: 163      Training Loss: 11.109469
Epoch: 164      Training Loss: 11.109431
Epoch: 165      Training Loss: 11.109393
Epoch: 166      Training Loss: 11.109359
Epoch: 167      Training Loss: 11.109326
Epoch: 168      Training Loss: 11.109293
Epoch: 169      Training Loss: 11.109265
Epoch: 170      Training Loss: 11.109234
Epoch: 171      Training Loss: 11.109206
Epoch: 172      Training Loss: 11.109180
Epoch: 173      Training Loss: 11.109153
Epoch: 174      Training Loss: 11.109132
Epoch: 175      Training Loss: 11.109106
Epoch: 176      Training Loss: 11.109085
Epoch: 177      Training Loss: 11.109065
Epoch: 178      Training Loss: 11.109043
Epoch: 179      Training Loss: 11.109023
Epoch: 180      Training Loss: 11.109006
Epoch: 181      Training Loss: 11.108989
Epoch: 182      Training Loss: 11.108973
Epoch: 183      Training Loss: 11.108955
Epoch: 184      Training Loss: 11.108941
Epoch: 185      Training Loss: 11.108928
Epoch: 186      Training Loss: 11.108913
Epoch: 187      Training Loss: 11.108900
Epoch: 188      Training Loss: 11.108890
Epoch: 189      Training Loss: 11.108875
Epoch: 190      Training Loss: 11.108865
Epoch: 191      Training Loss: 11.108853
Epoch: 192      Training Loss: 11.108842
Epoch: 193      Training Loss: 11.108831
Epoch: 194      Training Loss: 11.108823
```

```
Epoch: 195      Training Loss: 11.108813
Epoch: 196      Training Loss: 11.108804
Epoch: 197      Training Loss: 11.108795
Epoch: 198      Training Loss: 11.108788
Epoch: 199      Training Loss: 11.108782
Epoch: 200      Training Loss: 11.108774
Epoch: 201      Training Loss: 11.108768
Epoch: 202      Training Loss: 11.108761
Epoch: 203      Training Loss: 11.108754
Epoch: 204      Training Loss: 11.108747
Epoch: 205      Training Loss: 11.108742
Epoch: 206      Training Loss: 11.108737
Epoch: 207      Training Loss: 11.108730
Epoch: 208      Training Loss: 11.108728
Epoch: 209      Training Loss: 11.108721
Epoch: 210      Training Loss: 11.108716
Epoch: 211      Training Loss: 11.108711
Epoch: 212      Training Loss: 11.108706
Epoch: 213      Training Loss: 11.108704
Epoch: 214      Training Loss: 11.108700
Epoch: 215      Training Loss: 11.108698
Epoch: 216      Training Loss: 11.108694
Epoch: 217      Training Loss: 11.108691
Epoch: 218      Training Loss: 11.108687
Epoch: 219      Training Loss: 11.108685
Epoch: 220      Training Loss: 11.108682
Epoch: 221      Training Loss: 11.108678
Epoch: 222      Training Loss: 11.108674
Epoch: 223      Training Loss: 11.108673
Epoch: 224      Training Loss: 11.108670
Epoch: 225      Training Loss: 11.108669
Epoch: 226      Training Loss: 11.108665
Epoch: 227      Training Loss: 11.108663
Epoch: 228      Training Loss: 11.108661
Epoch: 229      Training Loss: 11.108661
Epoch: 230      Training Loss: 11.108656
Epoch: 231      Training Loss: 11.108655
Epoch: 232      Training Loss: 11.108655
Epoch: 233      Training Loss: 11.108654
Epoch: 234      Training Loss: 11.108652
Epoch: 235      Training Loss: 11.108650
Epoch: 236      Training Loss: 11.108648
Epoch: 237      Training Loss: 11.108647
Epoch: 238      Training Loss: 11.108646
Epoch: 239      Training Loss: 11.108644
Epoch: 240      Training Loss: 11.108643
Epoch: 241      Training Loss: 11.108642
Epoch: 242      Training Loss: 11.108641
```

```
Epoch: 243      Training Loss: 11.108642
Epoch: 244      Training Loss: 11.108639
Epoch: 245      Training Loss: 11.108637
Epoch: 246      Training Loss: 11.108636
Epoch: 247      Training Loss: 11.108636
Epoch: 248      Training Loss: 11.108636
Epoch: 249      Training Loss: 11.108635
Epoch: 250      Training Loss: 11.108635
Epoch: 251      Training Loss: 11.108630
Epoch: 252      Training Loss: 11.108631
Epoch: 253      Training Loss: 11.108631
Epoch: 254      Training Loss: 11.108629
Epoch: 255      Training Loss: 11.108627
Epoch: 256      Training Loss: 11.108629
Epoch: 257      Training Loss: 11.108627
Epoch: 258      Training Loss: 11.108628
Epoch: 259      Training Loss: 11.108625
Epoch: 260      Training Loss: 11.108624
Epoch: 261      Training Loss: 11.108624
Epoch: 262      Training Loss: 11.108624
Epoch: 263      Training Loss: 11.108624
Epoch: 264      Training Loss: 11.108623
Epoch: 265      Training Loss: 11.108624
Epoch: 266      Training Loss: 11.108624
Epoch: 267      Training Loss: 11.108623
Epoch: 268      Training Loss: 11.108623
Epoch: 269      Training Loss: 11.108621
Epoch: 270      Training Loss: 11.108621
Epoch: 271      Training Loss: 11.108619
Epoch: 272      Training Loss: 11.108621
Epoch: 273      Training Loss: 11.108620
Epoch: 274      Training Loss: 11.108620
Epoch: 275      Training Loss: 11.108620
Epoch: 276      Training Loss: 11.108618
Epoch: 277      Training Loss: 11.108619
Epoch: 278      Training Loss: 11.108618
Epoch: 279      Training Loss: 11.108618
Epoch: 280      Training Loss: 11.108617
Epoch: 281      Training Loss: 11.108618
Epoch: 282      Training Loss: 11.108617
Epoch: 283      Training Loss: 11.108616
Epoch: 284      Training Loss: 11.108615
Epoch: 285      Training Loss: 11.108616
Epoch: 286      Training Loss: 11.108614
Epoch: 287      Training Loss: 11.108615
Epoch: 288      Training Loss: 11.108615
Epoch: 289      Training Loss: 11.108615
Epoch: 290      Training Loss: 11.108614
```

```
Epoch: 291      Training Loss: 11.108615
Epoch: 292      Training Loss: 11.108614
Epoch: 293      Training Loss: 11.108614
Epoch: 294      Training Loss: 11.108614
Epoch: 295      Training Loss: 11.108614
Epoch: 296      Training Loss: 11.108614
Epoch: 297      Training Loss: 11.108613
Epoch: 298      Training Loss: 11.108612
Epoch: 299      Training Loss: 11.108612
Epoch: 300      Training Loss: 11.108612
Epoch: 301      Training Loss: 11.108613
Epoch: 302      Training Loss: 11.108611
Epoch: 303      Training Loss: 11.108611
Epoch: 304      Training Loss: 11.108610
Epoch: 305      Training Loss: 11.108609
Epoch: 306      Training Loss: 11.108610
Epoch: 307      Training Loss: 11.108610
Epoch: 308      Training Loss: 11.108610
Epoch: 309      Training Loss: 11.108610
Epoch: 310      Training Loss: 11.108609
Epoch: 311      Training Loss: 11.108608
Epoch: 312      Training Loss: 11.108608
Epoch: 313      Training Loss: 11.108608
Epoch: 314      Training Loss: 11.108609
Epoch: 315      Training Loss: 11.108609
Epoch: 316      Training Loss: 11.108608
Epoch: 317      Training Loss: 11.108608
Epoch: 318      Training Loss: 11.108607
Epoch: 319      Training Loss: 11.108606
Epoch: 320      Training Loss: 11.108606
Epoch: 321      Training Loss: 11.108606
Epoch: 322      Training Loss: 11.108606
Epoch: 323      Training Loss: 11.108606
Epoch: 324      Training Loss: 11.108607
Epoch: 325      Training Loss: 11.108606
Epoch: 326      Training Loss: 11.108607
Epoch: 327      Training Loss: 11.108606
Epoch: 328      Training Loss: 11.108606
Epoch: 329      Training Loss: 11.108605
Epoch: 330      Training Loss: 11.108606
Epoch: 331      Training Loss: 11.108605
Epoch: 332      Training Loss: 11.108605
Epoch: 333      Training Loss: 11.108604
Epoch: 334      Training Loss: 11.108604
Epoch: 335      Training Loss: 11.108604
Epoch: 336      Training Loss: 11.108604
Epoch: 337      Training Loss: 11.108603
Epoch: 338      Training Loss: 11.108604
```

```
Epoch: 339      Training Loss: 11.108603
Epoch: 340      Training Loss: 11.108604
Epoch: 341      Training Loss: 11.108604
Epoch: 342      Training Loss: 11.108603
Epoch: 343      Training Loss: 11.108603
Epoch: 344      Training Loss: 11.108603
Epoch: 345      Training Loss: 11.108603
Epoch: 346      Training Loss: 11.108603
Epoch: 347      Training Loss: 11.108603
Epoch: 348      Training Loss: 11.108602
Epoch: 349      Training Loss: 11.108601
Epoch: 350      Training Loss: 11.108601
Epoch: 351      Training Loss: 11.108601
Epoch: 352      Training Loss: 11.108601
Epoch: 353      Training Loss: 11.108601
Epoch: 354      Training Loss: 11.108601
Epoch: 355      Training Loss: 11.108601
Epoch: 356      Training Loss: 11.108601
Epoch: 357      Training Loss: 11.108601
Epoch: 358      Training Loss: 11.108601
Epoch: 359      Training Loss: 11.108601
Epoch: 360      Training Loss: 11.108600
Epoch: 361      Training Loss: 11.108600
Epoch: 362      Training Loss: 11.108600
Epoch: 363      Training Loss: 11.108600
Epoch: 364      Training Loss: 11.108600
Epoch: 365      Training Loss: 11.108600
Epoch: 366      Training Loss: 11.108598
Epoch: 367      Training Loss: 11.108598
Epoch: 368      Training Loss: 11.108598
Epoch: 369      Training Loss: 11.108598
Epoch: 370      Training Loss: 11.108597
Epoch: 371      Training Loss: 11.108598
Epoch: 372      Training Loss: 11.108597
Epoch: 373      Training Loss: 11.108598
Epoch: 374      Training Loss: 11.108596
Epoch: 375      Training Loss: 11.108597
Epoch: 376      Training Loss: 11.108595
Epoch: 377      Training Loss: 11.108595
Epoch: 378      Training Loss: 11.108595
Epoch: 379      Training Loss: 11.108595
Epoch: 380      Training Loss: 11.108595
Epoch: 381      Training Loss: 11.108595
Epoch: 382      Training Loss: 11.108595
Epoch: 383      Training Loss: 11.108594
Epoch: 384      Training Loss: 11.108593
Epoch: 385      Training Loss: 11.108595
Epoch: 386      Training Loss: 11.108594
```

```
Epoch: 387      Training Loss: 11.108594
Epoch: 388      Training Loss: 11.108594
Epoch: 389      Training Loss: 11.108593
Epoch: 390      Training Loss: 11.108592
Epoch: 391      Training Loss: 11.108593
Epoch: 392      Training Loss: 11.108592
Epoch: 393      Training Loss: 11.108591
Epoch: 394      Training Loss: 11.108593
Epoch: 395      Training Loss: 11.108592
Epoch: 396      Training Loss: 11.108591
Epoch: 397      Training Loss: 11.108591
Epoch: 398      Training Loss: 11.108591
Epoch: 399      Training Loss: 11.108591
Epoch: 400      Training Loss: 11.108590
Epoch: 401      Training Loss: 11.108591
Epoch: 402      Training Loss: 11.108591
Epoch: 403      Training Loss: 11.108589
Epoch: 404      Training Loss: 11.108589
Epoch: 405      Training Loss: 11.108589
Epoch: 406      Training Loss: 11.108589
Epoch: 407      Training Loss: 11.108589
Epoch: 408      Training Loss: 11.108589
Epoch: 409      Training Loss: 11.108588
Epoch: 410      Training Loss: 11.108589
Epoch: 411      Training Loss: 11.108589
Epoch: 412      Training Loss: 11.108589
Epoch: 413      Training Loss: 11.108588
Epoch: 414      Training Loss: 11.108588
Epoch: 415      Training Loss: 11.108588
Epoch: 416      Training Loss: 11.108588
Epoch: 417      Training Loss: 11.108587
Epoch: 418      Training Loss: 11.108588
Epoch: 419      Training Loss: 11.108587
Epoch: 420      Training Loss: 11.108586
Epoch: 421      Training Loss: 11.108585
Epoch: 422      Training Loss: 11.108585
Epoch: 423      Training Loss: 11.108586
Epoch: 424      Training Loss: 11.108586
Epoch: 425      Training Loss: 11.108586
Epoch: 426      Training Loss: 11.108586
Epoch: 427      Training Loss: 11.108585
Epoch: 428      Training Loss: 11.108585
Epoch: 429      Training Loss: 11.108585
Epoch: 430      Training Loss: 11.108585
Epoch: 431      Training Loss: 11.108584
Epoch: 432      Training Loss: 11.108584
Epoch: 433      Training Loss: 11.108583
Epoch: 434      Training Loss: 11.108584
```

```
Epoch: 435      Training Loss: 11.108584
Epoch: 436      Training Loss: 11.108584
Epoch: 437      Training Loss: 11.108583
Epoch: 438      Training Loss: 11.108584
Epoch: 439      Training Loss: 11.108583
Epoch: 440      Training Loss: 11.108583
Epoch: 441      Training Loss: 11.108583
Epoch: 442      Training Loss: 11.108583
Epoch: 443      Training Loss: 11.108583
Epoch: 444      Training Loss: 11.108582
Epoch: 445      Training Loss: 11.108582
Epoch: 446      Training Loss: 11.108582
Epoch: 447      Training Loss: 11.108582
Epoch: 448      Training Loss: 11.108582
Epoch: 449      Training Loss: 11.108582
Epoch: 450      Training Loss: 11.108582
Epoch: 451      Training Loss: 11.108581
Epoch: 452      Training Loss: 11.108581
Epoch: 453      Training Loss: 11.108581
Epoch: 454      Training Loss: 11.108580
Epoch: 455      Training Loss: 11.108582
Epoch: 456      Training Loss: 11.108580
Epoch: 457      Training Loss: 11.108580
Epoch: 458      Training Loss: 11.108578
Epoch: 459      Training Loss: 11.108580
Epoch: 460      Training Loss: 11.108578
Epoch: 461      Training Loss: 11.108578
Epoch: 462      Training Loss: 11.108578
Epoch: 463      Training Loss: 11.108578
Epoch: 464      Training Loss: 11.108578
Epoch: 465      Training Loss: 11.108577
Epoch: 466      Training Loss: 11.108577
Epoch: 467      Training Loss: 11.108578
Epoch: 468      Training Loss: 11.108577
Epoch: 469      Training Loss: 11.108576
Epoch: 470      Training Loss: 11.108576
Epoch: 471      Training Loss: 11.108576
Epoch: 472      Training Loss: 11.108576
Epoch: 473      Training Loss: 11.108576
Epoch: 474      Training Loss: 11.108576
Epoch: 475      Training Loss: 11.108576
Epoch: 476      Training Loss: 11.108575
Epoch: 477      Training Loss: 11.108575
Epoch: 478      Training Loss: 11.108575
Epoch: 479      Training Loss: 11.108574
Epoch: 480      Training Loss: 11.108574
Epoch: 481      Training Loss: 11.108575
Epoch: 482      Training Loss: 11.108574
```

```
Epoch: 483      Training Loss: 11.108574
Epoch: 484      Training Loss: 11.108574
Epoch: 485      Training Loss: 11.108573
Epoch: 486      Training Loss: 11.108573
Epoch: 487      Training Loss: 11.108572
Epoch: 488      Training Loss: 11.108572
Epoch: 489      Training Loss: 11.108572
Epoch: 490      Training Loss: 11.108572
Epoch: 491      Training Loss: 11.108572
Epoch: 492      Training Loss: 11.108572
Epoch: 493      Training Loss: 11.108572
Epoch: 494      Training Loss: 11.108570
Epoch: 495      Training Loss: 11.108571
Epoch: 496      Training Loss: 11.108570
Epoch: 497      Training Loss: 11.108570
Epoch: 498      Training Loss: 11.108571
Epoch: 499      Training Loss: 11.108569
Epoch: 500      Training Loss: 11.108569
Epoch: 501      Training Loss: 11.108569
Epoch: 502      Training Loss: 11.108569
Epoch: 503      Training Loss: 11.108570
Epoch: 504      Training Loss: 11.108569
Epoch: 505      Training Loss: 11.108569
Epoch: 506      Training Loss: 11.108569
Epoch: 507      Training Loss: 11.108568
Epoch: 508      Training Loss: 11.108568
Epoch: 509      Training Loss: 11.108568
Epoch: 510      Training Loss: 11.108568
Epoch: 511      Training Loss: 11.108567
Epoch: 512      Training Loss: 11.108568
Epoch: 513      Training Loss: 11.108566
Epoch: 514      Training Loss: 11.108566
Epoch: 515      Training Loss: 11.108567
Epoch: 516      Training Loss: 11.108567
Epoch: 517      Training Loss: 11.108566
Epoch: 518      Training Loss: 11.108567
Epoch: 519      Training Loss: 11.108567
Epoch: 520      Training Loss: 11.108566
Epoch: 521      Training Loss: 11.108566
Epoch: 522      Training Loss: 11.108565
Epoch: 523      Training Loss: 11.108565
Epoch: 524      Training Loss: 11.108565
Epoch: 525      Training Loss: 11.108565
Epoch: 526      Training Loss: 11.108565
Epoch: 527      Training Loss: 11.108565
Epoch: 528      Training Loss: 11.108565
Epoch: 529      Training Loss: 11.108565
Epoch: 530      Training Loss: 11.108565
```

```
Epoch: 531      Training Loss: 11.108565
Epoch: 532      Training Loss: 11.108565
Epoch: 533      Training Loss: 11.108565
Epoch: 534      Training Loss: 11.108565
Epoch: 535      Training Loss: 11.108565
Epoch: 536      Training Loss: 11.108562
Epoch: 537      Training Loss: 11.108563
Epoch: 538      Training Loss: 11.108564
Epoch: 539      Training Loss: 11.108562
Epoch: 540      Training Loss: 11.108562
Epoch: 541      Training Loss: 11.108562
Epoch: 542      Training Loss: 11.108562
Epoch: 543      Training Loss: 11.108562
Epoch: 544      Training Loss: 11.108561
Epoch: 545      Training Loss: 11.108562
Epoch: 546      Training Loss: 11.108561
Epoch: 547      Training Loss: 11.108562
Epoch: 548      Training Loss: 11.108562
Epoch: 549      Training Loss: 11.108562
Epoch: 550      Training Loss: 11.108562
Epoch: 551      Training Loss: 11.108562
Epoch: 552      Training Loss: 11.108560
Epoch: 553      Training Loss: 11.108561
Epoch: 554      Training Loss: 11.108561
Epoch: 555      Training Loss: 11.108560
Epoch: 556      Training Loss: 11.108561
Epoch: 557      Training Loss: 11.108559
Epoch: 558      Training Loss: 11.108559
Epoch: 559      Training Loss: 11.108559
Epoch: 560      Training Loss: 11.108560
Epoch: 561      Training Loss: 11.108559
Epoch: 562      Training Loss: 11.108559
Epoch: 563      Training Loss: 11.108557
Epoch: 564      Training Loss: 11.108558
Epoch: 565      Training Loss: 11.108559
Epoch: 566      Training Loss: 11.108559
Epoch: 567      Training Loss: 11.108559
Epoch: 568      Training Loss: 11.108557
Epoch: 569      Training Loss: 11.108557
Epoch: 570      Training Loss: 11.108557
Epoch: 571      Training Loss: 11.108557
Epoch: 572      Training Loss: 11.108558
Epoch: 573      Training Loss: 11.108558
Epoch: 574      Training Loss: 11.108557
Epoch: 575      Training Loss: 11.108556
Epoch: 576      Training Loss: 11.108557
Epoch: 577      Training Loss: 11.108556
Epoch: 578      Training Loss: 11.108554
```

```
Epoch: 579      Training Loss: 11.108555
Epoch: 580      Training Loss: 11.108555
Epoch: 581      Training Loss: 11.108555
Epoch: 582      Training Loss: 11.108554
Epoch: 583      Training Loss: 11.108555
Epoch: 584      Training Loss: 11.108554
Epoch: 585      Training Loss: 11.108555
Epoch: 586      Training Loss: 11.108555
Epoch: 587      Training Loss: 11.108554
Epoch: 588      Training Loss: 11.108554
Epoch: 589      Training Loss: 11.108554
Epoch: 590      Training Loss: 11.108554
Epoch: 591      Training Loss: 11.108554
Epoch: 592      Training Loss: 11.108553
Epoch: 593      Training Loss: 11.108553
Epoch: 594      Training Loss: 11.108553
Epoch: 595      Training Loss: 11.108553
Epoch: 596      Training Loss: 11.108553
Epoch: 597      Training Loss: 11.108552
Epoch: 598      Training Loss: 11.108552
Epoch: 599      Training Loss: 11.108551
Epoch: 600      Training Loss: 11.108552
Epoch: 601      Training Loss: 11.108552
Epoch: 602      Training Loss: 11.108551
Epoch: 603      Training Loss: 11.108551
Epoch: 604      Training Loss: 11.108552
Epoch: 605      Training Loss: 11.108549
Epoch: 606      Training Loss: 11.108549
Epoch: 607      Training Loss: 11.108549
Epoch: 608      Training Loss: 11.108549
Epoch: 609      Training Loss: 11.108548
Epoch: 610      Training Loss: 11.108548
Epoch: 611      Training Loss: 11.108548
Epoch: 612      Training Loss: 11.108548
Epoch: 613      Training Loss: 11.108548
Epoch: 614      Training Loss: 11.108548
Epoch: 615      Training Loss: 11.108547
Epoch: 616      Training Loss: 11.108548
Epoch: 617      Training Loss: 11.108547
Epoch: 618      Training Loss: 11.108547
Epoch: 619      Training Loss: 11.108547
Epoch: 620      Training Loss: 11.108547
Epoch: 621      Training Loss: 11.108546
Epoch: 622      Training Loss: 11.108545
Epoch: 623      Training Loss: 11.108544
Epoch: 624      Training Loss: 11.108544
Epoch: 625      Training Loss: 11.108544
Epoch: 626      Training Loss: 11.108544
```

```
Epoch: 627      Training Loss: 11.108544
Epoch: 628      Training Loss: 11.108544
Epoch: 629      Training Loss: 11.108544
Epoch: 630      Training Loss: 11.108544
Epoch: 631      Training Loss: 11.108545
Epoch: 632      Training Loss: 11.108544
Epoch: 633      Training Loss: 11.108543
Epoch: 634      Training Loss: 11.108543
Epoch: 635      Training Loss: 11.108543
Epoch: 636      Training Loss: 11.108543
Epoch: 637      Training Loss: 11.108543
Epoch: 638      Training Loss: 11.108543
Epoch: 639      Training Loss: 11.108543
Epoch: 640      Training Loss: 11.108543
Epoch: 641      Training Loss: 11.108543
Epoch: 642      Training Loss: 11.108542
Epoch: 643      Training Loss: 11.108542
Epoch: 644      Training Loss: 11.108541
Epoch: 645      Training Loss: 11.108541
Epoch: 646      Training Loss: 11.108541
Epoch: 647      Training Loss: 11.108541
Epoch: 648      Training Loss: 11.108541
Epoch: 649      Training Loss: 11.108541
Epoch: 650      Training Loss: 11.108541
Epoch: 651      Training Loss: 11.108541
Epoch: 652      Training Loss: 11.108541
Epoch: 653      Training Loss: 11.108541
Epoch: 654      Training Loss: 11.108541
Epoch: 655      Training Loss: 11.108541
Epoch: 656      Training Loss: 11.108541
Epoch: 657      Training Loss: 11.108541
Epoch: 658      Training Loss: 11.108539
Epoch: 659      Training Loss: 11.108539
Epoch: 660      Training Loss: 11.108539
Epoch: 661      Training Loss: 11.108539
Epoch: 662      Training Loss: 11.108539
Epoch: 663      Training Loss: 11.108539
Epoch: 664      Training Loss: 11.108538
Epoch: 665      Training Loss: 11.108537
Epoch: 666      Training Loss: 11.108537
Epoch: 667      Training Loss: 11.108537
Epoch: 668      Training Loss: 11.108537
Epoch: 669      Training Loss: 11.108537
Epoch: 670      Training Loss: 11.108537
Epoch: 671      Training Loss: 11.108537
Epoch: 672      Training Loss: 11.108536
Epoch: 673      Training Loss: 11.108536
Epoch: 674      Training Loss: 11.108536
```

```
Epoch: 675      Training Loss: 11.108536
Epoch: 676      Training Loss: 11.108537
Epoch: 677      Training Loss: 11.108535
Epoch: 678      Training Loss: 11.108536
Epoch: 679      Training Loss: 11.108535
Epoch: 680      Training Loss: 11.108534
Epoch: 681      Training Loss: 11.108535
Epoch: 682      Training Loss: 11.108534
Epoch: 683      Training Loss: 11.108534
Epoch: 684      Training Loss: 11.108534
Epoch: 685      Training Loss: 11.108534
Epoch: 686      Training Loss: 11.108533
Epoch: 687      Training Loss: 11.108533
Epoch: 688      Training Loss: 11.108533
Epoch: 689      Training Loss: 11.108532
Epoch: 690      Training Loss: 11.108531
Epoch: 691      Training Loss: 11.108531
Epoch: 692      Training Loss: 11.108531
Epoch: 693      Training Loss: 11.108531
Epoch: 694      Training Loss: 11.108531
Epoch: 695      Training Loss: 11.108531
Epoch: 696      Training Loss: 11.108531
Epoch: 697      Training Loss: 11.108531
Epoch: 698      Training Loss: 11.108531
Epoch: 699      Training Loss: 11.108531
Epoch: 700      Training Loss: 11.108531
Epoch: 701      Training Loss: 11.108531
Epoch: 702      Training Loss: 11.108531
Epoch: 703      Training Loss: 11.108531
Epoch: 704      Training Loss: 11.108531
Epoch: 705      Training Loss: 11.108530
Epoch: 706      Training Loss: 11.108530
Epoch: 707      Training Loss: 11.108530
Epoch: 708      Training Loss: 11.108531
Epoch: 709      Training Loss: 11.108530
Epoch: 710      Training Loss: 11.108529
Epoch: 711      Training Loss: 11.108529
Epoch: 712      Training Loss: 11.108529
Epoch: 713      Training Loss: 11.108528
Epoch: 714      Training Loss: 11.108528
Epoch: 715      Training Loss: 11.108527
Epoch: 716      Training Loss: 11.108527
Epoch: 717      Training Loss: 11.108527
Epoch: 718      Training Loss: 11.108525
Epoch: 719      Training Loss: 11.108525
Epoch: 720      Training Loss: 11.108526
Epoch: 721      Training Loss: 11.108526
Epoch: 722      Training Loss: 11.108526
```

```
Epoch: 723      Training Loss: 11.108526
Epoch: 724      Training Loss: 11.108525
Epoch: 725      Training Loss: 11.108524
Epoch: 726      Training Loss: 11.108524
Epoch: 727      Training Loss: 11.108524
Epoch: 728      Training Loss: 11.108524
Epoch: 729      Training Loss: 11.108524
Epoch: 730      Training Loss: 11.108524
Epoch: 731      Training Loss: 11.108523
Epoch: 732      Training Loss: 11.108524
Epoch: 733      Training Loss: 11.108524
Epoch: 734      Training Loss: 11.108522
Epoch: 735      Training Loss: 11.108522
Epoch: 736      Training Loss: 11.108521
Epoch: 737      Training Loss: 11.108521
Epoch: 738      Training Loss: 11.108522
Epoch: 739      Training Loss: 11.108522
Epoch: 740      Training Loss: 11.108522
Epoch: 741      Training Loss: 11.108521
Epoch: 742      Training Loss: 11.108521
Epoch: 743      Training Loss: 11.108521
Epoch: 744      Training Loss: 11.108521
Epoch: 745      Training Loss: 11.108521
Epoch: 746      Training Loss: 11.108521
Epoch: 747      Training Loss: 11.108521
Epoch: 748      Training Loss: 11.108521
Epoch: 749      Training Loss: 11.108520
Epoch: 750      Training Loss: 11.108521
Epoch: 751      Training Loss: 11.108520
Epoch: 752      Training Loss: 11.108520
Epoch: 753      Training Loss: 11.108519
Epoch: 754      Training Loss: 11.108519
Epoch: 755      Training Loss: 11.108519
Epoch: 756      Training Loss: 11.108519
Epoch: 757      Training Loss: 11.108518
Epoch: 758      Training Loss: 11.108518
Epoch: 759      Training Loss: 11.108518
Epoch: 760      Training Loss: 11.108518
Epoch: 761      Training Loss: 11.108517
Epoch: 762      Training Loss: 11.108517
Epoch: 763      Training Loss: 11.108518
Epoch: 764      Training Loss: 11.108518
Epoch: 765      Training Loss: 11.108517
Epoch: 766      Training Loss: 11.108517
Epoch: 767      Training Loss: 11.108517
Epoch: 768      Training Loss: 11.108517
Epoch: 769      Training Loss: 11.108515
Epoch: 770      Training Loss: 11.108517
```

```
Epoch: 771      Training Loss: 11.108515
Epoch: 772      Training Loss: 11.108515
Epoch: 773      Training Loss: 11.108515
Epoch: 774      Training Loss: 11.108514
Epoch: 775      Training Loss: 11.108514
Epoch: 776      Training Loss: 11.108513
Epoch: 777      Training Loss: 11.108513
Epoch: 778      Training Loss: 11.108513
Epoch: 779      Training Loss: 11.108513
Epoch: 780      Training Loss: 11.108513
Epoch: 781      Training Loss: 11.108513
Epoch: 782      Training Loss: 11.108513
Epoch: 783      Training Loss: 11.108512
Epoch: 784      Training Loss: 11.108512
Epoch: 785      Training Loss: 11.108512
Epoch: 786      Training Loss: 11.108511
Epoch: 787      Training Loss: 11.108512
Epoch: 788      Training Loss: 11.108512
Epoch: 789      Training Loss: 11.108511
Epoch: 790      Training Loss: 11.108511
Epoch: 791      Training Loss: 11.108510
Epoch: 792      Training Loss: 11.108510
Epoch: 793      Training Loss: 11.108510
Epoch: 794      Training Loss: 11.108509
Epoch: 795      Training Loss: 11.108510
Epoch: 796      Training Loss: 11.108509
Epoch: 797      Training Loss: 11.108509
Epoch: 798      Training Loss: 11.108509
Epoch: 799      Training Loss: 11.108509
Epoch: 800      Training Loss: 11.108509
Epoch: 801      Training Loss: 11.108507
Epoch: 802      Training Loss: 11.108509
Epoch: 803      Training Loss: 11.108508
Epoch: 804      Training Loss: 11.108508
Epoch: 805      Training Loss: 11.108507
Epoch: 806      Training Loss: 11.108507
Epoch: 807      Training Loss: 11.108507
Epoch: 808      Training Loss: 11.108507
Epoch: 809      Training Loss: 11.108507
Epoch: 810      Training Loss: 11.108507
Epoch: 811      Training Loss: 11.108507
Epoch: 812      Training Loss: 11.108506
Epoch: 813      Training Loss: 11.108506
Epoch: 814      Training Loss: 11.108506
Epoch: 815      Training Loss: 11.108507
Epoch: 816      Training Loss: 11.108506
Epoch: 817      Training Loss: 11.108506
Epoch: 818      Training Loss: 11.108506
```

```
Epoch: 819      Training Loss: 11.108505
Epoch: 820      Training Loss: 11.108504
Epoch: 821      Training Loss: 11.108504
Epoch: 822      Training Loss: 11.108504
Epoch: 823      Training Loss: 11.108504
Epoch: 824      Training Loss: 11.108504
Epoch: 825      Training Loss: 11.108504
Epoch: 826      Training Loss: 11.108504
Epoch: 827      Training Loss: 11.108502
Epoch: 828      Training Loss: 11.108502
Epoch: 829      Training Loss: 11.108502
Epoch: 830      Training Loss: 11.108502
Epoch: 831      Training Loss: 11.108502
Epoch: 832      Training Loss: 11.108501
Epoch: 833      Training Loss: 11.108502
Epoch: 834      Training Loss: 11.108501
Epoch: 835      Training Loss: 11.108501
Epoch: 836      Training Loss: 11.108501
Epoch: 837      Training Loss: 11.108501
Epoch: 838      Training Loss: 11.108501
Epoch: 839      Training Loss: 11.108501
Epoch: 840      Training Loss: 11.108501
Epoch: 841      Training Loss: 11.108500
Epoch: 842      Training Loss: 11.108500
Epoch: 843      Training Loss: 11.108500
Epoch: 844      Training Loss: 11.108501
Epoch: 845      Training Loss: 11.108500
Epoch: 846      Training Loss: 11.108500
Epoch: 847      Training Loss: 11.108500
Epoch: 848      Training Loss: 11.108500
Epoch: 849      Training Loss: 11.108499
Epoch: 850      Training Loss: 11.108499
Epoch: 851      Training Loss: 11.108499
Epoch: 852      Training Loss: 11.108499
Epoch: 853      Training Loss: 11.108499
Epoch: 854      Training Loss: 11.108499
Epoch: 855      Training Loss: 11.108499
Epoch: 856      Training Loss: 11.108499
Epoch: 857      Training Loss: 11.108498
Epoch: 858      Training Loss: 11.108499
Epoch: 859      Training Loss: 11.108500
Epoch: 860      Training Loss: 11.108498
Epoch: 861      Training Loss: 11.108497
Epoch: 862      Training Loss: 11.108498
Epoch: 863      Training Loss: 11.108498
Epoch: 864      Training Loss: 11.108498
Epoch: 865      Training Loss: 11.108498
Epoch: 866      Training Loss: 11.108496
```

```
Epoch: 867      Training Loss: 11.108496
Epoch: 868      Training Loss: 11.108496
Epoch: 869      Training Loss: 11.108496
Epoch: 870      Training Loss: 11.108496
Epoch: 871      Training Loss: 11.108496
Epoch: 872      Training Loss: 11.108496
Epoch: 873      Training Loss: 11.108496
Epoch: 874      Training Loss: 11.108495
Epoch: 875      Training Loss: 11.108496
Epoch: 876      Training Loss: 11.108495
Epoch: 877      Training Loss: 11.108495
Epoch: 878      Training Loss: 11.108494
Epoch: 879      Training Loss: 11.108495
Epoch: 880      Training Loss: 11.108495
Epoch: 881      Training Loss: 11.108495
Epoch: 882      Training Loss: 11.108493
Epoch: 883      Training Loss: 11.108493
Epoch: 884      Training Loss: 11.108493
Epoch: 885      Training Loss: 11.108493
Epoch: 886      Training Loss: 11.108491
Epoch: 887      Training Loss: 11.108491
Epoch: 888      Training Loss: 11.108492
Epoch: 889      Training Loss: 11.108491
Epoch: 890      Training Loss: 11.108491
Epoch: 891      Training Loss: 11.108491
Epoch: 892      Training Loss: 11.108490
Epoch: 893      Training Loss: 11.108490
Epoch: 894      Training Loss: 11.108490
Epoch: 895      Training Loss: 11.108490
Epoch: 896      Training Loss: 11.108489
Epoch: 897      Training Loss: 11.108489
Epoch: 898      Training Loss: 11.108490
Epoch: 899      Training Loss: 11.108490
Epoch: 900      Training Loss: 11.108489
Epoch: 901      Training Loss: 11.108488
Epoch: 902      Training Loss: 11.108489
Epoch: 903      Training Loss: 11.108488
Epoch: 904      Training Loss: 11.108488
Epoch: 905      Training Loss: 11.108488
Epoch: 906      Training Loss: 11.108488
Epoch: 907      Training Loss: 11.108486
Epoch: 908      Training Loss: 11.108486
Epoch: 909      Training Loss: 11.108487
Epoch: 910      Training Loss: 11.108486
Epoch: 911      Training Loss: 11.108486
Epoch: 912      Training Loss: 11.108486
Epoch: 913      Training Loss: 11.108486
Epoch: 914      Training Loss: 11.108484
```

```
Epoch: 915      Training Loss: 11.108485
Epoch: 916      Training Loss: 11.108484
Epoch: 917      Training Loss: 11.108485
Epoch: 918      Training Loss: 11.108485
Epoch: 919      Training Loss: 11.108483
Epoch: 920      Training Loss: 11.108483
Epoch: 921      Training Loss: 11.108483
Epoch: 922      Training Loss: 11.108483
Epoch: 923      Training Loss: 11.108483
Epoch: 924      Training Loss: 11.108483
Epoch: 925      Training Loss: 11.108483
Epoch: 926      Training Loss: 11.108482
Epoch: 927      Training Loss: 11.108482
Epoch: 928      Training Loss: 11.108482
Epoch: 929      Training Loss: 11.108483
Epoch: 930      Training Loss: 11.108482
Epoch: 931      Training Loss: 11.108482
Epoch: 932      Training Loss: 11.108482
Epoch: 933      Training Loss: 11.108482
Epoch: 934      Training Loss: 11.108482
Epoch: 935      Training Loss: 11.108482
Epoch: 936      Training Loss: 11.108481
Epoch: 937      Training Loss: 11.108481
Epoch: 938      Training Loss: 11.108481
Epoch: 939      Training Loss: 11.108481
Epoch: 940      Training Loss: 11.108479
Epoch: 941      Training Loss: 11.108479
Epoch: 942      Training Loss: 11.108479
Epoch: 943      Training Loss: 11.108479
Epoch: 944      Training Loss: 11.108479
Epoch: 945      Training Loss: 11.108479
Epoch: 946      Training Loss: 11.108479
Epoch: 947      Training Loss: 11.108480
Epoch: 948      Training Loss: 11.108478
Epoch: 949      Training Loss: 11.108479
Epoch: 950      Training Loss: 11.108479
Epoch: 951      Training Loss: 11.108478
Epoch: 952      Training Loss: 11.108478
Epoch: 953      Training Loss: 11.108478
Epoch: 954      Training Loss: 11.108478
Epoch: 955      Training Loss: 11.108478
Epoch: 956      Training Loss: 11.108478
Epoch: 957      Training Loss: 11.108478
Epoch: 958      Training Loss: 11.108476
Epoch: 959      Training Loss: 11.108477
Epoch: 960      Training Loss: 11.108477
Epoch: 961      Training Loss: 11.108477
Epoch: 962      Training Loss: 11.108476
```

```
Epoch: 963      Training Loss: 11.108477
Epoch: 964      Training Loss: 11.108476
Epoch: 965      Training Loss: 11.108476
Epoch: 966      Training Loss: 11.108475
Epoch: 967      Training Loss: 11.108476
Epoch: 968      Training Loss: 11.108477
Epoch: 969      Training Loss: 11.108476
Epoch: 970      Training Loss: 11.108477
Epoch: 971      Training Loss: 11.108475
Epoch: 972      Training Loss: 11.108475
Epoch: 973      Training Loss: 11.108475
Epoch: 974      Training Loss: 11.108475
Epoch: 975      Training Loss: 11.108474
Epoch: 976      Training Loss: 11.108475
Epoch: 977      Training Loss: 11.108475
Epoch: 978      Training Loss: 11.108473
Epoch: 979      Training Loss: 11.108473
Epoch: 980      Training Loss: 11.108473
Epoch: 981      Training Loss: 11.108472
Epoch: 982      Training Loss: 11.108472
Epoch: 983      Training Loss: 11.108473
Epoch: 984      Training Loss: 11.108473
Epoch: 985      Training Loss: 11.108472
Epoch: 986      Training Loss: 11.108472
Epoch: 987      Training Loss: 11.108471
Epoch: 988      Training Loss: 11.108471
Epoch: 989      Training Loss: 11.108471
Epoch: 990      Training Loss: 11.108471
Epoch: 991      Training Loss: 11.108470
Epoch: 992      Training Loss: 11.108472
Epoch: 993      Training Loss: 11.108470
Epoch: 994      Training Loss: 11.108469
Epoch: 995      Training Loss: 11.108470
Epoch: 996      Training Loss: 11.108470
Epoch: 997      Training Loss: 11.108470
Epoch: 998      Training Loss: 11.108470
Epoch: 999      Training Loss: 11.108468
Epoch: 1000     Training Loss: 11.108469
Epoch: 1001     Training Loss: 11.108468
Epoch: 1002     Training Loss: 11.108469
Epoch: 1003     Training Loss: 11.108469
Epoch: 1004     Training Loss: 11.108468
Epoch: 1005     Training Loss: 11.108467
Epoch: 1006     Training Loss: 11.108468
Epoch: 1007     Training Loss: 11.108467
Epoch: 1008     Training Loss: 11.108467
Epoch: 1009     Training Loss: 11.108466
Epoch: 1010     Training Loss: 11.108466
```

```
Epoch: 1011     Training Loss: 11.108465
Epoch: 1012     Training Loss: 11.108465
Epoch: 1013     Training Loss: 11.108466
Epoch: 1014     Training Loss: 11.108465
Epoch: 1015     Training Loss: 11.108465
Epoch: 1016     Training Loss: 11.108465
Epoch: 1017     Training Loss: 11.108465
Epoch: 1018     Training Loss: 11.108463
Epoch: 1019     Training Loss: 11.108464
Epoch: 1020     Training Loss: 11.108465
Epoch: 1021     Training Loss: 11.108463
Epoch: 1022     Training Loss: 11.108463
Epoch: 1023     Training Loss: 11.108463
Epoch: 1024     Training Loss: 11.108463
Epoch: 1025     Training Loss: 11.108463
Epoch: 1026     Training Loss: 11.108463
Epoch: 1027     Training Loss: 11.108462
Epoch: 1028     Training Loss: 11.108462
Epoch: 1029     Training Loss: 11.108462
Epoch: 1030     Training Loss: 11.108461
Epoch: 1031     Training Loss: 11.108461
Epoch: 1032     Training Loss: 11.108462
Epoch: 1033     Training Loss: 11.108461
Epoch: 1034     Training Loss: 11.108462
Epoch: 1035     Training Loss: 11.108461
Epoch: 1036     Training Loss: 11.108461
Epoch: 1037     Training Loss: 11.108461
Epoch: 1038     Training Loss: 11.108461
Epoch: 1039     Training Loss: 11.108460
Epoch: 1040     Training Loss: 11.108460
Epoch: 1041     Training Loss: 11.108460
Epoch: 1042     Training Loss: 11.108460
Epoch: 1043     Training Loss: 11.108460
Epoch: 1044     Training Loss: 11.108460
Epoch: 1045     Training Loss: 11.108460
Epoch: 1046     Training Loss: 11.108460
Epoch: 1047     Training Loss: 11.108459
Epoch: 1048     Training Loss: 11.108459
Epoch: 1049     Training Loss: 11.108459
Epoch: 1050     Training Loss: 11.108459
Epoch: 1051     Training Loss: 11.108459
Epoch: 1052     Training Loss: 11.108459
Epoch: 1053     Training Loss: 11.108458
Epoch: 1054     Training Loss: 11.108458
Epoch: 1055     Training Loss: 11.108458
Epoch: 1056     Training Loss: 11.108458
Epoch: 1057     Training Loss: 11.108458
Epoch: 1058     Training Loss: 11.108456
```

```
Epoch: 1059     Training Loss: 11.108456
Epoch: 1060     Training Loss: 11.108456
Epoch: 1061     Training Loss: 11.108456
Epoch: 1062     Training Loss: 11.108456
Epoch: 1063     Training Loss: 11.108455
Epoch: 1064     Training Loss: 11.108455
Epoch: 1065     Training Loss: 11.108455
Epoch: 1066     Training Loss: 11.108455
Epoch: 1067     Training Loss: 11.108455
Epoch: 1068     Training Loss: 11.108455
Epoch: 1069     Training Loss: 11.108455
Epoch: 1070     Training Loss: 11.108454
Epoch: 1071     Training Loss: 11.108454
Epoch: 1072     Training Loss: 11.108454
Epoch: 1073     Training Loss: 11.108454
Epoch: 1074     Training Loss: 11.108455
Epoch: 1075     Training Loss: 11.108454
Epoch: 1076     Training Loss: 11.108452
Epoch: 1077     Training Loss: 11.108452
Epoch: 1078     Training Loss: 11.108454
Epoch: 1079     Training Loss: 11.108452
Epoch: 1080     Training Loss: 11.108452
Epoch: 1081     Training Loss: 11.108452
Epoch: 1082     Training Loss: 11.108452
Epoch: 1083     Training Loss: 11.108451
Epoch: 1084     Training Loss: 11.108451
Epoch: 1085     Training Loss: 11.108451
Epoch: 1086     Training Loss: 11.108451
Epoch: 1087     Training Loss: 11.108451
Epoch: 1088     Training Loss: 11.108451
Epoch: 1089     Training Loss: 11.108451
Epoch: 1090     Training Loss: 11.108450
Epoch: 1091     Training Loss: 11.108450
Epoch: 1092     Training Loss: 11.108450
Epoch: 1093     Training Loss: 11.108449
Epoch: 1094     Training Loss: 11.108449
Epoch: 1095     Training Loss: 11.108449
Epoch: 1096     Training Loss: 11.108449
Epoch: 1097     Training Loss: 11.108448
Epoch: 1098     Training Loss: 11.108448
Epoch: 1099     Training Loss: 11.108448
Epoch: 1100     Training Loss: 11.108448
Epoch: 1101     Training Loss: 11.108448
Epoch: 1102     Training Loss: 11.108448
Epoch: 1103     Training Loss: 11.108448
Epoch: 1104     Training Loss: 11.108448
Epoch: 1105     Training Loss: 11.108447
Epoch: 1106     Training Loss: 11.108447
```

```
Epoch: 1107     Training Loss: 11.108447
Epoch: 1108     Training Loss: 11.108446
Epoch: 1109     Training Loss: 11.108447
Epoch: 1110     Training Loss: 11.108446
Epoch: 1111     Training Loss: 11.108445
Epoch: 1112     Training Loss: 11.108446
Epoch: 1113     Training Loss: 11.108445
Epoch: 1114     Training Loss: 11.108445
Epoch: 1115     Training Loss: 11.108444
Epoch: 1116     Training Loss: 11.108443
Epoch: 1117     Training Loss: 11.108443
Epoch: 1118     Training Loss: 11.108443
Epoch: 1119     Training Loss: 11.108443
Epoch: 1120     Training Loss: 11.108443
Epoch: 1121     Training Loss: 11.108443
Epoch: 1122     Training Loss: 11.108442
Epoch: 1123     Training Loss: 11.108442
Epoch: 1124     Training Loss: 11.108443
Epoch: 1125     Training Loss: 11.108442
Epoch: 1126     Training Loss: 11.108442
Epoch: 1127     Training Loss: 11.108442
Epoch: 1128     Training Loss: 11.108442
Epoch: 1129     Training Loss: 11.108442
Epoch: 1130     Training Loss: 11.108442
Epoch: 1131     Training Loss: 11.108442
Epoch: 1132     Training Loss: 11.108442
Epoch: 1133     Training Loss: 11.108441
Epoch: 1134     Training Loss: 11.108440
Epoch: 1135     Training Loss: 11.108440
Epoch: 1136     Training Loss: 11.108441
Epoch: 1137     Training Loss: 11.108441
Epoch: 1138     Training Loss: 11.108439
Epoch: 1139     Training Loss: 11.108439
Epoch: 1140     Training Loss: 11.108439
Epoch: 1141     Training Loss: 11.108438
Epoch: 1142     Training Loss: 11.108438
Epoch: 1143     Training Loss: 11.108439
Epoch: 1144     Training Loss: 11.108438
Epoch: 1145     Training Loss: 11.108438
Epoch: 1146     Training Loss: 11.108438
Epoch: 1147     Training Loss: 11.108438
Epoch: 1148     Training Loss: 11.108438
Epoch: 1149     Training Loss: 11.108438
Epoch: 1150     Training Loss: 11.108438
Epoch: 1151     Training Loss: 11.108438
Epoch: 1152     Training Loss: 11.108437
Epoch: 1153     Training Loss: 11.108437
Epoch: 1154     Training Loss: 11.108437
```

```
Epoch: 1155     Training Loss: 11.108437
Epoch: 1156     Training Loss: 11.108437
Epoch: 1157     Training Loss: 11.108437
Epoch: 1158     Training Loss: 11.108437
Epoch: 1159     Training Loss: 11.108437
Epoch: 1160     Training Loss: 11.108436
Epoch: 1161     Training Loss: 11.108434
Epoch: 1162     Training Loss: 11.108436
Epoch: 1163     Training Loss: 11.108436
Epoch: 1164     Training Loss: 11.108436
Epoch: 1165     Training Loss: 11.108434
Epoch: 1166     Training Loss: 11.108435
Epoch: 1167     Training Loss: 11.108434
Epoch: 1168     Training Loss: 11.108434
Epoch: 1169     Training Loss: 11.108434
Epoch: 1170     Training Loss: 11.108434
Epoch: 1171     Training Loss: 11.108432
Epoch: 1172     Training Loss: 11.108434
Epoch: 1173     Training Loss: 11.108433
Epoch: 1174     Training Loss: 11.108432
Epoch: 1175     Training Loss: 11.108432
Epoch: 1176     Training Loss: 11.108432
Epoch: 1177     Training Loss: 11.108432
Epoch: 1178     Training Loss: 11.108432
Epoch: 1179     Training Loss: 11.108432
Epoch: 1180     Training Loss: 11.108432
Epoch: 1181     Training Loss: 11.108431
Epoch: 1182     Training Loss: 11.108430
Epoch: 1183     Training Loss: 11.108431
Epoch: 1184     Training Loss: 11.108431
Epoch: 1185     Training Loss: 11.108431
Epoch: 1186     Training Loss: 11.108431
Epoch: 1187     Training Loss: 11.108431
Epoch: 1188     Training Loss: 11.108430
Epoch: 1189     Training Loss: 11.108430
Epoch: 1190     Training Loss: 11.108429
Epoch: 1191     Training Loss: 11.108430
Epoch: 1192     Training Loss: 11.108429
Epoch: 1193     Training Loss: 11.108429
Epoch: 1194     Training Loss: 11.108429
Epoch: 1195     Training Loss: 11.108428
Epoch: 1196     Training Loss: 11.108428
Epoch: 1197     Training Loss: 11.108427
Epoch: 1198     Training Loss: 11.108427
Epoch: 1199     Training Loss: 11.108427
Epoch: 1200     Training Loss: 11.108427
Epoch: 1201     Training Loss: 11.108427
Epoch: 1202     Training Loss: 11.108427
```

```
Epoch: 1203     Training Loss: 11.108426
Epoch: 1204     Training Loss: 11.108425
Epoch: 1205     Training Loss: 11.108425
Epoch: 1206     Training Loss: 11.108425
Epoch: 1207     Training Loss: 11.108425
Epoch: 1208     Training Loss: 11.108423
Epoch: 1209     Training Loss: 11.108425
Epoch: 1210     Training Loss: 11.108423
Epoch: 1211     Training Loss: 11.108425
Epoch: 1212     Training Loss: 11.108424
Epoch: 1213     Training Loss: 11.108425
Epoch: 1214     Training Loss: 11.108423
Epoch: 1215     Training Loss: 11.108423
Epoch: 1216     Training Loss: 11.108423
Epoch: 1217     Training Loss: 11.108423
Epoch: 1218     Training Loss: 11.108423
Epoch: 1219     Training Loss: 11.108423
Epoch: 1220     Training Loss: 11.108423
Epoch: 1221     Training Loss: 11.108421
Epoch: 1222     Training Loss: 11.108421
Epoch: 1223     Training Loss: 11.108421
Epoch: 1224     Training Loss: 11.108421
Epoch: 1225     Training Loss: 11.108421
Epoch: 1226     Training Loss: 11.108421
Epoch: 1227     Training Loss: 11.108421
Epoch: 1228     Training Loss: 11.108419
Epoch: 1229     Training Loss: 11.108419
Epoch: 1230     Training Loss: 11.108419
Epoch: 1231     Training Loss: 11.108419
Epoch: 1232     Training Loss: 11.108419
Epoch: 1233     Training Loss: 11.108419
Epoch: 1234     Training Loss: 11.108419
Epoch: 1235     Training Loss: 11.108419
Epoch: 1236     Training Loss: 11.108418
Epoch: 1237     Training Loss: 11.108418
Epoch: 1238     Training Loss: 11.108418
Epoch: 1239     Training Loss: 11.108418
Epoch: 1240     Training Loss: 11.108418
Epoch: 1241     Training Loss: 11.108417
Epoch: 1242     Training Loss: 11.108417
Epoch: 1243     Training Loss: 11.108417
Epoch: 1244     Training Loss: 11.108417
Epoch: 1245     Training Loss: 11.108417
Epoch: 1246     Training Loss: 11.108417
Epoch: 1247     Training Loss: 11.108416
Epoch: 1248     Training Loss: 11.108417
Epoch: 1249     Training Loss: 11.108417
Epoch: 1250     Training Loss: 11.108415
```

```
Epoch: 1251     Training Loss: 11.108414
Epoch: 1252     Training Loss: 11.108414
Epoch: 1253     Training Loss: 11.108415
Epoch: 1254     Training Loss: 11.108414
Epoch: 1255     Training Loss: 11.108415
Epoch: 1256     Training Loss: 11.108415
Epoch: 1257     Training Loss: 11.108414
Epoch: 1258     Training Loss: 11.108414
Epoch: 1259     Training Loss: 11.108414
Epoch: 1260     Training Loss: 11.108414
Epoch: 1261     Training Loss: 11.108414
Epoch: 1262     Training Loss: 11.108414
Epoch: 1263     Training Loss: 11.108414
Epoch: 1264     Training Loss: 11.108414
Epoch: 1265     Training Loss: 11.108412
Epoch: 1266     Training Loss: 11.108412
Epoch: 1267     Training Loss: 11.108412
Epoch: 1268     Training Loss: 11.108412
Epoch: 1269     Training Loss: 11.108412
Epoch: 1270     Training Loss: 11.108412
Epoch: 1271     Training Loss: 11.108412
Epoch: 1272     Training Loss: 11.108411
Epoch: 1273     Training Loss: 11.108410
Epoch: 1274     Training Loss: 11.108410
Epoch: 1275     Training Loss: 11.108410
Epoch: 1276     Training Loss: 11.108410
Epoch: 1277     Training Loss: 11.108409
Epoch: 1278     Training Loss: 11.108409
Epoch: 1279     Training Loss: 11.108410
Epoch: 1280     Training Loss: 11.108409
Epoch: 1281     Training Loss: 11.108409
Epoch: 1282     Training Loss: 11.108407
Epoch: 1283     Training Loss: 11.108408
Epoch: 1284     Training Loss: 11.108408
Epoch: 1285     Training Loss: 11.108409
Epoch: 1286     Training Loss: 11.108409
Epoch: 1287     Training Loss: 11.108409
Epoch: 1288     Training Loss: 11.108409
Epoch: 1289     Training Loss: 11.108407
Epoch: 1290     Training Loss: 11.108407
Epoch: 1291     Training Loss: 11.108407
Epoch: 1292     Training Loss: 11.108407
Epoch: 1293     Training Loss: 11.108405
Epoch: 1294     Training Loss: 11.108406
Epoch: 1295     Training Loss: 11.108407
Epoch: 1296     Training Loss: 11.108406
Epoch: 1297     Training Loss: 11.108406
Epoch: 1298     Training Loss: 11.108404
```

```
Epoch: 1299     Training Loss: 11.108404
Epoch: 1300     Training Loss: 11.108405
Epoch: 1301     Training Loss: 11.108404
Epoch: 1302     Training Loss: 11.108405
Epoch: 1303     Training Loss: 11.108405
Epoch: 1304     Training Loss: 11.108404
Epoch: 1305     Training Loss: 11.108403
Epoch: 1306     Training Loss: 11.108404
Epoch: 1307     Training Loss: 11.108404
Epoch: 1308     Training Loss: 11.108403
Epoch: 1309     Training Loss: 11.108403
Epoch: 1310     Training Loss: 11.108403
Epoch: 1311     Training Loss: 11.108403
Epoch: 1312     Training Loss: 11.108403
Epoch: 1313     Training Loss: 11.108403
Epoch: 1314     Training Loss: 11.108403
Epoch: 1315     Training Loss: 11.108402
Epoch: 1316     Training Loss: 11.108402
Epoch: 1317     Training Loss: 11.108402
Epoch: 1318     Training Loss: 11.108401
Epoch: 1319     Training Loss: 11.108401
Epoch: 1320     Training Loss: 11.108401
Epoch: 1321     Training Loss: 11.108400
Epoch: 1322     Training Loss: 11.108401
Epoch: 1323     Training Loss: 11.108400
Epoch: 1324     Training Loss: 11.108400
Epoch: 1325     Training Loss: 11.108400
Epoch: 1326     Training Loss: 11.108400
Epoch: 1327     Training Loss: 11.108399
Epoch: 1328     Training Loss: 11.108400
Epoch: 1329     Training Loss: 11.108399
Epoch: 1330     Training Loss: 11.108399
Epoch: 1331     Training Loss: 11.108398
Epoch: 1332     Training Loss: 11.108397
Epoch: 1333     Training Loss: 11.108398
Epoch: 1334     Training Loss: 11.108398
Epoch: 1335     Training Loss: 11.108397
Epoch: 1336     Training Loss: 11.108398
Epoch: 1337     Training Loss: 11.108397
Epoch: 1338     Training Loss: 11.108398
Epoch: 1339     Training Loss: 11.108397
Epoch: 1340     Training Loss: 11.108397
Epoch: 1341     Training Loss: 11.108397
Epoch: 1342     Training Loss: 11.108396
Epoch: 1343     Training Loss: 11.108396
Epoch: 1344     Training Loss: 11.108396
Epoch: 1345     Training Loss: 11.108396
Epoch: 1346     Training Loss: 11.108396
```

```
Epoch: 1347	Training Loss: 11.108396
Epoch: 1348	Training Loss: 11.108396
Epoch: 1349	Training Loss: 11.108396
Epoch: 1350	Training Loss: 11.108396
Epoch: 1351	Training Loss: 11.108396
Epoch: 1352	Training Loss: 11.108396
Epoch: 1353	Training Loss: 11.108395
Epoch: 1354	Training Loss: 11.108395
Epoch: 1355	Training Loss: 11.108395
Epoch: 1356	Training Loss: 11.108393
Epoch: 1357	Training Loss: 11.108393
Epoch: 1358	Training Loss: 11.108394
Epoch: 1359	Training Loss: 11.108393
Epoch: 1360	Training Loss: 11.108394
Epoch: 1361	Training Loss: 11.108393
Epoch: 1362	Training Loss: 11.108392
Epoch: 1363	Training Loss: 11.108390
Epoch: 1364	Training Loss: 11.108392
Epoch: 1365	Training Loss: 11.108391
Epoch: 1366	Training Loss: 11.108391
Epoch: 1367	Training Loss: 11.108390
Epoch: 1368	Training Loss: 11.108390
Epoch: 1369	Training Loss: 11.108390
Epoch: 1370	Training Loss: 11.108390
Epoch: 1371	Training Loss: 11.108390
Epoch: 1372	Training Loss: 11.108390
Epoch: 1373	Training Loss: 11.108390
Epoch: 1374	Training Loss: 11.108390
Epoch: 1375	Training Loss: 11.108390
Epoch: 1376	Training Loss: 11.108390
Epoch: 1377	Training Loss: 11.108390
Epoch: 1378	Training Loss: 11.108390
Epoch: 1379	Training Loss: 11.108389
Epoch: 1380	Training Loss: 11.108389
Epoch: 1381	Training Loss: 11.108388
Epoch: 1382	Training Loss: 11.108388
Epoch: 1383	Training Loss: 11.108388
Epoch: 1384	Training Loss: 11.108386
Epoch: 1385	Training Loss: 11.108387
Epoch: 1386	Training Loss: 11.108387
Epoch: 1387	Training Loss: 11.108387
Epoch: 1388	Training Loss: 11.108385
Epoch: 1389	Training Loss: 11.108386
Epoch: 1390	Training Loss: 11.108385
Epoch: 1391	Training Loss: 11.108385
Epoch: 1392	Training Loss: 11.108385
Epoch: 1393	Training Loss: 11.108385
Epoch: 1394	Training Loss: 11.108385
```

```
Epoch: 1395     Training Loss: 11.108385
Epoch: 1396     Training Loss: 11.108385
Epoch: 1397     Training Loss: 11.108384
Epoch: 1398     Training Loss: 11.108384
Epoch: 1399     Training Loss: 11.108384
Epoch: 1400     Training Loss: 11.108384
Epoch: 1401     Training Loss: 11.108384
Epoch: 1402     Training Loss: 11.108384
Epoch: 1403     Training Loss: 11.108384
Epoch: 1404     Training Loss: 11.108384
Epoch: 1405     Training Loss: 11.108383
Epoch: 1406     Training Loss: 11.108382
Epoch: 1407     Training Loss: 11.108382
Epoch: 1408     Training Loss: 11.108382
Epoch: 1409     Training Loss: 11.108381
Epoch: 1410     Training Loss: 11.108381
Epoch: 1411     Training Loss: 11.108380
Epoch: 1412     Training Loss: 11.108380
Epoch: 1413     Training Loss: 11.108379
Epoch: 1414     Training Loss: 11.108379
Epoch: 1415     Training Loss: 11.108379
Epoch: 1416     Training Loss: 11.108380
Epoch: 1417     Training Loss: 11.108378
Epoch: 1418     Training Loss: 11.108378
Epoch: 1419     Training Loss: 11.108378
Epoch: 1420     Training Loss: 11.108379
Epoch: 1421     Training Loss: 11.108378
Epoch: 1422     Training Loss: 11.108378
Epoch: 1423     Training Loss: 11.108378
Epoch: 1424     Training Loss: 11.108378
Epoch: 1425     Training Loss: 11.108378
Epoch: 1426     Training Loss: 11.108378
Epoch: 1427     Training Loss: 11.108377
Epoch: 1428     Training Loss: 11.108377
Epoch: 1429     Training Loss: 11.108377
Epoch: 1430     Training Loss: 11.108376
Epoch: 1431     Training Loss: 11.108376
Epoch: 1432     Training Loss: 11.108376
Epoch: 1433     Training Loss: 11.108375
Epoch: 1434     Training Loss: 11.108377
Epoch: 1435     Training Loss: 11.108375
Epoch: 1436     Training Loss: 11.108376
Epoch: 1437     Training Loss: 11.108375
Epoch: 1438     Training Loss: 11.108374
Epoch: 1439     Training Loss: 11.108375
Epoch: 1440     Training Loss: 11.108375
Epoch: 1441     Training Loss: 11.108375
Epoch: 1442     Training Loss: 11.108375
```

```
Epoch: 1443     Training Loss: 11.108374
Epoch: 1444     Training Loss: 11.108374
Epoch: 1445     Training Loss: 11.108374
Epoch: 1446     Training Loss: 11.108374
Epoch: 1447     Training Loss: 11.108374
Epoch: 1448     Training Loss: 11.108373
Epoch: 1449     Training Loss: 11.108373
Epoch: 1450     Training Loss: 11.108373
Epoch: 1451     Training Loss: 11.108372
Epoch: 1452     Training Loss: 11.108371
Epoch: 1453     Training Loss: 11.108372
Epoch: 1454     Training Loss: 11.108372
Epoch: 1455     Training Loss: 11.108372
Epoch: 1456     Training Loss: 11.108371
Epoch: 1457     Training Loss: 11.108371
Epoch: 1458     Training Loss: 11.108371
Epoch: 1459     Training Loss: 11.108370
Epoch: 1460     Training Loss: 11.108371
Epoch: 1461     Training Loss: 11.108371
Epoch: 1462     Training Loss: 11.108370
Epoch: 1463     Training Loss: 11.108370
Epoch: 1464     Training Loss: 11.108370
Epoch: 1465     Training Loss: 11.108370
Epoch: 1466     Training Loss: 11.108369
Epoch: 1467     Training Loss: 11.108370
Epoch: 1468     Training Loss: 11.108369
Epoch: 1469     Training Loss: 11.108369
Epoch: 1470     Training Loss: 11.108369
Epoch: 1471     Training Loss: 11.108369
Epoch: 1472     Training Loss: 11.108368
Epoch: 1473     Training Loss: 11.108367
Epoch: 1474     Training Loss: 11.108367
Epoch: 1475     Training Loss: 11.108366
Epoch: 1476     Training Loss: 11.108366
Epoch: 1477     Training Loss: 11.108366
Epoch: 1478     Training Loss: 11.108366
Epoch: 1479     Training Loss: 11.108366
Epoch: 1480     Training Loss: 11.108366
Epoch: 1481     Training Loss: 11.108366
Epoch: 1482     Training Loss: 11.108366
Epoch: 1483     Training Loss: 11.108366
Epoch: 1484     Training Loss: 11.108366
Epoch: 1485     Training Loss: 11.108366
Epoch: 1486     Training Loss: 11.108366
Epoch: 1487     Training Loss: 11.108366
Epoch: 1488     Training Loss: 11.108366
Epoch: 1489     Training Loss: 11.108366
Epoch: 1490     Training Loss: 11.108366
```

```
Epoch: 1491     Training Loss: 11.108366
Epoch: 1492     Training Loss: 11.108365
Epoch: 1493     Training Loss: 11.108364
Epoch: 1494     Training Loss: 11.108362
Epoch: 1495     Training Loss: 11.108363
Epoch: 1496     Training Loss: 11.108363
Epoch: 1497     Training Loss: 11.108363
Epoch: 1498     Training Loss: 11.108363
Epoch: 1499     Training Loss: 11.108363
Epoch: 1500     Training Loss: 11.108362
Epoch: 1501     Training Loss: 11.108362
Epoch: 1502     Training Loss: 11.108362
Epoch: 1503     Training Loss: 11.108360
Epoch: 1504     Training Loss: 11.108361
Epoch: 1505     Training Loss: 11.108360
Epoch: 1506     Training Loss: 11.108360
Epoch: 1507     Training Loss: 11.108361
Epoch: 1508     Training Loss: 11.108360
Epoch: 1509     Training Loss: 11.108360
Epoch: 1510     Training Loss: 11.108360
Epoch: 1511     Training Loss: 11.108360
Epoch: 1512     Training Loss: 11.108360
Epoch: 1513     Training Loss: 11.108359
Epoch: 1514     Training Loss: 11.108359
Epoch: 1515     Training Loss: 11.108357
Epoch: 1516     Training Loss: 11.108358
Epoch: 1517     Training Loss: 11.108358
Epoch: 1518     Training Loss: 11.108358
Epoch: 1519     Training Loss: 11.108358
Epoch: 1520     Training Loss: 11.108358
Epoch: 1521     Training Loss: 11.108356
Epoch: 1522     Training Loss: 11.108356
Epoch: 1523     Training Loss: 11.108356
Epoch: 1524     Training Loss: 11.108356
Epoch: 1525     Training Loss: 11.108356
Epoch: 1526     Training Loss: 11.108356
Epoch: 1527     Training Loss: 11.108356
Epoch: 1528     Training Loss: 11.108356
Epoch: 1529     Training Loss: 11.108355
Epoch: 1530     Training Loss: 11.108355
Epoch: 1531     Training Loss: 11.108355
Epoch: 1532     Training Loss: 11.108355
Epoch: 1533     Training Loss: 11.108355
Epoch: 1534     Training Loss: 11.108355
Epoch: 1535     Training Loss: 11.108355
Epoch: 1536     Training Loss: 11.108353
Epoch: 1537     Training Loss: 11.108353
Epoch: 1538     Training Loss: 11.108352
```

```
Epoch: 1539     Training Loss: 11.108352
Epoch: 1540     Training Loss: 11.108352
Epoch: 1541     Training Loss: 11.108352
Epoch: 1542     Training Loss: 11.108352
Epoch: 1543     Training Loss: 11.108352
Epoch: 1544     Training Loss: 11.108351
Epoch: 1545     Training Loss: 11.108352
Epoch: 1546     Training Loss: 11.108351
Epoch: 1547     Training Loss: 11.108351
Epoch: 1548     Training Loss: 11.108350
Epoch: 1549     Training Loss: 11.108350
Epoch: 1550     Training Loss: 11.108350
Epoch: 1551     Training Loss: 11.108350
Epoch: 1552     Training Loss: 11.108350
Epoch: 1553     Training Loss: 11.108350
Epoch: 1554     Training Loss: 11.108350
Epoch: 1555     Training Loss: 11.108350
Epoch: 1556     Training Loss: 11.108349
Epoch: 1557     Training Loss: 11.108349
Epoch: 1558     Training Loss: 11.108349
Epoch: 1559     Training Loss: 11.108348
Epoch: 1560     Training Loss: 11.108348
Epoch: 1561     Training Loss: 11.108348
Epoch: 1562     Training Loss: 11.108348
Epoch: 1563     Training Loss: 11.108348
Epoch: 1564     Training Loss: 11.108348
Epoch: 1565     Training Loss: 11.108348
Epoch: 1566     Training Loss: 11.108347
Epoch: 1567     Training Loss: 11.108347
Epoch: 1568     Training Loss: 11.108346
Epoch: 1569     Training Loss: 11.108346
Epoch: 1570     Training Loss: 11.108345
Epoch: 1571     Training Loss: 11.108345
Epoch: 1572     Training Loss: 11.108345
Epoch: 1573     Training Loss: 11.108345
Epoch: 1574     Training Loss: 11.108345
Epoch: 1575     Training Loss: 11.108345
Epoch: 1576     Training Loss: 11.108344
Epoch: 1577     Training Loss: 11.108344
Epoch: 1578     Training Loss: 11.108343
Epoch: 1579     Training Loss: 11.108343
Epoch: 1580     Training Loss: 11.108343
Epoch: 1581     Training Loss: 11.108343
Epoch: 1582     Training Loss: 11.108343
Epoch: 1583     Training Loss: 11.108343
Epoch: 1584     Training Loss: 11.108343
Epoch: 1585     Training Loss: 11.108343
Epoch: 1586     Training Loss: 11.108343
```

```
Epoch: 1587     Training Loss: 11.108342
Epoch: 1588     Training Loss: 11.108343
Epoch: 1589     Training Loss: 11.108342
Epoch: 1590     Training Loss: 11.108342
Epoch: 1591     Training Loss: 11.108342
Epoch: 1592     Training Loss: 11.108342
Epoch: 1593     Training Loss: 11.108342
Epoch: 1594     Training Loss: 11.108341
Epoch: 1595     Training Loss: 11.108342
Epoch: 1596     Training Loss: 11.108341
Epoch: 1597     Training Loss: 11.108341
Epoch: 1598     Training Loss: 11.108340
Epoch: 1599     Training Loss: 11.108339
Epoch: 1600     Training Loss: 11.108342
Epoch: 1601     Training Loss: 11.108339
Epoch: 1602     Training Loss: 11.108340
Epoch: 1603     Training Loss: 11.108340
Epoch: 1604     Training Loss: 11.108340
Epoch: 1605     Training Loss: 11.108340
Epoch: 1606     Training Loss: 11.108340
Epoch: 1607     Training Loss: 11.108340
Epoch: 1608     Training Loss: 11.108340
Epoch: 1609     Training Loss: 11.108338
Epoch: 1610     Training Loss: 11.108338
Epoch: 1611     Training Loss: 11.108338
Epoch: 1612     Training Loss: 11.108338
Epoch: 1613     Training Loss: 11.108338
Epoch: 1614     Training Loss: 11.108338
Epoch: 1615     Training Loss: 11.108338
Epoch: 1616     Training Loss: 11.108337
Epoch: 1617     Training Loss: 11.108338
Epoch: 1618     Training Loss: 11.108338
Epoch: 1619     Training Loss: 11.108337
Epoch: 1620     Training Loss: 11.108337
Epoch: 1621     Training Loss: 11.108336
Epoch: 1622     Training Loss: 11.108336
Epoch: 1623     Training Loss: 11.108336
Epoch: 1624     Training Loss: 11.108336
Epoch: 1625     Training Loss: 11.108336
Epoch: 1626     Training Loss: 11.108335
Epoch: 1627     Training Loss: 11.108336
Epoch: 1628     Training Loss: 11.108335
Epoch: 1629     Training Loss: 11.108335
Epoch: 1630     Training Loss: 11.108335
Epoch: 1631     Training Loss: 11.108335
Epoch: 1632     Training Loss: 11.108335
Epoch: 1633     Training Loss: 11.108335
Epoch: 1634     Training Loss: 11.108335
```

```
Epoch: 1635     Training Loss: 11.108334
Epoch: 1636     Training Loss: 11.108333
Epoch: 1637     Training Loss: 11.108333
Epoch: 1638     Training Loss: 11.108333
Epoch: 1639     Training Loss: 11.108334
Epoch: 1640     Training Loss: 11.108333
Epoch: 1641     Training Loss: 11.108333
Epoch: 1642     Training Loss: 11.108333
Epoch: 1643     Training Loss: 11.108331
Epoch: 1644     Training Loss: 11.108331
Epoch: 1645     Training Loss: 11.108331
Epoch: 1646     Training Loss: 11.108331
Epoch: 1647     Training Loss: 11.108331
Epoch: 1648     Training Loss: 11.108331
Epoch: 1649     Training Loss: 11.108331
Epoch: 1650     Training Loss: 11.108331
Epoch: 1651     Training Loss: 11.108331
Epoch: 1652     Training Loss: 11.108331
Epoch: 1653     Training Loss: 11.108331
Epoch: 1654     Training Loss: 11.108331
Epoch: 1655     Training Loss: 11.108330
Epoch: 1656     Training Loss: 11.108330
Epoch: 1657     Training Loss: 11.108330
Epoch: 1658     Training Loss: 11.108330
Epoch: 1659     Training Loss: 11.108329
Epoch: 1660     Training Loss: 11.108329
Epoch: 1661     Training Loss: 11.108328
Epoch: 1662     Training Loss: 11.108328
Epoch: 1663     Training Loss: 11.108328
Epoch: 1664     Training Loss: 11.108328
Epoch: 1665     Training Loss: 11.108327
Epoch: 1666     Training Loss: 11.108326
Epoch: 1667     Training Loss: 11.108326
Epoch: 1668     Training Loss: 11.108326
Epoch: 1669     Training Loss: 11.108326
Epoch: 1670     Training Loss: 11.108325
Epoch: 1671     Training Loss: 11.108325
Epoch: 1672     Training Loss: 11.108325
Epoch: 1673     Training Loss: 11.108325
Epoch: 1674     Training Loss: 11.108325
Epoch: 1675     Training Loss: 11.108325
Epoch: 1676     Training Loss: 11.108325
Epoch: 1677     Training Loss: 11.108325
Epoch: 1678     Training Loss: 11.108324
Epoch: 1679     Training Loss: 11.108322
Epoch: 1680     Training Loss: 11.108324
Epoch: 1681     Training Loss: 11.108322
Epoch: 1682     Training Loss: 11.108322
```

```
Epoch: 1683     Training Loss: 11.108322
Epoch: 1684     Training Loss: 11.108322
Epoch: 1685     Training Loss: 11.108322
Epoch: 1686     Training Loss: 11.108322
Epoch: 1687     Training Loss: 11.108321
Epoch: 1688     Training Loss: 11.108321
Epoch: 1689     Training Loss: 11.108321
Epoch: 1690     Training Loss: 11.108320
Epoch: 1691     Training Loss: 11.108320
Epoch: 1692     Training Loss: 11.108319
Epoch: 1693     Training Loss: 11.108319
Epoch: 1694     Training Loss: 11.108319
Epoch: 1695     Training Loss: 11.108319
Epoch: 1696     Training Loss: 11.108319
Epoch: 1697     Training Loss: 11.108319
Epoch: 1698     Training Loss: 11.108318
Epoch: 1699     Training Loss: 11.108318
Epoch: 1700     Training Loss: 11.108318
Epoch: 1701     Training Loss: 11.108318
Epoch: 1702     Training Loss: 11.108318
Epoch: 1703     Training Loss: 11.108317
Epoch: 1704     Training Loss: 11.108316
Epoch: 1705     Training Loss: 11.108316
Epoch: 1706     Training Loss: 11.108316
Epoch: 1707     Training Loss: 11.108316
Epoch: 1708     Training Loss: 11.108315
Epoch: 1709     Training Loss: 11.108315
Epoch: 1710     Training Loss: 11.108315
Epoch: 1711     Training Loss: 11.108315
Epoch: 1712     Training Loss: 11.108315
Epoch: 1713     Training Loss: 11.108315
Epoch: 1714     Training Loss: 11.108315
Epoch: 1715     Training Loss: 11.108315
Epoch: 1716     Training Loss: 11.108315
Epoch: 1717     Training Loss: 11.108315
Epoch: 1718     Training Loss: 11.108315
Epoch: 1719     Training Loss: 11.108315
Epoch: 1720     Training Loss: 11.108315
Epoch: 1721     Training Loss: 11.108315
Epoch: 1722     Training Loss: 11.108314
Epoch: 1723     Training Loss: 11.108314
Epoch: 1724     Training Loss: 11.108314
Epoch: 1725     Training Loss: 11.108315
Epoch: 1726     Training Loss: 11.108314
Epoch: 1727     Training Loss: 11.108314
Epoch: 1728     Training Loss: 11.108314
Epoch: 1729     Training Loss: 11.108314
Epoch: 1730     Training Loss: 11.108314
```

```
Epoch: 1731     Training Loss: 11.108314
Epoch: 1732     Training Loss: 11.108311
Epoch: 1733     Training Loss: 11.108313
Epoch: 1734     Training Loss: 11.108310
Epoch: 1735     Training Loss: 11.108311
Epoch: 1736     Training Loss: 11.108311
Epoch: 1737     Training Loss: 11.108311
Epoch: 1738     Training Loss: 11.108311
Epoch: 1739     Training Loss: 11.108311
Epoch: 1740     Training Loss: 11.108310
Epoch: 1741     Training Loss: 11.108309
Epoch: 1742     Training Loss: 11.108309
Epoch: 1743     Training Loss: 11.108309
Epoch: 1744     Training Loss: 11.108309
Epoch: 1745     Training Loss: 11.108309
Epoch: 1746     Training Loss: 11.108309
Epoch: 1747     Training Loss: 11.108308
Epoch: 1748     Training Loss: 11.108309
Epoch: 1749     Training Loss: 11.108308
Epoch: 1750     Training Loss: 11.108308
Epoch: 1751     Training Loss: 11.108308
Epoch: 1752     Training Loss: 11.108308
Epoch: 1753     Training Loss: 11.108307
Epoch: 1754     Training Loss: 11.108307
Epoch: 1755     Training Loss: 11.108307
Epoch: 1756     Training Loss: 11.108307
Epoch: 1757     Training Loss: 11.108306
Epoch: 1758     Training Loss: 11.108306
Epoch: 1759     Training Loss: 11.108306
Epoch: 1760     Training Loss: 11.108306
Epoch: 1761     Training Loss: 11.108306
Epoch: 1762     Training Loss: 11.108305
Epoch: 1763     Training Loss: 11.108304
Epoch: 1764     Training Loss: 11.108304
Epoch: 1765     Training Loss: 11.108304
Epoch: 1766     Training Loss: 11.108305
Epoch: 1767     Training Loss: 11.108305
Epoch: 1768     Training Loss: 11.108305
Epoch: 1769     Training Loss: 11.108304
Epoch: 1770     Training Loss: 11.108304
Epoch: 1771     Training Loss: 11.108303
Epoch: 1772     Training Loss: 11.108303
Epoch: 1773     Training Loss: 11.108303
Epoch: 1774     Training Loss: 11.108303
Epoch: 1775     Training Loss: 11.108302
Epoch: 1776     Training Loss: 11.108302
Epoch: 1777     Training Loss: 11.108302
Epoch: 1778     Training Loss: 11.108302
```

```
Epoch: 1779     Training Loss: 11.108302
Epoch: 1780     Training Loss: 11.108302
Epoch: 1781     Training Loss: 11.108301
Epoch: 1782     Training Loss: 11.108301
Epoch: 1783     Training Loss: 11.108301
Epoch: 1784     Training Loss: 11.108301
Epoch: 1785     Training Loss: 11.108301
Epoch: 1786     Training Loss: 11.108300
Epoch: 1787     Training Loss: 11.108301
Epoch: 1788     Training Loss: 11.108300
Epoch: 1789     Training Loss: 11.108300
Epoch: 1790     Training Loss: 11.108300
Epoch: 1791     Training Loss: 11.108299
Epoch: 1792     Training Loss: 11.108299
Epoch: 1793     Training Loss: 11.108298
Epoch: 1794     Training Loss: 11.108298
Epoch: 1795     Training Loss: 11.108298
Epoch: 1796     Training Loss: 11.108298
Epoch: 1797     Training Loss: 11.108298
Epoch: 1798     Training Loss: 11.108297
Epoch: 1799     Training Loss: 11.108296
Epoch: 1800     Training Loss: 11.108296
Epoch: 1801     Training Loss: 11.108296
Epoch: 1802     Training Loss: 11.108296
Epoch: 1803     Training Loss: 11.108295
Epoch: 1804     Training Loss: 11.108296
Epoch: 1805     Training Loss: 11.108295
Epoch: 1806     Training Loss: 11.108295
Epoch: 1807     Training Loss: 11.108295
Epoch: 1808     Training Loss: 11.108295
Epoch: 1809     Training Loss: 11.108295
Epoch: 1810     Training Loss: 11.108295
Epoch: 1811     Training Loss: 11.108296
Epoch: 1812     Training Loss: 11.108294
Epoch: 1813     Training Loss: 11.108294
Epoch: 1814     Training Loss: 11.108294
Epoch: 1815     Training Loss: 11.108294
Epoch: 1816     Training Loss: 11.108292
Epoch: 1817     Training Loss: 11.108294
Epoch: 1818     Training Loss: 11.108292
Epoch: 1819     Training Loss: 11.108292
Epoch: 1820     Training Loss: 11.108291
Epoch: 1821     Training Loss: 11.108291
Epoch: 1822     Training Loss: 11.108291
Epoch: 1823     Training Loss: 11.108291
Epoch: 1824     Training Loss: 11.108291
Epoch: 1825     Training Loss: 11.108291
Epoch: 1826     Training Loss: 11.108292
```

```
Epoch: 1827	Training Loss: 11.108290
Epoch: 1828	Training Loss: 11.108291
Epoch: 1829	Training Loss: 11.108290
Epoch: 1830	Training Loss: 11.108290
Epoch: 1831	Training Loss: 11.108291
Epoch: 1832	Training Loss: 11.108290
Epoch: 1833	Training Loss: 11.108290
Epoch: 1834	Training Loss: 11.108289
Epoch: 1835	Training Loss: 11.108288
Epoch: 1836	Training Loss: 11.108290
Epoch: 1837	Training Loss: 11.108289
Epoch: 1838	Training Loss: 11.108288
Epoch: 1839	Training Loss: 11.108288
Epoch: 1840	Training Loss: 11.108287
Epoch: 1841	Training Loss: 11.108287
Epoch: 1842	Training Loss: 11.108287
Epoch: 1843	Training Loss: 11.108287
Epoch: 1844	Training Loss: 11.108288
Epoch: 1845	Training Loss: 11.108287
Epoch: 1846	Training Loss: 11.108287
Epoch: 1847	Training Loss: 11.108287
Epoch: 1848	Training Loss: 11.108287
Epoch: 1849	Training Loss: 11.108286
Epoch: 1850	Training Loss: 11.108286
Epoch: 1851	Training Loss: 11.108286
Epoch: 1852	Training Loss: 11.108285
Epoch: 1853	Training Loss: 11.108286
Epoch: 1854	Training Loss: 11.108284
Epoch: 1855	Training Loss: 11.108284
Epoch: 1856	Training Loss: 11.108284
Epoch: 1857	Training Loss: 11.108284
Epoch: 1858	Training Loss: 11.108284
Epoch: 1859	Training Loss: 11.108284
Epoch: 1860	Training Loss: 11.108284
Epoch: 1861	Training Loss: 11.108284
Epoch: 1862	Training Loss: 11.108284
Epoch: 1863	Training Loss: 11.108283
Epoch: 1864	Training Loss: 11.108284
Epoch: 1865	Training Loss: 11.108282
Epoch: 1866	Training Loss: 11.108284
Epoch: 1867	Training Loss: 11.108283
Epoch: 1868	Training Loss: 11.108283
Epoch: 1869	Training Loss: 11.108282
Epoch: 1870	Training Loss: 11.108283
Epoch: 1871	Training Loss: 11.108283
Epoch: 1872	Training Loss: 11.108282
Epoch: 1873	Training Loss: 11.108282
Epoch: 1874	Training Loss: 11.108283
```

```
Epoch: 1875     Training Loss: 11.108282
Epoch: 1876     Training Loss: 11.108282
Epoch: 1877     Training Loss: 11.108282
Epoch: 1878     Training Loss: 11.108282
Epoch: 1879     Training Loss: 11.108282
Epoch: 1880     Training Loss: 11.108281
Epoch: 1881     Training Loss: 11.108280
Epoch: 1882     Training Loss: 11.108280
Epoch: 1883     Training Loss: 11.108279
Epoch: 1884     Training Loss: 11.108278
Epoch: 1885     Training Loss: 11.108278
Epoch: 1886     Training Loss: 11.108278
Epoch: 1887     Training Loss: 11.108278
Epoch: 1888     Training Loss: 11.108278
Epoch: 1889     Training Loss: 11.108278
Epoch: 1890     Training Loss: 11.108278
Epoch: 1891     Training Loss: 11.108278
Epoch: 1892     Training Loss: 11.108278
Epoch: 1893     Training Loss: 11.108278
Epoch: 1894     Training Loss: 11.108278
Epoch: 1895     Training Loss: 11.108277
Epoch: 1896     Training Loss: 11.108277
Epoch: 1897     Training Loss: 11.108277
Epoch: 1898     Training Loss: 11.108277
Epoch: 1899     Training Loss: 11.108276
Epoch: 1900     Training Loss: 11.108276
Epoch: 1901     Training Loss: 11.108275
Epoch: 1902     Training Loss: 11.108275
Epoch: 1903     Training Loss: 11.108274
Epoch: 1904     Training Loss: 11.108274
Epoch: 1905     Training Loss: 11.108274
Epoch: 1906     Training Loss: 11.108274
Epoch: 1907     Training Loss: 11.108274
Epoch: 1908     Training Loss: 11.108274
Epoch: 1909     Training Loss: 11.108274
Epoch: 1910     Training Loss: 11.108274
Epoch: 1911     Training Loss: 11.108273
Epoch: 1912     Training Loss: 11.108273
Epoch: 1913     Training Loss: 11.108273
Epoch: 1914     Training Loss: 11.108273
Epoch: 1915     Training Loss: 11.108273
Epoch: 1916     Training Loss: 11.108273
Epoch: 1917     Training Loss: 11.108273
Epoch: 1918     Training Loss: 11.108273
Epoch: 1919     Training Loss: 11.108273
Epoch: 1920     Training Loss: 11.108273
Epoch: 1921     Training Loss: 11.108272
Epoch: 1922     Training Loss: 11.108271
```

```
Epoch: 1923     Training Loss: 11.108271
Epoch: 1924     Training Loss: 11.108271
Epoch: 1925     Training Loss: 11.108270
Epoch: 1926     Training Loss: 11.108270
Epoch: 1927     Training Loss: 11.108270
Epoch: 1928     Training Loss: 11.108270
Epoch: 1929     Training Loss: 11.108271
Epoch: 1930     Training Loss: 11.108270
Epoch: 1931     Training Loss: 11.108270
Epoch: 1932     Training Loss: 11.108269
Epoch: 1933     Training Loss: 11.108269
Epoch: 1934     Training Loss: 11.108269
Epoch: 1935     Training Loss: 11.108269
Epoch: 1936     Training Loss: 11.108269
Epoch: 1937     Training Loss: 11.108269
Epoch: 1938     Training Loss: 11.108269
Epoch: 1939     Training Loss: 11.108268
Epoch: 1940     Training Loss: 11.108268
Epoch: 1941     Training Loss: 11.108267
Epoch: 1942     Training Loss: 11.108267
Epoch: 1943     Training Loss: 11.108267
Epoch: 1944     Training Loss: 11.108267
Epoch: 1945     Training Loss: 11.108267
Epoch: 1946     Training Loss: 11.108266
Epoch: 1947     Training Loss: 11.108266
Epoch: 1948     Training Loss: 11.108266
Epoch: 1949     Training Loss: 11.108265
Epoch: 1950     Training Loss: 11.108264
Epoch: 1951     Training Loss: 11.108265
Epoch: 1952     Training Loss: 11.108264
Epoch: 1953     Training Loss: 11.108264
Epoch: 1954     Training Loss: 11.108264
Epoch: 1955     Training Loss: 11.108265
Epoch: 1956     Training Loss: 11.108265
Epoch: 1957     Training Loss: 11.108263
Epoch: 1958     Training Loss: 11.108263
Epoch: 1959     Training Loss: 11.108263
Epoch: 1960     Training Loss: 11.108263
Epoch: 1961     Training Loss: 11.108262
Epoch: 1962     Training Loss: 11.108261
Epoch: 1963     Training Loss: 11.108261
Epoch: 1964     Training Loss: 11.108261
Epoch: 1965     Training Loss: 11.108261
Epoch: 1966     Training Loss: 11.108261
Epoch: 1967     Training Loss: 11.108259
Epoch: 1968     Training Loss: 11.108259
Epoch: 1969     Training Loss: 11.108259
Epoch: 1970     Training Loss: 11.108259
```

```
Epoch: 1971     Training Loss: 11.108259
Epoch: 1972     Training Loss: 11.108258
Epoch: 1973     Training Loss: 11.108258
Epoch: 1974     Training Loss: 11.108258
Epoch: 1975     Training Loss: 11.108257
Epoch: 1976     Training Loss: 11.108257
Epoch: 1977     Training Loss: 11.108257
Epoch: 1978     Training Loss: 11.108257
Epoch: 1979     Training Loss: 11.108257
Epoch: 1980     Training Loss: 11.108257
Epoch: 1981     Training Loss: 11.108257
Epoch: 1982     Training Loss: 11.108257
Epoch: 1983     Training Loss: 11.108257
Epoch: 1984     Training Loss: 11.108257
Epoch: 1985     Training Loss: 11.108257
Epoch: 1986     Training Loss: 11.108255
Epoch: 1987     Training Loss: 11.108255
Epoch: 1988     Training Loss: 11.108255
Epoch: 1989     Training Loss: 11.108255
Epoch: 1990     Training Loss: 11.108255
Epoch: 1991     Training Loss: 11.108255
Epoch: 1992     Training Loss: 11.108255
Epoch: 1993     Training Loss: 11.108254
Epoch: 1994     Training Loss: 11.108254
Epoch: 1995     Training Loss: 11.108254
Epoch: 1996     Training Loss: 11.108253
Epoch: 1997     Training Loss: 11.108254
Epoch: 1998     Training Loss: 11.108254
Epoch: 1999     Training Loss: 11.108254
Epoch: 2000     Training Loss: 11.108254
```

[ ]:

```
# Test the performance of the trained model

teacher.eval()
test_loss = 0.0

for idx, (data, labels) in enumerate(test_loader):
    # forward pass
    output = teacher(data)
    test_loss += criterion(output, labels).item()

# print the mean squared loss of the test dataset normalized by the mean square␣
 ↪of the test  labels
print('Average mean squared error {:.6f}'.format(test_loss / torch.mean(torch.
 ↪pow(y_test, 2))))
```

4 Part 3: Vary the width parameter, and plot the test error for different widths (5 points)

1. How does the test error vary as we change the width? In particular, consider varying the width of the student network from 1 to 20.
2. [Bonus] What happens if we vary the sample size?
3. [Bonus] How about adding a small amount of noise to the labels of the training dataset?

Report what you found and include the results in your submission.

[10]:
```
n_epochs = 2000 # the number of epochs can be tuned for better performance

criterion = torch.nn.MSELoss()
```

[14]:
```
def student_net(w):
    std_net = Net(d_input,w)
    std_net.train() # prep model for training
    optimizer = torch.optim.SGD(std_net.parameters(), lr=0.01)

    for epoch in range(n_epochs):
        train_loss = 0.0

        # train the model
        for idx, (data, labels) in enumerate(train_loader):
            # ------------------
            # Write your implementation here.

            # clear the gradients of all optimized variables
            optimizer.zero_grad()
            # forward pass: compute predicted outputs by passing inputs to the␣
↪model
            output = std_net(data)
            # calculate the loss
            loss = criterion(output,labels)
            # backward pass: compute gradient of the loss with respect to model␣
↪parameters
            loss.backward()
            # perform a single optimization step (parameter update)
            optimizer.step()
            # update running training loss
            train_loss += loss

            t_loss = train_loss / torch.mean(torch.pow(y_train, 2))

            # ------------------

    # Test the performance of the trained model
    std_net.eval()
```

```
    test_loss = 0.0

    for idx, (data, labels) in enumerate(test_loader):
        # forward pass
        output = std_net(data)
        test_loss += criterion(output, labels).item()

    # print the mean squared loss of the test dataset normalized by the mean␣
↪square of the test  labels
    print('Average mean squared error {:.6f}'.format(test_loss / torch.
↪mean(torch.pow(y_test, 2))))
```

[17]:
```
test_error = []
for w in range(1,21):
    print("width:",w)
    test_error.append(student_net(w))
```

```
width: 1
Average mean squared error 10.930061
width: 2
Average mean squared error 10.916376
width: 3
Average mean squared error 10.914468
width: 4
Average mean squared error 10.910889
width: 5
Average mean squared error 10.913124
width: 6
Average mean squared error 10.910731
width: 7
Average mean squared error 10.910593
width: 8
Average mean squared error 10.910300
width: 9
Average mean squared error 10.910397
width: 10
Average mean squared error 10.910455
width: 11
Average mean squared error 10.910155
width: 12
Average mean squared error 10.910148
width: 13
Average mean squared error 10.909992
width: 14
Average mean squared error 10.909336
width: 15
Average mean squared error 10.910934
```

```
width: 16
Average mean squared error 10.909252
width: 17
Average mean squared error 10.909606
width: 18
Average mean squared error 10.909761
width: 19
Average mean squared error 10.908634
width: 20
Average mean squared error 10.908721
```

Varying Sample size

Having width as 20 and changing the Sample size

[18]:
```python
# sample size
N1 = 5 * 20 * d_input

# random data from standard normal distribution
x_train = torch.randn(N1, d_input)
x_test = torch.randn(N1, d_input)

# teacher network with random weights
teacher = Net(d_input, width)

# generate labels using the teacher network
y_train = torch.FloatTensor([teacher.forward(x) for x in x_train])
y_test = torch.FloatTensor([teacher.forward(x) for x in x_test])

# combine the data and labels into pytorch friendly format
train_data = torch.utils.data.TensorDataset(x_train, y_train)
test_data = torch.utils.data.TensorDataset(x_test, y_test)

# prepare data loaders
train_loader = torch.utils.data.DataLoader(train_data, batch_size=batch_size)
test_loader = torch.utils.data.DataLoader(test_data, batch_size=batch_size)
```

[19]:
```python
def student_net(w):
    std_net = Net(d_input,w)
    std_net.train() # prep model for training
    optimizer = torch.optim.SGD(std_net.parameters(), lr=0.01)

    for epoch in range(n_epochs):
        train_loss = 0.0

        # train the model
        for idx, (data, labels) in enumerate(train_loader):
            # ------------------
```

```
            # Write your implementation here.

            # clear the gradients of all optimized variables
            optimizer.zero_grad()
            # forward pass: compute predicted outputs by passing inputs to the␣
↪model
            output = std_net(data)
            # calculate the loss
            loss = criterion(output,labels)
            # backward pass: compute gradient of the loss with respect to model␣
↪parameters
            loss.backward()
            # perform a single optimization step (parameter update)
            optimizer.step()
            # update running training loss
            train_loss += loss

            t_loss = train_loss / torch.mean(torch.pow(y_train, 2))

            # ------------------

    # Test the performance of the trained model
    std_net.eval()
    test_loss = 0.0

    for idx, (data, labels) in enumerate(test_loader):
        # forward pass
        output = std_net(data)
        test_loss += criterion(output, labels).item()

    # print the mean squared loss of the test dataset normalized by the mean␣
↪square of the test  labels
    print('Average mean squared error {:.6f}'.format(test_loss / torch.
↪mean(torch.pow(y_test, 2))))
```

[20]:
```
test_error = []
test_error.append(student_net(20))
```

```
Average mean squared error 87.164070
```

Varying sample size is increasing the *mean squared error*. However we can try for different vlaues of *Width* and *d_input* to experiment what actually happens for smaller to larger values of *N(Sample Size)*

[25]:
```
# generate labels using the teacher network
y_train = torch.FloatTensor([teacher.forward(x) for x in x_train])
x_train = torch.randn(N, d_input)
```

```python
dim1 = x_train.shape #to get the dimesion of the data
dim2 = y_train.shape

noise1 = np.random.rand(dim1)
noise2 = np.random.rand(dim2)

noisy_data_x = x_train + noise1
noisy_data_y = y_train + noise2 # to add noise the existing data

noisy_train_data = torch.utils.data.TensorDataset(noisy_data_x, noisy_data_y)

noisy_data_loader = torch.utils.data.DataLoader(noisy_train_data,␣
 ↪batch_size=batch_size)
```

```
      ␣
↪---------------------------------------------------------------------------

       TypeError                                 Traceback (most recent call␣
↪last)

       <ipython-input-25-fa668226c435> in <module>
         6 dim1 = x_train.shape #to get the dimesion of the data
         7 dim2 = y_train.shape
   ----> 8 noise1 = np.random.rand(dim1)
         9 noise2 = np.random.rand(dim2)
        10


       mtrand.pyx in mtrand.RandomState.rand()


       mtrand.pyx in mtrand.RandomState.random_sample()


       mtrand.pyx in mtrand.cont0_array()


       TypeError: 'torch.Size' object cannot be interpreted as an integer
```