classifiers_step_by_step

September 30, 2024

1 St124092

2 Kaung Htet Cho

3 Linear Classifier with Gradient Descent

In this lab, we will explore the implementation of a linear classifier from scratch. The topics covered include:

- Initialization of weights and bias
- Matrix multiplication of inputs (X) and weights (theta) with bias
- Loss (cost) function calculation
- Gradient Descent (both batch and stochastic)
- Weight update
- Use case for binomial and multinomial classification using sigmoid and softmax

---

3.1 0. Import necessary libraries

[1]:
```
import os
import time
import numpy as np
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import torch
from torch.utils.data import DataLoader, Subset
from torchvision import datasets, transforms

import torch.nn as nn
# import torchvision.transforms as transforms
# import torchvision.datasets as datasets
from torchvision import models
```

3.2 1. Initialization of Weights and Bias

Before training, we need to initialize our weights (`theta`) and bias (`b`). This can be done randomly or using a small constant value. In most cases, initializing weights with small random values works best to break symmetry, while bias can be initialized to zero.

3.3 2. Matrix Multiplication of X and θ with Bias

In linear models, the prediction is computed as the dot product between the input features X and the weight vector θ, plus the bias b. Mathematically, this is expressed as:

$$y = X\theta + b$$

To incorporate the bias term into the matrix multiplication, we can augment the input matrix X and the weight vector θ.

3.3.1 i. Augmenting X

Add a column of ones to the input matrix X to account for the bias term. Let X have dimensions $m \times n$ (where m is the number of samples and n is the number of features). The augmented matrix X_{bias} will have dimensions $m \times (n+1)$:

$$X_{\text{bias}} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

3.3.2 ii. Augmenting θ

Extend θ to include the bias term. The extended vector θ_{bias} will have dimensions $(n+1) \times c$ (where c is the no. of classes):

$$\theta_{\text{bias}} = \begin{bmatrix} b \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

3.3.3 iii. Matrix Multiplication

With the augmented matrix X_{bias} and the extended vector θ_{bias}, the prediction y can be computed as:

$$y = X_{\text{bias}}\theta_{\text{bias}}$$

This approach simplifies the computation by integrating the bias term directly into the matrix multiplication, which can be more efficient and straightforward in practice, especially when using matrix operations libraries.

[2]:
```python
# Add bias term (column of 1s) to X
def add_bias_term(X):
    return np.c_[np.ones((X.shape[0], 1)), X]
```

[3]:
```python
# Initialize weights with X updated to handle bias
def initialize_parameters(X, y, multiclass):
    n_features = X.shape[1]  # Number of features from the input X with bias␣
↪term
    if multiclass:
        n_classes = y.shape[1]
        theta = np.random.randn(n_features, n_classes) * 0.01  # Small random␣
↪weights
    else:
        theta = np.random.randn(n_features, 1) * 0.01  # Small random weights
    return theta
```

[4]:
```python
# Linear prediction
def linear_prediction(X, theta):
    return np.dot(X, theta)
```

3.4 3. Loss Functions for Classification

For classification tasks, different loss functions are used depending on whether the task is binary classification or multinomial classification.

3.4.1 Binary Classification (Sigmoid/Logistic Regression)

The loss function used is binary cross-entropy, also known as log loss. It is defined as:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

where: - m is the number of samples. - y_i is the true label for the i-th sample. - $\hat{y}_i$ is the predicted probability for the i-th sample.

3.4.2 Multinomial Classification (Softmax)

For multinomial classification, especially after one-hot encoding the labels, the loss function is categorical cross-entropy. It is defined as:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} y_{ik} \log(\hat{y}_{ik})$$

where: - m is the number of samples. - K is the number of classes. - y_{ik} is the one-hot encoded label for the i-th sample and k-th class (binary indicator: 0 or 1). - $\hat{y}_{ik}$ is the predicted probability of class k for the i-th sample.

In one-hot encoding, y_{ik} is 1 if the i-th sample belongs to class k, and 0 otherwise. This loss function measures how well the predicted probabilities match the one-hot encoded true labels.

```
[5]: # Binary Cross Entropy Loss
def binary_cross_entropy_loss(y_true, y_pred):
    m = y_true.shape[0]
    epsilon = 1e-15  # To avoid log(0) this is a very small value
    y_pred = np.clip(y_pred, epsilon, 1 - epsilon)
    return -np.sum(y_true * np.log(y_pred) + (1 - y_true) * np.log(1 - y_pred))
 ↪/ m

# Categorical Cross Entropy Loss
def categorical_cross_entropy_loss(y_true, y_pred):
    m = y_true.shape[0]
    epsilon = 1e-15  # To avoid log(0)
    y_pred = np.clip(y_pred, epsilon, 1 - epsilon)
    return -np.sum(y_true * np.log(y_pred)) / m
```

3.5 4. Gradient Descent for Minimizing the Loss Function with weight updates

Gradient descent is used to minimize the loss function by iteratively updating the weights based on the gradient of the loss function.

3.5.1 Batch Gradient Descent

In Batch Gradient Descent, the gradient is computed using all examples in the dataset:

$$\theta = \theta - \alpha \frac{\partial J(\theta)}{\partial \theta}$$

where: - α is the learning rate. - $\frac{\partial J(\theta)}{\partial \theta}$ is the gradient of the loss function with respect to the weights.

3.5.2 Stochastic Gradient Descent (SGD)

In Stochastic Gradient Descent, the gradient is computed using only one example at a time:

$$\theta = \theta - \alpha \frac{\partial J(\theta)}{\partial \theta}$$

where: - α is the learning rate. - $\frac{\partial J(\theta)}{\partial \theta}$ is the gradient of the loss function with respect to the weights, computed for a single example.

In both cases, the update rule for weights is the same, but the difference lies in how the gradient is computed: either over the entire dataset (Batch Gradient Descent) or a single example (Stochastic Gradient Descent).

3.6 5. Weight Update

After calculating the gradient, we update the weights using the formula mentioned above. Depending on whether we are using batch gradient descent or stochastic gradient descent, the weight update happens differently.

[6]:
```python
def gradient_descent_step(X, y, predictions, theta, learning_rate, multiclass):
    m = X.shape[0]
    # predictions = linear_prediction(X, theta)
    if multiclass:
        # predictions = softmax(predictions)
        gradients = (1/m) * np.dot(X.T, (predictions - y))
    else:
        # predictions = sigmoid(predictions)
        gradients = (1/m) * np.dot(X.T, (predictions - y))
    theta = theta - learning_rate * gradients
    return theta
```

[7]:
```python
# Stochastic Gradient Descent (SGD) Step
def stochastic_gradient_descent_step(X, y, theta, learning_rate, multiclass):
    m = X.shape[0]
    for i in range(m):
        xi = X[i:i+1]
        yi = y[i:i+1]
        # print (theta.shape)
        prediction = linear_prediction(xi, theta)
        if multiclass:
            prediction = softmax(prediction)
        else:
            prediction = sigmoid(prediction)
        # print(prediction.shape)
        # print(xi.T.shape)
        # print(yi.shape)
        gradients = np.dot(xi.T, (prediction - yi))
        theta = theta - learning_rate * gradients
    return theta
```

3.7 Activation: Binomial Classification (Sigmoid)

In binary classification, the sigmoid function is applied to the linear output to obtain the predicted probability. The sigmoid function $\sigma(z)$ is defined as:

$$\hat{y} = \sigma(X\theta) = \frac{1}{1 + e^{-X\theta}}$$

where: - $\hat{y}$ is the predicted probability. - $X\theta$ represents the linear combination of the input features X and the weights θ. - e is the base of the natural logarithm.

The sigmoid function maps the linear output to a probability value between 0 and 1, which can then be used to make a classification decision.

3.8 Activation: Multinomial Classification (Softmax)

In multinomial classification, the softmax function is used to compute probabilities across multiple classes. The softmax function $\text{softmax}(z_i)$ for class k is defined as:

$$\hat{y}_{ik} = \frac{e^{(X\theta_k)}}{\sum_{j=1}^{K} e^{(X\theta_j)}}$$

where: - $\hat{y}_{ik}$ is the predicted probability of the i-th sample belonging to class k. - $X\theta_k$ is the linear combination of the input features X and the weights θ_k for class k. - K is the total number of classes. - The denominator is the sum of the exponentials of the linear combinations for all classes, ensuring that the probabilities sum up to 1.

The softmax function converts the linear outputs into a probability distribution over multiple classes, which is useful for making predictions in multiclass classification problems.

[8]:
```
# Sigmoid function for binary classification
def sigmoid(z):
    return 1 / (1 + np.exp(-z))

# Softmax function for multi-class
def softmax(z):
    exp_scores = np.exp(z - np.max(z, axis=1, keepdims=True))  # For numerical␣
 ↪stability
    return exp_scores / np.sum(exp_scores, axis=1, keepdims=True)
```

3.8.1 OKAY LETS GO A HEAD AND TRAIN OUR MODEL !!

[9]:
```
# Training function
def train_model(X, y, learning_rate=0.01, iterations=1000, batch=True,␣
 ↪multiclass=False):
    # Add bias term to X
    X = add_bias_term(X)

    # Initialize theta
    theta = initialize_parameters(X, y, multiclass)

    # Track loss over iterations
    losses = []

    for i in range(iterations):
        # Compute predictions and loss
        if multiclass:
            predictions = softmax(linear_prediction(X, theta))
            loss = categorical_cross_entropy_loss(y, predictions)
```

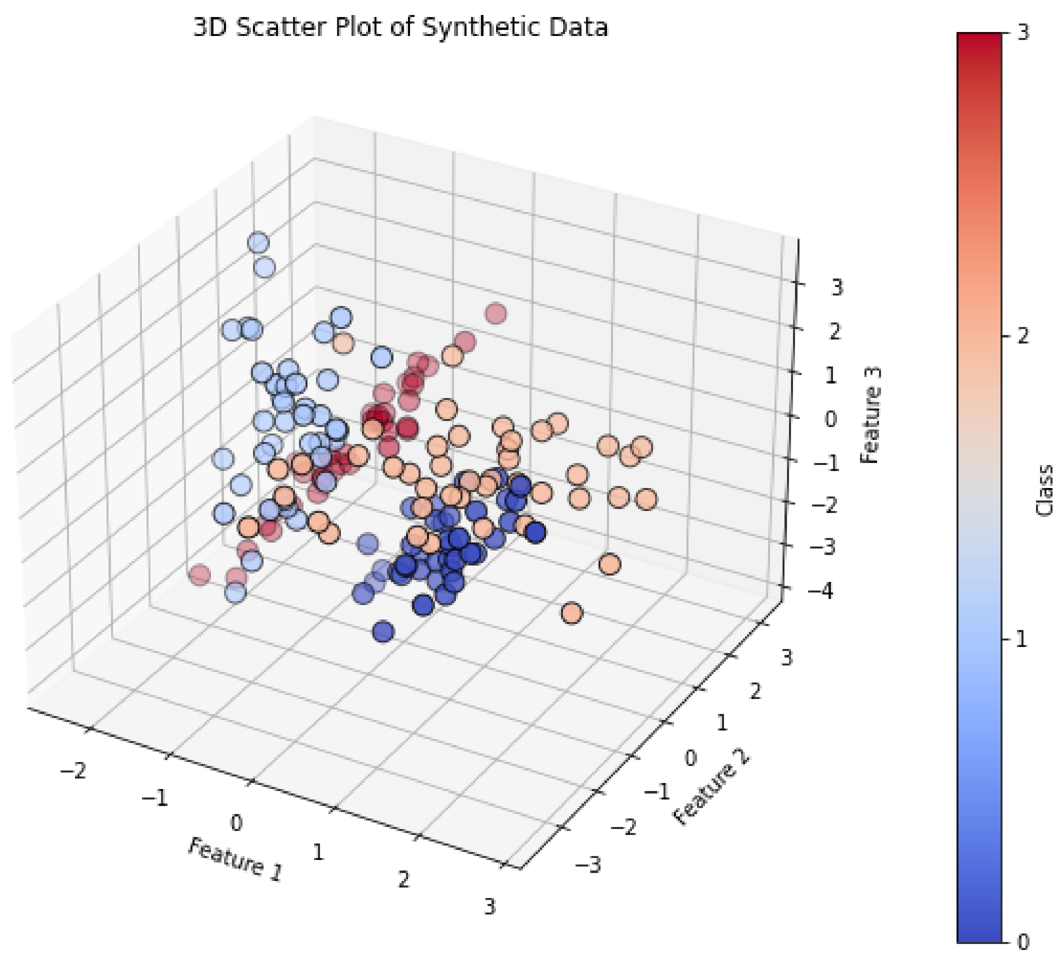
```
        else:
            predictions = sigmoid(linear_prediction(X, theta))
            loss = binary_cross_entropy_loss(y, predictions)

        losses.append(loss)

        # Update weights
        if batch:
            theta = gradient_descent_step(X, y, predictions, theta,␣
 ↪learning_rate, multiclass)
        else:
            # Use stochastic gradient descent
            theta = stochastic_gradient_descent_step(X, y, theta,␣
 ↪learning_rate, multiclass)

        # Print loss every 100 iterations
        if i % 100 == 0:
            print(f"Iteration {i}/{iterations}, Loss: {loss:.4f}")

    return theta, losses
```

3.8.2 Lets generate some data!!

[10]:
```
# Set multiclass to True or False
multiclass = True  # Set to True for multiclass classification

if multiclass:
    # For multiclass classification
    X_syn, y_syn = make_classification(n_samples=200, n_features=3,␣
 ↪n_informative=3,
                                   n_redundant=0, n_clusters_per_class=1,␣
 ↪n_classes=4, random_state=100)
    # Convert y to one-hot encoding
    y_syn = np.eye(np.max(y_syn) + 1)[y_syn]
else:
    # For binary classification
    X_syn, y_syn = make_classification(n_samples=200, n_features=2,␣
 ↪n_classes=2, n_informative=2, n_redundant=0, random_state=100)
    y_syn = y_syn.reshape(-1, 1)  # Reshape y to be a column vector

# Split into training and test sets
X_train_syn, X_test_syn, y_train_syn, y_test_syn = train_test_split(X_syn,␣
 ↪y_syn, test_size=0.2, random_state=100)
```

[11]:
```
if multiclass:
    fig = plt.figure(figsize=(10, 8))
```

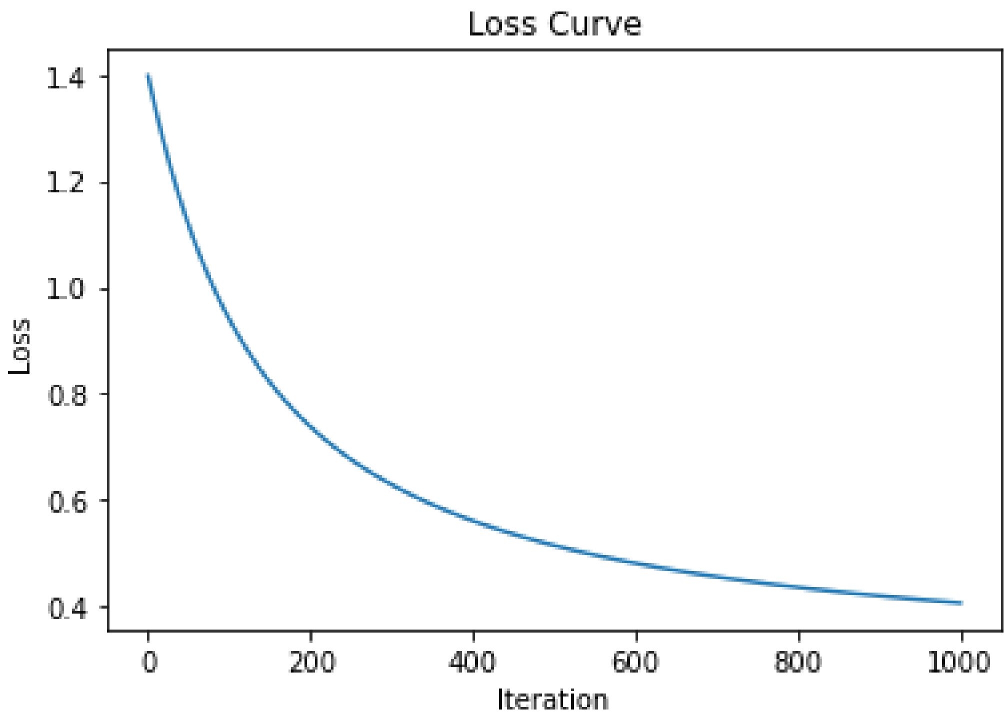
```python
    ax = fig.add_subplot(111, projection='3d')
    # Convert one-hot encoding to class labels for plotting
    y_train_labels = np.argmax(y_train_syn, axis=1)

    # Use the 3 features for the scatter plot
    scatter = ax.scatter(X_train_syn[:, 0], X_train_syn[:, 1], X_train_syn[:,␣
↪2],
                         c=y_train_labels, cmap='coolwarm', edgecolor='k',␣
↪s=100)

    # Add labels
    ax.set_title("3D Scatter Plot of Synthetic Data")
    ax.set_xlabel("Feature 1")
    ax.set_ylabel("Feature 2")
    ax.set_zlabel("Feature 3")

    # Add color bar to represent class labels
    cbar = fig.colorbar(scatter, ax=ax, pad=0.1)
    cbar.set_label('Class')

    # Set ticks to be integers corresponding to class labels
    cbar.set_ticks(np.arange(np.min(y_train_labels), np.max(y_train_labels) +␣
↪1))

    plt.show()
else:
    plt.figure(figsize=(10, 8))
    plt.scatter(X_train_syn[:, 0], X_train_syn[:, 1], c=y_train_syn,␣
↪cmap='coolwarm', edgecolor='k', s=100)
    plt.title("Scatter Plot of Training Data")
    plt.xlabel("Feature 1")
    plt.ylabel("Feature 2")
    plt.show()
```



[12]:
```
start_time = time.time()
theta, losses = train_model(X_train_syn, y_train_syn, learning_rate=0.01,␣
    ↪iterations=1000, batch=True, multiclass=multiclass)
print(f"Time Taken Using Batch gradient descent :{time.time() - start_time}")
```

```
Iteration 0/1000, Loss: 1.3973
Iteration 100/1000, Loss: 0.9391
Iteration 200/1000, Loss: 0.7377
Iteration 300/1000, Loss: 0.6285
Iteration 400/1000, Loss: 0.5605
Iteration 500/1000, Loss: 0.5143
Iteration 600/1000, Loss: 0.4807
Iteration 700/1000, Loss: 0.4554
Iteration 800/1000, Loss: 0.4354
Iteration 900/1000, Loss: 0.4194
Time Taken Using Batch gradient descent :0.14931321144104004
```

[13]:
```python
plt.plot(losses)
plt.xlabel('Iteration')
plt.ylabel('Loss')
plt.title('Loss Curve')
plt.show()
```



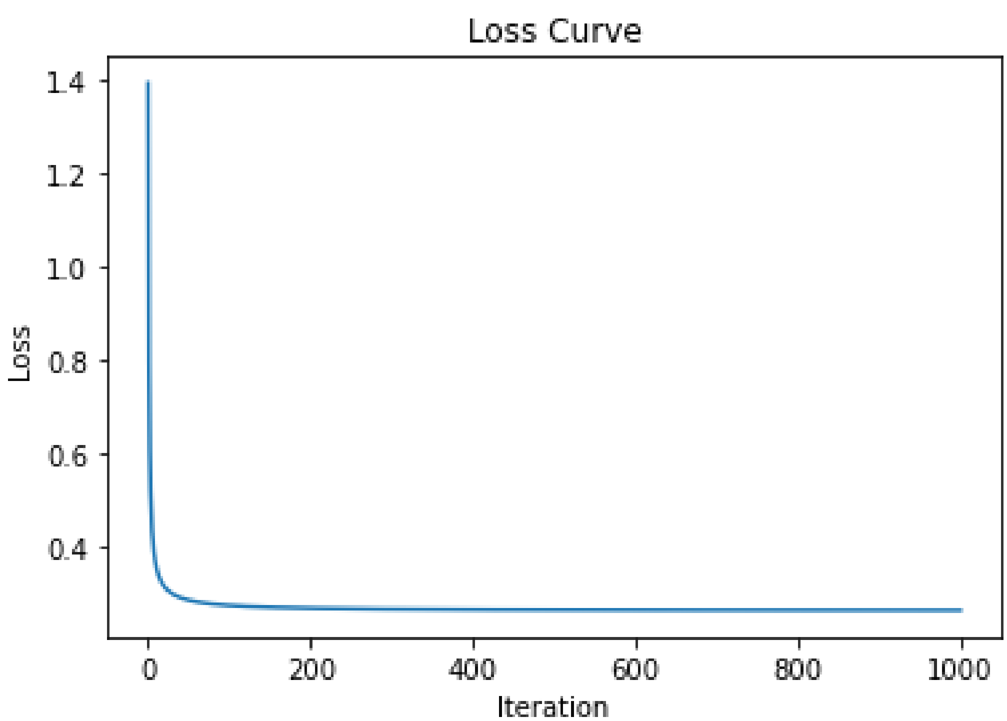
[14]:
```python
# Add bias term to test data
X_test_bias = add_bias_term(X_test_syn)

if multiclass:
    predictions = softmax(linear_prediction(X_test_bias, theta))
    predicted_classes = np.argmax(predictions, axis=1)
    true_classes = np.argmax(y_test_syn, axis=1)
else:
    predictions = sigmoid(linear_prediction(X_test_bias, theta))
    predicted_classes = (predictions >= 0.5).astype(int)
    true_classes = y_test_syn

# Calculate accuracy
accuracy = np.mean(predicted_classes.flatten() == true_classes.flatten())
print(f"Test Accuracy: {accuracy * 100:.2f}%")
```

```
Test Accuracy: 90.00%
```

[15]:
```
start_time = time.time()
theta, losses = train_model(X_train_syn, y_train_syn, learning_rate=0.01,
  ↪iterations=1000, batch=False, multiclass=multiclass)
print(f"Time Taken Using Stochastic gradient descent :{time.time() -
  ↪start_time}")
```

```
Iteration 0/1000, Loss: 1.3929
Iteration 100/1000, Loss: 0.2756
Iteration 200/1000, Loss: 0.2695
Iteration 300/1000, Loss: 0.2675
Iteration 400/1000, Loss: 0.2666
Iteration 500/1000, Loss: 0.2660
Iteration 600/1000, Loss: 0.2657
Iteration 700/1000, Loss: 0.2654
Iteration 800/1000, Loss: 0.2653
Iteration 900/1000, Loss: 0.2652
Time Taken Using Stochastic gradient descent :5.1505231857299805
```

[22]:
```
plt.plot(losses)
plt.xlabel('Iteration')
plt.ylabel('Loss')
plt.title('Loss Curve')
plt.show()
```



[23]:
```
# Add bias term to test data
X_test_bias = add_bias_term(X_test_syn)

if multiclass:
    predictions = softmax(linear_prediction(X_test_bias, theta))
    predicted_classes = np.argmax(predictions, axis=1)
    true_classes = np.argmax(y_test_syn, axis=1)
else:
    predictions = sigmoid(linear_prediction(X_test_bias, theta))
    predicted_classes = (predictions >= 0.5).astype(int)
    true_classes = y_test_syn

# Calculate accuracy
accuracy = np.mean(predicted_classes.flatten() == true_classes.flatten())
print(f"Test Accuracy: {accuracy * 100:.2f}%")
```

```
Test Accuracy: 92.50%
```

3.9 For Image what could be possible changes??

[24]:
```
# For our puffer surver we need to browse via a proxy!!

# Set HTTP and HTTPS proxy
os.environ['http_proxy'] = 'http://192.41.170.23:3128'
os.environ['https_proxy'] = 'http://192.41.170.23:3128'
```

[25]:
```
# Check if GPU is available
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")
```

```
Using device: cpu
```

```
/opt/conda/lib/python3.9/site-packages/torch/cuda/__init__.py:118: UserWarning:
CUDA initialization: The NVIDIA driver on your system is too old (found version
11060). Please update your GPU driver by downloading and installing a new
version from the URL: http://www.nvidia.com/Download/index.aspx Alternatively,
go to: https://pytorch.org to install a PyTorch version that has been compiled
with your version of the CUDA driver. (Triggered internally at
../c10/cuda/CUDAFunctions.cpp:108.)
  return torch._C._cuda_getDeviceCount() > 0
```

[26]:
```
cifar_train = datasets.CIFAR10('data', train=True, download=True␣
 ↪,transform=transforms.ToTensor())
cifar_test = datasets.CIFAR10('data', train=False, download=True␣
 ↪,transform=transforms.ToTensor())
```

```
Files already downloaded and verified
Files already downloaded and verified
```

[27]:
```python
# Function to subsample CIFAR-10 dataset
def subsample_dataset(dataset, sample_size=1000):
    indices = np.random.choice(len(dataset), sample_size, replace=False)
    subset = Subset(dataset, indices)
    return subset

# Subsample the training and test datasets
sample_size = 1000
train_subset = subsample_dataset(cifar_train, sample_size=sample_size)
test_subset = subsample_dataset(cifar_test, sample_size=int(sample_size * 0.4))

# Load data into PyTorch DataLoader
train_loader = DataLoader(train_subset, batch_size=sample_size, shuffle=True)
test_loader = DataLoader(test_subset, batch_size=int(sample_size * 0.4),␣
 ↪shuffle=False)

# Fetch all data and labels for easier handling
X_train, y_train = next(iter(train_loader))
X_test, y_test = next(iter(test_loader))

print("Before Flattening")
print(f"Training data shape: {X_train.shape}")
print(f"Test data shape: {X_test.shape}")

# Reshape the images to 2D for the KNN algorithm
X_train = X_train.view(X_train.size(0), -1).to(device)  # Flatten
X_test = X_test.view(X_test.size(0), -1).to(device)
y_train = y_train.to(device)
y_test = y_test.to(device)

print("After Flattening")
print(f"Training data shape: {X_train.shape}")
print(f"Test data shape: {X_test.shape}")
```

```
Before Flattening
Training data shape: torch.Size([1000, 3, 32, 32])
Test data shape: torch.Size([400, 3, 32, 32])
After Flattening
Training data shape: torch.Size([1000, 3072])
Test data shape: torch.Size([400, 3072])
```

[28]:
```python
class ImageLinearClassifier:
    def __init__(self, input_size, n_classes):
        self.W = np.random.randn(n_classes, input_size) * 0.01  # Small random␣
 ↪weights (10,3072)
        self.b = np.zeros((n_classes, 1))  # Bias initialized to zero (10,1)
```

```
    def predict(self, X):
        # Reshape X to (input_size, batch_size)
        X=X.T  # Transpose to shape (3072,1000)
        return np.dot(self.W, X) + self.b

    def compute_loss(self, X, y):
        """
            X: (batch_size, input_size) = (1000, 3072)
            y: (batch_size,) = (1000,) with class labels (0-9)
        """
        m = X.shape[0]
        z = self.predict(X)
        probs = self.softmax(z)
        log_likelihood = -np.log(probs[y, range(m)])
        return np.sum(log_likelihood) / m

    def softmax(self, z):
        exp_z = np.exp(z - np.max(z, axis=0, keepdims=True))  # Numerical␣
 ↪stability
        return exp_z / np.sum(exp_z, axis=0, keepdims=True)

    def gradient_descent(self, X, y, learning_rate=0.001):
        # Compute the gradient and update W, b
        m = X.shape[0]
        z = self.predict(X)
        probs = self.softmax(z)
        probs[y, range(m)] -= 1  # Gradient of softmax loss wrt z
        dW = np.dot(probs, X) / m  #Gradient wrt weights
        db = np.sum(probs, axis=1, keepdims=True) / m

        # Update weights and bias
        self.W -= learning_rate * dW
        self.b -= learning_rate * db
```

[29]:
```
def train(classifier, X_train, y_train, epochs, learning_rate):
    losses = []
    for i in range(epochs):
        loss = classifier.compute_loss(X_train, y_train)
        losses.append(loss)
        print(f'Epoch {i+1}, Loss: {loss}')
        classifier.gradient_descent(X_train, y_train, learning_rate)
    return losses
```

[30]:
```
print(f"Training data: {len(cifar_train)}")
print(f"Test data: {len(cifar_test)}")

image, label = cifar_train[0]
```

```
# Now you can check the shape of the image
print(f"Image shape: {image.shape}")
```

```
Training data: 50000
Test data: 10000
Image shape: torch.Size([3, 32, 32])
```

[31]:

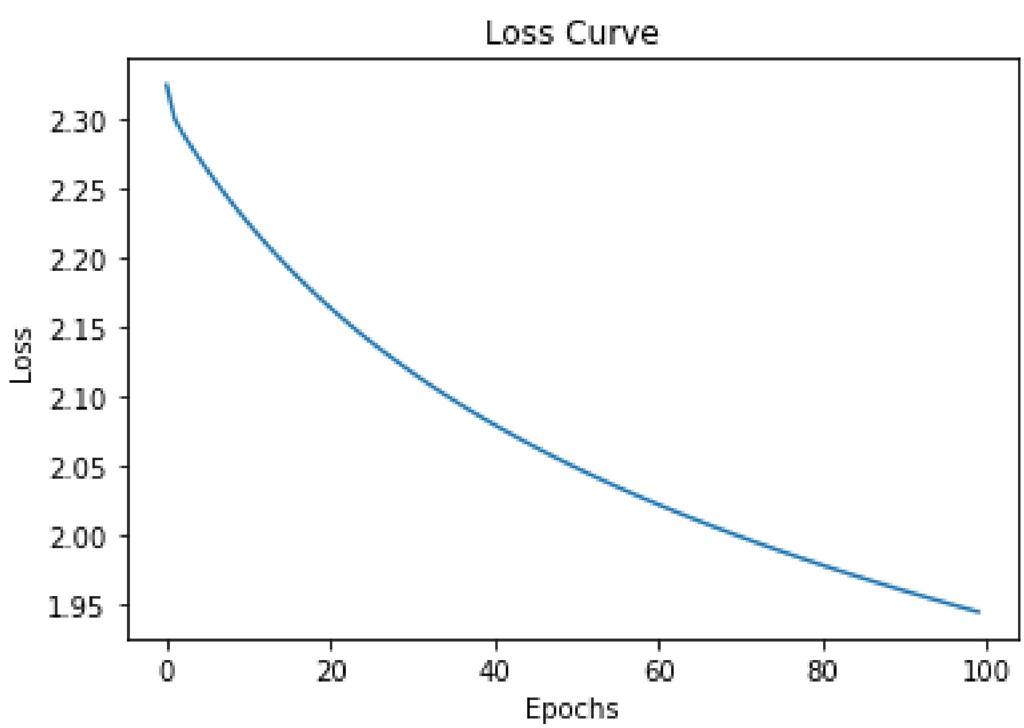
```
# Example usage
n_classes = 10  # For CIFAR-10
image_size = 32 * 32 * 3  # CIFAR-10 images are 32x32x3
classifier = ImageLinearClassifier(input_size=image_size, n_classes=n_classes)

# X_train is shape (image_size, batch_size) and y_train is (batch_size,)
losses = train(classifier, X_train, y_train, epochs=100, learning_rate=0.01)
```

```
Epoch 1, Loss: 2.3244961981300367
Epoch 2, Loss: 2.2997889815893306
Epoch 3, Loss: 2.289259543337674
Epoch 4, Loss: 2.2801174848191055
Epoch 5, Loss: 2.2713609350461
Epoch 6, Loss: 2.262906242493645
Epoch 7, Loss: 2.2547342921396396
Epoch 8, Loss: 2.2468314493952577
Epoch 9, Loss: 2.239185109292782
Epoch 10, Loss: 2.231783285579562
Epoch 11, Loss: 2.224614558107142
Epoch 12, Loss: 2.217668054751691
Epoch 13, Loss: 2.2109334354131662
Epoch 14, Loss: 2.2044008748110646
Epoch 15, Loss: 2.1980610440313995
Epoch 16, Loss: 2.1919050911519107
Epoch 17, Loss: 2.1859246212841756
Epoch 18, Loss: 2.1801116763302115
Epoch 19, Loss: 2.174458714703721
Epoch 20, Loss: 2.16895859122133
Epoch 21, Loss: 2.163604537329091
Epoch 22, Loss: 2.158390141794516
Epoch 23, Loss: 2.1533093319642607
Epoch 24, Loss: 2.1483563556618948
Epoch 25, Loss: 2.1435257637785807
Epoch 26, Loss: 2.1388123935914547
Epoch 27, Loss: 2.1342113528296007
Epoch 28, Loss: 2.1297180044954005
Epoch 29, Loss: 2.1253279524392354
Epoch 30, Loss: 2.121037027677783
Epoch 31, Loss: 2.1168412754400796
Epoch 32, Loss: 2.112736942920921
Epoch 33, Loss: 2.1087204677177533
```

```
Epoch 34, Loss: 2.1047884669247896
Epoch 35, Loss: 2.100937726856475
Epoch 36, Loss: 2.097165193371493
Epoch 37, Loss: 2.09346796276809
Epoch 38, Loss: 2.089843273221514
Epoch 39, Loss: 2.0862884967347086
Epoch 40, Loss: 2.0828011315740347
Epoch 41, Loss: 2.079378795162563
Epoch 42, Loss: 2.0760192174044594
Epoch 43, Loss: 2.072720234415001
Epoch 44, Loss: 2.0694797826319036
Epoch 45, Loss: 2.066295893284762
Epoch 46, Loss: 2.063166687200586
Epoch 47, Loss: 2.0600903699245765
Epoch 48, Loss: 2.0570652271364205
Epoch 49, Loss: 2.054089620343528
Epoch 50, Loss: 2.0511619828336873
Epoch 51, Loss: 2.0482808158707098
Epoch 52, Loss: 2.045444685117578
Epoch 53, Loss: 2.042652217272638
Epoch 54, Loss: 2.0399020969052306
Epoch 55, Loss: 2.0371930634780577
Epoch 56, Loss: 2.0345239085443767
Epoch 57, Loss: 2.0318934731089033
Epoch 58, Loss: 2.0293006451420066
Epoch 59, Loss: 2.026744357237483
Epoch 60, Loss: 2.0242235844048304
Epoch 61, Loss: 2.021737341987524
Epoch 62, Loss: 2.019284683699393
Epoch 63, Loss: 2.0168646997716815
Epoch 64, Loss: 2.0144765152038984
Epoch 65, Loss: 2.0121192881120042
Epoch 66, Loss: 2.0097922081679105
Epoch 67, Loss: 2.007494495124667
Epoch 68, Loss: 2.005225397422096
Epoch 69, Loss: 2.002984190867946
Epoch 70, Loss: 2.0007701773900095
Epoch 71, Loss: 1.998582683854897
Epoch 72, Loss: 1.9964210609494848
Epoch 73, Loss: 1.994284682121282
Epoch 74, Loss: 1.9921729425742292
Epoch 75, Loss: 1.99008525831665
Epoch 76, Loss: 1.9880210652583017
Epoch 77, Loss: 1.9859798183536554
Epoch 78, Loss: 1.9839609907887301
Epoch 79, Loss: 1.9819640732089676
Epoch 80, Loss: 1.9799885729857998
Epoch 81, Loss: 1.9780340135197056
```

```
Epoch 82, Loss: 1.9760999335776916
Epoch 83, Loss: 1.9741858866632642
Epoch 84, Loss: 1.9722914404170768
Epoch 85, Loss: 1.9704161760465466
Epoch 86, Loss: 1.9685596877828515
Epoch 87, Loss: 1.9667215823637958
Epoch 88, Loss: 1.9649014785411427
Epoch 89, Loss: 1.9630990066110918
Epoch 90, Loss: 1.96131380796665
Epoch 91, Loss: 1.9595455346707313
Epoch 92, Loss: 1.9577938490488873
Epoch 93, Loss: 1.9560584233006288
Epoch 94, Loss: 1.9543389391283645
Epoch 95, Loss: 1.9526350873830467
Epoch 96, Loss: 1.9509465677256517
Epoch 97, Loss: 1.9492730883036848
Epoch 98, Loss: 1.9476143654419438
Epoch 99, Loss: 1.945970123346818
Epoch 100, Loss: 1.9443400938234363
```

[32]:
```python
plt.plot(losses)
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Loss Curve')
plt.show()
```



3.10 Key Components:

1. Input Layer: The input data, similar to your previous setup.
2. Hidden Layer(s): These layers will have weights, biases, and non-linear activations like ReLU.
3. Output Layer: This will have a softmax activation for classification.
4. Loss Function: Cross-entropy loss for classification.
5. Backpropagation: To update weights using gradients from the loss.

3.11 MLP Structure Example:

1. Input Layer: (3072 neurons, corresponding to image size 32x32x3 in CIFAR-10)
2. Hidden Layer 1: Fully connected, with a non-linear activation like ReLU.
3. Hidden Layer 2: Another fully connected layer (optional).
4. Output Layer: A fully connected layer with 10 neurons (for 10 classes) and softmax activation.

[33]:
```python
class MLPClassifier:
    def __init__(self, input_size, hidden_size, output_size):
        # Weight initialization
        self.W1 = np.random.randn(hidden_size, input_size) * 0.01  #␣
 ↪(hidden_size, input_size)
        self.b1 = np.zeros((hidden_size, 1))  # (hidden_size, 1)
        self.W2 = np.random.randn(output_size, hidden_size) * 0.01  #␣
 ↪(output_size, hidden_size)
        self.b2 = np.zeros((output_size, 1))  # (output_size, 1)

    def relu(self, z):
        return np.maximum(0, z)

    def relu_derivative(self, z):
        return np.where(z > 0, 1, 0)

    def softmax(self, z):
        exp_z = np.exp(z - np.max(z, axis=0, keepdims=True))  # Numerical␣
 ↪stability
        return exp_z / np.sum(exp_z, axis=0, keepdims=True)

    def forward(self, X):
        """
        Forward pass through the network.
        X: input data of shape (input_size, batch_size)
        """
        # Layer 1 (hidden layer)
        self.Z1 = np.dot(self.W1, X) + self.b1  # (hidden_size, batch_size)
        self.A1 = self.relu(self.Z1)  # Apply ReLU activation

        ## ADD ANOTHER HIDDEN LAYER IN YOUR TAKE HOME EXERCISE
```

```
        # Layer 2 (output layer)
        self.Z2 = np.dot(self.W2, self.A1) + self.b2  # (output_size,␣
↪batch_size)
        self.A2 = self.softmax(self.Z2)  # Apply softmax activation
        return self.A2

    def compute_loss(self, A2, y):
        """
        Compute cross-entropy loss.
        A2: output from softmax, shape (output_size, batch_size)
        y: true labels, shape (batch_size,)
        """
        m = y.shape[0]  # batch size
        log_likelihood = -np.log(A2[y, range(m)])
        loss = np.sum(log_likelihood) / m
        return loss

    def backward(self, X, y, learning_rate=0.01):
        """
        Perform backward propagation and update weights.
        X: input data of shape (input_size, batch_size)
        y: true labels of shape (batch_size,)
        """
        m = X.shape[1]  # Batch size

        # Gradient of the loss w.r.t. Z2
        dZ2 = self.A2  # Softmax probabilities
        dZ2[y, range(m)] -= 1  # Subtract 1 from the correct class probabilities
        dZ2 /= m

        # Gradients for W2 and b2
        dW2 = np.dot(dZ2, self.A1.T)  # (output_size, hidden_size)
        db2 = np.sum(dZ2, axis=1, keepdims=True)  # (output_size, 1)

        # Gradients for the hidden layer (backprop through ReLU)
        dA1 = np.dot(self.W2.T, dZ2)  # (hidden_size, batch_size)
        dZ1 = dA1 * self.relu_derivative(self.Z1)  # Backprop through ReLU

        # Gradients for W1 and b1
        dW1 = np.dot(dZ1, X.T)  # (hidden_size, input_size)
        db1 = np.sum(dZ1, axis=1, keepdims=True)  # (hidden_size, 1)

        # Update weights and biases
        self.W1 -= learning_rate * dW1
        self.b1 -= learning_rate * db1
        self.W2 -= learning_rate * dW2
```

```
        self.b2 -= learning_rate * db2

    def train(self, X_train, y_train, epochs=100, learning_rate=0.01):
        """
        Train the network.
        X_train: input data, shape (input_size, batch_size)
        y_train: true labels, shape (batch_size,)
        """
        losses = []
        for i in range(epochs):
            # Forward pass
            A2 = self.forward(X_train)

            # Compute the loss
            loss = self.compute_loss(A2, y_train)
            print(f'Epoch {i+1}, Loss: {loss}')
            losses.append(loss)

            # Backward pass
            self.backward(X_train, y_train, learning_rate)
        return losses
```

[34]:

```
input_size = 3072  # CIFAR-10 images are 32x32x3
hidden_size = 100  # Arbitrary hidden layer size
output_size = 10  # 10 classes for CIFAR-10

mlp = MLPClassifier(input_size, hidden_size, output_size)

X_train_copy = X_train.T

losses = mlp.train(X_train_copy, y_train, epochs=100, learning_rate=0.1)
```

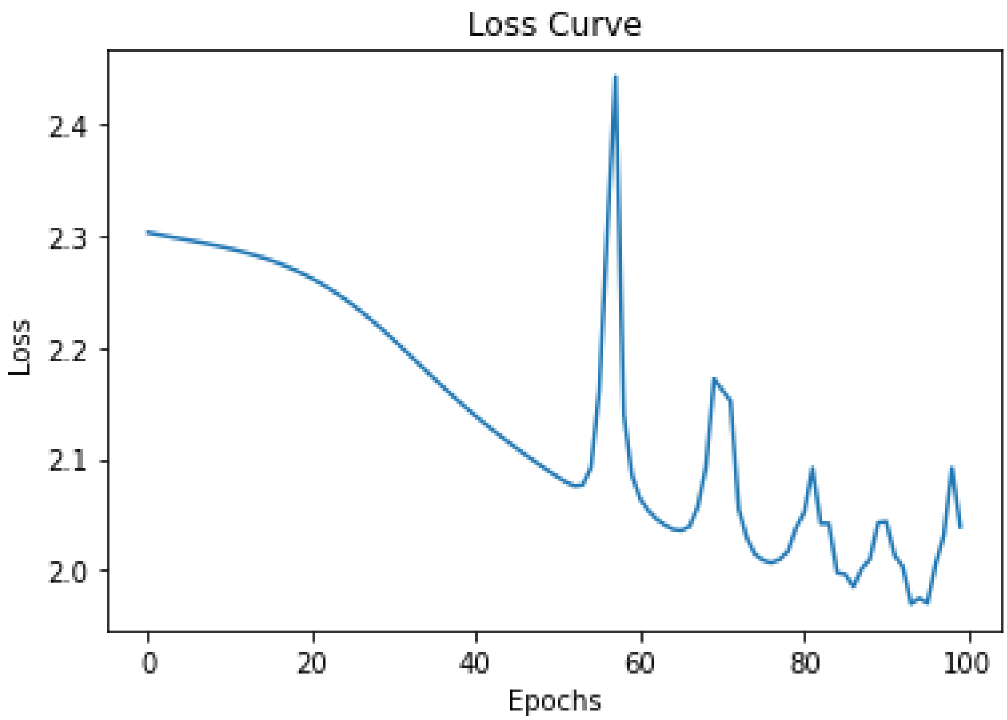
```
Epoch 1, Loss: 2.3030842819548556
Epoch 2, Loss: 2.3015363917786082
Epoch 3, Loss: 2.3001489169449023
Epoch 4, Loss: 2.2988126765041725
Epoch 5, Loss: 2.2974839138784398
Epoch 6, Loss: 2.2961367103012567
Epoch 7, Loss: 2.2947441601867316
Epoch 8, Loss: 2.2933161857146662
Epoch 9, Loss: 2.2918307818550905
Epoch 10, Loss: 2.2902501004056566
Epoch 11, Loss: 2.2885728603025077
Epoch 12, Loss: 2.2867710420391965
Epoch 13, Loss: 2.28482650068963
Epoch 14, Loss: 2.2827300708284843
Epoch 15, Loss: 2.280460710587007
```

```
Epoch 16, Loss: 2.2779983382655264
Epoch 17, Loss: 2.275326904526805
Epoch 18, Loss: 2.2724156703272897
Epoch 19, Loss: 2.2692446092081937
Epoch 20, Loss: 2.265783052904521
Epoch 21, Loss: 2.2620051510956265
Epoch 22, Loss: 2.2579024896852204
Epoch 23, Loss: 2.2534667474726295
Epoch 24, Loss: 2.248682941816581
Epoch 25, Loss: 2.2435555664856865
Epoch 26, Loss: 2.238090286404641
Epoch 27, Loss: 2.2322957284771894
Epoch 28, Loss: 2.2262025786151445
Epoch 29, Loss: 2.2198324810472227
Epoch 30, Loss: 2.2132179915121273
Epoch 31, Loss: 2.206416949219848
Epoch 32, Loss: 2.1994763747070616
Epoch 33, Loss: 2.192437497097516
Epoch 34, Loss: 2.1853609016848776
Epoch 35, Loss: 2.1782909408264683
Epoch 36, Loss: 2.1712730117358614
Epoch 37, Loss: 2.1643497285573385
Epoch 38, Loss: 2.157556602481427
Epoch 39, Loss: 2.1509162284781893
Epoch 40, Loss: 2.1444427930730012
Epoch 41, Loss: 2.1381434886247273
Epoch 42, Loss: 2.132020210285066
Epoch 43, Loss: 2.1260657210102036
Epoch 44, Loss: 2.120274094185845
Epoch 45, Loss: 2.1146335261458753
Epoch 46, Loss: 2.109128955132295
Epoch 47, Loss: 2.103754483315264
Epoch 48, Loss: 2.098500736771581
Epoch 49, Loss: 2.0933548635043633
Epoch 50, Loss: 2.0883227338421984
Epoch 51, Loss: 2.0834511096716968
Epoch 52, Loss: 2.078984017258503
Epoch 53, Loss: 2.0756599452569615
Epoch 54, Loss: 2.0766689980371162
Epoch 55, Loss: 2.0922967733155784
Epoch 56, Loss: 2.1605329255061276
Epoch 57, Loss: 2.3063294360181126
Epoch 58, Loss: 2.4432081865291178
Epoch 59, Loss: 2.1402261809858656
Epoch 60, Loss: 2.0850022093239007
Epoch 61, Loss: 2.063904411841294
Epoch 62, Loss: 2.0534114278925797
Epoch 63, Loss: 2.0458858579825296
```

```
Epoch 64, Loss: 2.040643518047492
Epoch 65, Loss: 2.036787965923209
Epoch 66, Loss: 2.035744161692922
Epoch 67, Loss: 2.0393307334187862
Epoch 68, Loss: 2.0568660448642038
Epoch 69, Loss: 2.0926139243383277
Epoch 70, Loss: 2.1718530733184362
Epoch 71, Loss: 2.161771067275567
Epoch 72, Loss: 2.152213679004437
Epoch 73, Loss: 2.0549837227282937
Epoch 74, Loss: 2.028929863298653
Epoch 75, Loss: 2.0141321556124683
Epoch 76, Loss: 2.0089562882497916
Epoch 77, Loss: 2.0067044504295204
Epoch 78, Loss: 2.0097589556757574
Epoch 79, Loss: 2.0169725989662055
Epoch 80, Loss: 2.0382322836450326
Epoch 81, Loss: 2.0519732200847813
Epoch 82, Loss: 2.091827524119243
Epoch 83, Loss: 2.042009079822729
Epoch 84, Loss: 2.041895822834455
Epoch 85, Loss: 1.9978903533093897
Epoch 86, Loss: 1.9963852140600225
Epoch 87, Loss: 1.9850632734249232
Epoch 88, Loss: 2.0013546612603754
Epoch 89, Loss: 2.009867436888828
Epoch 90, Loss: 2.0426687524440896
Epoch 91, Loss: 2.0438123110144897
Epoch 92, Loss: 2.01328808574113
Epoch 93, Loss: 2.003191913323788
Epoch 94, Loss: 1.970003500667602
Epoch 95, Loss: 1.974904061402372
Epoch 96, Loss: 1.9705977217876876
Epoch 97, Loss: 2.0057133630704165
Epoch 98, Loss: 2.0313346602072584
Epoch 99, Loss: 2.0918011395176954
Epoch 100, Loss: 2.039101904821624
```

[35]:

```
plt.plot(losses)
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Loss Curve')
plt.show()
```



3.12 TAKE AWAY EXERCISE

- In the picture above the learning rate is high so the model is not able to find a minima and is overshooting.
- Use a learning rate scheduler in the above implementation, increase the no. of epochs and train on full data.
- Also increase the complexity of structure by adding another hidden layer

[30]:
```
# your code here
```

[38]:
```
class MLPClassifier:
    def __init__(self, input_size, hidden_size, output_size, hidden_size_2):
        # Initialize weights and biases for two hidden layers
        self.W1 = np.random.randn(hidden_size, input_size) * 0.01
        self.b1 = np.zeros((hidden_size, 1))
        self.W2 = np.random.randn(hidden_size_2, hidden_size) * 0.01
        self.b2 = np.zeros((hidden_size_2, 1))
        self.W3 = np.random.randn(output_size, hidden_size_2) * 0.01
        self.b3 = np.zeros((output_size, 1))

    def relu(self, z):
        return np.maximum(0, z)

    def relu_derivative(self, z):
```

```
        return np.where(z > 0, 1, 0)

    def softmax(self, z):
        exp_z = np.exp(z - np.max(z, axis=0, keepdims=True))  # Numerical␣
↪stability
        return exp_z / np.sum(exp_z, axis=0, keepdims=True)

    def forward(self, X):
        self.Z1 = np.dot(self.W1, X) + self.b1
        self.A1 = self.relu(self.Z1)
        self.Z2 = np.dot(self.W2, self.A1) + self.b2
        self.A2 = self.relu(self.Z2)
        self.Z3 = np.dot(self.W3, self.A2) + self.b3
        self.A3 = self.softmax(self.Z3)
        return self.A3

    def compute_loss(self, A3, y):
        m = y.shape[0]
        log_likelihood = -np.log(A3[y, range(m)])
        loss = np.sum(log_likelihood) / m
        return loss

    def backward(self, X, y, learning_rate):
        m = X.shape[1]
        dZ3 = self.A3
        dZ3[y, range(m)] -= 1
        dZ3 /= m

        dW3 = np.dot(dZ3, self.A2.T)
        db3 = np.sum(dZ3, axis=1, keepdims=True)

        dA2 = np.dot(self.W3.T, dZ3)
        dZ2 = dA2 * self.relu_derivative(self.Z2)

        dW2 = np.dot(dZ2, self.A1.T)
        db2 = np.sum(dZ2, axis=1, keepdims=True)

        dA1 = np.dot(self.W2.T, dZ2)
        dZ1 = dA1 * self.relu_derivative(self.Z1)

        dW1 = np.dot(dZ1, X.T)
        db1 = np.sum(dZ1, axis=1, keepdims=True)

        # Update weights and biases
        self.W1 -= learning_rate * dW1
        self.b1 -= learning_rate * db1
        self.W2 -= learning_rate * dW2
```

```
        self.b2 -= learning_rate * db2
        self.W3 -= learning_rate * dW3
        self.b3 -= learning_rate * db3

    def train(self, X_train, y_train, epochs=100, initial_lr=0.01,␣
↪step_size=10, lr_decay=0.1, early_stopping_patience=10):
        """
        Train the model with a step learning rate scheduler and early stopping.

        Parameters:
        - epochs: Number of epochs for training.
        - initial_lr: Starting learning rate.
        - step_size: After how many epochs to reduce the learning rate.
        - lr_decay: The factor by which to multiply the learning rate after␣
↪step_size epochs.
        """
        losses = []
        best_loss = float('inf')
        patience_counter = 0
        learning_rate = initial_lr

        for i in range(epochs):
            # Forward pass
            A3 = self.forward(X_train)
            # Compute loss
            loss = self.compute_loss(A3, y_train)
            print(f'Epoch {i+1}, Loss: {loss}, Learning Rate: {learning_rate}')
            losses.append(loss)

            # Check for early stopping condition
            if loss < best_loss:
                best_loss = loss
                patience_counter = 0
            else:
                patience_counter += 1
                if patience_counter >= early_stopping_patience:
                    print(f"Early stopping at epoch {i+1}")
                    break

            # Backward pass and weight updates
            self.backward(X_train, y_train, learning_rate)

            # Step Learning Rate Scheduler
            if (i+1) % step_size == 0:
                learning_rate *= lr_decay

        return losses
```

[46]:
```
train_loader = DataLoader(cifar_train, batch_size=64, shuffle=True)
X_train, y_train = next(iter(train_loader))

print("Before Flattening")
print(f"Training data shape: {X_train.shape}")

# Reshape the images to 2D for the KNN algorithm
X_train = X_train.view(X_train.size(0), -1)  # Flatten
X_train = X_train.T.to(device)
y_train = y_train.to(device)

print("After Flattening")
print(f"Training data shape: {X_train.shape}")
```

```
Before Flattening
Training data shape: torch.Size([64, 3, 32, 32])
After Flattening
Training data shape: torch.Size([3072, 64])
```

[66]:
```
# Training the model using the step learning rate scheduler
model = MLPClassifier(input_size=32*32*3, hidden_size=128, hidden_size_2=64,␣
↪output_size=10)
losses = model.train(X_train, y_train, epochs=1000, initial_lr=0.01,␣
↪step_size=100, lr_decay=0.5, early_stopping_patience=10)
```

```
Epoch 1, Loss: 2.3024745706888172, Learning Rate: 0.01
Epoch 2, Loss: 2.30233318353272, Learning Rate: 0.01
Epoch 3, Loss: 2.302192092327501, Learning Rate: 0.01
Epoch 4, Loss: 2.3020512184434656, Learning Rate: 0.01
Epoch 5, Loss: 2.301910698371731, Learning Rate: 0.01
Epoch 6, Loss: 2.301770550944966, Learning Rate: 0.01
Epoch 7, Loss: 2.3016307471512496, Learning Rate: 0.01
Epoch 8, Loss: 2.3014912063666877, Learning Rate: 0.01
Epoch 9, Loss: 2.3013519876400643, Learning Rate: 0.01
Epoch 10, Loss: 2.301213095284651, Learning Rate: 0.01
Epoch 11, Loss: 2.301074487061726, Learning Rate: 0.01
Epoch 12, Loss: 2.300936105104025, Learning Rate: 0.01
Epoch 13, Loss: 2.300798030744547, Learning Rate: 0.01
Epoch 14, Loss: 2.3006601792323473, Learning Rate: 0.01
Epoch 15, Loss: 2.3005226712229323, Learning Rate: 0.01
Epoch 16, Loss: 2.300385519720482, Learning Rate: 0.01
Epoch 17, Loss: 2.300248697250098, Learning Rate: 0.01
Epoch 18, Loss: 2.3001122342679405, Learning Rate: 0.01
Epoch 19, Loss: 2.2999760619700456, Learning Rate: 0.01
Epoch 20, Loss: 2.2998402279592933, Learning Rate: 0.01
Epoch 21, Loss: 2.2997046359101727, Learning Rate: 0.01
Epoch 22, Loss: 2.299569292311642, Learning Rate: 0.01
```

```
Epoch 23, Loss: 2.2994342950118947, Learning Rate: 0.01
Epoch 24, Loss: 2.299299614109505, Learning Rate: 0.01
Epoch 25, Loss: 2.299165240061179, Learning Rate: 0.01
Epoch 26, Loss: 2.299031124152446, Learning Rate: 0.01
Epoch 27, Loss: 2.2988972201419244, Learning Rate: 0.01
Epoch 28, Loss: 2.2987634538667323, Learning Rate: 0.01
Epoch 29, Loss: 2.2986299496076397, Learning Rate: 0.01
Epoch 30, Loss: 2.298496729550528, Learning Rate: 0.01
Epoch 31, Loss: 2.2983637949213307, Learning Rate: 0.01
Epoch 32, Loss: 2.298231156862896, Learning Rate: 0.01
Epoch 33, Loss: 2.298098730458618, Learning Rate: 0.01
Epoch 34, Loss: 2.297966596286144, Learning Rate: 0.01
Epoch 35, Loss: 2.2978348154920685, Learning Rate: 0.01
Epoch 36, Loss: 2.2977033124084447, Learning Rate: 0.01
Epoch 37, Loss: 2.297572136030042, Learning Rate: 0.01
Epoch 38, Loss: 2.29744121971299, Learning Rate: 0.01
Epoch 39, Loss: 2.297310603157541, Learning Rate: 0.01
Epoch 40, Loss: 2.297180231260405, Learning Rate: 0.01
Epoch 41, Loss: 2.2970501762718385, Learning Rate: 0.01
Epoch 42, Loss: 2.2969203131957627, Learning Rate: 0.01
Epoch 43, Loss: 2.2967907026145618, Learning Rate: 0.01
Epoch 44, Loss: 2.2966613099034143, Learning Rate: 0.01
Epoch 45, Loss: 2.2965321440427355, Learning Rate: 0.01
Epoch 46, Loss: 2.29640321687172, Learning Rate: 0.01
Epoch 47, Loss: 2.2962745028150833, Learning Rate: 0.01
Epoch 48, Loss: 2.2961460487480876, Learning Rate: 0.01
Epoch 49, Loss: 2.296017823602359, Learning Rate: 0.01
Epoch 50, Loss: 2.2958897262076174, Learning Rate: 0.01
Epoch 51, Loss: 2.2957619331352834, Learning Rate: 0.01
Epoch 52, Loss: 2.2956344476779997, Learning Rate: 0.01
Epoch 53, Loss: 2.2955072160441747, Learning Rate: 0.01
Epoch 54, Loss: 2.295380244926892, Learning Rate: 0.01
Epoch 55, Loss: 2.2952535717581335, Learning Rate: 0.01
Epoch 56, Loss: 2.2951271184164685, Learning Rate: 0.01
Epoch 57, Loss: 2.295000836815534, Learning Rate: 0.01
Epoch 58, Loss: 2.2948747734138433, Learning Rate: 0.01
Epoch 59, Loss: 2.294748902462565, Learning Rate: 0.01
Epoch 60, Loss: 2.2946232941508127, Learning Rate: 0.01
Epoch 61, Loss: 2.2944979569122923, Learning Rate: 0.01
Epoch 62, Loss: 2.2943728570066275, Learning Rate: 0.01
Epoch 63, Loss: 2.294247991672849, Learning Rate: 0.01
Epoch 64, Loss: 2.294123234378228, Learning Rate: 0.01
Epoch 65, Loss: 2.293998635179289, Learning Rate: 0.01
Epoch 66, Loss: 2.293874284141575, Learning Rate: 0.01
Epoch 67, Loss: 2.29375019648890l, Learning Rate: 0.01
Epoch 68, Loss: 2.293626344350314, Learning Rate: 0.01
Epoch 69, Loss: 2.293502654117604, Learning Rate: 0.01
Epoch 70, Loss: 2.293379173795622, Learning Rate: 0.01
```

```
Epoch 71, Loss: 2.2932559515012967, Learning Rate: 0.01
Epoch 72, Loss: 2.2931329421240934, Learning Rate: 0.01
Epoch 73, Loss: 2.293010124066777, Learning Rate: 0.01
Epoch 74, Loss: 2.2928875044013592, Learning Rate: 0.01
Epoch 75, Loss: 2.292765072430335, Learning Rate: 0.01
Epoch 76, Loss: 2.2926428551365157, Learning Rate: 0.01
Epoch 77, Loss: 2.2925208644947865, Learning Rate: 0.01
Epoch 78, Loss: 2.292399057245729, Learning Rate: 0.01
Epoch 79, Loss: 2.292277478471553, Learning Rate: 0.01
Epoch 80, Loss: 2.292156140113035, Learning Rate: 0.01
Epoch 81, Loss: 2.292035019600354, Learning Rate: 0.01
Epoch 82, Loss: 2.2919141942095878, Learning Rate: 0.01
Epoch 83, Loss: 2.291793580488328, Learning Rate: 0.01
Epoch 84, Loss: 2.291673209833389, Learning Rate: 0.01
Epoch 85, Loss: 2.2915530807516564, Learning Rate: 0.01
Epoch 86, Loss: 2.291433181961388, Learning Rate: 0.01
Epoch 87, Loss: 2.2913134919147846, Learning Rate: 0.01
Epoch 88, Loss: 2.2911940229792345, Learning Rate: 0.01
Epoch 89, Loss: 2.29107482262415, Learning Rate: 0.01
Epoch 90, Loss: 2.2909558172242543, Learning Rate: 0.01
Epoch 91, Loss: 2.2908370400843667, Learning Rate: 0.01
Epoch 92, Loss: 2.290718490060338, Learning Rate: 0.01
Epoch 93, Loss: 2.2906001817368145, Learning Rate: 0.01
Epoch 94, Loss: 2.290482147220536, Learning Rate: 0.01
Epoch 95, Loss: 2.2903643275084855, Learning Rate: 0.01
Epoch 96, Loss: 2.290246699523875, Learning Rate: 0.01
Epoch 97, Loss: 2.2901292852841513, Learning Rate: 0.01
Epoch 98, Loss: 2.290012077234409, Learning Rate: 0.01
Epoch 99, Loss: 2.289895106253912, Learning Rate: 0.01
Epoch 100, Loss: 2.2897784046252134, Learning Rate: 0.01
Epoch 101, Loss: 2.289661929047903, Learning Rate: 0.005
Epoch 102, Loss: 2.289603783296311, Learning Rate: 0.005
Epoch 103, Loss: 2.2895456896493727, Learning Rate: 0.005
Epoch 104, Loss: 2.289487651414767, Learning Rate: 0.005
Epoch 105, Loss: 2.289429664518779, Learning Rate: 0.005
Epoch 106, Loss: 2.2893717170703454, Learning Rate: 0.005
Epoch 107, Loss: 2.289313810584312, Learning Rate: 0.005
Epoch 108, Loss: 2.289255953636747, Learning Rate: 0.005
Epoch 109, Loss: 2.2891981516769793, Learning Rate: 0.005
Epoch 110, Loss: 2.2891404021927246, Learning Rate: 0.005
Epoch 111, Loss: 2.289082693615068, Learning Rate: 0.005
Epoch 112, Loss: 2.289025033258004, Learning Rate: 0.005
Epoch 113, Loss: 2.2889674227906975, Learning Rate: 0.005
Epoch 114, Loss: 2.2889098689021345, Learning Rate: 0.005
Epoch 115, Loss: 2.288852395113949, Learning Rate: 0.005
Epoch 116, Loss: 2.288794977964047, Learning Rate: 0.005
Epoch 117, Loss: 2.2887375998376296, Learning Rate: 0.005
Epoch 118, Loss: 2.288680269340486, Learning Rate: 0.005
```

```
Epoch 119, Loss: 2.2886229875709727, Learning Rate: 0.005
Epoch 120, Loss: 2.288565743813895, Learning Rate: 0.005
Epoch 121, Loss: 2.2885085565357377, Learning Rate: 0.005
Epoch 122, Loss: 2.2884514197618375, Learning Rate: 0.005
Epoch 123, Loss: 2.2883943281616834, Learning Rate: 0.005
Epoch 124, Loss: 2.2883372751747943, Learning Rate: 0.005
Epoch 125, Loss: 2.288280267391192, Learning Rate: 0.005
Epoch 126, Loss: 2.2882233057431884, Learning Rate: 0.005
Epoch 127, Loss: 2.288166387136215, Learning Rate: 0.005
Epoch 128, Loss: 2.288109508003307, Learning Rate: 0.005
Epoch 129, Loss: 2.288052672511577, Learning Rate: 0.005
Epoch 130, Loss: 2.2879958909131317, Learning Rate: 0.005
Epoch 131, Loss: 2.2879391557866766, Learning Rate: 0.005
Epoch 132, Loss: 2.2878824678580716, Learning Rate: 0.005
Epoch 133, Loss: 2.2878258229734785, Learning Rate: 0.005
Epoch 134, Loss: 2.28776922350795, Learning Rate: 0.005
Epoch 135, Loss: 2.2877126733707693, Learning Rate: 0.005
Epoch 136, Loss: 2.2876561644731606, Learning Rate: 0.005
Epoch 137, Loss: 2.2875996998897348, Learning Rate: 0.005
Epoch 138, Loss: 2.28754328102613, Learning Rate: 0.005
Epoch 139, Loss: 2.287486909710055, Learning Rate: 0.005
Epoch 140, Loss: 2.2874305856526904, Learning Rate: 0.005
Epoch 141, Loss: 2.2873743097425487, Learning Rate: 0.005
Epoch 142, Loss: 2.287318076519527, Learning Rate: 0.005
Epoch 143, Loss: 2.2872618899914707, Learning Rate: 0.005
Epoch 144, Loss: 2.2872057528912957, Learning Rate: 0.005
Epoch 145, Loss: 2.287149649146021, Learning Rate: 0.005
Epoch 146, Loss: 2.287093595287419, Learning Rate: 0.005
Epoch 147, Loss: 2.2870375887784187, Learning Rate: 0.005
Epoch 148, Loss: 2.2869816200902013, Learning Rate: 0.005
Epoch 149, Loss: 2.2869256816324963, Learning Rate: 0.005
Epoch 150, Loss: 2.286869756177894, Learning Rate: 0.005
Epoch 151, Loss: 2.2868138698466325, Learning Rate: 0.005
Epoch 152, Loss: 2.2867580205628313, Learning Rate: 0.005
Epoch 153, Loss: 2.28670221765287, Learning Rate: 0.005
Epoch 154, Loss: 2.2866464446172206, Learning Rate: 0.005
Epoch 155, Loss: 2.2865907146518483, Learning Rate: 0.005
Epoch 156, Loss: 2.286535029331163, Learning Rate: 0.005
Epoch 157, Loss: 2.286479391308996, Learning Rate: 0.005
Epoch 158, Loss: 2.2864238022835686, Learning Rate: 0.005
Epoch 159, Loss: 2.28636826183953, Learning Rate: 0.005
Epoch 160, Loss: 2.286312780713563, Learning Rate: 0.005
Epoch 161, Loss: 2.286257348921094, Learning Rate: 0.005
Epoch 162, Loss: 2.286201957384213, Learning Rate: 0.005
Epoch 163, Loss: 2.286146618291469, Learning Rate: 0.005
Epoch 164, Loss: 2.2860913084385386, Learning Rate: 0.005
Epoch 165, Loss: 2.2860360493083975, Learning Rate: 0.005
Epoch 166, Loss: 2.2859808259014054, Learning Rate: 0.005
```

```
Epoch 167, Loss: 2.2859256328452635, Learning Rate: 0.005
Epoch 168, Loss: 2.28587048102767, Learning Rate: 0.005
Epoch 169, Loss: 2.285815375937843, Learning Rate: 0.005
Epoch 170, Loss: 2.285760308777253, Learning Rate: 0.005
Epoch 171, Loss: 2.2857052885186286, Learning Rate: 0.005
Epoch 172, Loss: 2.285650303505434, Learning Rate: 0.005
Epoch 173, Loss: 2.285595368362504, Learning Rate: 0.005
Epoch 174, Loss: 2.2855404731524382, Learning Rate: 0.005
Epoch 175, Loss: 2.2854856249753746, Learning Rate: 0.005
Epoch 176, Loss: 2.2854308128507625, Learning Rate: 0.005
Epoch 177, Loss: 2.285376067827786, Learning Rate: 0.005
Epoch 178, Loss: 2.285321354485796, Learning Rate: 0.005
Epoch 179, Loss: 2.2852666713426606, Learning Rate: 0.005
Epoch 180, Loss: 2.285212040236674, Learning Rate: 0.005
Epoch 181, Loss: 2.2851574644233312, Learning Rate: 0.005
Epoch 182, Loss: 2.285102921968346, Learning Rate: 0.005
Epoch 183, Loss: 2.2850484492024155, Learning Rate: 0.005
Epoch 184, Loss: 2.284994037327837, Learning Rate: 0.005
Epoch 185, Loss: 2.2849396695243107, Learning Rate: 0.005
Epoch 186, Loss: 2.2848853413947214, Learning Rate: 0.005
Epoch 187, Loss: 2.284831079537108, Learning Rate: 0.005
Epoch 188, Loss: 2.2847768531536166, Learning Rate: 0.005
Epoch 189, Loss: 2.2847226695703444, Learning Rate: 0.005
Epoch 190, Loss: 2.284668505107792, Learning Rate: 0.005
Epoch 191, Loss: 2.2846143778097048, Learning Rate: 0.005
Epoch 192, Loss: 2.2845602873543482, Learning Rate: 0.005
Epoch 193, Loss: 2.2845062343029743, Learning Rate: 0.005
Epoch 194, Loss: 2.2844522278333352, Learning Rate: 0.005
Epoch 195, Loss: 2.2843982693644147, Learning Rate: 0.005
Epoch 196, Loss: 2.2843443409356077, Learning Rate: 0.005
Epoch 197, Loss: 2.284290454528257, Learning Rate: 0.005
Epoch 198, Loss: 2.284236603763482, Learning Rate: 0.005
Epoch 199, Loss: 2.2841828022027277, Learning Rate: 0.005
Epoch 200, Loss: 2.284129025783332, Learning Rate: 0.005
Epoch 201, Loss: 2.284075282176598, Learning Rate: 0.0025
Epoch 202, Loss: 2.284048425829382, Learning Rate: 0.0025
Epoch 203, Loss: 2.284021579155984, Learning Rate: 0.0025
Epoch 204, Loss: 2.283994745247191, Learning Rate: 0.0025
Epoch 205, Loss: 2.2839679138983744, Learning Rate: 0.0025
Epoch 206, Loss: 2.2839410931251756, Learning Rate: 0.0025
Epoch 207, Loss: 2.283914280624424, Learning Rate: 0.0025
Epoch 208, Loss: 2.283887479833913, Learning Rate: 0.0025
Epoch 209, Loss: 2.2838606871040366, Learning Rate: 0.0025
Epoch 210, Loss: 2.2838339063677586, Learning Rate: 0.0025
Epoch 211, Loss: 2.283807134697537, Learning Rate: 0.0025
Epoch 212, Loss: 2.283780373015854, Learning Rate: 0.0025
Epoch 213, Loss: 2.2837535880986426, Learning Rate: 0.0025
Epoch 214, Loss: 2.283726815253744, Learning Rate: 0.0025
```

```
Epoch 215, Loss: 2.283700050540936, Learning Rate: 0.0025
Epoch 216, Loss: 2.283673296366124, Learning Rate: 0.0025
Epoch 217, Loss: 2.283646552276934, Learning Rate: 0.0025
Epoch 218, Loss: 2.283619818931819, Learning Rate: 0.0025
Epoch 219, Loss: 2.283593098810912, Learning Rate: 0.0025
Epoch 220, Loss: 2.283566387936861, Learning Rate: 0.0025
Epoch 221, Loss: 2.2835396862519937, Learning Rate: 0.0025
Epoch 222, Loss: 2.2835129933184826, Learning Rate: 0.0025
Epoch 223, Loss: 2.2834863171836006, Learning Rate: 0.0025
Epoch 224, Loss: 2.28345964602624, Learning Rate: 0.0025
Epoch 225, Loss: 2.28343298462441, Learning Rate: 0.0025
Epoch 226, Loss: 2.2834063272316665, Learning Rate: 0.0025
Epoch 227, Loss: 2.2833796819650125, Learning Rate: 0.0025
Epoch 228, Loss: 2.283353041275422, Learning Rate: 0.0025
Epoch 229, Loss: 2.283326408739019, Learning Rate: 0.0025
Epoch 230, Loss: 2.2832997860765456, Learning Rate: 0.0025
Epoch 231, Loss: 2.2832731722764987, Learning Rate: 0.0025
Epoch 232, Loss: 2.283246573837125, Learning Rate: 0.0025
Epoch 233, Loss: 2.2832199820874157, Learning Rate: 0.0025
Epoch 234, Loss: 2.283193401363076, Learning Rate: 0.0025
Epoch 235, Loss: 2.2831668286627775, Learning Rate: 0.0025
Epoch 236, Loss: 2.283140266884355, Learning Rate: 0.0025
Epoch 237, Loss: 2.2831137128662653, Learning Rate: 0.0025
Epoch 238, Loss: 2.283087169560888, Learning Rate: 0.0025
Epoch 239, Loss: 2.283060634184054, Learning Rate: 0.0025
Epoch 240, Loss: 2.2830341052681113, Learning Rate: 0.0025
Epoch 241, Loss: 2.2830075857013648, Learning Rate: 0.0025
Epoch 242, Loss: 2.282981072321326, Learning Rate: 0.0025
Epoch 243, Loss: 2.2829545725100293, Learning Rate: 0.0025
Epoch 244, Loss: 2.2829280843148134, Learning Rate: 0.0025
Epoch 245, Loss: 2.2829016028175513, Learning Rate: 0.0025
Epoch 246, Loss: 2.2828751294840046, Learning Rate: 0.0025
Epoch 247, Loss: 2.282848665717025, Learning Rate: 0.0025
Epoch 248, Loss: 2.282822213555249, Learning Rate: 0.0025
Epoch 249, Loss: 2.2827957680900224, Learning Rate: 0.0025
Epoch 250, Loss: 2.2827693375488507, Learning Rate: 0.0025
Epoch 251, Loss: 2.2827429136510613, Learning Rate: 0.0025
Epoch 252, Loss: 2.2827165123820223, Learning Rate: 0.0025
Epoch 253, Loss: 2.2826901185546116, Learning Rate: 0.0025
Epoch 254, Loss: 2.2826637343099225, Learning Rate: 0.0025
Epoch 255, Loss: 2.282637358511118, Learning Rate: 0.0025
Epoch 256, Loss: 2.282610989827414, Learning Rate: 0.0025
Epoch 257, Loss: 2.2825846337128937, Learning Rate: 0.0025
Epoch 258, Loss: 2.282558289671413, Learning Rate: 0.0025
Epoch 259, Loss: 2.2825319549730656, Learning Rate: 0.0025
Epoch 260, Loss: 2.2825056256702867, Learning Rate: 0.0025
Epoch 261, Loss: 2.282479304983636, Learning Rate: 0.0025
Epoch 262, Loss: 2.2824529882180746, Learning Rate: 0.0025
```

```
Epoch 263, Loss: 2.282426679097793, Learning Rate: 0.0025
Epoch 264, Loss: 2.2824003829774924, Learning Rate: 0.0025
Epoch 265, Loss: 2.2823741003671723, Learning Rate: 0.0025
Epoch 266, Loss: 2.2823478227423664, Learning Rate: 0.0025
Epoch 267, Loss: 2.282321559025976, Learning Rate: 0.0025
Epoch 268, Loss: 2.2822952996587165, Learning Rate: 0.0025
Epoch 269, Loss: 2.282269050206134, Learning Rate: 0.0025
Epoch 270, Loss: 2.282242809596149, Learning Rate: 0.0025
Epoch 271, Loss: 2.2822165761780058, Learning Rate: 0.0025
Epoch 272, Loss: 2.2821903555747274, Learning Rate: 0.0025
Epoch 273, Loss: 2.2821641360958576, Learning Rate: 0.0025
Epoch 274, Loss: 2.2821379191369013, Learning Rate: 0.0025
Epoch 275, Loss: 2.2821117107835365, Learning Rate: 0.0025
Epoch 276, Loss: 2.282085509686567, Learning Rate: 0.0025
Epoch 277, Loss: 2.2820593227278465, Learning Rate: 0.0025
Epoch 278, Loss: 2.282033149376402, Learning Rate: 0.0025
Epoch 279, Loss: 2.282006985943527, Learning Rate: 0.0025
Epoch 280, Loss: 2.2819808335679963, Learning Rate: 0.0025
Epoch 281, Loss: 2.281954688680318, Learning Rate: 0.0025
Epoch 282, Loss: 2.2819285557070526, Learning Rate: 0.0025
Epoch 283, Loss: 2.2819024273591815, Learning Rate: 0.0025
Epoch 284, Loss: 2.2818763123836283, Learning Rate: 0.0025
Epoch 285, Loss: 2.2818502051832192, Learning Rate: 0.0025
Epoch 286, Loss: 2.2818241115156432, Learning Rate: 0.0025
Epoch 287, Loss: 2.2817980257771144, Learning Rate: 0.0025
Epoch 288, Loss: 2.2817719481973846, Learning Rate: 0.0025
Epoch 289, Loss: 2.2817458811981925, Learning Rate: 0.0025
Epoch 290, Loss: 2.2817198202389224, Learning Rate: 0.0025
Epoch 291, Loss: 2.281693758958247, Learning Rate: 0.0025
Epoch 292, Loss: 2.281667705871228, Learning Rate: 0.0025
Epoch 293, Loss: 2.281641660840156, Learning Rate: 0.0025
Epoch 294, Loss: 2.281615625949729, Learning Rate: 0.0025
Epoch 295, Loss: 2.2815895999523175, Learning Rate: 0.0025
Epoch 296, Loss: 2.281563584334796, Learning Rate: 0.0025
Epoch 297, Loss: 2.281537580266801, Learning Rate: 0.0025
Epoch 298, Loss: 2.281511583212617, Learning Rate: 0.0025
Epoch 299, Loss: 2.2814855959990226, Learning Rate: 0.0025
Epoch 300, Loss: 2.281459613509367, Learning Rate: 0.0025
Epoch 301, Loss: 2.2814336423339903, Learning Rate: 0.00125
Epoch 302, Loss: 2.281420658584804, Learning Rate: 0.00125
Epoch 303, Loss: 2.2814076776538954, Learning Rate: 0.00125
Epoch 304, Loss: 2.281394699080198, Learning Rate: 0.00125
Epoch 305, Loss: 2.281381721588919, Learning Rate: 0.00125
Epoch 306, Loss: 2.2813687469581754, Learning Rate: 0.00125
Epoch 307, Loss: 2.28135577574545, Learning Rate: 0.00125
Epoch 308, Loss: 2.2813428092365218, Learning Rate: 0.00125
Epoch 309, Loss: 2.2813298447326114, Learning Rate: 0.00125
Epoch 310, Loss: 2.281316882269868, Learning Rate: 0.00125
```

```
Epoch 311, Loss: 2.2813039206492984, Learning Rate: 0.00125
Epoch 312, Loss: 2.281290965212447, Learning Rate: 0.00125
Epoch 313, Loss: 2.281278009632037, Learning Rate: 0.00125
Epoch 314, Loss: 2.281265055041029, Learning Rate: 0.00125
Epoch 315, Loss: 2.281252101394, Learning Rate: 0.00125
Epoch 316, Loss: 2.281239150456037, Learning Rate: 0.00125
Epoch 317, Loss: 2.281226204571265, Learning Rate: 0.00125
Epoch 318, Loss: 2.2812132605017528, Learning Rate: 0.00125
Epoch 319, Loss: 2.2812003157972747, Learning Rate: 0.00125
Epoch 320, Loss: 2.2811873745432893, Learning Rate: 0.00125
Epoch 321, Loss: 2.281174435965018, Learning Rate: 0.00125
Epoch 322, Loss: 2.28116149824569, Learning Rate: 0.00125
Epoch 323, Loss: 2.2811485620702063, Learning Rate: 0.00125
Epoch 324, Loss: 2.2811356306629387, Learning Rate: 0.00125
Epoch 325, Loss: 2.281122698588332, Learning Rate: 0.00125
Epoch 326, Loss: 2.2811097706949455, Learning Rate: 0.00125
Epoch 327, Loss: 2.2810968462176247, Learning Rate: 0.00125
Epoch 328, Loss: 2.281083924917594, Learning Rate: 0.00125
Epoch 329, Loss: 2.2810710065122484, Learning Rate: 0.00125
Epoch 330, Loss: 2.28105808747393, Learning Rate: 0.00125
Epoch 331, Loss: 2.2810451712772686, Learning Rate: 0.00125
Epoch 332, Loss: 2.2810322564302994, Learning Rate: 0.00125
Epoch 333, Loss: 2.281019345197725, Learning Rate: 0.00125
Epoch 334, Loss: 2.2810064349538983, Learning Rate: 0.00125
Epoch 335, Loss: 2.2809935239904746, Learning Rate: 0.00125
Epoch 336, Loss: 2.280980615497507, Learning Rate: 0.00125
Epoch 337, Loss: 2.280967709648488, Learning Rate: 0.00125
Epoch 338, Loss: 2.2809548069072068, Learning Rate: 0.00125
Epoch 339, Loss: 2.2809419051274373, Learning Rate: 0.00125
Epoch 340, Loss: 2.2809290054111786, Learning Rate: 0.00125
Epoch 341, Loss: 2.2809161073981556, Learning Rate: 0.00125
Epoch 342, Loss: 2.280903212962631, Learning Rate: 0.00125
Epoch 343, Loss: 2.2808903185775176, Learning Rate: 0.00125
Epoch 344, Loss: 2.280877426301243, Learning Rate: 0.00125
Epoch 345, Loss: 2.2808645385761546, Learning Rate: 0.00125
Epoch 346, Loss: 2.280851649861412, Learning Rate: 0.00125
Epoch 347, Loss: 2.280838764263572, Learning Rate: 0.00125
Epoch 348, Loss: 2.280825880671123, Learning Rate: 0.00125
Epoch 349, Loss: 2.2808130002395948, Learning Rate: 0.00125
Epoch 350, Loss: 2.2808001193421683, Learning Rate: 0.00125
Epoch 351, Loss: 2.280787240025532, Learning Rate: 0.00125
Epoch 352, Loss: 2.280774363759254, Learning Rate: 0.00125
Epoch 353, Loss: 2.2807614890224333, Learning Rate: 0.00125
Epoch 354, Loss: 2.2807486173191407, Learning Rate: 0.00125
Epoch 355, Loss: 2.280735747020424, Learning Rate: 0.00125
Epoch 356, Loss: 2.2807228795977883, Learning Rate: 0.00125
Epoch 357, Loss: 2.2807100156610867, Learning Rate: 0.00125
Epoch 358, Loss: 2.2806971529567095, Learning Rate: 0.00125
```

```
Epoch 359, Loss: 2.2806842922577673, Learning Rate: 0.00125
Epoch 360, Loss: 2.2806714334353257, Learning Rate: 0.00125
Epoch 361, Loss: 2.28065857698744, Learning Rate: 0.00125
Epoch 362, Loss: 2.2806457221940244, Learning Rate: 0.00125
Epoch 363, Loss: 2.2806328707503294, Learning Rate: 0.00125
Epoch 364, Loss: 2.280620020175906, Learning Rate: 0.00125
Epoch 365, Loss: 2.2806071718610794, Learning Rate: 0.00125
Epoch 366, Loss: 2.2805943257464794, Learning Rate: 0.00125
Epoch 367, Loss: 2.280581482346861, Learning Rate: 0.00125
Epoch 368, Loss: 2.280568642080602, Learning Rate: 0.00125
Epoch 369, Loss: 2.280555803135981, Learning Rate: 0.00125
Epoch 370, Loss: 2.280542966952384, Learning Rate: 0.00125
Epoch 371, Loss: 2.2805301337813626, Learning Rate: 0.00125
Epoch 372, Loss: 2.2805173027706083, Learning Rate: 0.00125
Epoch 373, Loss: 2.2805044731642012, Learning Rate: 0.00125
Epoch 374, Loss: 2.280491645117403, Learning Rate: 0.00125
Epoch 375, Loss: 2.2804788198626214, Learning Rate: 0.00125
Epoch 376, Loss: 2.2804659959008116, Learning Rate: 0.00125
Epoch 377, Loss: 2.280453175017292, Learning Rate: 0.00125
Epoch 378, Loss: 2.2804403542046767, Learning Rate: 0.00125
Epoch 379, Loss: 2.280427534391582, Learning Rate: 0.00125
Epoch 380, Loss: 2.280414701165293, Learning Rate: 0.00125
Epoch 381, Loss: 2.2804018702359588, Learning Rate: 0.00125
Epoch 382, Loss: 2.280389040700281, Learning Rate: 0.00125
Epoch 383, Loss: 2.280376213815808, Learning Rate: 0.00125
Epoch 384, Loss: 2.2803633884933414, Learning Rate: 0.00125
Epoch 385, Loss: 2.280350564736053, Learning Rate: 0.00125
Epoch 386, Loss: 2.2803377439722117, Learning Rate: 0.00125
Epoch 387, Loss: 2.28032492517161, Learning Rate: 0.00125
Epoch 388, Loss: 2.2803121111112437, Learning Rate: 0.00125
Epoch 389, Loss: 2.280299298121501, Learning Rate: 0.00125
Epoch 390, Loss: 2.280286486388047, Learning Rate: 0.00125
Epoch 391, Loss: 2.280273677153538, Learning Rate: 0.00125
Epoch 392, Loss: 2.2802608706371723, Learning Rate: 0.00125
Epoch 393, Loss: 2.2802480671202234, Learning Rate: 0.00125
Epoch 394, Loss: 2.280235265759801, Learning Rate: 0.00125
Epoch 395, Loss: 2.280222467461933, Learning Rate: 0.00125
Epoch 396, Loss: 2.2802096693404312, Learning Rate: 0.00125
Epoch 397, Loss: 2.2801968729013913, Learning Rate: 0.00125
Epoch 398, Loss: 2.2801840789995564, Learning Rate: 0.00125
Epoch 399, Loss: 2.2801712882426326, Learning Rate: 0.00125
Epoch 400, Loss: 2.2801585016541344, Learning Rate: 0.00125
Epoch 401, Loss: 2.280145717510445, Learning Rate: 0.000625
Epoch 402, Loss: 2.2801393260361458, Learning Rate: 0.000625
Epoch 403, Loss: 2.2801329348016317, Learning Rate: 0.000625
Epoch 404, Loss: 2.280126544742857, Learning Rate: 0.000625
Epoch 405, Loss: 2.280120155135784, Learning Rate: 0.000625
Epoch 406, Loss: 2.280113765330499, Learning Rate: 0.000625
```

```
Epoch 407, Loss: 2.2801073761444255, Learning Rate: 0.000625
Epoch 408, Loss: 2.280100987747558, Learning Rate: 0.000625
Epoch 409, Loss: 2.2800945997420476, Learning Rate: 0.000625
Epoch 410, Loss: 2.280088212012627, Learning Rate: 0.000625
Epoch 411, Loss: 2.280081823815099, Learning Rate: 0.000625
Epoch 412, Loss: 2.2800754363702347, Learning Rate: 0.000625
Epoch 413, Loss: 2.2800690495754674, Learning Rate: 0.000625
Epoch 414, Loss: 2.28006266322891, Learning Rate: 0.000625
Epoch 415, Loss: 2.280056278185312, Learning Rate: 0.000625
Epoch 416, Loss: 2.2800498930926754, Learning Rate: 0.000625
Epoch 417, Loss: 2.280043508235189, Learning Rate: 0.000625
Epoch 418, Loss: 2.280037124065062, Learning Rate: 0.000625
Epoch 419, Loss: 2.2800307406887095, Learning Rate: 0.000625
Epoch 420, Loss: 2.2800243575198857, Learning Rate: 0.000625
Epoch 421, Loss: 2.2800179749689367, Learning Rate: 0.000625
Epoch 422, Loss: 2.2800115926427367, Learning Rate: 0.000625
Epoch 423, Loss: 2.2800052112451987, Learning Rate: 0.000625
Epoch 424, Loss: 2.2799988297992786, Learning Rate: 0.000625
Epoch 425, Loss: 2.2799924497457633, Learning Rate: 0.000625
Epoch 426, Loss: 2.2799860703107644, Learning Rate: 0.000625
Epoch 427, Loss: 2.2799796921043876, Learning Rate: 0.000625
Epoch 428, Loss: 2.2799733140055274, Learning Rate: 0.000625
Epoch 429, Loss: 2.2799669364985053, Learning Rate: 0.000625
Epoch 430, Loss: 2.2799605594826238, Learning Rate: 0.000625
Epoch 431, Loss: 2.279954183025487, Learning Rate: 0.000625
Epoch 432, Loss: 2.2799478066117063, Learning Rate: 0.000625
Epoch 433, Loss: 2.279941431366801, Learning Rate: 0.000625
Epoch 434, Loss: 2.2799350559578437, Learning Rate: 0.000625
Epoch 435, Loss: 2.2799286818268634, Learning Rate: 0.000625
Epoch 436, Loss: 2.2799223075974013, Learning Rate: 0.000625
Epoch 437, Loss: 2.279915933473243, Learning Rate: 0.000625
Epoch 438, Loss: 2.279909559874816, Learning Rate: 0.000625
Epoch 439, Loss: 2.2799031873749884, Learning Rate: 0.000625
Epoch 440, Loss: 2.2798968147532177, Learning Rate: 0.000625
Epoch 441, Loss: 2.279890443470116, Learning Rate: 0.000625
Epoch 442, Loss: 2.279884071932277, Learning Rate: 0.000625
Epoch 443, Loss: 2.2798777006393514, Learning Rate: 0.000625
Epoch 444, Loss: 2.279871330020803, Learning Rate: 0.000625
Epoch 445, Loss: 2.2798649600539846, Learning Rate: 0.000625
Epoch 446, Loss: 2.279858590471486, Learning Rate: 0.000625
Epoch 447, Loss: 2.2798522221194193, Learning Rate: 0.000625
Epoch 448, Loss: 2.27984585552566, Learning Rate: 0.000625
Epoch 449, Loss: 2.2798394905103123, Learning Rate: 0.000625
Epoch 450, Loss: 2.2798331254509576, Learning Rate: 0.000625
Epoch 451, Loss: 2.2798267610706775, Learning Rate: 0.000625
Epoch 452, Loss: 2.2798203974802957, Learning Rate: 0.000625
Epoch 453, Loss: 2.2798140343886395, Learning Rate: 0.000625
Epoch 454, Loss: 2.2798076713958815, Learning Rate: 0.000625
```

```
Epoch 455, Loss: 2.2798013095190908, Learning Rate: 0.000625
Epoch 456, Loss: 2.2797949477899557, Learning Rate: 0.000625
Epoch 457, Loss: 2.2797885870886248, Learning Rate: 0.000625
Epoch 458, Loss: 2.2797822267992407, Learning Rate: 0.000625
Epoch 459, Loss: 2.2797758676388242, Learning Rate: 0.000625
Epoch 460, Loss: 2.2797695079365647, Learning Rate: 0.000625
Epoch 461, Loss: 2.2797631489475108, Learning Rate: 0.000625
Epoch 462, Loss: 2.279756790795328, Learning Rate: 0.000625
Epoch 463, Loss: 2.279750433875704, Learning Rate: 0.000625
Epoch 464, Loss: 2.2797440775300446, Learning Rate: 0.000625
Epoch 465, Loss: 2.2797377216380745, Learning Rate: 0.000625
Epoch 466, Loss: 2.2797313661801613, Learning Rate: 0.000625
Epoch 467, Loss: 2.279725011847133, Learning Rate: 0.000625
Epoch 468, Loss: 2.2797186568094108, Learning Rate: 0.000625
Epoch 469, Loss: 2.27971230250505, Learning Rate: 0.000625
Epoch 470, Loss: 2.2797059490652565, Learning Rate: 0.000625
Epoch 471, Loss: 2.279699596272814, Learning Rate: 0.000625
Epoch 472, Loss: 2.27969324348899, Learning Rate: 0.000625
Epoch 473, Loss: 2.279686891396297, Learning Rate: 0.000625
Epoch 474, Loss: 2.279680540029724, Learning Rate: 0.000625
Epoch 475, Loss: 2.2796741891346644, Learning Rate: 0.000625
Epoch 476, Loss: 2.279667838768004, Learning Rate: 0.000625
Epoch 477, Loss: 2.279661488454158, Learning Rate: 0.000625
Epoch 478, Loss: 2.2796551396040914, Learning Rate: 0.000625
Epoch 479, Loss: 2.279648790805274, Learning Rate: 0.000625
Epoch 480, Loss: 2.2796424422199233, Learning Rate: 0.000625
Epoch 481, Loss: 2.2796360942965386, Learning Rate: 0.000625
Epoch 482, Loss: 2.279629746947265, Learning Rate: 0.000625
Epoch 483, Loss: 2.279623399823604, Learning Rate: 0.000625
Epoch 484, Loss: 2.279617053713892, Learning Rate: 0.000625
Epoch 485, Loss: 2.2796107073489944, Learning Rate: 0.000625
Epoch 486, Loss: 2.2796043621069835, Learning Rate: 0.000625
Epoch 487, Loss: 2.279598016699209, Learning Rate: 0.000625
Epoch 488, Loss: 2.2795916724231544, Learning Rate: 0.000625
Epoch 489, Loss: 2.2795853285600804, Learning Rate: 0.000625
Epoch 490, Loss: 2.2795789851934396, Learning Rate: 0.000625
Epoch 491, Loss: 2.279572642402384, Learning Rate: 0.000625
Epoch 492, Loss: 2.2795663002616062, Learning Rate: 0.000625
Epoch 493, Loss: 2.279559958031924, Learning Rate: 0.000625
Epoch 494, Loss: 2.279553616847709, Learning Rate: 0.000625
Epoch 495, Loss: 2.279547275866126, Learning Rate: 0.000625
Epoch 496, Loss: 2.279540936029116, Learning Rate: 0.000625
Epoch 497, Loss: 2.27953459556971, Learning Rate: 0.000625
Epoch 498, Loss: 2.279528256399067, Learning Rate: 0.000625
Epoch 499, Loss: 2.2795219186828586, Learning Rate: 0.000625
Epoch 500, Loss: 2.2795155816446053, Learning Rate: 0.000625
Epoch 501, Loss: 2.2795092445969978, Learning Rate: 0.0003125
Epoch 502, Loss: 2.2795060760638126, Learning Rate: 0.0003125
```

```
Epoch 503, Loss: 2.2795029078650373, Learning Rate: 0.0003125
Epoch 504, Loss: 2.279499739754728, Learning Rate: 0.0003125
Epoch 505, Loss: 2.2794965718508378, Learning Rate: 0.0003125
Epoch 506, Loss: 2.2794934040617227, Learning Rate: 0.0003125
Epoch 507, Loss: 2.2794902363410725, Learning Rate: 0.0003125
Epoch 508, Loss: 2.2794870687947175, Learning Rate: 0.0003125
Epoch 509, Loss: 2.2794839013433337, Learning Rate: 0.0003125
Epoch 510, Loss: 2.279480733960356, Learning Rate: 0.0003125
Epoch 511, Loss: 2.2794775668035303, Learning Rate: 0.0003125
Epoch 512, Loss: 2.279474399634716, Learning Rate: 0.0003125
Epoch 513, Loss: 2.2794712326385493, Learning Rate: 0.0003125
Epoch 514, Loss: 2.2794680657908932, Learning Rate: 0.0003125
Epoch 515, Loss: 2.279464898922467, Learning Rate: 0.0003125
Epoch 516, Loss: 2.2794617324610513, Learning Rate: 0.0003125
Epoch 517, Loss: 2.2794585657626136, Learning Rate: 0.0003125
Epoch 518, Loss: 2.2794553993297786, Learning Rate: 0.0003125
Epoch 519, Loss: 2.279452233101476, Learning Rate: 0.0003125
Epoch 520, Loss: 2.279449067210112, Learning Rate: 0.0003125
Epoch 521, Loss: 2.2794459020707727, Learning Rate: 0.0003125
Epoch 522, Loss: 2.2794427365994565, Learning Rate: 0.0003125
Epoch 523, Loss: 2.279439571331298, Learning Rate: 0.0003125
Epoch 524, Loss: 2.2794364063987445, Learning Rate: 0.0003125
Epoch 525, Loss: 2.2794332412881033, Learning Rate: 0.0003125
Epoch 526, Loss: 2.279430076670088, Learning Rate: 0.0003125
Epoch 527, Loss: 2.2794269118202184, Learning Rate: 0.0003125
Epoch 528, Loss: 2.2794237473179586, Learning Rate: 0.0003125
Epoch 529, Loss: 2.279420582767819, Learning Rate: 0.0003125
Epoch 530, Loss: 2.2794174183799174, Learning Rate: 0.0003125
Epoch 531, Loss: 2.279414254122951, Learning Rate: 0.0003125
Epoch 532, Loss: 2.2794110900794697, Learning Rate: 0.0003125
Epoch 533, Loss: 2.2794079259013715, Learning Rate: 0.0003125
Epoch 534, Loss: 2.279404762365724, Learning Rate: 0.0003125
Epoch 535, Loss: 2.2794015983885383, Learning Rate: 0.0003125
Epoch 536, Loss: 2.2793984345939666, Learning Rate: 0.0003125
Epoch 537, Loss: 2.279395271043456, Learning Rate: 0.0003125
Epoch 538, Loss: 2.2793921075290626, Learning Rate: 0.0003125
Epoch 539, Loss: 2.2793889442906137, Learning Rate: 0.0003125
Epoch 540, Loss: 2.2793857811520213, Learning Rate: 0.0003125
Epoch 541, Loss: 2.2793826179678707, Learning Rate: 0.0003125
Epoch 542, Loss: 2.2793794547738235, Learning Rate: 0.0003125
Epoch 543, Loss: 2.2793762919805034, Learning Rate: 0.0003125
Epoch 544, Loss: 2.279373129323374, Learning Rate: 0.0003125
Epoch 545, Loss: 2.279369966843082, Learning Rate: 0.0003125
Epoch 546, Loss: 2.2793668043757425, Learning Rate: 0.0003125
Epoch 547, Loss: 2.27936364231787, Learning Rate: 0.0003125
Epoch 548, Loss: 2.279360480073984, Learning Rate: 0.0003125
Epoch 549, Loss: 2.2793573179280355, Learning Rate: 0.0003125
Epoch 550, Loss: 2.2793541558602124, Learning Rate: 0.0003125
```

```
Epoch 551, Loss: 2.279350994119281, Learning Rate: 0.0003125
Epoch 552, Loss: 2.2793478323256346, Learning Rate: 0.0003125
Epoch 553, Loss: 2.27934467066558, Learning Rate: 0.0003125
Epoch 554, Loss: 2.2793415092335394, Learning Rate: 0.0003125
Epoch 555, Loss: 2.279338347765956, Learning Rate: 0.0003125
Epoch 556, Loss: 2.279335186548302, Learning Rate: 0.0003125
Epoch 557, Loss: 2.279332025332639, Learning Rate: 0.0003125
Epoch 558, Loss: 2.279328864182419, Learning Rate: 0.0003125
Epoch 559, Loss: 2.279325703423498, Learning Rate: 0.0003125
Epoch 560, Loss: 2.2793225423825714, Learning Rate: 0.0003125
Epoch 561, Loss: 2.279319381455661, Learning Rate: 0.0003125
Epoch 562, Loss: 2.279316220916494, Learning Rate: 0.0003125
Epoch 563, Loss: 2.2793130602417797, Learning Rate: 0.0003125
Epoch 564, Loss: 2.279309899828162, Learning Rate: 0.0003125
Epoch 565, Loss: 2.279306739446831, Learning Rate: 0.0003125
Epoch 566, Loss: 2.2793035794188468, Learning Rate: 0.0003125
Epoch 567, Loss: 2.2793004191911352, Learning Rate: 0.0003125
Epoch 568, Loss: 2.2792972594677723, Learning Rate: 0.0003125
Epoch 569, Loss: 2.2792941001998215, Learning Rate: 0.0003125
Epoch 570, Loss: 2.2792909411666455, Learning Rate: 0.0003125
Epoch 571, Loss: 2.279287782032499, Learning Rate: 0.0003125
Epoch 572, Loss: 2.2792846233381137, Learning Rate: 0.0003125
Epoch 573, Loss: 2.279281464530033, Learning Rate: 0.0003125
Epoch 574, Loss: 2.2792783058405366, Learning Rate: 0.0003125
Epoch 575, Loss: 2.279275147448275, Learning Rate: 0.0003125
Epoch 576, Loss: 2.2792719891075843, Learning Rate: 0.0003125
Epoch 577, Loss: 2.279268830812635, Learning Rate: 0.0003125
Epoch 578, Loss: 2.2792656727881493, Learning Rate: 0.0003125
Epoch 579, Loss: 2.27926251486355, Learning Rate: 0.0003125
Epoch 580, Loss: 2.2792593568521644, Learning Rate: 0.0003125
Epoch 581, Loss: 2.2792561991565554, Learning Rate: 0.0003125
Epoch 582, Loss: 2.2792530414566547, Learning Rate: 0.0003125
Epoch 583, Loss: 2.279249883964145, Learning Rate: 0.0003125
Epoch 584, Loss: 2.279246726731337, Learning Rate: 0.0003125
Epoch 585, Loss: 2.2792435693073996, Learning Rate: 0.0003125
Epoch 586, Loss: 2.279240411914686, Learning Rate: 0.0003125
Epoch 587, Loss: 2.2792372545925144, Learning Rate: 0.0003125
Epoch 588, Loss: 2.279234097339122, Learning Rate: 0.0003125
Epoch 589, Loss: 2.279230939604868, Learning Rate: 0.0003125
Epoch 590, Loss: 2.279227781864423, Learning Rate: 0.0003125
Epoch 591, Loss: 2.27922462435434, Learning Rate: 0.0003125
Epoch 592, Loss: 2.2792214669534747, Learning Rate: 0.0003125
Epoch 593, Loss: 2.279218309465727, Learning Rate: 0.0003125
Epoch 594, Loss: 2.2792151523242072, Learning Rate: 0.0003125
Epoch 595, Loss: 2.279211995145662, Learning Rate: 0.0003125
Epoch 596, Loss: 2.279208838276048, Learning Rate: 0.0003125
Epoch 597, Loss: 2.279205681516309, Learning Rate: 0.0003125
Epoch 598, Loss: 2.2792025244605556, Learning Rate: 0.0003125
```

```
Epoch 599, Loss: 2.2791993673698077, Learning Rate: 0.0003125
Epoch 600, Loss: 2.2791962104706105, Learning Rate: 0.0003125
Epoch 601, Loss: 2.279193053726595, Learning Rate: 0.00015625
Epoch 602, Loss: 2.2791914752694753, Learning Rate: 0.00015625
Epoch 603, Loss: 2.27918989685281, Learning Rate: 0.00015625
Epoch 604, Loss: 2.2791883185226336, Learning Rate: 0.00015625
Epoch 605, Loss: 2.279186740291139, Learning Rate: 0.00015625
Epoch 606, Loss: 2.2791851620050068, Learning Rate: 0.00015625
Epoch 607, Loss: 2.2791835838722436, Learning Rate: 0.00015625
Epoch 608, Loss: 2.279182005680706, Learning Rate: 0.00015625
Epoch 609, Loss: 2.27918042719069, Learning Rate: 0.00015625
Epoch 610, Loss: 2.2791788488879208, Learning Rate: 0.00015625
Epoch 611, Loss: 2.27917727052369, Learning Rate: 0.00015625
Epoch 612, Loss: 2.2791756921628084, Learning Rate: 0.00015625
Epoch 613, Loss: 2.279174113913827, Learning Rate: 0.00015625
Epoch 614, Loss: 2.2791725356214725, Learning Rate: 0.00015625
Epoch 615, Loss: 2.2791709573645225, Learning Rate: 0.00015625
Epoch 616, Loss: 2.2791693791484575, Learning Rate: 0.00015625
Epoch 617, Loss: 2.279167800981538, Learning Rate: 0.00015625
Epoch 618, Loss: 2.279166222830134, Learning Rate: 0.00015625
Epoch 619, Loss: 2.279164644671858, Learning Rate: 0.00015625
Epoch 620, Loss: 2.279163066594698, Learning Rate: 0.00015625
Epoch 621, Loss: 2.2791614885128184, Learning Rate: 0.00015625
Epoch 622, Loss: 2.279159910430249, Learning Rate: 0.00015625
Epoch 623, Loss: 2.2791583324546663, Learning Rate: 0.00015625
Epoch 624, Loss: 2.279156754443327, Learning Rate: 0.00015625
Epoch 625, Loss: 2.279155176468318, Learning Rate: 0.00015625
Epoch 626, Loss: 2.279153598577069, Learning Rate: 0.00015625
Epoch 627, Loss: 2.279152020610825, Learning Rate: 0.00015625
Epoch 628, Loss: 2.2791504427662814, Learning Rate: 0.00015625
Epoch 629, Loss: 2.2791488649281035, Learning Rate: 0.00015625
Epoch 630, Loss: 2.279147287017321, Learning Rate: 0.00015625
Epoch 631, Loss: 2.2791457092998066, Learning Rate: 0.00015625
Epoch 632, Loss: 2.279144131541331, Learning Rate: 0.00015625
Epoch 633, Loss: 2.2791425537881547, Learning Rate: 0.00015625
Epoch 634, Loss: 2.2791409761945194, Learning Rate: 0.00015625
Epoch 635, Loss: 2.279139398513991, Learning Rate: 0.00015625
Epoch 636, Loss: 2.27913782086914, Learning Rate: 0.00015625
Epoch 637, Loss: 2.2791362433381415, Learning Rate: 0.00015625
Epoch 638, Loss: 2.2791346657621574, Learning Rate: 0.00015625
Epoch 639, Loss: 2.2791330882459917, Learning Rate: 0.00015625
Epoch 640, Loss: 2.2791315107387335, Learning Rate: 0.00015625
Epoch 641, Loss: 2.2791299332139965, Learning Rate: 0.00015625
Epoch 642, Loss: 2.2791283558361695, Learning Rate: 0.00015625
Epoch 643, Loss: 2.27912677839878, Learning Rate: 0.00015625
Epoch 644, Loss: 2.2791252008060896, Learning Rate: 0.00015625
Epoch 645, Loss: 2.2791236232750594, Learning Rate: 0.00015625
Epoch 646, Loss: 2.2791220457843315, Learning Rate: 0.00015625
```

Epoch 647, Loss: 2.2791204683437107, Learning Rate: 0.00015625
Epoch 648, Loss: 2.2791188908435887, Learning Rate: 0.00015625
Epoch 649, Loss: 2.279117313440821, Learning Rate: 0.00015625
Epoch 650, Loss: 2.2791157360562058, Learning Rate: 0.00015625
Epoch 651, Loss: 2.279114158649, Learning Rate: 0.00015625
Epoch 652, Loss: 2.2791125813813875, Learning Rate: 0.00015625
Epoch 653, Loss: 2.2791110040428864, Learning Rate: 0.00015625
Epoch 654, Loss: 2.2791094267051824, Learning Rate: 0.00015625
Epoch 655, Loss: 2.27910784954436, Learning Rate: 0.00015625
Epoch 656, Loss: 2.2791062722529607, Learning Rate: 0.00015625
Epoch 657, Loss: 2.279104695027792, Learning Rate: 0.00015625
Epoch 658, Loss: 2.2791031180063754, Learning Rate: 0.00015625
Epoch 659, Loss: 2.2791015409849047, Learning Rate: 0.00015625
Epoch 660, Loss: 2.2790999640536453, Learning Rate: 0.00015625
Epoch 661, Loss: 2.279098387207257, Learning Rate: 0.00015625
Epoch 662, Loss: 2.279096810282218, Learning Rate: 0.00015625
Epoch 663, Loss: 2.27909523347537, Learning Rate: 0.00015625
Epoch 664, Loss: 2.279093656680277, Learning Rate: 0.00015625
Epoch 665, Loss: 2.2790920798598737, Learning Rate: 0.00015625
Epoch 666, Loss: 2.2790905031174558, Learning Rate: 0.00015625
Epoch 667, Loss: 2.279088926253425, Learning Rate: 0.00015625
Epoch 668, Loss: 2.2790873492948087, Learning Rate: 0.00015625
Epoch 669, Loss: 2.2790857723220235, Learning Rate: 0.00015625
Epoch 670, Loss: 2.2790841953256087, Learning Rate: 0.00015625
Epoch 671, Loss: 2.2790826184936943, Learning Rate: 0.00015625
Epoch 672, Loss: 2.2790810415882925, Learning Rate: 0.00015625
Epoch 673, Loss: 2.279079464669775, Learning Rate: 0.00015625
Epoch 674, Loss: 2.279077887916576, Learning Rate: 0.00015625
Epoch 675, Loss: 2.279076311099809, Learning Rate: 0.00015625
Epoch 676, Loss: 2.2790747343199373, Learning Rate: 0.00015625
Epoch 677, Loss: 2.2790731575862107, Learning Rate: 0.00015625
Epoch 678, Loss: 2.279071580845468, Learning Rate: 0.00015625
Epoch 679, Loss: 2.2790700042152396, Learning Rate: 0.00015625
Epoch 680, Loss: 2.279068427553873, Learning Rate: 0.00015625
Epoch 681, Loss: 2.279066850964157, Learning Rate: 0.00015625
Epoch 682, Loss: 2.279065274396197, Learning Rate: 0.00015625
Epoch 683, Loss: 2.2790636978259444, Learning Rate: 0.00015625
Epoch 684, Loss: 2.2790621213158597, Learning Rate: 0.00015625
Epoch 685, Loss: 2.2790605448535723, Learning Rate: 0.00015625
Epoch 686, Loss: 2.279058968336516, Learning Rate: 0.00015625
Epoch 687, Loss: 2.2790573919436463, Learning Rate: 0.00015625
Epoch 688, Loss: 2.2790558155203096, Learning Rate: 0.00015625
Epoch 689, Loss: 2.2790542390859723, Learning Rate: 0.00015625
Epoch 690, Loss: 2.2790526628145495, Learning Rate: 0.00015625
Epoch 691, Loss: 2.2790510864506377, Learning Rate: 0.00015625
Epoch 692, Loss: 2.279049510086174, Learning Rate: 0.00015625
Epoch 693, Loss: 2.27904793391344, Learning Rate: 0.00015625
Epoch 694, Loss: 2.2790463576318714, Learning Rate: 0.00015625

```
Epoch 695, Loss: 2.2790447813520345, Learning Rate: 0.00015625
Epoch 696, Loss: 2.279043205237798, Learning Rate: 0.00015625
Epoch 697, Loss: 2.279041629043564, Learning Rate: 0.00015625
Epoch 698, Loss: 2.279040052899333, Learning Rate: 0.00015625
Epoch 699, Loss: 2.2790384768187337, Learning Rate: 0.00015625
Epoch 700, Loss: 2.2790369006937263, Learning Rate: 0.00015625
Epoch 701, Loss: 2.279035324646271, Learning Rate: 7.8125e-05
Epoch 702, Loss: 2.2790345365965354, Learning Rate: 7.8125e-05
Epoch 703, Loss: 2.27903374858134, Learning Rate: 7.8125e-05
Epoch 704, Loss: 2.2790329605925588, Learning Rate: 7.8125e-05
Epoch 705, Loss: 2.2790321725797904, Learning Rate: 7.8125e-05
Epoch 706, Loss: 2.2790313846100876, Learning Rate: 7.8125e-05
Epoch 707, Loss: 2.2790305966215367, Learning Rate: 7.8125e-05
Epoch 708, Loss: 2.2790298086194674, Learning Rate: 7.8125e-05
Epoch 709, Loss: 2.279029020694505, Learning Rate: 7.8125e-05
Epoch 710, Loss: 2.279028232718119, Learning Rate: 7.8125e-05
Epoch 711, Loss: 2.279027444735046, Learning Rate: 7.8125e-05
Epoch 712, Loss: 2.2790266568367086, Learning Rate: 7.8125e-05
Epoch 713, Loss: 2.27902586887006, Learning Rate: 7.8125e-05
Epoch 714, Loss: 2.2790250809176893, Learning Rate: 7.8125e-05
Epoch 715, Loss: 2.2790242930292726, Learning Rate: 7.8125e-05
Epoch 716, Loss: 2.279023505086249, Learning Rate: 7.8125e-05
Epoch 717, Loss: 2.2790227171655464, Learning Rate: 7.8125e-05
Epoch 718, Loss: 2.2790219292812077, Learning Rate: 7.8125e-05
Epoch 719, Loss: 2.279021141369231, Learning Rate: 7.8125e-05
Epoch 720, Loss: 2.2790203534780007, Learning Rate: 7.8125e-05
Epoch 721, Loss: 2.279019565596758, Learning Rate: 7.8125e-05
Epoch 722, Loss: 2.2790187777042545, Learning Rate: 7.8125e-05
Epoch 723, Loss: 2.279017989850405, Learning Rate: 7.8125e-05
Epoch 724, Loss: 2.2790172019788852, Learning Rate: 7.8125e-05
Epoch 725, Loss: 2.279016414105523, Learning Rate: 7.8125e-05
Epoch 726, Loss: 2.2790156262708896, Learning Rate: 7.8125e-05
Epoch 727, Loss: 2.2790148384177904, Learning Rate: 7.8125e-05
Epoch 728, Loss: 2.2790140505826244, Learning Rate: 7.8125e-05
Epoch 729, Loss: 2.2790132627531277, Learning Rate: 7.8125e-05
Epoch 730, Loss: 2.2790124749152314, Learning Rate: 7.8125e-05
Epoch 731, Loss: 2.279011687120227, Learning Rate: 7.8125e-05
Epoch 732, Loss: 2.279010899302614, Learning Rate: 7.8125e-05
Epoch 733, Loss: 2.279010111471587, Learning Rate: 7.8125e-05
Epoch 734, Loss: 2.279009323703648, Learning Rate: 7.8125e-05
Epoch 735, Loss: 2.2790085359118724, Learning Rate: 7.8125e-05
Epoch 736, Loss: 2.2790077481046143, Learning Rate: 7.8125e-05
Epoch 737, Loss: 2.279006960387397, Learning Rate: 7.8125e-05
Epoch 738, Loss: 2.2790061726251256, Learning Rate: 7.8125e-05
Epoch 739, Loss: 2.279005384868066, Learning Rate: 7.8125e-05
Epoch 740, Loss: 2.2790045971808834, Learning Rate: 7.8125e-05
Epoch 741, Loss: 2.279003809423666, Learning Rate: 7.8125e-05
Epoch 742, Loss: 2.2790030216887054, Learning Rate: 7.8125e-05
```

```
Epoch 743, Loss: 2.2790022340178644, Learning Rate: 7.8125e-05
Epoch 744, Loss: 2.2790014462840817, Learning Rate: 7.8125e-05
Epoch 745, Loss: 2.279000658587878, Learning Rate: 7.8125e-05
Epoch 746, Loss: 2.278999870904734, Learning Rate: 7.8125e-05
Epoch 747, Loss: 2.2789990831994134, Learning Rate: 7.8125e-05
Epoch 748, Loss: 2.2789982955446906, Learning Rate: 7.8125e-05
Epoch 749, Loss: 2.278997507869342, Learning Rate: 7.8125e-05
Epoch 750, Loss: 2.278996720186093, Learning Rate: 7.8125e-05
Epoch 751, Loss: 2.2789959325537694, Learning Rate: 7.8125e-05
Epoch 752, Loss: 2.2789951448886203, Learning Rate: 7.8125e-05
Epoch 753, Loss: 2.2789943572422295, Learning Rate: 7.8125e-05
Epoch 754, Loss: 2.2789935696162096, Learning Rate: 7.8125e-05
Epoch 755, Loss: 2.278992781962846, Learning Rate: 7.8125e-05
Epoch 756, Loss: 2.2789919943551884, Learning Rate: 7.8125e-05
Epoch 757, Loss: 2.278991206750149, Learning Rate: 7.8125e-05
Epoch 758, Loss: 2.278990419104023, Learning Rate: 7.8125e-05
Epoch 759, Loss: 2.2789896315341904, Learning Rate: 7.8125e-05
Epoch 760, Loss: 2.278988843933128, Learning Rate: 7.8125e-05
Epoch 761, Loss: 2.2789880563128633, Learning Rate: 7.8125e-05
Epoch 762, Loss: 2.278987268777926, Learning Rate: 7.8125e-05
Epoch 763, Loss: 2.278986481181271, Learning Rate: 7.8125e-05
Epoch 764, Loss: 2.2789856935921513, Learning Rate: 7.8125e-05
Epoch 765, Loss: 2.2789849060648404, Learning Rate: 7.8125e-05
Epoch 766, Loss: 2.2789841184834465, Learning Rate: 7.8125e-05
Epoch 767, Loss: 2.2789833309336895, Learning Rate: 7.8125e-05
Epoch 768, Loss: 2.278982543413659, Learning Rate: 7.8125e-05
Epoch 769, Loss: 2.278981755855903, Learning Rate: 7.8125e-05
Epoch 770, Loss: 2.2789809683303015, Learning Rate: 7.8125e-05
Epoch 771, Loss: 2.2789801808285173, Learning Rate: 7.8125e-05
Epoch 772, Loss: 2.2789793932808338, Learning Rate: 7.8125e-05
Epoch 773, Loss: 2.27897860579585, Learning Rate: 7.8125e-05
Epoch 774, Loss: 2.278977818290565, Learning Rate: 7.8125e-05
Epoch 775, Loss: 2.2789770307870496, Learning Rate: 7.8125e-05
Epoch 776, Loss: 2.278976243315947, Learning Rate: 7.8125e-05
Epoch 777, Loss: 2.2789754558229482, Learning Rate: 7.8125e-05
Epoch 778, Loss: 2.2789746683575816, Learning Rate: 7.8125e-05
Epoch 779, Loss: 2.2789738809017246, Learning Rate: 7.8125e-05
Epoch 780, Loss: 2.2789730934162886, Learning Rate: 7.8125e-05
Epoch 781, Loss: 2.27897230598285, Learning Rate: 7.8125e-05
Epoch 782, Loss: 2.2789715185397244, Learning Rate: 7.8125e-05
Epoch 783, Loss: 2.278970731081191, Learning Rate: 7.8125e-05
Epoch 784, Loss: 2.27896994367611, Learning Rate: 7.8125e-05
Epoch 785, Loss: 2.278969156236549, Learning Rate: 7.8125e-05
Epoch 786, Loss: 2.278968368807969, Learning Rate: 7.8125e-05
Epoch 787, Loss: 2.2789675814336174, Learning Rate: 7.8125e-05
Epoch 788, Loss: 2.2789667940008966, Learning Rate: 7.8125e-05
Epoch 789, Loss: 2.278966006584797, Learning Rate: 7.8125e-05
Epoch 790, Loss: 2.2789652192433616, Learning Rate: 7.8125e-05
```

```
Epoch 791, Loss: 2.278964431832642, Learning Rate: 7.8125e-05
Epoch 792, Loss: 2.2789636444396266, Learning Rate: 7.8125e-05
Epoch 793, Loss: 2.278962857093198, Learning Rate: 7.8125e-05
Epoch 794, Loss: 2.2789620697235278, Learning Rate: 7.8125e-05
Epoch 795, Loss: 2.2789612823672316, Learning Rate: 7.8125e-05
Epoch 796, Loss: 2.2789604950430347, Learning Rate: 7.8125e-05
Epoch 797, Loss: 2.2789597077191486, Learning Rate: 7.8125e-05
Epoch 798, Loss: 2.2789589204413234, Learning Rate: 7.8125e-05
Epoch 799, Loss: 2.2789581331459066, Learning Rate: 7.8125e-05
Epoch 800, Loss: 2.2789573458591814, Learning Rate: 7.8125e-05
Epoch 801, Loss: 2.27895655858776, Learning Rate: 3.90625e-05
Epoch 802, Loss: 2.2789561649710848, Learning Rate: 3.90625e-05
Epoch 803, Loss: 2.27895577135451, Learning Rate: 3.90625e-05
Epoch 804, Loss: 2.2789553777420215, Learning Rate: 3.90625e-05
Epoch 805, Loss: 2.2789549841337813, Learning Rate: 3.90625e-05
Epoch 806, Loss: 2.278954590536535, Learning Rate: 3.90625e-05
Epoch 807, Loss: 2.2789541969387566, Learning Rate: 3.90625e-05
Epoch 808, Loss: 2.2789538033312784, Learning Rate: 3.90625e-05
Epoch 809, Loss: 2.2789534097339477, Learning Rate: 3.90625e-05
Epoch 810, Loss: 2.2789530161416893, Learning Rate: 3.90625e-05
Epoch 811, Loss: 2.27895262254853, Learning Rate: 3.90625e-05
Epoch 812, Loss: 2.2789522289466007, Learning Rate: 3.90625e-05
Epoch 813, Loss: 2.27895183535796, Learning Rate: 3.90625e-05
Epoch 814, Loss: 2.2789514417813073, Learning Rate: 3.90625e-05
Epoch 815, Loss: 2.2789510481782482, Learning Rate: 3.90625e-05
Epoch 816, Loss: 2.2789506545879616, Learning Rate: 3.90625e-05
Epoch 817, Loss: 2.2789502610256824, Learning Rate: 3.90625e-05
Epoch 818, Loss: 2.2789498674279005, Learning Rate: 3.90625e-05
Epoch 819, Loss: 2.2789494738380904, Learning Rate: 3.90625e-05
Epoch 820, Loss: 2.278949080281267, Learning Rate: 3.90625e-05
Epoch 821, Loss: 2.2789486866892137, Learning Rate: 3.90625e-05
Epoch 822, Loss: 2.278948293109797, Learning Rate: 3.90625e-05
Epoch 823, Loss: 2.27894789954525, Learning Rate: 3.90625e-05
Epoch 824, Loss: 2.2789475059622197, Learning Rate: 3.90625e-05
Epoch 825, Loss: 2.2789471123969527, Learning Rate: 3.90625e-05
Epoch 826, Loss: 2.27894671883054, Learning Rate: 3.90625e-05
Epoch 827, Loss: 2.278946325253904, Learning Rate: 3.90625e-05
Epoch 828, Loss: 2.278945931696433, Learning Rate: 3.90625e-05
Epoch 829, Loss: 2.2789455381287436, Learning Rate: 3.90625e-05
Epoch 830, Loss: 2.278945144566208, Learning Rate: 3.90625e-05
Epoch 831, Loss: 2.2789447510081224, Learning Rate: 3.90625e-05
Epoch 832, Loss: 2.2789443574462105, Learning Rate: 3.90625e-05
Epoch 833, Loss: 2.2789439638931404, Learning Rate: 3.90625e-05
Epoch 834, Loss: 2.2789435703373893, Learning Rate: 3.90625e-05
Epoch 835, Loss: 2.2789431767325534, Learning Rate: 3.90625e-05
Epoch 836, Loss: 2.2789427831530444, Learning Rate: 3.90625e-05
Epoch 837, Loss: 2.278942389573224, Learning Rate: 3.90625e-05
Epoch 838, Loss: 2.2789419959743755, Learning Rate: 3.90625e-05
```

```
Epoch 839, Loss: 2.2789416023893647, Learning Rate: 3.90625e-05
Epoch 840, Loss: 2.278941208824371, Learning Rate: 3.90625e-05
Epoch 841, Loss: 2.2789408152304556, Learning Rate: 3.90625e-05
Epoch 842, Loss: 2.2789404216479165, Learning Rate: 3.90625e-05
Epoch 843, Loss: 2.2789400280818652, Learning Rate: 3.90625e-05
Epoch 844, Loss: 2.2789396345049537, Learning Rate: 3.90625e-05
Epoch 845, Loss: 2.278939240926876, Learning Rate: 3.90625e-05
Epoch 846, Loss: 2.278938847353139, Learning Rate: 3.90625e-05
Epoch 847, Loss: 2.27893845378612, Learning Rate: 3.90625e-05
Epoch 848, Loss: 2.278938060222669, Learning Rate: 3.90625e-05
Epoch 849, Loss: 2.2789376666458687, Learning Rate: 3.90625e-05
Epoch 850, Loss: 2.2789372730815236, Learning Rate: 3.90625e-05
Epoch 851, Loss: 2.2789368795327443, Learning Rate: 3.90625e-05
Epoch 852, Loss: 2.2789364859524923, Learning Rate: 3.90625e-05
Epoch 853, Loss: 2.2789360923982693, Learning Rate: 3.90625e-05
Epoch 854, Loss: 2.2789356988476372, Learning Rate: 3.90625e-05
Epoch 855, Loss: 2.2789353052811325, Learning Rate: 3.90625e-05
Epoch 856, Loss: 2.2789349117323985, Learning Rate: 3.90625e-05
Epoch 857, Loss: 2.278934518183103, Learning Rate: 3.90625e-05
Epoch 858, Loss: 2.2789341246193624, Learning Rate: 3.90625e-05
Epoch 859, Loss: 2.2789337310804556, Learning Rate: 3.90625e-05
Epoch 860, Loss: 2.2789333375363143, Learning Rate: 3.90625e-05
Epoch 861, Loss: 2.2789329439723662, Learning Rate: 3.90625e-05
Epoch 862, Loss: 2.2789325504380553, Learning Rate: 3.90625e-05
Epoch 863, Loss: 2.278932156903425, Learning Rate: 3.90625e-05
Epoch 864, Loss: 2.2789317633460007, Learning Rate: 3.90625e-05
Epoch 865, Loss: 2.2789313698072413, Learning Rate: 3.90625e-05
Epoch 866, Loss: 2.2789309762846486, Learning Rate: 3.90625e-05
Epoch 867, Loss: 2.2789305827332638, Learning Rate: 3.90625e-05
Epoch 868, Loss: 2.278930189205449, Learning Rate: 3.90625e-05
Epoch 869, Loss: 2.278929795668411, Learning Rate: 3.90625e-05
Epoch 870, Loss: 2.2789294021357893, Learning Rate: 3.90625e-05
Epoch 871, Loss: 2.2789290086149494, Learning Rate: 3.90625e-05
Epoch 872, Loss: 2.278928615076298, Learning Rate: 3.90625e-05
Epoch 873, Loss: 2.278928221552288, Learning Rate: 3.90625e-05
Epoch 874, Loss: 2.27892782803779, Learning Rate: 3.90625e-05
Epoch 875, Loss: 2.278927434503445, Learning Rate: 3.90625e-05
Epoch 876, Loss: 2.2789270409883557, Learning Rate: 3.90625e-05
Epoch 877, Loss: 2.2789266474742362, Learning Rate: 3.90625e-05
Epoch 878, Loss: 2.2789262539466546, Learning Rate: 3.90625e-05
Epoch 879, Loss: 2.278925860439521, Learning Rate: 3.90625e-05
Epoch 880, Loss: 2.2789254669218564, Learning Rate: 3.90625e-05
Epoch 881, Loss: 2.278925073399657, Learning Rate: 3.90625e-05
Epoch 882, Loss: 2.278924679903519, Learning Rate: 3.90625e-05
Epoch 883, Loss: 2.2789242863908603, Learning Rate: 3.90625e-05
Epoch 884, Loss: 2.2789238928679296, Learning Rate: 3.90625e-05
Epoch 885, Loss: 2.278923499365661, Learning Rate: 3.90625e-05
Epoch 886, Loss: 2.2789231058522494, Learning Rate: 3.90625e-05
```

Epoch 887, Loss: 2.2789227123216365, Learning Rate: 3.90625e-05
Epoch 888, Loss: 2.2789223188155687, Learning Rate: 3.90625e-05
Epoch 889, Loss: 2.2789219253125648, Learning Rate: 3.90625e-05
Epoch 890, Loss: 2.2789215316487703, Learning Rate: 3.90625e-05
Epoch 891, Loss: 2.2789211374556, Learning Rate: 3.90625e-05
Epoch 892, Loss: 2.2789207432670224, Learning Rate: 3.90625e-05
Epoch 893, Loss: 2.278920349080341, Learning Rate: 3.90625e-05
Epoch 894, Loss: 2.2789199548796435, Learning Rate: 3.90625e-05
Epoch 895, Loss: 2.2789195606945505, Learning Rate: 3.90625e-05
Epoch 896, Loss: 2.2789191665156108, Learning Rate: 3.90625e-05
Epoch 897, Loss: 2.2789187722640714, Learning Rate: 3.90625e-05
Epoch 898, Loss: 2.2789183780295548, Learning Rate: 3.90625e-05
Epoch 899, Loss: 2.2789179837930593, Learning Rate: 3.90625e-05
Epoch 900, Loss: 2.2789175895588833, Learning Rate: 3.90625e-05
Epoch 901, Loss: 2.278917195321034, Learning Rate: 1.953125e-05
Epoch 902, Loss: 2.278916998195701, Learning Rate: 1.953125e-05
Epoch 903, Loss: 2.2789168010849394, Learning Rate: 1.953125e-05
Epoch 904, Loss: 2.278916603960959, Learning Rate: 1.953125e-05
Epoch 905, Loss: 2.2789164068525016, Learning Rate: 1.953125e-05
Epoch 906, Loss: 2.2789162097384876, Learning Rate: 1.953125e-05
Epoch 907, Loss: 2.278916012618933, Learning Rate: 1.953125e-05
Epoch 908, Loss: 2.278915815510192, Learning Rate: 1.953125e-05
Epoch 909, Loss: 2.278915618397289, Learning Rate: 1.953125e-05
Epoch 910, Loss: 2.278915421282554, Learning Rate: 1.953125e-05
Epoch 911, Loss: 2.2789152241690696, Learning Rate: 1.953125e-05
Epoch 912, Loss: 2.27891502706022, Learning Rate: 1.953125e-05
Epoch 913, Loss: 2.2789148299478414, Learning Rate: 1.953125e-05
Epoch 914, Loss: 2.2789146328348218, Learning Rate: 1.953125e-05
Epoch 915, Loss: 2.278914435725214, Learning Rate: 1.953125e-05
Epoch 916, Loss: 2.2789142386211823, Learning Rate: 1.953125e-05
Epoch 917, Loss: 2.2789140415019222, Learning Rate: 1.953125e-05
Epoch 918, Loss: 2.2789138443999395, Learning Rate: 1.953125e-05
Epoch 919, Loss: 2.27891364728959, Learning Rate: 1.953125e-05
Epoch 920, Loss: 2.2789134501768875, Learning Rate: 1.953125e-05
Epoch 921, Loss: 2.2789132530742178, Learning Rate: 1.953125e-05
Epoch 922, Loss: 2.278913055964715, Learning Rate: 1.953125e-05
Epoch 923, Loss: 2.2789128588582543, Learning Rate: 1.953125e-05
Epoch 924, Loss: 2.2789126617477193, Learning Rate: 1.953125e-05
Epoch 925, Loss: 2.278912464644102, Learning Rate: 1.953125e-05
Epoch 926, Loss: 2.278912267542574, Learning Rate: 1.953125e-05
Epoch 927, Loss: 2.278912070430114, Learning Rate: 1.953125e-05
Epoch 928, Loss: 2.278911873326374, Learning Rate: 1.953125e-05
Epoch 929, Loss: 2.2789116762262935, Learning Rate: 1.953125e-05
Epoch 930, Loss: 2.2789114791183795, Learning Rate: 1.953125e-05
Epoch 931, Loss: 2.278911282013889, Learning Rate: 1.953125e-05
Epoch 932, Loss: 2.278911084915446, Learning Rate: 1.953125e-05
Epoch 933, Loss: 2.278910887804595, Learning Rate: 1.953125e-05
Epoch 934, Loss: 2.2789106907078858, Learning Rate: 1.953125e-05

```
Epoch 935, Loss: 2.278910493606091, Learning Rate: 1.953125e-05
Epoch 936, Loss: 2.278910296500766, Learning Rate: 1.953125e-05
Epoch 937, Loss: 2.2789100994021148, Learning Rate: 1.953125e-05
Epoch 938, Loss: 2.2789099022977015, Learning Rate: 1.953125e-05
Epoch 939, Loss: 2.278909705203125, Learning Rate: 1.953125e-05
Epoch 940, Loss: 2.278909508099412, Learning Rate: 1.953125e-05
Epoch 941, Loss: 2.2789093109968857, Learning Rate: 1.953125e-05
Epoch 942, Loss: 2.278909113907087, Learning Rate: 1.953125e-05
Epoch 943, Loss: 2.2789089168021985, Learning Rate: 1.953125e-05
Epoch 944, Loss: 2.2789087196998903, Learning Rate: 1.953125e-05
Epoch 945, Loss: 2.2789085226104238, Learning Rate: 1.953125e-05
Epoch 946, Loss: 2.2789083255076865, Learning Rate: 1.953125e-05
Epoch 947, Loss: 2.2789081284115786, Learning Rate: 1.953125e-05
Epoch 948, Loss: 2.278907931317571, Learning Rate: 1.953125e-05
Epoch 949, Loss: 2.278907734216526, Learning Rate: 1.953125e-05
Epoch 950, Loss: 2.278907537126975, Learning Rate: 1.953125e-05
Epoch 951, Loss: 2.278907340023074, Learning Rate: 1.953125e-05
Epoch 952, Loss: 2.27890714293435, Learning Rate: 1.953125e-05
Epoch 953, Loss: 2.2789069458396574, Learning Rate: 1.953125e-05
Epoch 954, Loss: 2.278906748737504, Learning Rate: 1.953125e-05
Epoch 955, Loss: 2.2789065516576152, Learning Rate: 1.953125e-05
Epoch 956, Loss: 2.2789063545550006, Learning Rate: 1.953125e-05
Epoch 957, Loss: 2.2789061574625067, Learning Rate: 1.953125e-05
Epoch 958, Loss: 2.2789059603750217, Learning Rate: 1.953125e-05
Epoch 959, Loss: 2.2789057632767076, Learning Rate: 1.953125e-05
Epoch 960, Loss: 2.2789055661903834, Learning Rate: 1.953125e-05
Epoch 961, Loss: 2.278905369094568, Learning Rate: 1.953125e-05
Epoch 962, Loss: 2.278905172007584, Learning Rate: 1.953125e-05
Epoch 963, Loss: 2.2789049749162706, Learning Rate: 1.953125e-05
Epoch 964, Loss: 2.2789047778203297, Learning Rate: 1.953125e-05
Epoch 965, Loss: 2.2789045807406842, Learning Rate: 1.953125e-05
Epoch 966, Loss: 2.278904383647321, Learning Rate: 1.953125e-05
Epoch 967, Loss: 2.2789041865547723, Learning Rate: 1.953125e-05
Epoch 968, Loss: 2.278903989473023, Learning Rate: 1.953125e-05
Epoch 969, Loss: 2.278903792379981, Learning Rate: 1.953125e-05
Epoch 970, Loss: 2.2789035952961263, Learning Rate: 1.953125e-05
Epoch 971, Loss: 2.278903398207386, Learning Rate: 1.953125e-05
Epoch 972, Loss: 2.2789032011177195, Learning Rate: 1.953125e-05
Epoch 973, Loss: 2.2789030040392744, Learning Rate: 1.953125e-05
Epoch 974, Loss: 2.278902806943393, Learning Rate: 1.953125e-05
Epoch 975, Loss: 2.2789026098682594, Learning Rate: 1.953125e-05
Epoch 976, Loss: 2.278902412779702, Learning Rate: 1.953125e-05
Epoch 977, Loss: 2.2789022156866894, Learning Rate: 1.953125e-05
Epoch 978, Loss: 2.2789020186198234, Learning Rate: 1.953125e-05
Epoch 979, Loss: 2.278901821523758, Learning Rate: 1.953125e-05
Epoch 980, Loss: 2.278901624441978, Learning Rate: 1.953125e-05
Epoch 981, Loss: 2.2789014273615926, Learning Rate: 1.953125e-05
Epoch 982, Loss: 2.2789012302772202, Learning Rate: 1.953125e-05
```

```
Epoch 983, Loss: 2.2789010331985224, Learning Rate: 1.953125e-05
Epoch 984, Loss: 2.2789008361119176, Learning Rate: 1.953125e-05
Epoch 985, Loss: 2.278900639031516, Learning Rate: 1.953125e-05
Epoch 986, Loss: 2.2789004419571794, Learning Rate: 1.953125e-05
Epoch 987, Loss: 2.2789002448686118, Learning Rate: 1.953125e-05
Epoch 988, Loss: 2.2789000477963253, Learning Rate: 1.953125e-05
Epoch 989, Loss: 2.278899850714163, Learning Rate: 1.953125e-05
Epoch 990, Loss: 2.278899653630048, Learning Rate: 1.953125e-05
Epoch 991, Loss: 2.2788994565602607, Learning Rate: 1.953125e-05
Epoch 992, Loss: 2.2788992594780018, Learning Rate: 1.953125e-05
Epoch 993, Loss: 2.278899062397226, Learning Rate: 1.953125e-05
Epoch 994, Loss: 2.2788988653234448, Learning Rate: 1.953125e-05
Epoch 995, Loss: 2.2788986682451684, Learning Rate: 1.953125e-05
Epoch 996, Loss: 2.2788984711697067, Learning Rate: 1.953125e-05
Epoch 997, Loss: 2.278898274090926, Learning Rate: 1.953125e-05
Epoch 998, Loss: 2.2788980770195195, Learning Rate: 1.953125e-05
Epoch 999, Loss: 2.2788978799427584, Learning Rate: 1.953125e-05
Epoch 1000, Loss: 2.278897682863559, Learning Rate: 1.953125e-05
```

[67]:

```
plt.plot(losses)
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Loss Curve')
plt.show()
```