import numpy as np

import matplotlib.pyplot as plt

from minisom import MiniSom

# Generate synthetic data for demonstration (e.g., 10x10 pixel images of characters)

# Here we create random data to simulate character images

def generate\_character\_data(num\_samples=100):

# Each character is represented as a flattened 10x10 image (100 pixels)

return np.random.rand(num\_samples, 100)

# Load dataset

data = generate\_character\_data(100) # Generate 100 random character images

# Initialize and train the Self-Organizing Map

som\_size = 10 # Size of the SOM grid (10x10)

som = MiniSom(som\_size, som\_size, input\_len=100, sigma=1.0, learning\_rate=0.5)

# Initialize weights and train

som.random\_weights\_init(data)

som.train\_random(data, num\_iteration=500)

# Visualize the results

def plot\_som(som):

plt.figure(figsize=(8, 8))

for i in range(som\_size):

for j in range(som\_size):

plt.text(i, j, f'{i},{j}', ha='center', va='center', fontsize=12,

bbox=dict(facecolor='white', alpha=0.5))

# Draw lines between neurons

if i < som\_size - 1:

plt.plot([i, i + 1], [j, j], color='gray', alpha=0.5)

if j < som\_size - 1:

plt.plot([i, i], [j, j + 1], color='gray', alpha=0.5)

plt.xlim(-0.5, som\_size - 0.5)

plt.ylim(-0.5, som\_size - 0.5)

plt.title('Self-Organizing Map')

plt.grid()

plt.show()

plot\_som(som)

# Map each sample to its closest neuron

mapped\_indices = np.array([som.winner(d) for d in data])

# Plotting the mapping of characters to SOM grid

plt.figure(figsize=(12, 8))

plt.scatter(mapped\_indices[:, 0], mapped\_indices[:, 1], alpha=0.7)

plt.title('Mapping of Characters to SOM Grid')

plt.xlabel('SOM X Index')

plt.ylabel('SOM Y Index')

plt.grid()

plt.show()

#OUTPUT



