Using my own dataset:  
Epoch [1/10000], Train Loss: 5.1221, Train Accuracy: 1.56%

Validation Accuracy: 1.97%, Validation Loss: 5.0497

Epoch [2/10000], Train Loss: 4.9778, Train Accuracy: 2.26%

Validation Accuracy: 2.49%, Validation Loss: 4.9793

Epoch [3/10000], Train Loss: 4.9159, Train Accuracy: 2.89%

Validation Accuracy: 2.23%, Validation Loss: 5.0065

Epoch [4/10000], Train Loss: 4.8896, Train Accuracy: 3.02%

Validation Accuracy: 3.04%, Validation Loss: 4.9289

Epoch [5/10000], Train Loss: 4.8786, Train Accuracy: 3.20%

Validation Accuracy: 1.74%, Validation Loss: 5.0969

Epoch [6/10000], Train Loss: 4.8718, Train Accuracy: 3.21%

Validation Accuracy: 0.98%, Validation Loss: 6.2225

Using tiny imagenet:  
Epoch [1/10], Train Loss: 5.0684, Train Accuracy: 2.37% Validation Accuracy: 4.04%, Validation Loss: 4.7328

Epoch [2/10], Train Loss: 4.5978, Train Accuracy: 5.36% Validation Accuracy: 7.26%, Validation Loss: 4.4122 Epoch

[3/10], Train Loss: 4.3582, Train Accuracy: 7.84% Validation Accuracy: 8.98%, Validation Loss: 4.2329 Epoch

[4/10], Train Loss: 4.2156, Train Accuracy: 9.55% Validation Accuracy: 10.69%, Validation Loss: 4.1189 Epoch

[5/10], Train Loss: 4.0994, Train Accuracy: 10.80% Validation Accuracy: 12.12%, Validation Loss: 4.0186 Epoch

[6/10], Train Loss: 3.9795, Train Accuracy: 12.54% Validation Accuracy: 14.45%, Validation Loss: 3.8623 Epoch

[7/10], Train Loss: 3.8619, Train Accuracy: 14.28% Validation Accuracy: 15.11%, Validation Loss: 3.7965 Epoch

[8/10], Train Loss: 3.7658, Train Accuracy: 15.61% Validation Accuracy: 16.36%, Validation Loss: 3.7135  
  
  
  
using imagenet 5, and optimizer = optim.Adam(model.parameters(), lr=0.001)  # Adam optimizer  
Epoch [1/100], Train Loss: 4.0524, Train Accuracy: 13.76%

Validation Accuracy: 18.09%, Validation Loss: 3.6925

Epoch [2/100], Train Loss: 3.2865, Train Accuracy: 25.65%

Validation Accuracy: 27.61%, Validation Loss: 3.1561

Epoch [3/100], Train Loss: 2.8863, Train Accuracy: 32.88%

Validation Accuracy: 32.76%, Validation Loss: 2.8978

Epoch [4/100], Train Loss: 2.5930, Train Accuracy: 38.60%

Validation Accuracy: 37.28%, Validation Loss: 2.6849

Epoch [5/100], Train Loss: 2.3453, Train Accuracy: 43.46%

Validation Accuracy: 39.64%, Validation Loss: 2.5772

Epoch [6/100], Train Loss: 2.1154, Train Accuracy: 48.05%

Validation Accuracy: 41.82%, Validation Loss: 2.4692

Epoch [7/100], Train Loss: 1.8927, Train Accuracy: 52.74%

Validation Accuracy: 42.10%, Validation Loss: 2.4927

Epoch [8/100], Train Loss: 1.6746, Train Accuracy: 57.32%

Validation Accuracy: 43.34%, Validation Loss: 2.5009

Epoch [9/100], Train Loss: 1.4356, Train Accuracy: 62.50%

Validation Accuracy: 43.94%, Validation Loss: 2.5334

Epoch [10/100], Train Loss: 1.1993, Train Accuracy: 67.77%

Validation Accuracy: 43.40%, Validation Loss: 2.6803  
  
overfitting

optimizer = optim.Adam(model.parameters(), lr=0.01, weight\_decay=5e-3)  # adjust weight decay for overfitting (5e-3 learn too slow)

Epoch [1/100], Train Loss: 5.1449, Train Accuracy: 1.32%

Validation Accuracy: 1.19%, Validation Loss: 5.1706

Epoch [2/100], Train Loss: 5.0720, Train Accuracy: 1.60%

Validation Accuracy: 1.55%, Validation Loss: 5.1962

Epoch [3/100], Train Loss: 5.0488, Train Accuracy: 1.72%

Validation Accuracy: 1.15%, Validation Loss: 5.1432

Epoch [4/100], Train Loss: 5.0389, Train Accuracy: 1.81%

Validation Accuracy: 1.60%, Validation Loss: 5.1591

Epoch 5: reducing learning rate of group 0 to 5.0000e-03.

Epoch [5/100], Train Loss: 5.0256, Train Accuracy: 1.99%

Validation Accuracy: 0.52%, Validation Loss: 6.2658

Epoch [6/100], Train Loss: 5.0036, Train Accuracy: 2.06%

Validation Accuracy: 1.67%, Validation Loss: 5.1069

Epoch [7/100], Train Loss: 5.0016, Train Accuracy: 2.16%

Validation Accuracy: 0.58%, Validation Loss: 5.7758

Epoch 8: reducing learning rate of group 0 to 2.5000e-03.

Epoch [8/100], Train Loss: 4.9980, Train Accuracy: 2.17%

Validation Accuracy: 1.06%, Validation Loss: 5.2270

import torch.nn as nn

import torch.optim as optim

import matplotlib.pyplot as plt  # For plotting

# Define loss function and optimizer

criterion = nn.CrossEntropyLoss()  # Cross-entropy loss for classification

optimizer = optim.Adam(model.parameters(), lr=0.01, weight\_decay=1e-4)  # adjust weight decay for overfitting (5e-3 learn too slow)

# Initialize lists to store metrics

train\_losses = []

val\_losses = []

train\_accuracies = []

val\_accuracies = []

num\_epochs = 10000  # Number of epochs

patience = 5  # Patience for early stopping

best\_val\_loss = float('inf')  # Initialize best validation loss as infinity

epochs\_without\_improvement = 0  # Counter for how many epochs passed without improvement

early\_stop = False

# Initialize the learning rate scheduler

scheduler = optim.lr\_scheduler.ReduceLROnPlateau(optimizer,

                                                 mode='min',      # Reduce when the quantity monitored has stopped decreasing

                                                 factor=0.5,     # Factor by which the learning rate will be reduced

                                                 patience=1,     # Number of epochs with no improvement after which learning rate will be reduced

                                                 verbose=True)   # Print a message when the learning rate is reduced

for epoch in range(num\_epochs):

    if early\_stop:

        print("Early stopping triggered.")

        break

    model.train()  # Set model to training mode

    running\_loss = 0.0

    correct\_train = 0

    total\_train = 0

    for images, labels in train\_loader:

        images, labels = images.to(device), labels.to(device)  # Move data to device

        # Zero the parameter gradients

        optimizer.zero\_grad()

        # Forward pass

        outputs = model(images)

        loss = criterion(outputs, labels)

        # Backward pass and optimize

        loss.backward()

        optimizer.step()

        # Update running loss

        running\_loss += loss.item()

        # Calculate training accuracy

        \_, predicted = torch.max(outputs.data, 1)

        total\_train += labels.size(0)

        correct\_train += (predicted == labels).sum().item()

    # Calculate average training loss and accuracy

    avg\_train\_loss = running\_loss / len(train\_loader)

    train\_accuracy = 100 \* correct\_train / total\_train

    train\_losses.append(avg\_train\_loss)

    train\_accuracies.append(train\_accuracy)

    # Validation loop

    model.eval()  # Set model to evaluation mode

    correct\_val = 0

    total\_val = 0

    val\_loss = 0.0

    with torch.no\_grad():  # No need to calculate gradients during validation

        for images, labels in val\_loader:

            images, labels = images.to(device), labels.to(device)

            outputs = model(images)

            loss = criterion(outputs, labels)

            val\_loss += loss.item()

            # Calculate validation accuracy

            \_, predicted = torch.max(outputs.data, 1)

            total\_val += labels.size(0)

            correct\_val += (predicted == labels).sum().item()

    # Calculate average validation loss and accuracy

    avg\_val\_loss = val\_loss / len(val\_loader)

    val\_accuracy = 100 \* correct\_val / total\_val

    val\_losses.append(avg\_val\_loss)

    val\_accuracies.append(val\_accuracy)

    # Step the scheduler

    scheduler.step(avg\_val\_loss)  # Update the learning rate based on validation loss

    # Print epoch statistics

    print(f'Epoch [{epoch+1}/{num\_epochs}], Train Loss: {avg\_train\_loss:.4f}, Train Accuracy: {train\_accuracy:.2f}%')

    print(f'Validation Accuracy: {val\_accuracy:.2f}%, Validation Loss: {avg\_val\_loss:.4f}')

    # Early stopping logic

    if avg\_val\_loss < best\_val\_loss:

        best\_val\_loss = avg\_val\_loss  # Update the best validation loss

        epochs\_without\_improvement = 0  # Reset the counter if we have improvement

        best\_model\_wts = model.state\_dict()  # Optionally save the best model

    else:

        epochs\_without\_improvement += 1  # Increment the counter

    if epochs\_without\_improvement >= patience:

        print(f'Early stopping after {epoch+1} epochs without improvement.')

        early\_stop = True

        # load the best model weights

        model.load\_state\_dict(best\_model\_wts)

print('Training complete!')

# Plotting the results

epochs = range(1, len(train\_losses) + 1)

plt.figure(figsize=(12, 5))

# Plot training & validation loss

plt.subplot(1, 2, 1)

plt.plot(epochs, train\_losses, 'bo-', label='Train Loss')

plt.plot(epochs, val\_losses, 'ro-', label='Validation Loss')

plt.title('Loss')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

# Plot training & validation accuracy

plt.subplot(1, 2, 2)

plt.plot(epochs, train\_accuracies, 'bo-', label='Train Accuracy')

plt.plot(epochs, val\_accuracies, 'ro-', label='Validation Accuracy')

plt.title('Accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy (%)')

plt.legend()

plt.show()

# Save the model with the best performance

if best\_model\_wts is not None:

    model.load\_state\_dict(best\_model\_wts)  # Ensure we load the best model weights before saving

    torch.save(model.state\_dict(), 'best\_model.pth')  # Save model to a file

    print("Best model saved as 'best\_model.pth'")

else:

    print("No improvement in validation loss, model not saved.")

Epoch [1/10000], Train Loss: 4.8647, Train Accuracy: 3.51%

Validation Accuracy: 3.79%, Validation Loss: 4.7462

Epoch [2/10000], Train Loss: 4.4787, Train Accuracy: 6.57%

Validation Accuracy: 7.00%, Validation Loss: 4.4190

Epoch [3/10000], Train Loss: 4.2598, Train Accuracy: 9.40%

Validation Accuracy: 9.22%, Validation Loss: 4.2630

Epoch [4/10000], Train Loss: 4.1085, Train Accuracy: 11.46%

Validation Accuracy: 11.02%, Validation Loss: 4.0932

Epoch [5/10000], Train Loss: 4.0058, Train Accuracy: 12.83%

Validation Accuracy: 12.37%, Validation Loss: 4.0022

Epoch [6/10000], Train Loss: 3.9256, Train Accuracy: 14.28%

Validation Accuracy: 12.92%, Validation Loss: 4.0542

Epoch [7/10000], Train Loss: 3.8472, Train Accuracy: 15.57%

Validation Accuracy: 15.26%, Validation Loss: 3.8465

Epoch [8/10000], Train Loss: 3.7897, Train Accuracy: 16.33%

Validation Accuracy: 16.68%, Validation Loss: 3.7639

Epoch [9/10000], Train Loss: 3.7513, Train Accuracy: 17.32%

Validation Accuracy: 15.59%, Validation Loss: 3.8653

Epoch [10/10000], Train Loss: 3.7222, Train Accuracy: 17.65%

Validation Accuracy: 17.17%, Validation Loss: 3.7619  
  
no overfitting at all, decrease weight decay

optimizer = optim.Adam(model.parameters(), lr=0.01, weight\_decay=1e-5)  # adjust weight decay for overfitting (5e-3 learn too slow)

Epoch [1/10000], Train Loss: 4.7635, Train Accuracy: 4.38%

Validation Accuracy: 6.71%, Validation Loss: 4.5068

Epoch [2/10000], Train Loss: 4.2935, Train Accuracy: 8.78%

Validation Accuracy: 9.51%, Validation Loss: 4.2168

Epoch [3/10000], Train Loss: 4.0519, Train Accuracy: 12.11%

Validation Accuracy: 12.19%, Validation Loss: 3.9910  
  
maybe weight decay is unnecessary  
  
  
---

Without any weight decay:  
Epoch [1/10000], Train Loss: 4.7047, Train Accuracy: 5.19%

Validation Accuracy: 8.02%, Validation Loss: 4.4061

Epoch [2/10000], Train Loss: 4.0731, Train Accuracy: 12.01%

Validation Accuracy: 13.20%, Validation Loss: 4.0001

Epoch [3/10000], Train Loss: 3.7520, Train Accuracy: 16.83%

Validation Accuracy: 17.38%, Validation Loss: 3.6903

Epoch [4/10000], Train Loss: 3.5503, Train Accuracy: 20.11%

Validation Accuracy: 19.98%, Validation Loss: 3.5930

Epoch [5/10000], Train Loss: 3.3892, Train Accuracy: 22.95%

Validation Accuracy: 23.40%, Validation Loss: 3.3838

Epoch [6/10000], Train Loss: 3.2517, Train Accuracy: 25.49%

Validation Accuracy: 25.11%, Validation Loss: 3.2776

Epoch [7/10000], Train Loss: 3.1315, Train Accuracy: 27.65%

Validation Accuracy: 28.13%, Validation Loss: 4.1708

Epoch [8/10000], Train Loss: 3.0268, Train Accuracy: 29.46%

Validation Accuracy: 28.87%, Validation Loss: 3.0979

Epoch [9/10000], Train Loss: 2.9256, Train Accuracy: 31.51%

Validation Accuracy: 30.72%, Validation Loss: 3.0117

Epoch [10/10000], Train Loss: 2.8301, Train Accuracy: 33.45%

Validation Accuracy: 31.68%, Validation Loss: 3.1471

Epoch [11/10000], Train Loss: 2.7450, Train Accuracy: 34.98%

Validation Accuracy: 33.55%, Validation Loss: 3.0078

Epoch [12/10000], Train Loss: 2.6602, Train Accuracy: 36.53%

Validation Accuracy: 33.58%, Validation Loss: 3.5296

Epoch [13/10000], Train Loss: 2.5800, Train Accuracy: 38.04%

Validation Accuracy: 34.98%, Validation Loss: 2.9170

Epoch [14/10000], Train Loss: 2.5112, Train Accuracy: 39.38%

Validation Accuracy: 35.17%, Validation Loss: 2.8180

Epoch [15/10000], Train Loss: 2.4330, Train Accuracy: 40.88%

Validation Accuracy: 36.12%, Validation Loss: 2.9577

Epoch [16/10000], Train Loss: 2.3591, Train Accuracy: 42.37%

Validation Accuracy: 36.74%, Validation Loss: 2.7487

Epoch [17/10000], Train Loss: 2.2946, Train Accuracy: 43.73%

Validation Accuracy: 36.38%, Validation Loss: 2.7970

Epoch 18: reducing learning rate of group 0 to 5.0000e-03.

Epoch [18/10000], Train Loss: 2.2232, Train Accuracy: 44.98%

Validation Accuracy: 37.57%, Validation Loss: 2.7583

Epoch [19/10000], Train Loss: 1.9472, Train Accuracy: 50.98%

Validation Accuracy: 40.51%, Validation Loss: 2.7428

Epoch [20/10000], Train Loss: 1.8457, Train Accuracy: 53.04%

Validation Accuracy: 40.20%, Validation Loss: 2.6504

Epoch [21/10000], Train Loss: 1.7734, Train Accuracy: 54.58%

Validation Accuracy: 40.16%, Validation Loss: 3.1880

Epoch 22: reducing learning rate of group 0 to 2.5000e-03.

Epoch [22/10000], Train Loss: 1.7094, Train Accuracy: 55.87%

Validation Accuracy: 39.98%, Validation Loss: 2.7090  
  
we need decay, start with 1e-4

1e-4  
Epoch [1/10000], Train Loss: 4.8820, Train Accuracy: 3.25%

Validation Accuracy: 4.44%, Validation Loss: 4.6883

Epoch [2/10000], Train Loss: 4.5399, Train Accuracy: 6.02%

Validation Accuracy: 5.45%, Validation Loss: 4.5088

Epoch [3/10000], Train Loss: 4.3455, Train Accuracy: 8.30%

Validation Accuracy: 8.46%, Validation Loss: 4.3220

Epoch [4/10000], Train Loss: 4.1711, Train Accuracy: 10.48%

Validation Accuracy: 10.83%, Validation Loss: 4.1520

Epoch [5/10000], Train Loss: 4.0343, Train Accuracy: 12.32%

Validation Accuracy: 12.51%, Validation Loss: 4.0269

Epoch [6/10000], Train Loss: 3.9479, Train Accuracy: 13.94%

Validation Accuracy: 13.77%, Validation Loss: 3.9553

Epoch [7/10000], Train Loss: 3.8680, Train Accuracy: 15.23%

Validation Accuracy: 15.84%, Validation Loss: 3.8377

Epoch [8/10000], Train Loss: 3.8089, Train Accuracy: 16.14%

Validation Accuracy: 16.17%, Validation Loss: 3.8021

Epoch [9/10000], Train Loss: 3.7627, Train Accuracy: 17.09%

Validation Accuracy: 15.76%, Validation Loss: 3.7952

Epoch [10/10000], Train Loss: 3.7349, Train Accuracy: 17.46%

Validation Accuracy: 17.09%, Validation Loss: 3.7292

Epoch [11/10000], Train Loss: 3.7027, Train Accuracy: 17.98%

Validation Accuracy: 17.20%, Validation Loss: 3.7712

Epoch [12/10000], Train Loss: 3.6821, Train Accuracy: 18.34%

Validation Accuracy: 17.97%, Validation Loss: 3.7042

Epoch [13/10000], Train Loss: 3.6656, Train Accuracy: 18.61%

Validation Accuracy: 18.18%, Validation Loss: 3.6500

Epoch [14/10000], Train Loss: 3.6495, Train Accuracy: 18.83%

Validation Accuracy: 18.56%, Validation Loss: 3.6436

Epoch [15/10000], Train Loss: 3.6314, Train Accuracy: 19.23%

Validation Accuracy: 19.07%, Validation Loss: 3.6352

Epoch [16/10000], Train Loss: 3.6112, Train Accuracy: 19.62%

Validation Accuracy: 17.90%, Validation Loss: 3.7457

Epoch 17: reducing learning rate of group 0 to 5.0000e-03.

Epoch [17/10000], Train Loss: 3.5977, Train Accuracy: 19.  
90%

Validation Accuracy: 18.62%, Validation Loss: 3.6590

---

1e-6

Epoch [1/10000], Train Loss: 4.7639, Train Accuracy: 4.47%

Validation Accuracy: 6.69%, Validation Loss: 4.4648

Epoch [2/10000], Train Loss: 4.1690, Train Accuracy: 10.95%

Validation Accuracy: 11.07%, Validation Loss: 4.1969

Epoch [3/10000], Train Loss: 3.8662, Train Accuracy: 15.25%

Validation Accuracy: 15.52%, Validation Loss: 3.8417

Epoch [4/10000], Train Loss: 3.6944, Train Accuracy: 17.90%

Validation Accuracy: 16.68%, Validation Loss: 3.7502

Epoch [5/10000], Train Loss: 3.5763, Train Accuracy: 19.89%

Validation Accuracy: 20.11%, Validation Loss: 3.5700

Epoch [6/10000], Train Loss: 3.4737, Train Accuracy: 21.53%

Validation Accuracy: 21.23%, Validation Loss: 3.5060

Epoch [7/10000], Train Loss: 3.3694, Train Accuracy: 23.20%

Validation Accuracy: 22.59%, Validation Loss: 3.4191

Epoch [8/10000], Train Loss: 3.2770, Train Accuracy: 25.12%

Validation Accuracy: 24.28%, Validation Loss: 3.3040

Epoch [9/10000], Train Loss: 3.1896, Train Accuracy: 26.60%

Validation Accuracy: 25.63%, Validation Loss: 3.2966

Epoch [10/10000], Train Loss: 3.1033, Train Accuracy: 28.24%

Validation Accuracy: 27.16%, Validation Loss: 3.1939

Epoch [11/10000], Train Loss: 3.0212, Train Accuracy: 29.59%

Validation Accuracy: 28.27%, Validation Loss: 3.1065

Epoch [12/10000], Train Loss: 2.9516, Train Accuracy: 30.96%

Validation Accuracy: 29.68%, Validation Loss: 3.0528

Epoch [13/10000], Train Loss: 2.8806, Train Accuracy: 32.35%

Validation Accuracy: 30.36%, Validation Loss: 3.0275

Epoch [14/10000], Train Loss: 2.8258, Train Accuracy: 33.17%

Validation Accuracy: 31.26%, Validation Loss: 2.9645

Epoch [15/10000], Train Loss: 2.7724, Train Accuracy: 34.29%

Validation Accuracy: 32.06%, Validation Loss: 2.9335

Epoch [16/10000], Train Loss: 2.7253, Train Accuracy: 35.22%

Validation Accuracy: 32.62%, Validation Loss: 2.9130

Epoch [17/10000], Train Loss: 2.6799, Train Accuracy: 35.95%

Validation Accuracy: 33.65%, Validation Loss: 2.8894

Epoch [18/10000], Train Loss: 2.6350, Train Accuracy: 37.00%

Validation Accuracy: 33.43%, Validation Loss: 2.8809

Epoch [19/10000], Train Loss: 2.5962, Train Accuracy: 37.79%

Validation Accuracy: 34.79%, Validation Loss: 2.8128

Epoch [20/10000], Train Loss: 2.5543, Train Accuracy: 38.70%

Validation Accuracy: 34.65%, Validation Loss: 2.8094

Epoch [21/10000], Train Loss: 2.5173, Train Accuracy: 39.29%

Validation Accuracy: 33.52%, Validation Loss: 2.9165

Epoch [22/10000], Train Loss: 2.4828, Train Accuracy: 39.81%

Validation Accuracy: 35.61%, Validation Loss: 2.7760

Epoch [23/10000], Train Loss: 2.4506, Train Accuracy: 40.58%

Validation Accuracy: 35.54%, Validation Loss: 2.7830

Epoch [24/10000], Train Loss: 2.4154, Train Accuracy: 41.10%

Validation Accuracy: 35.74%, Validation Loss: 2.7733

Epoch [25/10000], Train Loss: 2.3829, Train Accuracy: 41.93%

Validation Accuracy: 36.55%, Validation Loss: 2.7464

Epoch [26/10000], Train Loss: 2.3577, Train Accuracy: 42.29%

Validation Accuracy: 36.11%, Validation Loss: 2.7544

Epoch 27: reducing learning rate of group 0 to 5.0000e-03.

Epoch [27/10000], Train Loss: 2.3248, Train Accuracy: 42.90%

Validation Accuracy: 36.63%, Validation Loss: 2.7505

Epoch [28/10000], Train Loss: 2.1090, Train Accuracy: 47.45%

Validation Accuracy: 39.21%, Validation Loss: 2.6240

Epoch [29/10000], Train Loss: 2.0477, Train Accuracy: 48.78%

Validation Accuracy: 39.43%, Validation Loss: 2.6319

Epoch 30: reducing learning rate of group 0 to 2.5000e-03.

Epoch [30/10000], Train Loss: 2.0184, Train Accuracy: 49.49%

Validation Accuracy: 39.51%, Validation Loss: 2.6507

Epoch [31/10000], Train Loss: 1.8732, Train Accuracy: 52.58%

Validation Accuracy: 40.10%, Validation Loss: 2.6322

Epoch 32: reducing learning rate of group 0 to 1.2500e-03.

Epoch [32/10000], Train Loss: 1.8386, Train Accuracy: 53.27%

Validation Accuracy: 40.40%, Validation Loss: 2.6514

Epoch [33/10000], Train Loss: 1.7510, Train Accuracy: 55.23%

Validation Accuracy: 40.68%, Validation Loss: 2.6577

Early stopping after 33 epochs without improvement.

Early stopping triggered.

Training complete!

1e-3  
Epoch [1/10000], Train Loss: 5.3773, Train Accuracy: 1.00%

Validation Accuracy: 0.63%, Validation Loss: 5.4927

Epoch [2/10000], Train Loss: 5.3310, Train Accuracy: 0.93%

Validation Accuracy: 0.50%, Validation Loss: 5.3707

Epoch [3/10000], Train Loss: 5.3348, Train Accuracy: 0.93%

Validation Accuracy: 0.50%, Validation Loss: 5.6839

Epoch 4: reducing learning rate of group 0 to 5.0000e-02.

Epoch [4/10000], Train Loss: 5.3322, Train Accuracy: 0.94%

Validation Accuracy: 0.49%, Validation Loss: 5.3858

Epoch [5/10000], Train Loss: 5.1222, Train Accuracy: 1.44%

Validation Accuracy: 0.60%, Validation Loss: 5.3074

Epoch [6/10000], Train Loss: 5.1169, Train Accuracy: 1.46%

Validation Accuracy: 0.48%, Validation Loss: 5.4615

Epoch [7/10000], Train Loss: 5.1253, Train Accuracy: 1.44%

Validation Accuracy: 0.80%, Validation Loss: 5.2147

Epoch [8/10000], Train Loss: 5.1223, Train Accuracy: 1.43%

Validation Accuracy: 0.69%, Validation Loss: 5.3686

Epoch 9: reducing learning rate of group 0 to 2.5000e-02.

Epoch [9/10000], Train Loss: 5.1235, Train Accuracy: 1.44%

Validation Accuracy: 1.33%, Validation Loss: 5.2844

Epoch [10/10000], Train Loss: 5.0405, Train Accuracy: 1.87%

Validation Accuracy: 0.85%, Validation Loss: 5.3333

Epoch 11: reducing learning rate of group 0 to 1.2500e-02.

Epoch [11/10000], Train Loss: 5.0402, Train Accuracy: 1.80%

Validation Accuracy: 0.69%, Validation Loss: 6.6891

Epoch [12/10000], Train Loss: 4.9963, Train Accuracy: 2.09%

Validation Accuracy: 2.20%, Validation Loss: 5.0116

Epoch [13/10000], Train Loss: 4.9951, Train Accuracy: 2.17%

Validation Accuracy: 0.82%, Validation Loss: 5.9024

Epoch 14: reducing learning rate of group 0 to 6.2500e-03.

Epoch [14/10000], Train Loss: 4.9871, Train Accuracy: 2.13%

Validation Accuracy: 1.22%, Validation Loss: 5.1653

Epoch [15/10000], Train Loss: 4.9589, Train Accuracy: 2.29%

Validation Accuracy: 2.11%, Validation Loss: 4.9561

Epoch [16/10000], Train Loss: 4.9553, Train Accuracy: 2.29%

Validation Accuracy: 1.64%, Validation Loss: 5.0457  
  
learn like shit, try 1e-5