CIFAR10_CNN

October 25, 2023

1 CIFAR10_CNN

- ELEC 576 HW 2
- Robert Heeter
- 25 October 2023

1.1 Structure:

- Purpose: Implement a PyTorch image classsification CNN neural network (LeNet5) on the cifar10 dataset

1) Set PyTorch metada
    - Seed
    - TensorFlow output (logging)
    - Whether to transfer to gpu (cuda)
2) Import data
    - Download data
    - Create data loaders with batchsize, transforms, scaling
3) Define model architecture, loss, and optimizer
4) Define test and training loops
    - Train:
        - Get next batch
        - Forward pass through model-
        - Calculate loss
        - Backward pass from loss (calculates the gradient for each parameter)
        - Optimizer: performs weight updates
        - Calculate accuracy, other stats
    - Test:
        - Calculate loss, accuracy, other stats
5) Perform training over multiple epochs
    - Each epoch:
        - Call train loop
        - Call test loop

1.2 Acknowledgements:

- Worked with Arielle Sanford

[1]:
```
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torchvision import datasets, transforms
from torch.autograd import Variable
from torch.utils.data import DataLoader
import numpy as np

from torch.utils.tensorboard import SummaryWriter
from datetime import datetime
import os

import matplotlib.pyplot as plt

%load_ext tensorboard
```

```
2023-10-25 01:07:29.652648: I tensorflow/core/platform/cpu_feature_guard.cc:193]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  SSE4.1 SSE4.2 AVX AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
```

[7]:
```
# 1. Set PyTorch metadata

batch_size = 128
epochs = 10
lr = 0.0001
try_cuda = True
seed = 1000

num_classes = 10

logging_interval = 10 # how many batches to wait before logging
grayscale = True

# setting up the logging
log_dir = os.path.join(os.getcwd(),'log/CIFAR10', datetime.now().
 ↪strftime('%b%d_%H-%M-%S'))
writer = SummaryWriter(log_dir=log_dir)

# deciding whether to send to the cpu or not if available
if torch.cuda.is_available() and try_cuda:
    cuda = True
    torch.cuda.manual_seed(seed)
else:
    cuda = False
    torch.manual_seed(seed)
```

[8]:
```
# 2. Import data

transform = transforms.Compose([transforms.Grayscale(num_output_channels=1),␣
 ↪transforms.ToTensor()])

train_dataset = datasets.CIFAR10('data', train=True, transform=transform,␣
 ↪download=True)
test_dataset = datasets.CIFAR10('data', train=False, transform=transform,␣
 ↪download=True)

train_loader = torch.utils.data.DataLoader(train_dataset,␣
 ↪batch_size=batch_size, shuffle=True, num_workers=2)
test_loader = torch.utils.data.DataLoader(test_dataset, batch_size=batch_size,␣
 ↪shuffle=False, num_workers=2)

def check_data_loader_dim(loader):
    # checking the dataset
    for images, labels in loader:
        print('Image batch dimensions: ', images.shape)
        print('Image label dimensions: ', labels.shape)
        break

check_data_loader_dim(train_loader)
check_data_loader_dim(test_loader)
```

```
Files already downloaded and verified
Files already downloaded and verified
Image batch dimensions:  torch.Size([128, 1, 32, 32])
Image label dimensions:  torch.Size([128])
Image batch dimensions:  torch.Size([128, 1, 32, 32])
Image label dimensions:  torch.Size([128])
```

[18]:
```
# 3. Defining model architecture, loss, and optimizer

# define configuration
in_channels = 1 # grayscale images have one channel
cuda = False # not using GPU
verbose = True

layer_1_n_filters = 32
layer_2_n_filters = 64
fc_1_n_nodes = 1024
padding = 2
```

```
pooling_size = 2
kernel_size = 5

# calculating the side length of the final activation maps
# convolutional layer output width = [(input_width - kernel_size + 2*padding)/
 ↪stride] + 1
conv1_OW = ((layer_1_n_filters - kernel_size + 2*padding)/1) + 1

# max-pooling layer output width = [(input_width - pooling_size)/stride] + 1
maxpool1_OW = ((conv1_OW - pooling_size)/2) + 1

# convolutional layer output width = [(input_width - kernel_size + 2*padding)/
 ↪stride] + 1
conv2_OW = ((maxpool1_OW - kernel_size + 2*padding)/1) + 1

# max-pooling layer output width = [(input_width - pooling_size)/stride] + 1
maxpool2_OW = ((conv2_OW - 2)/2) + 1

final_length = int(maxpool2_OW)

if verbose:
    print(f"final_length = {final_length}")

# define architecture
class LeNet5(nn.Module):
    def __init__(self, num_classes, grayscale=False):
        super(LeNet5, self).__init__()

        self.grayscale = grayscale
        self.num_classes = num_classes

        if self.grayscale:
            in_channels = 1
        else:
            in_channels = 3

        self.features = nn.Sequential(
            nn.Conv2d(in_channels, layer_1_n_filters, kernel_size=5,␣
 ↪padding=2), # 32 filters with 5x5 kernel
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2),
            nn.Conv2d(layer_1_n_filters, layer_2_n_filters, kernel_size=5,␣
 ↪padding=2), # 64 filters with 5x5 kernel
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )
        self.classifier = nn.Sequential(
            nn.Linear(final_length*final_length*layer_2_n_filters*in_channels,␣
↪fc_1_n_nodes),
            nn.Tanh(),
            nn.Linear(fc_1_n_nodes, num_classes),
        )

    def forward(self, x):
        x = self.features(x) # send input through convolutional layers
        x = x.view(x.size(0), -1) # reshaping
        x = self.classifier(x) # send input through MLP layers

        logits = x # unnormalized
        probas = F.softmax(logits, dim=1) # normalized

        return logits, probas

model = LeNet5(num_classes=num_classes, grayscale=True)
print(model)

# optimizer = optim.Adam(model.parameters(), lr=lr)
optimizer = optim.SGD(model.parameters(), lr=lr)

print(optimizer)
```

```
final_length = 8
LeNet5(
  (features): Sequential(
    (0): Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (3): Conv2d(32, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): ReLU()
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (classifier): Sequential(
    (0): Linear(in_features=4096, out_features=1024, bias=True)
    (1): Tanh()
    (2): Linear(in_features=1024, out_features=10, bias=True)
  )
)
SGD (
Parameter Group 0
    dampening: 0
    differentiable: False
    foreach: None
    lr: 0.0001
    maximize: False
    momentum: 0
    nesterov: False
    weight_decay: 0
)
```

[10]:
```
# 4. Define test and training loops

def train(epoch):
    model.train()

    criterion = nn.CrossEntropyLoss()
    total_loss = 0.0
    correct = 0

    for batch_idx, (data, target) in enumerate(train_loader):
        if cuda:
            data, target = data.cuda(), target.cuda()
        optimizer.zero_grad()
        logits, probas = model(data) # forward pass
        loss = criterion(logits, target)
        loss.backward() # backward pass
        optimizer.step()
        total_loss += loss.item()

        # calculate training accuracy
        _, predicted = probas.max(1)
        correct += predicted.eq(target).sum().item()
        if batch_idx % logging_interval == 0:
            print(f'Train Epoch: {epoch} [{batch_idx * len(data)}/
 ↪{len(train_loader.dataset)} ({100. * batch_idx / len(train_loader):.
 ↪0f}%)]\tLoss: {loss.item()}')

    # calculate and log training metrics
    train_loss = total_loss / len(train_loader.dataset)
    train_accuracy = 100. * correct / len(train_loader.dataset)
    print(f"Epoch {epoch}: Training Loss: {train_loss:.4f}, Accuracy:␣
 ↪{train_accuracy:.2f}%")

    # add to TensorBoard
    writer.add_scalar('Loss/Train', train_loss, epoch)
    writer.add_scalar('Accuracy/Train', train_accuracy, epoch)

def test(epoch):
    model.eval()
    criterion = nn.CrossEntropyLoss()
    test_loss = 0
    correct = 0

    with torch.no_grad():
        for data, target in test_loader:
            if cuda:
                data, target = data.cuda(), target.cuda()
            logits, probas = model(data)
            test_loss += criterion(logits, target).item()
            _, predicted = probas.max(1)
            correct += predicted.eq(target).sum().item()

    # calculate and log testing metrics
    test_loss /= len(test_loader.dataset)
    test_accuracy = 100. * correct / len(test_loader.dataset)
    print(f"Test Loss: {test_loss:.4f}, Accuracy: {test_accuracy:.2f}%")

    # add to TensorBoard
    writer.add_scalar('Accuracy/Test', test_accuracy, epoch)
```

[19]:
```
# 5. Perform training over multiple epochs

# start training
for epoch in range(1, epochs + 1):
    train(epoch)
    test(epoch)

writer.close()
```

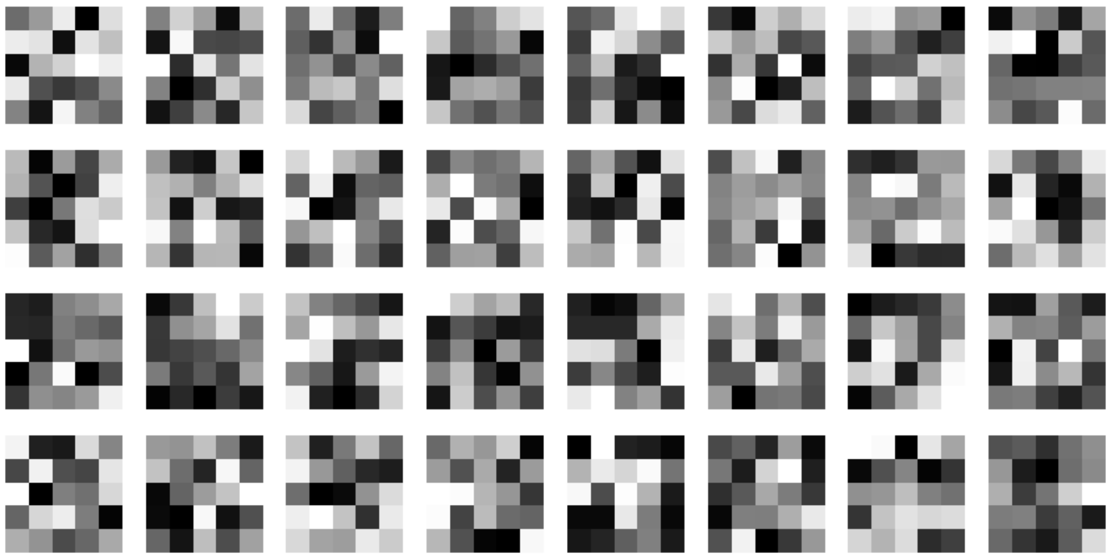
```
Train Epoch: 1 [0/50000 (0%)]   Loss: 2.308523654937744
Train Epoch: 1 [1280/50000 (3%)]        Loss: 2.303161859512329
Train Epoch: 1 [2560/50000 (5%)]        Loss: 2.3010594844818115
Train Epoch: 1 [3840/50000 (8%)]        Loss: 2.307061195373535
Train Epoch: 1 [5120/50000 (10%)]       Loss: 2.2996342182159424
Train Epoch: 1 [6400/50000 (13%)]       Loss: 2.301795244216919
Train Epoch: 1 [7680/50000 (15%)]       Loss: 2.304896116256714
Train Epoch: 1 [8960/50000 (18%)]       Loss: 2.302110433578491
Train Epoch: 1 [10240/50000 (20%)]      Loss: 2.297053098678589
Train Epoch: 1 [11520/50000 (23%)]      Loss: 2.2965855598449707
Train Epoch: 1 [12800/50000 (26%)]      Loss: 2.302279233932495
Train Epoch: 1 [14080/50000 (28%)]      Loss: 2.305506944656372
Train Epoch: 1 [15360/50000 (31%)]      Loss: 2.3001298904418945
Train Epoch: 1 [16640/50000 (33%)]      Loss: 2.3006937503814697
Train Epoch: 1 [17920/50000 (36%)]      Loss: 2.2962822914123535
Train Epoch: 1 [19200/50000 (38%)]      Loss: 2.2976248264312744
Train Epoch: 1 [20480/50000 (41%)]      Loss: 2.3011724948883057
Train Epoch: 1 [21760/50000 (43%)]      Loss: 2.3081719875335693
Train Epoch: 1 [23040/50000 (46%)]      Loss: 2.304274797439575
Train Epoch: 1 [24320/50000 (49%)]      Loss: 2.3074381351470947
Train Epoch: 1 [25600/50000 (51%)]      Loss: 2.305353879928589
Train Epoch: 1 [26880/50000 (54%)]      Loss: 2.308314561843872
Train Epoch: 1 [28160/50000 (56%)]      Loss: 2.305246353149414
Train Epoch: 1 [29440/50000 (59%)]      Loss: 2.3011577129364014
Train Epoch: 1 [30720/50000 (61%)]      Loss: 2.308297872543335
Train Epoch: 1 [32000/50000 (64%)]      Loss: 2.3107194900512695
Train Epoch: 1 [33280/50000 (66%)]      Loss: 2.301215410232544
Train Epoch: 1 [34560/50000 (69%)]      Loss: 2.3063387870788574
Train Epoch: 1 [35840/50000 (72%)]      Loss: 2.2952561378479004
Train Epoch: 1 [37120/50000 (74%)]      Loss: 2.3007757663726807
Train Epoch: 1 [38400/50000 (77%)]      Loss: 2.3017327785491943
Train Epoch: 1 [39680/50000 (79%)]      Loss: 2.3047306537628174
Train Epoch: 1 [40960/50000 (82%)]      Loss: 2.303074598312378
Train Epoch: 1 [42240/50000 (84%)]      Loss: 2.296360731124878
Train Epoch: 1 [43520/50000 (87%)]      Loss: 2.2991132736206055
Train Epoch: 1 [44800/50000 (90%)]      Loss: 2.307877779006958
Train Epoch: 1 [46080/50000 (92%)]      Loss: 2.3048758506774902
Train Epoch: 1 [47360/50000 (95%)]      Loss: 2.301509141921997
Train Epoch: 1 [48640/50000 (97%)]      Loss: 2.3040542602539062
Train Epoch: 1 [31200/50000 (100%)]     Loss: 2.302400827407837
Epoch 1: Training Loss: 0.0180, Accuracy: 9.66%
Test Loss: 0.0182, Accuracy: 9.50%
Train Epoch: 2 [0/50000 (0%)]   Loss: 2.3067781925201416
Train Epoch: 2 [1280/50000 (3%)]        Loss: 2.304879665374756
Train Epoch: 2 [2560/50000 (5%)]        Loss: 2.3060505390167236
Train Epoch: 2 [3840/50000 (8%)]        Loss: 2.3062267303466797
Train Epoch: 2 [5120/50000 (10%)]       Loss: 2.3040382862091064
Train Epoch: 2 [6400/50000 (13%)]       Loss: 2.3052570819854736
Train Epoch: 2 [7680/50000 (15%)]       Loss: 2.303212881088257
Train Epoch: 2 [8960/50000 (18%)]       Loss: 2.300882577896118
Train Epoch: 2 [10240/50000 (20%)]      Loss: 2.2979536056518555
Train Epoch: 2 [11520/50000 (23%)]      Loss: 2.2992618083953857
Train Epoch: 2 [12800/50000 (26%)]      Loss: 2.302790641784668
Train Epoch: 2 [14080/50000 (28%)]      Loss: 2.3039608001708984
Train Epoch: 2 [15360/50000 (31%)]      Loss: 2.306405782699585
Train Epoch: 2 [16640/50000 (33%)]      Loss: 2.303574562072754
Train Epoch: 2 [17920/50000 (36%)]      Loss: 2.30540132522583
Train Epoch: 2 [19200/50000 (38%)]      Loss: 2.308239698410034
Train Epoch: 2 [20480/50000 (41%)]      Loss: 2.3014488220214844
Train Epoch: 2 [21760/50000 (43%)]      Loss: 2.307694673538208
Train Epoch: 2 [23040/50000 (46%)]      Loss: 2.3042798042297363
Train Epoch: 2 [24320/50000 (49%)]      Loss: 2.3028931617736816
Train Epoch: 2 [25600/50000 (51%)]      Loss: 2.299886703491211
Train Epoch: 2 [26880/50000 (54%)]      Loss: 2.300490379333496
Train Epoch: 2 [28160/50000 (56%)]      Loss: 2.3001506328582764
Train Epoch: 2 [29440/50000 (59%)]      Loss: 2.299396276473999
Train Epoch: 2 [30720/50000 (61%)]      Loss: 2.303147554397583
Train Epoch: 2 [32000/50000 (64%)]      Loss: 2.305342674255371
Train Epoch: 2 [33280/50000 (66%)]      Loss: 2.304701089859009
Train Epoch: 2 [34560/50000 (69%)]      Loss: 2.300305128097534
Train Epoch: 2 [35840/50000 (72%)]      Loss: 2.304781913757324
Train Epoch: 2 [37120/50000 (74%)]      Loss: 2.302128791809082
Train Epoch: 2 [38400/50000 (77%)]      Loss: 2.2958977222442627
Train Epoch: 2 [39680/50000 (79%)]      Loss: 2.2982563972473145
Train Epoch: 2 [40960/50000 (82%)]      Loss: 2.302349805831909
Train Epoch: 2 [42240/50000 (84%)]      Loss: 2.3016774654388428
Train Epoch: 2 [43520/50000 (87%)]      Loss: 2.3072586059570312
Train Epoch: 2 [44800/50000 (90%)]      Loss: 2.3059604167938232
Train Epoch: 2 [46080/50000 (92%)]      Loss: 2.307875156402588
Train Epoch: 2 [47360/50000 (95%)]      Loss: 2.300769805908203
Train Epoch: 2 [48640/50000 (97%)]      Loss: 2.2975823879241943
Train Epoch: 2 [31200/50000 (100%)]     Loss: 2.2967493534088135
Epoch 2: Training Loss: 0.0180, Accuracy: 9.69%
Test Loss: 0.0182, Accuracy: 9.25%
Train Epoch: 3 [0/50000 (0%)]   Loss: 2.2987751960754395
Train Epoch: 3 [1280/50000 (3%)]        Loss: 2.306701421737671
Train Epoch: 3 [2560/50000 (5%)]        Loss: 2.306191921234131
Train Epoch: 3 [3840/50000 (8%)]        Loss: 2.3003571033477783
Train Epoch: 3 [5120/50000 (10%)]       Loss: 2.3016116619110107
Train Epoch: 3 [6400/50000 (13%)]       Loss: 2.3002681732177734
Train Epoch: 3 [7680/50000 (15%)]       Loss: 2.306448459625244
Train Epoch: 3 [8960/50000 (18%)]       Loss: 2.3050546646118164
Train Epoch: 3 [10240/50000 (20%)]      Loss: 2.300477981567383
Train Epoch: 3 [11520/50000 (23%)]      Loss: 2.307866334915161
Train Epoch: 3 [12800/50000 (26%)]      Loss: 2.2998809814453125
Train Epoch: 3 [14080/50000 (28%)]      Loss: 2.305934429168701
Train Epoch: 3 [15360/50000 (31%)]      Loss: 2.306894540786743
Train Epoch: 3 [16640/50000 (33%)]      Loss: 2.2994675636291504
Train Epoch: 3 [17920/50000 (36%)]      Loss: 2.304676055908203
Train Epoch: 3 [19200/50000 (38%)]      Loss: 2.2964870929718018
Train Epoch: 3 [20480/50000 (41%)]      Loss: 2.3003692626953125
Train Epoch: 3 [21760/50000 (43%)]      Loss: 2.296705961227417
Train Epoch: 3 [23040/50000 (46%)]      Loss: 2.3028602600097656
Train Epoch: 3 [24320/50000 (49%)]      Loss: 2.2971761226654053
Train Epoch: 3 [25600/50000 (51%)]      Loss: 2.3041305541992188
Train Epoch: 3 [26880/50000 (54%)]      Loss: 2.3029751777648926
Train Epoch: 3 [28160/50000 (56%)]      Loss: 2.2988953590393066
Train Epoch: 3 [29440/50000 (59%)]      Loss: 2.3012850284576416
Train Epoch: 3 [30720/50000 (61%)]      Loss: 2.302473545074463
Train Epoch: 3 [32000/50000 (64%)]      Loss: 2.2980525493621826
Train Epoch: 3 [33280/50000 (66%)]      Loss: 2.304413080215454
Train Epoch: 3 [34560/50000 (69%)]      Loss: 2.303335666656494
Train Epoch: 3 [35840/50000 (72%)]      Loss: 2.297560691833496
Train Epoch: 3 [37120/50000 (74%)]      Loss: 2.300300121307373
Train Epoch: 3 [38400/50000 (77%)]      Loss: 2.303713083267212
Train Epoch: 3 [39680/50000 (79%)]      Loss: 2.303368091583252
Train Epoch: 3 [40960/50000 (82%)]      Loss: 2.301318645477295
Train Epoch: 3 [42240/50000 (84%)]      Loss: 2.298973560333252
Train Epoch: 3 [43520/50000 (87%)]      Loss: 2.3018410205841064
Train Epoch: 3 [44800/50000 (90%)]      Loss: 2.2998645305633545
Train Epoch: 3 [46080/50000 (92%)]      Loss: 2.296520471572876
Train Epoch: 3 [47360/50000 (95%)]      Loss: 2.2980289459228516
Train Epoch: 3 [48640/50000 (97%)]      Loss: 2.298635482788086
Train Epoch: 3 [31200/50000 (100%)]     Loss: 2.3001456260681152
Epoch 3: Training Loss: 0.0180, Accuracy: 9.49%
Test Loss: 0.0182, Accuracy: 9.46%
Train Epoch: 4 [0/50000 (0%)]   Loss: 2.2986557483673096
Train Epoch: 4 [1280/50000 (3%)]        Loss: 2.2998852729797363
Train Epoch: 4 [2560/50000 (5%)]        Loss: 2.299736499786377
Train Epoch: 4 [3840/50000 (8%)]        Loss: 2.3046019077301025
Train Epoch: 4 [5120/50000 (10%)]       Loss: 2.2967638969421387
Train Epoch: 4 [6400/50000 (13%)]       Loss: 2.300973892211914
Train Epoch: 4 [7680/50000 (15%)]       Loss: 2.29852032661438
Train Epoch: 4 [8960/50000 (18%)]       Loss: 2.2990121841430664
Train Epoch: 4 [10240/50000 (20%)]      Loss: 2.298064947128296
Train Epoch: 4 [11520/50000 (23%)]      Loss: 2.301069736480713
Train Epoch: 4 [12800/50000 (26%)]      Loss: 2.300475835800171
Train Epoch: 4 [14080/50000 (28%)]      Loss: 2.3081986904144287
Train Epoch: 4 [15360/50000 (31%)]      Loss: 2.297729730606079
Train Epoch: 4 [16640/50000 (33%)]      Loss: 2.30611515045166
Train Epoch: 4 [17920/50000 (36%)]      Loss: 2.2973482608795166
Train Epoch: 4 [19200/50000 (38%)]      Loss: 2.300758123397827
Train Epoch: 4 [20480/50000 (41%)]      Loss: 2.3062961101531982
Train Epoch: 4 [21760/50000 (43%)]      Loss: 2.3069911003112793
Train Epoch: 4 [23040/50000 (46%)]      Loss: 2.297006845474243
Train Epoch: 4 [24320/50000 (49%)]      Loss: 2.299837589263916
Train Epoch: 4 [25600/50000 (51%)]      Loss: 2.3066041469573975
Train Epoch: 4 [26880/50000 (54%)]      Loss: 2.3049299716949463
Train Epoch: 4 [28160/50000 (56%)]      Loss: 2.3042449951171875
Train Epoch: 4 [29440/50000 (59%)]      Loss: 2.298288345336914
Train Epoch: 4 [30720/50000 (61%)]      Loss: 2.299752950668335
Train Epoch: 4 [32000/50000 (64%)]      Loss: 2.307558059692383
Train Epoch: 4 [33280/50000 (66%)]      Loss: 2.299715995788574
Train Epoch: 4 [34560/50000 (69%)]      Loss: 2.300227165222168
Train Epoch: 4 [35840/50000 (72%)]      Loss: 2.301628351211548
Train Epoch: 4 [37120/50000 (74%)]      Loss: 2.299304723739624
Train Epoch: 4 [38400/50000 (77%)]      Loss: 2.302314519882202
Train Epoch: 4 [39680/50000 (79%)]      Loss: 2.302924394607544
Train Epoch: 4 [40960/50000 (82%)]      Loss: 2.2973570823669434
Train Epoch: 4 [42240/50000 (84%)]      Loss: 2.2996137142181396
Train Epoch: 4 [43520/50000 (87%)]      Loss: 2.3024532794952393
Train Epoch: 4 [44800/50000 (90%)]      Loss: 2.3011159896850586
Train Epoch: 4 [46080/50000 (92%)]      Loss: 2.2988016605377197
Train Epoch: 4 [47360/50000 (95%)]      Loss: 2.311255931854248
Train Epoch: 4 [48640/50000 (97%)]      Loss: 2.2992069721221924
Train Epoch: 4 [31200/50000 (100%)]     Loss: 2.304290771484375
Epoch 4: Training Loss: 0.0180, Accuracy: 9.45%
Test Loss: 0.0182, Accuracy: 9.26%
Train Epoch: 5 [0/50000 (0%)]   Loss: 2.300523042678833
Train Epoch: 5 [1280/50000 (3%)]        Loss: 2.3028647899627686
Train Epoch: 5 [2560/50000 (5%)]        Loss: 2.303110361099243
Train Epoch: 5 [3840/50000 (8%)]        Loss: 2.3035295009613037
Train Epoch: 5 [5120/50000 (10%)]       Loss: 2.302757978439331
Train Epoch: 5 [6400/50000 (13%)]       Loss: 2.3040637969970703
Train Epoch: 5 [7680/50000 (15%)]       Loss: 2.2961785793304443
Train Epoch: 5 [8960/50000 (18%)]       Loss: 2.306713819503784
Train Epoch: 5 [10240/50000 (20%)]      Loss: 2.297666549682617
Train Epoch: 5 [11520/50000 (23%)]      Loss: 2.2998769283294678
Train Epoch: 5 [12800/50000 (26%)]      Loss: 2.3011186122894287
Train Epoch: 5 [14080/50000 (28%)]      Loss: 2.3005940914154053
Train Epoch: 5 [15360/50000 (31%)]      Loss: 2.3017895221710205
Train Epoch: 5 [16640/50000 (33%)]      Loss: 2.3085992336273193
Train Epoch: 5 [17920/50000 (36%)]      Loss: 2.30002498626709
Train Epoch: 5 [19200/50000 (38%)]      Loss: 2.3030896186828613
Train Epoch: 5 [20480/50000 (41%)]      Loss: 2.306706428527832
Train Epoch: 5 [21760/50000 (43%)]      Loss: 2.298903703689575
Train Epoch: 5 [23040/50000 (46%)]      Loss: 2.2993624210357666
Train Epoch: 5 [24320/50000 (49%)]      Loss: 2.3079028129577637
Train Epoch: 5 [25600/50000 (51%)]      Loss: 2.3028388023376465
Train Epoch: 5 [26880/50000 (54%)]      Loss: 2.29959774017334
Train Epoch: 5 [28160/50000 (56%)]      Loss: 2.299036741256714
Train Epoch: 5 [29440/50000 (59%)]      Loss: 2.3038330078125
Train Epoch: 5 [30720/50000 (61%)]      Loss: 2.302229642868042
Train Epoch: 5 [32000/50000 (64%)]      Loss: 2.3004343509674072
Train Epoch: 5 [33280/50000 (66%)]      Loss: 2.2991857528686523
Train Epoch: 5 [34560/50000 (69%)]      Loss: 2.3003408908843994
Train Epoch: 5 [35840/50000 (72%)]      Loss: 2.2969472408294678
Train Epoch: 5 [37120/50000 (74%)]      Loss: 2.299152374267578
Train Epoch: 5 [38400/50000 (77%)]      Loss: 2.301103115081787
Train Epoch: 5 [39680/50000 (79%)]      Loss: 2.300649404525757
Train Epoch: 5 [40960/50000 (82%)]      Loss: 2.2987170219421387
Train Epoch: 5 [42240/50000 (84%)]      Loss: 2.302837371826172
Train Epoch: 5 [43520/50000 (87%)]      Loss: 2.29823899269104
Train Epoch: 5 [44800/50000 (90%)]      Loss: 2.2989206314086914
Train Epoch: 5 [46080/50000 (92%)]      Loss: 2.300048351287842
Train Epoch: 5 [47360/50000 (95%)]      Loss: 2.308366537094116
Train Epoch: 5 [48640/50000 (97%)]      Loss: 2.2964916229248047
Train Epoch: 5 [31200/50000 (100%)]     Loss: 2.301330089569092
```

```
Epoch 5: Training Loss: 0.0180, Accuracy: 9.52%
Test Loss: 0.0182, Accuracy: 9.39%
Train Epoch: 6 [0/50000 (0%)]   Loss: 2.2989089488983154
Train Epoch: 6 [1280/50000 (3%)]        Loss: 2.302013635635376
Train Epoch: 6 [2560/50000 (5%)]        Loss: 2.3019590377807617
Train Epoch: 6 [3840/50000 (8%)]        Loss: 2.295210361480713
Train Epoch: 6 [5120/50000 (10%)]       Loss: 2.2987165451049805
Train Epoch: 6 [6400/50000 (13%)]       Loss: 2.2995078563690186
Train Epoch: 6 [7680/50000 (15%)]       Loss: 2.298574209213257
Train Epoch: 6 [8960/50000 (18%)]       Loss: 2.3053393363952637
Train Epoch: 6 [10240/50000 (20%)]      Loss: 2.3026065826416016
Train Epoch: 6 [11520/50000 (23%)]      Loss: 2.2969841957092285
Train Epoch: 6 [12800/50000 (26%)]      Loss: 2.298499584197998
Train Epoch: 6 [14080/50000 (28%)]      Loss: 2.3055715560913086
Train Epoch: 6 [15360/50000 (31%)]      Loss: 2.3042705059051514
Train Epoch: 6 [16640/50000 (33%)]      Loss: 2.301650047302246
Train Epoch: 6 [17920/50000 (36%)]      Loss: 2.301522970199585
Train Epoch: 6 [19200/50000 (38%)]      Loss: 2.3003783226013184
Train Epoch: 6 [20480/50000 (41%)]      Loss: 2.3008129596710205
Train Epoch: 6 [21760/50000 (43%)]      Loss: 2.298896551132202
Train Epoch: 6 [23040/50000 (46%)]      Loss: 2.299184560775757
Train Epoch: 6 [24320/50000 (49%)]      Loss: 2.302034854888916
Train Epoch: 6 [25600/50000 (51%)]      Loss: 2.300701379776001
Train Epoch: 6 [26880/50000 (54%)]      Loss: 2.3025059700012207
Train Epoch: 6 [28160/50000 (56%)]      Loss: 2.306375026702881
Train Epoch: 6 [29440/50000 (59%)]      Loss: 2.301527261734009
Train Epoch: 6 [30720/50000 (61%)]      Loss: 2.30118989944458
Train Epoch: 6 [32000/50000 (64%)]      Loss: 2.296562671661377
Train Epoch: 6 [33280/50000 (66%)]      Loss: 2.3015859127044678
Train Epoch: 6 [34560/50000 (69%)]      Loss: 2.30181622505188
Train Epoch: 6 [35840/50000 (72%)]      Loss: 2.3041727542877197
Train Epoch: 6 [37120/50000 (74%)]      Loss: 2.300201177597046
Train Epoch: 6 [38400/50000 (77%)]      Loss: 2.3031485080718994
Train Epoch: 6 [39680/50000 (79%)]      Loss: 2.299501657485962
Train Epoch: 6 [40960/50000 (82%)]      Loss: 2.3045291900634766
Train Epoch: 6 [42240/50000 (84%)]      Loss: 2.30053448677063
Train Epoch: 6 [43520/50000 (87%)]      Loss: 2.30022931098938
Train Epoch: 6 [44800/50000 (90%)]      Loss: 2.293468475341797
Train Epoch: 6 [46080/50000 (92%)]      Loss: 2.302288293838501
Train Epoch: 6 [47360/50000 (95%)]      Loss: 2.304013967514038
Train Epoch: 6 [48640/50000 (97%)]      Loss: 2.304385185241699
Train Epoch: 6 [31200/50000 (100%)]     Loss: 2.29689884185791
Epoch 6: Training Loss: 0.0180, Accuracy: 10.01%
Test Loss: 0.0182, Accuracy: 10.22%
Train Epoch: 7 [0/50000 (0%)]   Loss: 2.2966949939727783
Train Epoch: 7 [1280/50000 (3%)]        Loss: 2.302372694015503
Train Epoch: 7 [2560/50000 (5%)]        Loss: 2.3012821674346924
Train Epoch: 7 [3840/50000 (8%)]        Loss: 2.299772262573242
Train Epoch: 7 [5120/50000 (10%)]       Loss: 2.2972543239593506
Train Epoch: 7 [6400/50000 (13%)]       Loss: 2.2970597743988037
Train Epoch: 7 [7680/50000 (15%)]       Loss: 2.3036763668060303
Train Epoch: 7 [8960/50000 (18%)]       Loss: 2.298424482345581
Train Epoch: 7 [10240/50000 (20%)]      Loss: 2.3031349182128906
Train Epoch: 7 [11520/50000 (23%)]      Loss: 2.296616792678833
Train Epoch: 7 [12800/50000 (26%)]      Loss: 2.2999942302703857
Train Epoch: 7 [14080/50000 (28%)]      Loss: 2.299921751022339
Train Epoch: 7 [15360/50000 (31%)]      Loss: 2.2966887950897217
Train Epoch: 7 [16640/50000 (33%)]      Loss: 2.30391526222229
Train Epoch: 7 [17920/50000 (36%)]      Loss: 2.299422025680542
Train Epoch: 7 [19200/50000 (38%)]      Loss: 2.3056044578552246
Train Epoch: 7 [20480/50000 (41%)]      Loss: 2.3029630184173584
Train Epoch: 7 [21760/50000 (43%)]      Loss: 2.303532838821411
Train Epoch: 7 [23040/50000 (46%)]      Loss: 2.3022372722625732
Train Epoch: 7 [24320/50000 (49%)]      Loss: 2.30549955368042
Train Epoch: 7 [25600/50000 (51%)]      Loss: 2.2978787422180176
Train Epoch: 7 [26880/50000 (54%)]      Loss: 2.300690174102783
Train Epoch: 7 [28160/50000 (56%)]      Loss: 2.3035240173339844
Train Epoch: 7 [29440/50000 (59%)]      Loss: 2.307485342025757
Train Epoch: 7 [30720/50000 (61%)]      Loss: 2.2931506633758545
Train Epoch: 7 [32000/50000 (64%)]      Loss: 2.3001322746276855
Train Epoch: 7 [33280/50000 (66%)]      Loss: 2.302443265914917
Train Epoch: 7 [34560/50000 (69%)]      Loss: 2.297062635421753
Train Epoch: 7 [35840/50000 (72%)]      Loss: 2.3019027709960938
Train Epoch: 7 [37120/50000 (74%)]      Loss: 2.296800136566162
Train Epoch: 7 [38400/50000 (77%)]      Loss: 2.3062636852264404
Train Epoch: 7 [39680/50000 (79%)]      Loss: 2.3081493377685547
Train Epoch: 7 [40960/50000 (82%)]      Loss: 2.297466516494751
Train Epoch: 7 [42240/50000 (84%)]      Loss: 2.2969369888305664
Train Epoch: 7 [43520/50000 (87%)]      Loss: 2.296064615249634
Train Epoch: 7 [44800/50000 (90%)]      Loss: 2.303208589553833
Train Epoch: 7 [46080/50000 (92%)]      Loss: 2.3005154132843018
Train Epoch: 7 [47360/50000 (95%)]      Loss: 2.300978899002075
Train Epoch: 7 [48640/50000 (97%)]      Loss: 2.3004002571105957
Train Epoch: 7 [31200/50000 (100%)]     Loss: 2.3014910221099854
Epoch 7: Training Loss: 0.0180, Accuracy: 10.70%
Test Loss: 0.0182, Accuracy: 11.03%
Train Epoch: 8 [0/50000 (0%)]   Loss: 2.302677869796753
Train Epoch: 8 [1280/50000 (3%)]        Loss: 2.29691219329834
Train Epoch: 8 [2560/50000 (5%)]        Loss: 2.302215099334717
Train Epoch: 8 [3840/50000 (8%)]        Loss: 2.298563241958618
Train Epoch: 8 [5120/50000 (10%)]       Loss: 2.3016297817230225
Train Epoch: 8 [6400/50000 (13%)]       Loss: 2.298325538635254
Train Epoch: 8 [7680/50000 (15%)]       Loss: 2.298661947250366
Train Epoch: 8 [8960/50000 (18%)]       Loss: 2.3012309074401855
Train Epoch: 8 [10240/50000 (20%)]      Loss: 2.3022384643554688
Train Epoch: 8 [11520/50000 (23%)]      Loss: 2.3026490211486816
Train Epoch: 8 [12800/50000 (26%)]      Loss: 2.2999584674835205
Train Epoch: 8 [14080/50000 (28%)]      Loss: 2.2998874187469482
Train Epoch: 8 [15360/50000 (31%)]      Loss: 2.29787015914917
Train Epoch: 8 [16640/50000 (33%)]      Loss: 2.3048577308654785
Train Epoch: 8 [17920/50000 (36%)]      Loss: 2.3000905513763428
Train Epoch: 8 [19200/50000 (38%)]      Loss: 2.2995498180389404
Train Epoch: 8 [20480/50000 (41%)]      Loss: 2.3043642044067383
Train Epoch: 8 [21760/50000 (43%)]      Loss: 2.2982325553894043
Train Epoch: 8 [23040/50000 (46%)]      Loss: 2.3010053634643555
Train Epoch: 8 [24320/50000 (49%)]      Loss: 2.3037428855895996
Train Epoch: 8 [25600/50000 (51%)]      Loss: 2.300774097442627
Train Epoch: 8 [26880/50000 (54%)]      Loss: 2.299375534057617
Train Epoch: 8 [28160/50000 (56%)]      Loss: 2.295464038848877
Train Epoch: 8 [29440/50000 (59%)]      Loss: 2.2994182109832764
Train Epoch: 8 [30720/50000 (61%)]      Loss: 2.301452159881592
Train Epoch: 8 [32000/50000 (64%)]      Loss: 2.2940337657928467
Train Epoch: 8 [33280/50000 (66%)]      Loss: 2.2968266010284424
Train Epoch: 8 [34560/50000 (69%)]      Loss: 2.3000082969665527
Train Epoch: 8 [35840/50000 (72%)]      Loss: 2.3014113903045654
Train Epoch: 8 [37120/50000 (74%)]      Loss: 2.2989633083343506
Train Epoch: 8 [38400/50000 (77%)]      Loss: 2.2985503673553467
Train Epoch: 8 [39680/50000 (79%)]      Loss: 2.3011438846588135
Train Epoch: 8 [40960/50000 (82%)]      Loss: 2.299135684967041
Train Epoch: 8 [42240/50000 (84%)]      Loss: 2.2983031272888184
Train Epoch: 8 [43520/50000 (87%)]      Loss: 2.3015666007995605
Train Epoch: 8 [44800/50000 (90%)]      Loss: 2.296574354171753
Train Epoch: 8 [46080/50000 (92%)]      Loss: 2.3006181716918945
Train Epoch: 8 [47360/50000 (95%)]      Loss: 2.3004393577575684
Train Epoch: 8 [48640/50000 (97%)]      Loss: 2.303985118865967
Train Epoch: 8 [31200/50000 (100%)]     Loss: 2.2951433658599854
Epoch 8: Training Loss: 0.0180, Accuracy: 11.21%
Test Loss: 0.0182, Accuracy: 11.46%
Train Epoch: 9 [0/50000 (0%)]   Loss: 2.303758382797241
Train Epoch: 9 [1280/50000 (3%)]        Loss: 2.298842668533325
Train Epoch: 9 [2560/50000 (5%)]        Loss: 2.2975828647613525
Train Epoch: 9 [3840/50000 (8%)]        Loss: 2.30006742477417
Train Epoch: 9 [5120/50000 (10%)]       Loss: 2.3036093711853027
Train Epoch: 9 [6400/50000 (13%)]       Loss: 2.30169939994812
Train Epoch: 9 [7680/50000 (15%)]       Loss: 2.302971839904785
Train Epoch: 9 [8960/50000 (18%)]       Loss: 2.2967095375061035
Train Epoch: 9 [10240/50000 (20%)]      Loss: 2.2998929023742676
Train Epoch: 9 [11520/50000 (23%)]      Loss: 2.3013646602630615
Train Epoch: 9 [12800/50000 (26%)]      Loss: 2.302884817123413
Train Epoch: 9 [14080/50000 (28%)]      Loss: 2.29821515083313
Train Epoch: 9 [15360/50000 (31%)]      Loss: 2.3022119998931885
Train Epoch: 9 [16640/50000 (33%)]      Loss: 2.301847457885742
Train Epoch: 9 [17920/50000 (36%)]      Loss: 2.2986912727355957
Train Epoch: 9 [19200/50000 (38%)]      Loss: 2.3014976978302
Train Epoch: 9 [20480/50000 (41%)]      Loss: 2.3006479740142822
Train Epoch: 9 [21760/50000 (43%)]      Loss: 2.302426815032959
Train Epoch: 9 [23040/50000 (46%)]      Loss: 2.299532890319824
Train Epoch: 9 [24320/50000 (49%)]      Loss: 2.2977354526519775
Train Epoch: 9 [25600/50000 (51%)]      Loss: 2.303382158279419
Train Epoch: 9 [26880/50000 (54%)]      Loss: 2.2987170219421387
Train Epoch: 9 [28160/50000 (56%)]      Loss: 2.302346706390381
Train Epoch: 9 [29440/50000 (59%)]      Loss: 2.298816680908203
Train Epoch: 9 [30720/50000 (61%)]      Loss: 2.2979795932769775
Train Epoch: 9 [32000/50000 (64%)]      Loss: 2.2993459701538086
Train Epoch: 9 [33280/50000 (66%)]      Loss: 2.2989344596862793
Train Epoch: 9 [34560/50000 (69%)]      Loss: 2.300079345703125
Train Epoch: 9 [35840/50000 (72%)]      Loss: 2.3003363609313965
Train Epoch: 9 [37120/50000 (74%)]      Loss: 2.3096296787261963
Train Epoch: 9 [38400/50000 (77%)]      Loss: 2.2997887134552
Train Epoch: 9 [39680/50000 (79%)]      Loss: 2.3023054599761963
Train Epoch: 9 [40960/50000 (82%)]      Loss: 2.296722888946533
Train Epoch: 9 [42240/50000 (84%)]      Loss: 2.297020673751831
Train Epoch: 9 [43520/50000 (87%)]      Loss: 2.2994067668914795
Train Epoch: 9 [44800/50000 (90%)]      Loss: 2.3039379119873047
Train Epoch: 9 [46080/50000 (92%)]      Loss: 2.3043441772460938
Train Epoch: 9 [47360/50000 (95%)]      Loss: 2.2950546741485596
Train Epoch: 9 [48640/50000 (97%)]      Loss: 2.2979836463928223
Train Epoch: 9 [31200/50000 (100%)]     Loss: 2.2967684268951416
Epoch 9: Training Loss: 0.0180, Accuracy: 11.45%
Test Loss: 0.0182, Accuracy: 11.44%
Train Epoch: 10 [0/50000 (0%)]  Loss: 2.2938356399536133
Train Epoch: 10 [1280/50000 (3%)]       Loss: 2.300027370452881
Train Epoch: 10 [2560/50000 (5%)]       Loss: 2.3004868030548096
Train Epoch: 10 [3840/50000 (8%)]       Loss: 2.3000080585479736
Train Epoch: 10 [5120/50000 (10%)]      Loss: 2.297945976257324
Train Epoch: 10 [6400/50000 (13%)]      Loss: 2.3036718368530273
Train Epoch: 10 [7680/50000 (15%)]      Loss: 2.304182291030884
Train Epoch: 10 [8960/50000 (18%)]      Loss: 2.296170234680176
Train Epoch: 10 [10240/50000 (20%)]     Loss: 2.2996981143951416
Train Epoch: 10 [11520/50000 (23%)]     Loss: 2.30572772026062
Train Epoch: 10 [12800/50000 (26%)]     Loss: 2.2968389987945557
Train Epoch: 10 [14080/50000 (28%)]     Loss: 2.3013663291931152
Train Epoch: 10 [15360/50000 (31%)]     Loss: 2.2983014583587646
Train Epoch: 10 [16640/50000 (33%)]     Loss: 2.29931902885437
Train Epoch: 10 [17920/50000 (36%)]     Loss: 2.2976903915405273
Train Epoch: 10 [19200/50000 (38%)]     Loss: 2.2966160774230957
Train Epoch: 10 [20480/50000 (41%)]     Loss: 2.302746057510376
Train Epoch: 10 [21760/50000 (43%)]     Loss: 2.2986984252929688
Train Epoch: 10 [23040/50000 (46%)]     Loss: 2.295294761657715
Train Epoch: 10 [24320/50000 (49%)]     Loss: 2.306084394454956
Train Epoch: 10 [25600/50000 (51%)]     Loss: 2.3026583194732666
Train Epoch: 10 [26880/50000 (54%)]     Loss: 2.2933707237243652
Train Epoch: 10 [28160/50000 (56%)]     Loss: 2.3034281730651855
Train Epoch: 10 [29440/50000 (59%)]     Loss: 2.2986772060394287
Train Epoch: 10 [30720/50000 (61%)]     Loss: 2.2954587936401367
Train Epoch: 10 [32000/50000 (64%)]     Loss: 2.2966861724853516
Train Epoch: 10 [33280/50000 (66%)]     Loss: 2.294684648513794
Train Epoch: 10 [34560/50000 (69%)]     Loss: 2.294586181640625
Train Epoch: 10 [35840/50000 (72%)]     Loss: 2.303431510925293
Train Epoch: 10 [37120/50000 (74%)]     Loss: 2.3029026985168457
Train Epoch: 10 [38400/50000 (77%)]     Loss: 2.2977676391601562
Train Epoch: 10 [39680/50000 (79%)]     Loss: 2.296052932739258
Train Epoch: 10 [40960/50000 (82%)]     Loss: 2.3019068241119385
Train Epoch: 10 [42240/50000 (84%)]     Loss: 2.3017663955688477
Train Epoch: 10 [43520/50000 (87%)]     Loss: 2.3016550540924072
Train Epoch: 10 [44800/50000 (90%)]     Loss: 2.30092716217041
Train Epoch: 10 [46080/50000 (92%)]     Loss: 2.2985923290252686
Train Epoch: 10 [47360/50000 (95%)]     Loss: 2.302210569381714
Train Epoch: 10 [48640/50000 (97%)]     Loss: 2.3008060455322266
Train Epoch: 10 [31200/50000 (100%)]    Loss: 2.2963390350341797
Epoch 10: Training Loss: 0.0180, Accuracy: 11.65%
Test Loss: 0.0182, Accuracy: 11.33%
```

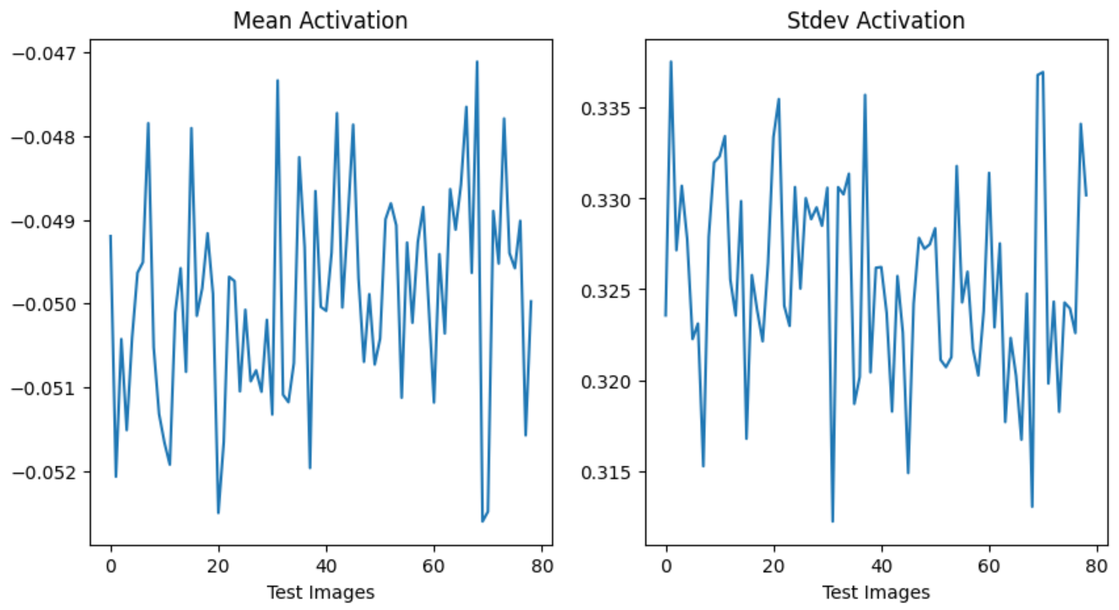
[15]:
```
%tensorboard --logdir log/CIFAR10 --port=8009
```

```
<IPython.core.display.HTML object>
```

[20]:
```
# display weight filters

# access the first convolutional layer
first_conv_layer = model.features[0]

# get the weight data from the layer
filters = first_conv_layer.weight.data

# normalize the weights to visualize them
filters = filters - filters.min()
filters = filters / filters.max()

# plot and visualize the filters
fig, axes = plt.subplots(4, 8, figsize=(12, 6))
for i, ax in enumerate(axes.ravel()):
    ax.imshow(filters[i].cpu().numpy()[0], cmap='gray')
    ax.axis('off')

plt.show()
```



[21]:
```python
# display activation statistics

model.eval()  # Set the model to evaluation mode

conv_layer = model.features[0]  # Assuming you want statistics for the first␣
↪convolutional layer
activation_stats = []  # To store statistics

with torch.no_grad():
    for data, _ in test_loader:
        if cuda:
            data = data.cuda()

        # pass the test data through the first convolutional layer
        activations = conv_layer(data)

        # calculate statistics (mean, standard deviation)
        mean = activations.mean().item()
        std = activations.std().item()

        activation_stats.append((mean, std))

activation_stats = torch.tensor(activation_stats)

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.plot(activation_stats[:, 0].numpy())
plt.title("Mean Activation")
plt.xlabel("Test Images")
plt.subplot(1, 2, 2)
plt.plot(activation_stats[:, 1].numpy())
plt.title("Stdev Activation")
plt.xlabel("Test Images")
plt.show()
```



```
[ ]:
```