Computer vision 2025 Assignment 3

Manh Ha Nguyen - a1840406

Deep Learning for Perception Tasks

This assignment contains 2 questions. The first question probes understanding of deep learning for classification. The second question requires you to write a short description of a Computer Vision method. You wil need to submit two separate PDF files, one for each question.

Question 1: A simple classifier, 20 marks (60% of the assignment)

For this exercise, we provide demo code showing how to train a network on a small dataset called Fashion-MNIST. Please run through the code "tutorial-style" to get a sense of what it is doing. Then use the code alongside lecture notes and other resources to understand how to use pytorch libraries to implement, train and use a neural network.

For the Fashion-MNIST dataset the labels from 0-9 correspond to various clothing classes so you might find it convenient to create a python list as follows:

class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']

You will need to answer various questions about the system, how it operates, the results of experiments with it and make modifications to it yourself. You can change the training scheme and the network structure.

Organize your own text and code cell to show the answer of each question below.

Detailed requirements:

Q1.1 (1 point)

Extract 3 images of different types of clothing from the training dataset, print out the size/shape of the training images, and display the three with their corresponding labels.

Q1.2 (2 point)

Run the training code for 10 epochs, for different values of the learning rate. Fill in the table below and plot the loss curves for each experiment:

Lr	Accuracy
1	19.97%
0.1	87.53%
0.01	83.56%
0.001	71.35%

Q1.3 (3 point)

Report the number of epochs when the network converges (or number of epochs for the best accuracy, if it fails to converge). Fill in the table below and plot the loss curve for each experiment. Please run the code for more than 10 epochs (e.g., 50 or 100) and report when you observe convergence.:

Lr	Accuracy	Epoch
1	10%	1
0.1	86.85%	13
0.01	86.88%	30
0.001	80.85%	30

Q1.4 (2 points)

Compare the results in table 1 and table 2, what is your observation and your understanding of learning rate? Observations:

- High Learning Rate (Lr=1):
  - Initial (10 epochs): Accuracy is very low (19.74%).
  - Extended training: The accuracy dropped further to 10% by epoch 1, indicating no meaningful learning and unstable training.
- Moderate Learning Rate (Lr=0.1):
  - Initial (10 epochs): Accuracy was relatively high (87.62%).
  - Extended training: Accuracy slightly decreased to 86.85% at epoch 13, suggesting quick convergence but potential slight overfitting or instability with further epochs.
- Lower Learning Rate (Lr=0.01):
  - Initial (10 epochs): Accuracy was decent (83.17%).
  - Extended training: Improved accuracy to 86.88% by epoch 30, indicating stable and steady convergence.
- Very Low Learning Rate (Lr=0.001):
  - Initial (10 epochs): Accuracy was lower (70.73%), indicating slower initial learning.
  - Extended training: Increased accuracy to 80.85% at epoch 30, demonstrating slow but stable learning.

Understanding of Learning Rate:

- The learning rate (Lr) significantly impacts training stability, convergence speed, and final accuracy.
- High learning rates (Lr=1) lead to rapid but unstable training, preventing effective learning and resulting in poor accuracy.
- Moderate learning rates (Lr=0.1) enable fast and good initial convergence, but risk instability or slight degradation if trained too long.
- Lower learning rates (Lr=0.01) offer balanced stability, achieving reliable and consistent improvement in accuracy over a moderate number of epochs.
- Very low learning rates (Lr=0.001) slow down training, requiring more epochs to achieve comparable performance, though they often yield stable and gradually improving results.

Q1.5 (5 points)

Build a wider network by modifying the code that constructs the network so that the hidden layer(s) contain more perceptrons, and record the accuracy along with the number of trainable parameters in your model. Now modify the original network to be deeper instead of wider (i.e. by adding more hidden layers). Record your accuracy and network size findings. Plot the loss curve for each experiment. Also plot the test accuracy and loss for both the wider and deeper architectures and discuss what you observe. Write down your conclusions about changing the network structure?

Structures	**Accuracy**	**Parameters**
Base	86.87%	669706
Deeper	86.60%	1195018
Wider	86.98%	1863690

Observations:

- Accuracy and Parameters:
  - Base Network: Accuracy = 86.87%, Parameters = 669,706.
  - Wider Network: Slight accuracy increase (86.98%) but significant parameter increase (1,863,690).
  - Deeper Network: Slight accuracy decrease (86.60%) with moderate parameter increase (1,195,018).
- Loss Curve Analysis:
  - Base vs. Wider Network: Loss curves for training and testing are virtually identical, suggesting that additional neurons (width) provided minimal improvements despite greatly increasing model complexity.
  - Deeper Network: Exhibited slightly slower or less effective convergence compared to both base and wider networks, suggesting challenges in optimizing additional depth.

Conclusions:

- Widening the network slightly improved accuracy but at a significant computational cost, with minimal visible changes in training dynamics.
- Deepening the network without optimization techniques slightly decreased performance and introduced potential optimization challenges.
- Practically, modest widening can be beneficial, but significant increases in parameters must be justified by notable accuracy gains.
- Simply adding depth without additional optimization strategies did not yield meaningful performance improvements, suggesting limited effectiveness of deeper architectures in this specific case.

Q1.6 (2 points)

Calculate the mean of the gradients of the loss to all trainable parameters. Plot the gradients curve for the first 100 training steps. What are your observations? Note that this gradients will be saved with the training weight automatically after you call loss.backwards(). Hint: the mean of the gradients decrease.

Discussion:

- Learning rate = 1e-3
    - Two-phase behavior:
        - Phase 1 (acceleration): in the first ~2 000 steps (2 epochs) the mean gradient drifts steadily away from zero, reaching its largest magnitude around –2.7e-4. This shows the model picking up speed once it escapes its random initialization.
        - Phase 2 (decay): after that peak, the gradient magnitude slowly shrinks back toward zero over the remaining 13 epochs, settling around –1e-4 by the end.
    - Interpretation: the “rise-then-fall” pattern reveals an initial exploratory phase (larger updates to escape plateaus) followed by a fine-tuning phase (smaller updates as the model homes in on a minimum).
- Learning rate = 1e-2
    - Rapid peak, noisy plateau:
        - The mean gradient ramps up even faster—peaking near -2.4e-4 within the first few hundred steps.
        - Rather than decaying cleanly, it then oscillates heavily around ±1e-4 for the rest of training.
    - Interpretation: you get a quicker “burst” of learning but with much more jitter. Updates remain large and noisy, indicating this LR is near the stability limit: it still converges, but less smoothly than 1e-3.
- Learning rate = 1e-1
    - Persistent instability:
        - From step 0 onward, gradients swing wildly between –2.5e-4 and +4e-4, with no discernible rise-or-fall trend.

        - There’s no phase where gradient magnitudes settle or decay—just constant high-frequency noise.
    - Interpretation: this LR is too large. The optimizer continually overshoots any basin and never enters a fine-tuning regime, so the network fails to converge.
- Conclusion:
    - For suitably small learning rates (1e-3 and 1e-2), the mean gradient magnitude decreases over training—reflecting progressively smaller parameter updates as the model approaches a minimum.
    - In contrast, a too-large rate (1e-1) yields noisy, non-decaying gradients and no convergence.

For more explanation of q1.7, you could refer to the following simple instructions: https://colab.research.google.com/drive/1XAsyNegGSvMf3_B6MrsXht7-fHqtJ7OW?usp=sharing

Q1.7 (5 points)

Modify the network structure and training/test to use a small convolutional neural network instead of an MLP. Discuss your findings with regard to convergence, accuracy and number of parameters, relative to MLPs.

Hint: Look at the structure of the CNN in the Workshop 3 examples.

Discussion:

Structures	Accuracy	Parameters	Learning Rate	Epochs
Base	86.87%	669706	0.01	30
CNN	87.79%	44374	0.01	30

Observations:

- Accuracy and Parameters Comparison:
    - Base MLP: Achieved accuracy of 86.87% with 669,706 parameters.
    - CNN: Improved accuracy (87.79%) with significantly fewer parameters (44,374).
- Loss Curve Analysis (CNN):
    - The CNN structure demonstrated rapid convergence in both training and test loss curves.
    - The CNN converged smoothly and quickly within the first few epochs, maintaining stable and closely matching train and test losses, indicating excellent generalization and minimal overfitting.

Key Insights:

- Convergence:

    - CNNs converge faster and more stably compared to MLPs. This likely results from CNNs' ability to effectively exploit spatial structures within image data.
- Accuracy:
    - CNN achieved a higher accuracy compared to the MLP despite having drastically fewer parameters, highlighting superior efficiency.
- Number of Parameters:
    - CNN architecture dramatically reduced model complexity while still improving performance, demonstrating superior parameter efficiency for image classification tasks.

Conclusions:

- CNN are significantly more effective than MLP.
- CNN architectures achieve better accuracy, faster and more stable convergence, and lower complexity (fewer parameters).
- The convolutional operation, combined with pooling layers, allows CNN to capture spatial hierarchies effectively, leading to enhanced generalization performance compared to fully-connected MLP architectures.

Q2 Optional Bonus Question (5 points, upto 20% bonus marks of the assignment)

Experiment with different activation functions (ReLU, Tanh, Sigmoid) and analyse their impact on training performance.(2 points)

In particular, focus your analysis on the Sigmoid activation function and discuss your finding of training with and without Xavier initialization. You may use the provided code for Xavier initialisation for this. (1 points)

Additionally, plot both the gradient and loss curves for your experiments. For gradient analysis, you may select one representative layer to monitor throughout training and briefly explain your choice. (1 points)

Discuss how gradients and loss behave across the network for different activation functions and initialisation methods if you see any difference. (1 points)

Note: the bonus marks wont replace marks for Q3.

Discussion: **Result Table** |Activation|Accuracy (No Xavier)|Accuracy (Xavier Init) |---|---|---| |ReLU|86.78%|87.40%| |Sigmoid|65.16%|76.66%| |Tanh|85.99%|87.07%|

Set up:

- Architecture & task: three-layer MLP trained for 30 epochs on the assignment dataset (same optimiser & learning-rate in all runs).
- Variants tested

    - Activations: ReLU, Tanh, Sigmoid
    - Initialisation: Kaiming (default) vs Xavier
- Monitored layer for gradients: first hidden layer
    - Nearest to the input, so any exploding/vanishing trend appears here earliest and then cascades; easy to compare across activations because its fan-in/out is identical in every run.

Observation:

- Xavier gives only approximately 0.6~1 pp boost for ReLU/Tanh, but a big +11.5 pp jump for Sigmoid.

- Sigmoid Activation:

    - Loss Curve:
        - Kaiming: 12-epoch flat plateau at around 2.29; sharp fall only after epoch 12.
        - Xavier: Smooth decline from epoch 1; hits around 0.6 by epoch 30.
    - Gradient:
        - Kaiming: Starts 2~3e-5 then collapses to 0 by around step 40 -> network “frozen”
        - Xavier: Starts 7e-5; decays slowly, still alive past step 60 -> continual learning.
    - Interpretation:
        - When the network starts with default Kaiming initialisation—which is tuned for ReLU but not for a bounded, S-shaped activation like sigmoid—many neurons receive large enough inputs that they sit in the flat “saturated” ends of the curve, where the slope of the sigmoid function is essentially zero. Because the back-propagated error signal is multiplied by that near-zero slope, the gradients all but vanish, learning stalls, and accuracy tops out around 65 percent.
        - In contrast, switching to Xavier initialisation scales the weights so that the pre-activation values cluster near the sigmoid’s steep, almost-linear centre. This avoids early saturation, preserves healthy gradient flow through the network, and ultimately yields an 11.5 pp improvement in final accuracy.

  => Variance-preserving initialisation is critical when using bounded, non-zero-centred activations like Sigmoid

- Gradient & Loss Curves:

    - ReLU: loss plunges in first 5 epochs with either init; gradients remain non-zero but taper slowly. Xavier simply gives a larger, steadier gradient band, shaving a few epochs off convergence.
    - Sigmoid: with Kaiming init, gradient plataue early -> flat loss; Xavier shifts entire trace up, preventing collapse.
    - Tanh: already zero-centred, so no severe saturation; Xavier mostly enlarges early gradients, giving a modest speed-up.

Conclusion:

- ReLU remains robust with both weights initialisation, but balanced variance still accelerates early optimisation.
- Tanh benefits modestly from Xavier; its zero-centred nature already mitigates gradient drift.
- Sigmoid requires variance-aware initialisation to avoid plateaus.
- Monitoring the first hidden layer is sufficient to diagnose exploding/vanishing trends because these propagate backward to deeper layers.
- For deeper nets or more stubborn datasets, combine good initialisation with normalisation layers or switch to non-saturating activations to have a better chance of stable training.

Question 3: Proposal for Practical Applications (40%)

In this part of the assignment you need to write a report about an application of a computer vision algorithm or technique. This can either be an application that you are aware of and possibly even use, such as a phone app, or it could be speculation -- an application that you think would benefit from using computer vision.

Begin by choosing a particular CV idea, method or problem area, such as:

a. removing noise in an image

b. increasing the resolution of an image

c. detecting and/or identifying objects in an image

d. segmenting images into constituents parts

e. estimating the depth of an object from one or more images

f. estimating the motion of two objects in different frames

g. others

Now think about various ways your chosen technique could be used. Here is a list of possible applications you could consider, but you are not restricted to this list, and there will be credit given for sensible invention outside this list (but no penalty if you don't want to be "inventive"): image editing systems in your phone; enhancement of images from old film; obstacle detection and avoidance for a domestic robot; facial recognition for phone security; cancer detection; person tracking and re-identification in security cameras; sport decision review systems; road-sign detection and interpretation for self-driving cars.

This is a little bit back-to-front from what might happen in real life in which the application usually motivates the solution, but the emphasis here is on an understanding of the CV technique.

You need to write a short report (800 words max) in which you do the following:

1. Clearly define the CV problem/area and describe its application scenarios
2. Briefly describe a solution based on image processing, computer vision and/or machine learning.
3. Discuss the advantages and the limitations of this method in various application scenarios.
4. It is important that you will define a useful metric to evaluate the performance of your method and discuss its tradeoff specific to the problem you have chosen.
5. You are welcome to cite existing work and take inspiration form literature addressing the problem you choose.

Hint1: Submit an individual pdf report for question 2.

Hint2: Organise your report well

Hint3: You can use diagrams, flow charts or other figures in your report for better understanding of your solution.

** For Q3, you do not need to implement your solution; just write the proposal/report and submit it as a separate PDF **

Set up and Import

In [16]:

```
import torch, torchvision
from torch import nn
from torch.utils.data import DataLoader
from torchvision import datasets, transforms
import matplotlib.pyplot as plt
import numpy as np, time, math, pprint
%matplotlib inline

if torch.cuda.is_available():
    device = "cuda"
else:
    device =  "cpu"

print("Using", device)
class_names = ['T-shirt/top','Trouser','Pullover','Dress','Coat',
               'Sandal','Shirt','Sneaker','Bag','Ankle boot']
```

```
Using cpu
```

Helper functions

In [17]:

```
# Define the neural network
class NeuralNetwork(nn.Module):
    def __init__(self):
        super().__init__()
        self.flatten = nn.Flatten()
        self.linear_relu_stack = nn.Sequential(
            nn.Linear(28 * 28, 512),
            nn.ReLU(),
            nn.Linear(512, 512),
            nn.ReLU(),
            nn.Linear(512, 10),
        )

    def forward(self, x):
        x = self.flatten(x)
        return self.linear_relu_stack(x)

# Xavier weight initialization
def init_weights(m):
    if isinstance(m, nn.Linear):
        nn.init.xavier_uniform_(m.weight)
        if m.bias is not None:
            nn.init.zeros_(m.bias)

# Training loop
def train(dataloader, model, loss_fn, optimizer):
    model.train()
    losses = []
    total = len(dataloader.dataset)
    for batch_idx, (X, y) in enumerate(dataloader):
        X, y = X.to(device), y.to(device)

        optimizer.zero_grad()
        preds = model(X)
        loss = loss_fn(preds, y)
        loss.backward()
        optimizer.step()

        losses.append(loss.item())
        if batch_idx % 100 == 0:
            seen = batch_idx * dataloader.batch_size
            print(f"  [Batch {batch_idx:03d}] loss: {loss:.6f}  ({seen}/{

    return np.mean(losses)

# Test loop
def test(dataloader, model, loss_fn):
    model.eval()
    losses, correct = [], 0
    with torch.no_grad():
        for X, y in dataloader:
            X, y = X.to(device), y.to(device)
            preds = model(X)
            losses.append(loss_fn(preds, y).item())
            correct += (preds.argmax(dim=1) == y).sum().item()

    avg_loss = np.mean(losses)
    accuracy = correct / len(dataloader.dataset)
    return accuracy, avg_loss

# Plotting function
def plot_train_test_loss(train_losses, test_losses, test_accuracies, lr):
    epochs = np.arange(1, len(train_losses) + 1)
    best_epoch = int(np.argmax(test_accuracies)) + 1
    best_acc = test_accuracies[best_epoch - 1]
    print(f"Best accuracy = {best_acc*100:.2f}% at epoch {best_epoch}\n")

    plt.figure()
    plt.plot(epochs, train_losses, label="Train Loss")
    plt.plot(epochs, test_losses,  label="Test Loss")
    plt.xlabel("Epoch")
    plt.ylabel("Loss")
    plt.title(f"Loss Curves (lr={lr})")
    plt.legend()
    plt.show()
```

Question 1

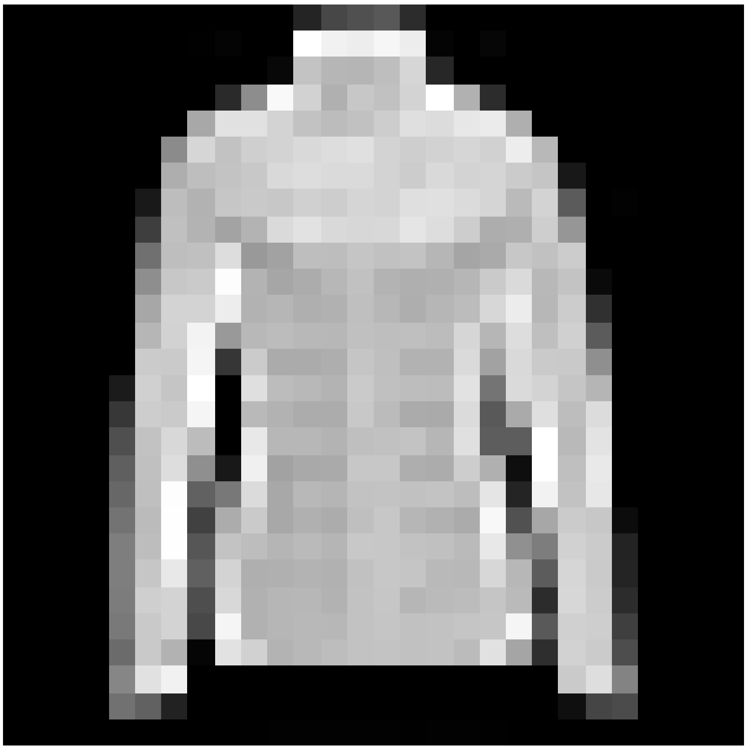
Q1.1

In [18]:

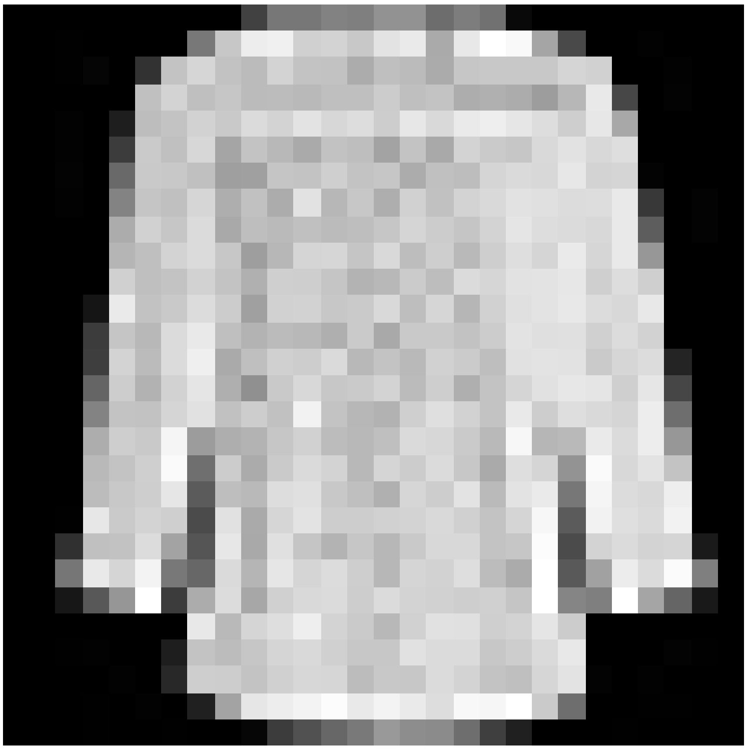
```
# load train and test data
transform = transforms.ToTensor()
train_data = datasets.FashionMNIST("data", train=True , download=True, tr
test_data  = datasets.FashionMNIST("data", train=False, download=True, tr

batch_size = 64

# Create data loaders.
train_dataloader = DataLoader(train_data, batch_size=batch_size)
test_dataloader = DataLoader(test_data, batch_size=batch_size)

for X, y in test_dataloader:
    print("Shape of X [N, C, H, W]: ", X.shape)
    print("Shape of y: ", y.shape, y.dtype)
    break

print(f"Training set: {len(train_data)} images shape {train_data[0][0].sh

# Extract three random distinct-class samples
idxs, labels, imgs = [], [], []
while len(labels) < 3:
    i = np.random.randint(0, len(train_data))
    img, lab = train_data[i]
    if lab not in labels:
        idxs.append(i); labels.append(lab); imgs.append(img)

for img, lab in zip(imgs, labels):
    plt.figure()
    plt.imshow(img.squeeze(), cmap="gray")
    plt.title(class_names[lab])
    plt.axis("off")
    plt.show()
```

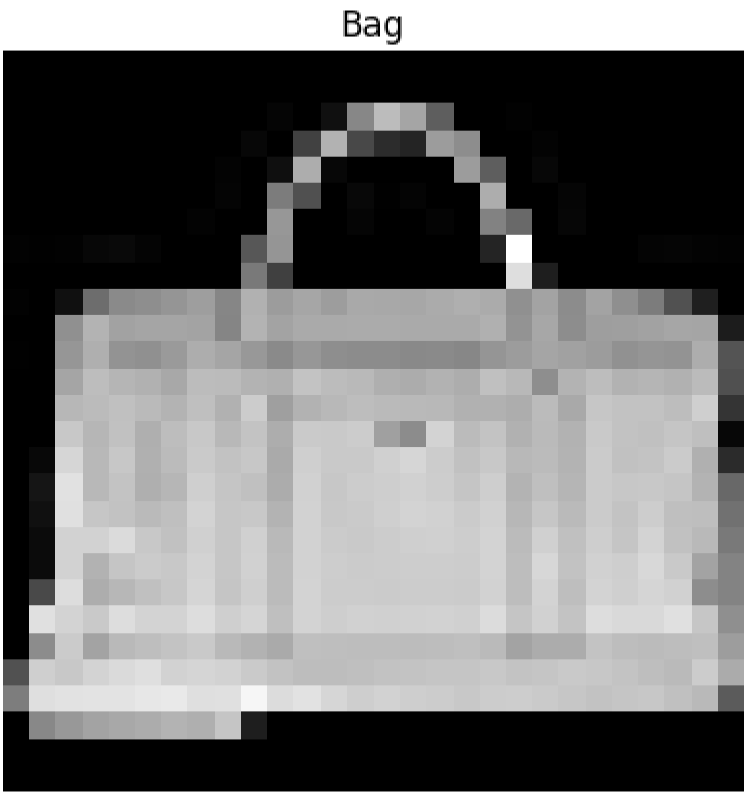
```
Shape of X [N, C, H, W]:  torch.Size([64, 1, 28, 28])
Shape of y:  torch.Size([64]) torch.int64
Training set: 60000 images shape torch.Size([1, 28, 28])
```

Coat



Shirt

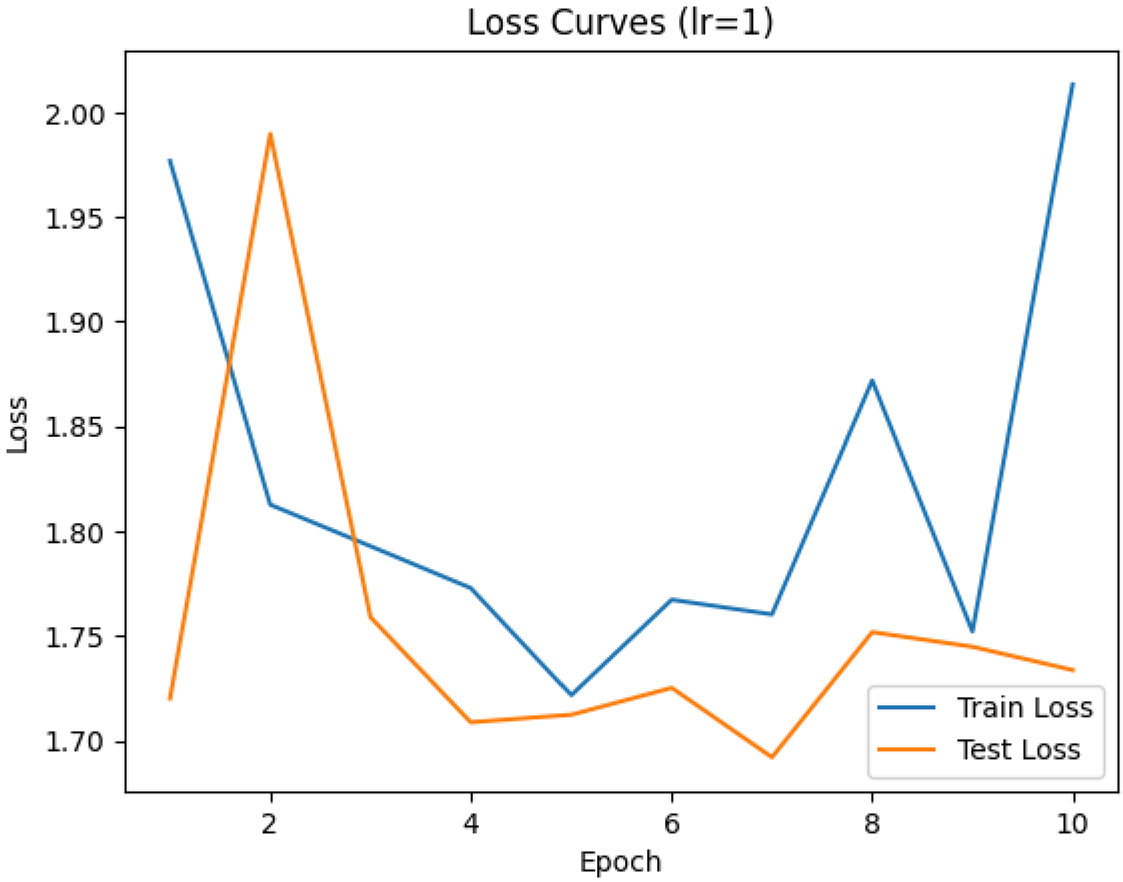




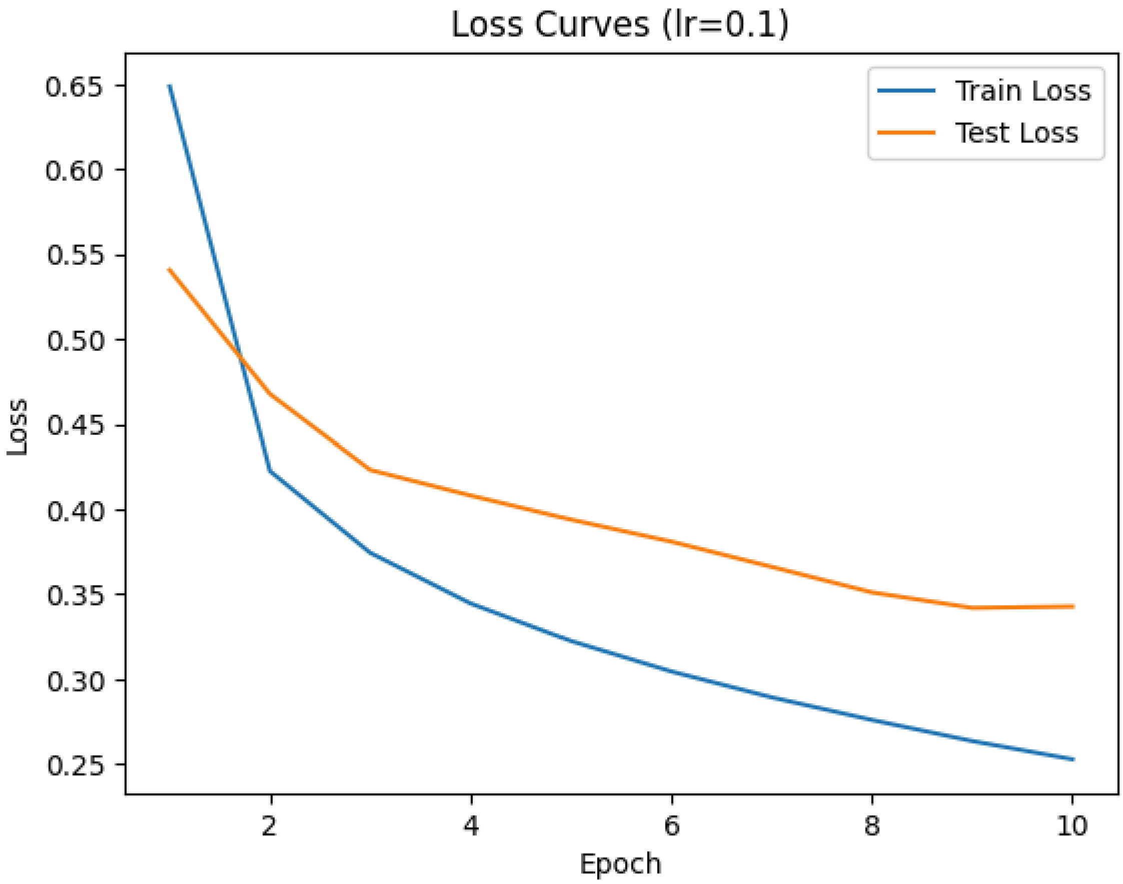
Q1.2

In [4]:
```
# Parameters
loss_fn = nn.CrossEntropyLoss()
learning_rates = [1, 1e-1, 1e-2, 1e-3]
epochs = 10

# Experiments
results = {}
for lr in learning_rates:
    print(f"Learning rate: {lr} \n")
    model = NeuralNetwork().to(device)
    # model.apply(init_weights)
    optimizer = torch.optim.SGD(model.parameters(), lr=lr)

    train_losses, test_losses, test_accs = [], [], []
    for epoch in range(1, epochs + 1):
        print(f"Epoch {epoch}/{epochs}")
        train_loss = train(train_dataloader, model, loss_fn, optimizer)
        acc, test_loss = test(test_dataloader, model, loss_fn)

        train_losses.append(train_loss)
        test_losses.append(test_loss)
        test_accs.append(acc)

    plot_train_test_loss(train_losses, test_losses, test_accs, lr)
    results[lr] = test_accs[-1]

# Final result
print("Final test accuracies:")
for lr, acc in results.items():
    print(f"  LR={lr:<7}: {acc*100:.2f}%")
```

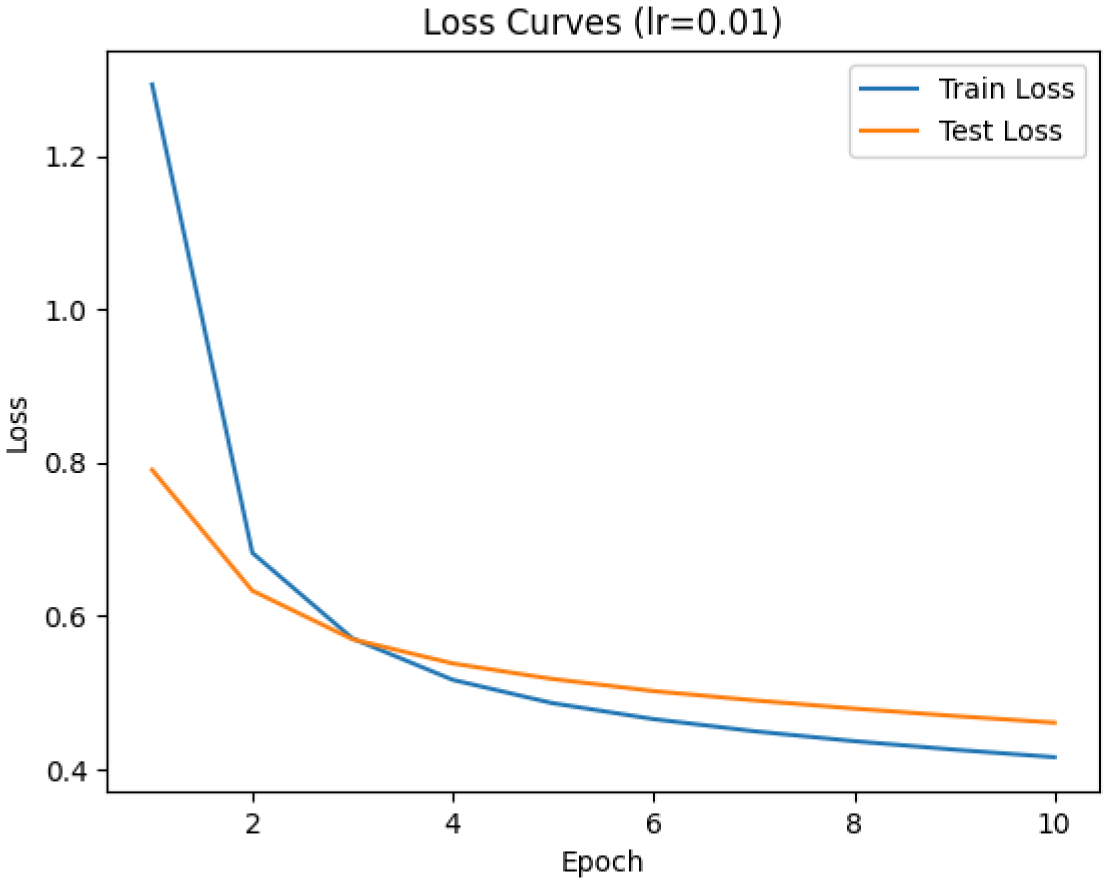
```
Learning rate: 1

Epoch 1/10
  [Batch 000] loss: 2.304520  (0/60000)
  [Batch 100] loss: 3.745166  (6400/60000)
  [Batch 200] loss: 1.867681  (12800/60000)
  [Batch 300] loss: 1.764732  (19200/60000)
  [Batch 400] loss: 1.871515  (25600/60000)
  [Batch 500] loss: 1.558125  (32000/60000)
  [Batch 600] loss: 1.648953  (38400/60000)
  [Batch 700] loss: 1.726286  (44800/60000)
  [Batch 800] loss: 1.694548  (51200/60000)
  [Batch 900] loss: 1.685721  (57600/60000)
Epoch 2/10
  [Batch 000] loss: 1.762296  (0/60000)
  [Batch 100] loss: 1.758461  (6400/60000)
  [Batch 200] loss: 1.814194  (12800/60000)
  [Batch 300] loss: 1.814709  (19200/60000)
  [Batch 400] loss: 1.648760  (25600/60000)
  [Batch 500] loss: 1.754205  (32000/60000)
  [Batch 600] loss: 1.694471  (38400/60000)
  [Batch 700] loss: 1.826227  (44800/60000)
  [Batch 800] loss: 1.692935  (51200/60000)
  [Batch 900] loss: 1.778063  (57600/60000)
Epoch 3/10
  [Batch 000] loss: 1.969276  (0/60000)
  [Batch 100] loss: 1.719424  (6400/60000)
  [Batch 200] loss: 1.779114  (12800/60000)
  [Batch 300] loss: 1.780869  (19200/60000)
  [Batch 400] loss: 1.836308  (25600/60000)
  [Batch 500] loss: 1.783931  (32000/60000)
  [Batch 600] loss: 1.702521  (38400/60000)
  [Batch 700] loss: 1.760956  (44800/60000)
  [Batch 800] loss: 1.743449  (51200/60000)
  [Batch 900] loss: 1.621454  (57600/60000)
Epoch 4/10
  [Batch 000] loss: 1.895454  (0/60000)
  [Batch 100] loss: 1.973461  (6400/60000)
  [Batch 200] loss: 1.654655  (12800/60000)
  [Batch 300] loss: 1.738359  (19200/60000)
  [Batch 400] loss: 1.667274  (25600/60000)
  [Batch 500] loss: 1.770437  (32000/60000)
  [Batch 600] loss: 1.695693  (38400/60000)
  [Batch 700] loss: 1.660135  (44800/60000)
  [Batch 800] loss: 1.684089  (51200/60000)
  [Batch 900] loss: 1.674419  (57600/60000)
Epoch 5/10
  [Batch 000] loss: 1.800173  (0/60000)
  [Batch 100] loss: 1.581471  (6400/60000)
  [Batch 200] loss: 1.734824  (12800/60000)
  [Batch 300] loss: 1.719339  (19200/60000)
  [Batch 400] loss: 1.698890  (25600/60000)
  [Batch 500] loss: 1.641073  (32000/60000)
  [Batch 600] loss: 1.475748  (38400/60000)
  [Batch 700] loss: 1.597176  (44800/60000)
  [Batch 800] loss: 1.780442  (51200/60000)
  [Batch 900] loss: 1.633255  (57600/60000)
Epoch 6/10
  [Batch 000] loss: 1.671549  (0/60000)
  [Batch 100] loss: 1.493161  (6400/60000)
  [Batch 200] loss: 1.736684  (12800/60000)
  [Batch 300] loss: 1.743098  (19200/60000)
  [Batch 400] loss: 1.700524  (25600/60000)
  [Batch 500] loss: 1.728392  (32000/60000)
  [Batch 600] loss: 1.681731  (38400/60000)
  [Batch 700] loss: 1.757722  (44800/60000)
  [Batch 800] loss: 1.670970  (51200/60000)
  [Batch 900] loss: 1.665831  (57600/60000)
Epoch 7/10
  [Batch 000] loss: 1.692327  (0/60000)
  [Batch 100] loss: 1.830861  (6400/60000)
  [Batch 200] loss: 1.783130  (12800/60000)
  [Batch 300] loss: 1.861868  (19200/60000)
  [Batch 400] loss: 1.683703  (25600/60000)
  [Batch 500] loss: 1.844684  (32000/60000)
  [Batch 600] loss: 1.727775  (38400/60000)
  [Batch 700] loss: 1.745950  (44800/60000)
  [Batch 800] loss: 1.659182  (51200/60000)
  [Batch 900] loss: 1.666198  (57600/60000)
Epoch 8/10
  [Batch 000] loss: 1.680739  (0/60000)
  [Batch 100] loss: 1.685716  (6400/60000)
  [Batch 200] loss: 1.696669  (12800/60000)
  [Batch 300] loss: 1.783770  (19200/60000)
  [Batch 400] loss: 1.520915  (25600/60000)
  [Batch 500] loss: 1.675599  (32000/60000)
  [Batch 600] loss: 1.563213  (38400/60000)
  [Batch 700] loss: 1.678965  (44800/60000)
  [Batch 800] loss: 1.729150  (51200/60000)
  [Batch 900] loss: 1.693452  (57600/60000)
Epoch 9/10
  [Batch 000] loss: 1.695339  (0/60000)
  [Batch 100] loss: 1.598138  (6400/60000)
  [Batch 200] loss: 1.747082  (12800/60000)
  [Batch 300] loss: 1.796686  (19200/60000)
  [Batch 400] loss: 1.649111  (25600/60000)
  [Batch 500] loss: 1.739203  (32000/60000)
  [Batch 600] loss: 1.668445  (38400/60000)
  [Batch 700] loss: 1.673965  (44800/60000)
  [Batch 800] loss: 1.708380  (51200/60000)
  [Batch 900] loss: 1.740823  (57600/60000)
Epoch 10/10
  [Batch 000] loss: 1.698370  (0/60000)
  [Batch 100] loss: 1.580893  (6400/60000)
  [Batch 200] loss: 1.683787  (12800/60000)
  [Batch 300] loss: 1.759484  (19200/60000)
  [Batch 400] loss: 1.642552  (25600/60000)
  [Batch 500] loss: 1.912857  (32000/60000)
  [Batch 600] loss: 1.596849  (38400/60000)
  [Batch 700] loss: 1.720926  (44800/60000)
  [Batch 800] loss: 1.709398  (51200/60000)
  [Batch 900] loss: 1.663048  (57600/60000)
Best accuracy = 26.87% at epoch 4
```


Loss Curves (lr=1)
Loss
2.00
1.95
1.90
1.85
1.80
1.75
1.70
Epoch
2
4
6
8
10
Train Loss
Test Loss

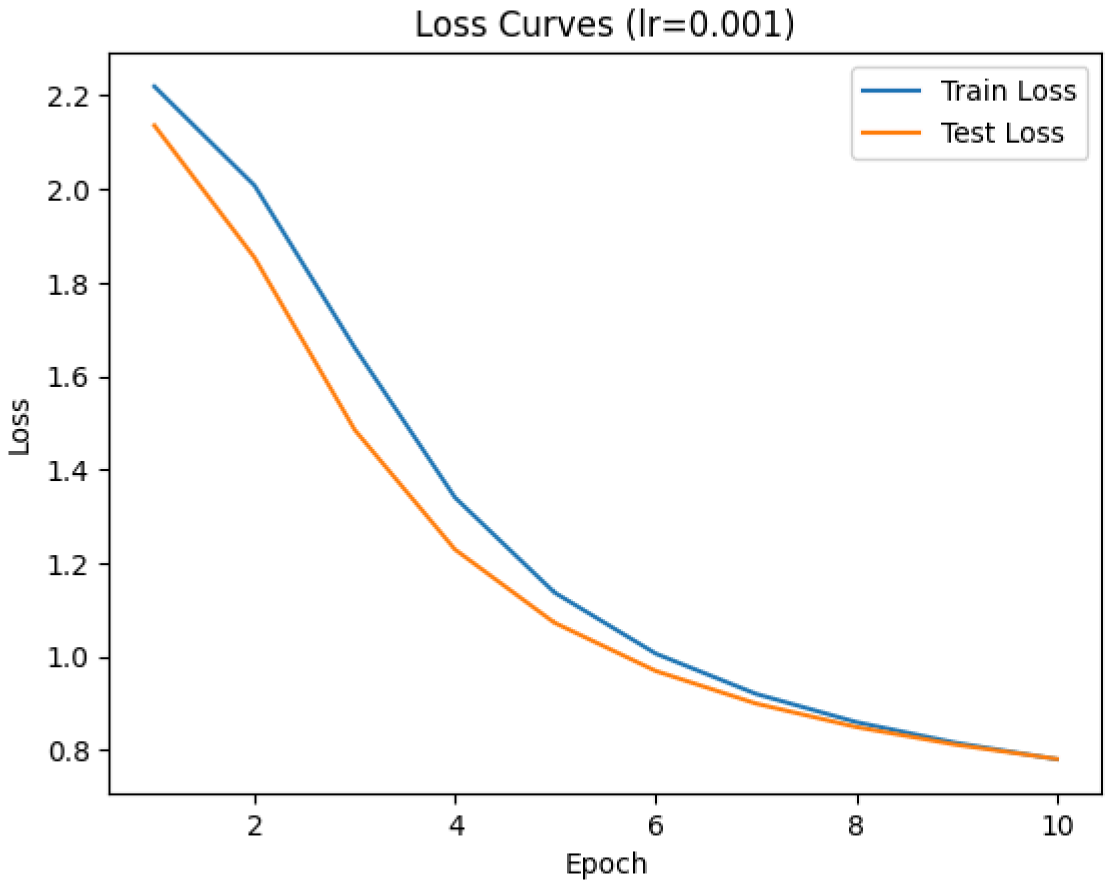
```
Learning rate: 0.1

Epoch 1/10
  [Batch 000] loss: 2.316946  (0/60000)
  [Batch 100] loss: 0.898326  (6400/60000)
  [Batch 200] loss: 0.574954  (12800/60000)
  [Batch 300] loss: 0.700685  (19200/60000)
  [Batch 400] loss: 0.597572  (25600/60000)
  [Batch 500] loss: 0.510775  (32000/60000)
  [Batch 600] loss: 0.536074  (38400/60000)
  [Batch 700] loss: 0.597469  (44800/60000)
  [Batch 800] loss: 0.590598  (51200/60000)
  [Batch 900] loss: 0.458736  (57600/60000)
Epoch 2/10
  [Batch 000] loss: 0.423019  (0/60000)
  [Batch 100] loss: 0.444391  (6400/60000)
  [Batch 200] loss: 0.376141  (12800/60000)
  [Batch 300] loss: 0.437111  (19200/60000)
  [Batch 400] loss: 0.407166  (25600/60000)
  [Batch 500] loss: 0.448467  (32000/60000)
  [Batch 600] loss: 0.410118  (38400/60000)
  [Batch 700] loss: 0.511516  (44800/60000)
  [Batch 800] loss: 0.491207  (51200/60000)
  [Batch 900] loss: 0.430814  (57600/60000)
Epoch 3/10
  [Batch 000] loss: 0.320760  (0/60000)
  [Batch 100] loss: 0.364480  (6400/60000)
  [Batch 200] loss: 0.318199  (12800/60000)
  [Batch 300] loss: 0.366680  (19200/60000)
  [Batch 400] loss: 0.346239  (25600/60000)
  [Batch 500] loss: 0.408265  (32000/60000)
  [Batch 600] loss: 0.345427  (38400/60000)
  [Batch 700] loss: 0.458336  (44800/60000)
  [Batch 800] loss: 0.445271  (51200/60000)
  [Batch 900] loss: 0.406637  (57600/60000)
Epoch 4/10
  [Batch 000] loss: 0.261887  (0/60000)
  [Batch 100] loss: 0.321079  (6400/60000)
  [Batch 200] loss: 0.266938  (12800/60000)
  [Batch 300] loss: 0.323000  (19200/60000)
  [Batch 400] loss: 0.316208  (25600/60000)
  [Batch 500] loss: 0.390194  (32000/60000)
  [Batch 600] loss: 0.311593  (38400/60000)
  [Batch 700] loss: 0.416510  (44800/60000)
  [Batch 800] loss: 0.408962  (51200/60000)
  [Batch 900] loss: 0.403410  (57600/60000)
Epoch 5/10
  [Batch 000] loss: 0.242293  (0/60000)
  [Batch 100] loss: 0.301595  (6400/60000)
  [Batch 200] loss: 0.238142  (12800/60000)
  [Batch 300] loss: 0.298192  (19200/60000)
  [Batch 400] loss: 0.309149  (25600/60000)
  [Batch 500] loss: 0.372646  (32000/60000)
  [Batch 600] loss: 0.288082  (38400/60000)
  [Batch 700] loss: 0.375683  (44800/60000)
  [Batch 800] loss: 0.386314  (51200/60000)
  [Batch 900] loss: 0.378248  (57600/60000)
Epoch 6/10
  [Batch 000] loss: 0.222909  (0/60000)
  [Batch 100] loss: 0.279430  (6400/60000)
  [Batch 200] loss: 0.218954  (12800/60000)
  [Batch 300] loss: 0.279866  (19200/60000)
  [Batch 400] loss: 0.304627  (25600/60000)
  [Batch 500] loss: 0.359317  (32000/60000)
  [Batch 600] loss: 0.275677  (38400/60000)
  [Batch 700] loss: 0.343308  (44800/60000)
  [Batch 800] loss: 0.366247  (51200/60000)
  [Batch 900] loss: 0.372765  (57600/60000)
Epoch 7/10
  [Batch 000] loss: 0.204944  (0/60000)
  [Batch 100] loss: 0.275024  (6400/60000)
  [Batch 200] loss: 0.193876  (12800/60000)
  [Batch 300] loss: 0.257652  (19200/60000)
  [Batch 400] loss: 0.291873  (25600/60000)
  [Batch 500] loss: 0.349696  (32000/60000)
  [Batch 600] loss: 0.249926  (38400/60000)
  [Batch 700] loss: 0.324659  (44800/60000)
  [Batch 800] loss: 0.350161  (51200/60000)
  [Batch 900] loss: 0.350018  (57600/60000)
Epoch 8/10
  [Batch 000] loss: 0.187513  (0/60000)
  [Batch 100] loss: 0.262625  (6400/60000)
  [Batch 200] loss: 0.179189  (12800/60000)
  [Batch 300] loss: 0.247960  (19200/60000)
  [Batch 400] loss: 0.291571  (25600/60000)
  [Batch 500] loss: 0.337740  (32000/60000)
  [Batch 600] loss: 0.238727  (38400/60000)
  [Batch 700] loss: 0.301217  (44800/60000)
  [Batch 800] loss: 0.323949  (51200/60000)
  [Batch 900] loss: 0.316012  (57600/60000)
Epoch 9/10
  [Batch 000] loss: 0.170709  (0/60000)
  [Batch 100] loss: 0.258079  (6400/60000)
  [Batch 200] loss: 0.183671  (12800/60000)
  [Batch 300] loss: 0.234766  (19200/60000)
  [Batch 400] loss: 0.281684  (25600/60000)
  [Batch 500] loss: 0.321049  (32000/60000)
  [Batch 600] loss: 0.228602  (38400/60000)
  [Batch 700] loss: 0.288358  (44800/60000)
  [Batch 800] loss: 0.305416  (51200/60000)
  [Batch 900] loss: 0.297309  (57600/60000)
Epoch 10/10
  [Batch 000] loss: 0.162483  (0/60000)
  [Batch 100] loss: 0.244329  (6400/60000)
  [Batch 200] loss: 0.172799  (12800/60000)
  [Batch 300] loss: 0.221684  (19200/60000)
  [Batch 400] loss: 0.283660  (25600/60000)
  [Batch 500] loss: 0.319394  (32000/60000)
  [Batch 600] loss: 0.225634  (38400/60000)
  [Batch 700] loss: 0.280357  (44800/60000)
  [Batch 800] loss: 0.301112  (51200/60000)
  [Batch 900] loss: 0.291500  (57600/60000)
Best accuracy = 87.67% at epoch 9
```


Loss Curves (lr=0.1)
Train Loss
Test Loss
Loss
Epoch
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
2
4
6
8
10

```
Learning rate: 0.01

Epoch 1/10
  [Batch 000] loss: 2.314329  (0/60000)
  [Batch 100] loss: 2.166001  (6400/60000)
  [Batch 200] loss: 1.809368  (12800/60000)
  [Batch 300] loss: 1.501394  (19200/60000)
  [Batch 400] loss: 1.130784  (25600/60000)
  [Batch 500] loss: 1.042473  (32000/60000)
  [Batch 600] loss: 0.998837  (38400/60000)
  [Batch 700] loss: 0.864243  (44800/60000)
  [Batch 800] loss: 0.868004  (51200/60000)
  [Batch 900] loss: 0.807506  (57600/60000)
Epoch 2/10
  [Batch 000] loss: 0.795473  (0/60000)
  [Batch 100] loss: 0.843838  (6400/60000)
  [Batch 200] loss: 0.589319  (12800/60000)
  [Batch 300] loss: 0.784176  (19200/60000)
  [Batch 400] loss: 0.661015  (25600/60000)
  [Batch 500] loss: 0.642479  (32000/60000)
  [Batch 600] loss: 0.719374  (38400/60000)
  [Batch 700] loss: 0.685976  (44800/60000)
  [Batch 800] loss: 0.695574  (51200/60000)
  [Batch 900] loss: 0.633607  (57600/60000)
Epoch 3/10
  [Batch 000] loss: 0.562946  (0/60000)
  [Batch 100] loss: 0.647219  (6400/60000)
  [Batch 200] loss: 0.441234  (12800/60000)
  [Batch 300] loss: 0.664088  (19200/60000)
  [Batch 400] loss: 0.582558  (25600/60000)
  [Batch 500] loss: 0.561089  (32000/60000)
  [Batch 600] loss: 0.603645  (38400/60000)
  [Batch 700] loss: 0.641111  (44800/60000)
  [Batch 800] loss: 0.660748  (51200/60000)
  [Batch 900] loss: 0.541132  (57600/60000)
Epoch 4/10
  [Batch 000] loss: 0.468428  (0/60000)
  [Batch 100] loss: 0.563518  (6400/60000)
  [Batch 200] loss: 0.381162  (12800/60000)
  [Batch 300] loss: 0.596225  (19200/60000)
  [Batch 400] loss: 0.530208  (25600/60000)
  [Batch 500] loss: 0.519761  (32000/60000)
  [Batch 600] loss: 0.546248  (38400/60000)
  [Batch 700] loss: 0.638229  (44800/60000)
  [Batch 800] loss: 0.644423  (51200/60000)
  [Batch 900] loss: 0.479617  (57600/60000)
Epoch 5/10
  [Batch 000] loss: 0.412060  (0/60000)
  [Batch 100] loss: 0.521560  (6400/60000)
  [Batch 200] loss: 0.349677  (12800/60000)
  [Batch 300] loss: 0.552302  (19200/60000)
  [Batch 400] loss: 0.487746  (25600/60000)
  [Batch 500] loss: 0.489107  (32000/60000)
  [Batch 600] loss: 0.510166  (38400/60000)
  [Batch 700] loss: 0.632951  (44800/60000)
  [Batch 800] loss: 0.621865  (51200/60000)
  [Batch 900] loss: 0.443293  (57600/60000)
Epoch 6/10
  [Batch 000] loss: 0.370650  (0/60000)
  [Batch 100] loss: 0.495788  (6400/60000)
  [Batch 200] loss: 0.328920  (12800/60000)
  [Batch 300] loss: 0.522104  (19200/60000)
  [Batch 400] loss: 0.459794  (25600/60000)
  [Batch 500] loss: 0.466965  (32000/60000)
  [Batch 600] loss: 0.484508  (38400/60000)
  [Batch 700] loss: 0.619549  (44800/60000)
  [Batch 800] loss: 0.599258  (51200/60000)
  [Batch 900] loss: 0.423255  (57600/60000)
Epoch 7/10
  [Batch 000] loss: 0.339036  (0/60000)
  [Batch 100] loss: 0.475743  (6400/60000)
  [Batch 200] loss: 0.311331  (12800/60000)
  [Batch 300] loss: 0.500448  (19200/60000)
  [Batch 400] loss: 0.438610  (25600/60000)
  [Batch 500] loss: 0.451461  (32000/60000)
  [Batch 600] loss: 0.464538  (38400/60000)
  [Batch 700] loss: 0.604142  (44800/60000)
  [Batch 800] loss: 0.578478  (51200/60000)
  [Batch 900] loss: 0.411515  (57600/60000)
Epoch 8/10
  [Batch 000] loss: 0.315521  (0/60000)
  [Batch 100] loss: 0.460181  (6400/60000)
  [Batch 200] loss: 0.297932  (12800/60000)
  [Batch 300] loss: 0.484918  (19200/60000)
  [Batch 400] loss: 0.421314  (25600/60000)
  [Batch 500] loss: 0.439312  (32000/60000)
  [Batch 600] loss: 0.449092  (38400/60000)
  [Batch 700] loss: 0.590426  (44800/60000)
  [Batch 800] loss: 0.562349  (51200/60000)
  [Batch 900] loss: 0.404418  (57600/60000)
Epoch 9/10
  [Batch 000] loss: 0.298808  (0/60000)
  [Batch 100] loss: 0.445493  (6400/60000)
  [Batch 200] loss: 0.287091  (12800/60000)
  [Batch 300] loss: 0.471347  (19200/60000)
  [Batch 400] loss: 0.406590  (25600/60000)
  [Batch 500] loss: 0.430698  (32000/60000)
  [Batch 600] loss: 0.436691  (38400/60000)
  [Batch 700] loss: 0.577311  (44800/60000)
  [Batch 800] loss: 0.548569  (51200/60000)
  [Batch 900] loss: 0.398853  (57600/60000)
Epoch 10/10
  [Batch 000] loss: 0.286025  (0/60000)
  [Batch 100] loss: 0.432210  (6400/60000)
  [Batch 200] loss: 0.277919  (12800/60000)
  [Batch 300] loss: 0.459433  (19200/60000)
  [Batch 400] loss: 0.392912  (25600/60000)
  [Batch 500] loss: 0.423137  (32000/60000)
  [Batch 600] loss: 0.425565  (38400/60000)
  [Batch 700] loss: 0.565257  (44800/60000)
  [Batch 800] loss: 0.535418  (51200/60000)
  [Batch 900] loss: 0.394587  (57600/60000)
Best accuracy = 83.56% at epoch 10
```


Loss Curves (lr=0.01)
Train Loss
Test Loss
Loss
1.2
1.0
0.8
0.6
0.4
Epoch
2
4
6
8
10

```
Learning rate: 0.001

Epoch 1/10
  [Batch 000] loss: 2.291207  (0/60000)
  [Batch 100] loss: 2.282664  (6400/60000)
  [Batch 200] loss: 2.261561  (12800/60000)
  [Batch 300] loss: 2.259733  (19200/60000)
  [Batch 400] loss: 2.240708  (25600/60000)
  [Batch 500] loss: 2.211135  (32000/60000)
  [Batch 600] loss: 2.221413  (38400/60000)
  [Batch 700] loss: 2.180183  (44800/60000)
  [Batch 800] loss: 2.168982  (51200/60000)
  [Batch 900] loss: 2.134594  (57600/60000)
Epoch 2/10
  [Batch 000] loss: 2.143305  (0/60000)
  [Batch 100] loss: 2.140811  (6400/60000)
  [Batch 200] loss: 2.076955  (12800/60000)
  [Batch 300] loss: 2.097295  (19200/60000)
  [Batch 400] loss: 2.043799  (25600/60000)
  [Batch 500] loss: 1.981938  (32000/60000)
  [Batch 600] loss: 2.014280  (38400/60000)
  [Batch 700] loss: 1.924833  (44800/60000)
  [Batch 800] loss: 1.920683  (51200/60000)
  [Batch 900] loss: 1.848292  (57600/60000)
Epoch 3/10
  [Batch 000] loss: 1.886089  (0/60000)
  [Batch 100] loss: 1.860739  (6400/60000)
  [Batch 200] loss: 1.736607  (12800/60000)
  [Batch 300] loss: 1.779799  (19200/60000)
  [Batch 400] loss: 1.668908  (25600/60000)
  [Batch 500] loss: 1.625564  (32000/60000)
  [Batch 600] loss: 1.653022  (38400/60000)
  [Batch 700] loss: 1.546031  (44800/60000)
  [Batch 800] loss: 1.566427  (51200/60000)
  [Batch 900] loss: 1.464037  (57600/60000)
Epoch 4/10
  [Batch 000] loss: 1.550664  (0/60000)
  [Batch 100] loss: 1.519670  (6400/60000)
  [Batch 200] loss: 1.367525  (12800/60000)
  [Batch 300] loss: 1.444642  (19200/60000)
  [Batch 400] loss: 1.326732  (25600/60000)
  [Batch 500] loss: 1.323069  (32000/60000)
  [Batch 600] loss: 1.349093  (38400/60000)
  [Batch 700] loss: 1.262155  (44800/60000)
  [Batch 800] loss: 1.298078  (51200/60000)
  [Batch 900] loss: 1.205571  (57600/60000)
Epoch 5/10
  [Batch 000] loss: 1.301303  (0/60000)
  [Batch 100] loss: 1.285281  (6400/60000)
  [Batch 200] loss: 1.120595  (12800/60000)
  [Batch 300] loss: 1.231754  (19200/60000)
  [Batch 400] loss: 1.110287  (25600/60000)
  [Batch 500] loss: 1.128477  (32000/60000)
  [Batch 600] loss: 1.164104  (38400/60000)
  [Batch 700] loss: 1.088291  (44800/60000)
  [Batch 800] loss: 1.130615  (51200/60000)
  [Batch 900] loss: 1.052869  (57600/60000)
Epoch 6/10
  [Batch 000] loss: 1.135910  (0/60000)
  [Batch 100] loss: 1.137760  (6400/60000)
  [Batch 200] loss: 0.958311  (12800/60000)
  [Batch 300] loss: 1.097929  (19200/60000)
  [Batch 400] loss: 0.978656  (25600/60000)
  [Batch 500] loss: 0.997987  (32000/60000)
  [Batch 600] loss: 1.049255  (38400/60000)
  [Batch 700] loss: 0.978399  (44800/60000)
  [Batch 800] loss: 1.020922  (51200/60000)
  [Batch 900] loss: 0.955989  (57600/60000)
Epoch 7/10
  [Batch 000] loss: 1.020586  (0/60000)
  [Batch 100] loss: 1.041784  (6400/60000)
  [Batch 200] loss: 0.846889  (12800/60000)
  [Batch 300] loss: 1.008241  (19200/60000)
  [Batch 400] loss: 0.896396  (25600/60000)
  [Batch 500] loss: 0.906205  (32000/60000)
  [Batch 600] loss: 0.973894  (38400/60000)
  [Batch 700] loss: 0.907450  (44800/60000)
  [Batch 800] loss: 0.944837  (51200/60000)
  [Batch 900] loss: 0.890592  (57600/60000)
Epoch 8/10
  [Batch 000] loss: 0.935713  (0/60000)
  [Batch 100] loss: 0.975010  (6400/60000)
  [Batch 200] loss: 0.767369  (12800/60000)
  [Batch 300] loss: 0.944887  (19200/60000)
  [Batch 400] loss: 0.841678  (25600/60000)
  [Batch 500] loss: 0.839613  (32000/60000)
  [Batch 600] loss: 0.920582  (38400/60000)
  [Batch 700] loss: 0.860503  (44800/60000)
  [Batch 800] loss: 0.889888  (51200/60000)
  [Batch 900] loss: 0.843562  (57600/60000)
Epoch 9/10
  [Batch 000] loss: 0.870365  (0/60000)
  [Batch 100] loss: 0.925107  (6400/60000)
  [Batch 200] loss: 0.708082  (12800/60000)
  [Batch 300] loss: 0.898015  (19200/60000)
  [Batch 400] loss: 0.802634  (25600/60000)
  [Batch 500] loss: 0.789818  (32000/60000)
  [Batch 600] loss: 0.880325  (38400/60000)
  [Batch 700] loss: 0.827971  (44800/60000)
  [Batch 800] loss: 0.848740  (51200/60000)
  [Batch 900] loss: 0.807776  (57600/60000)
Epoch 10/10
  [Batch 000] loss: 0.818067  (0/60000)
  [Batch 100] loss: 0.885113  (6400/60000)
  [Batch 200] loss: 0.662254  (12800/60000)
  [Batch 300] loss: 0.861844  (19200/60000)
  [Batch 400] loss: 0.772732  (25600/60000)
  [Batch 500] loss: 0.751643  (32000/60000)
  [Batch 600] loss: 0.847747  (38400/60000)
  [Batch 700] loss: 0.803974  (44800/60000)
  [Batch 800] loss: 0.816698  (51200/60000)
  [Batch 900] loss: 0.778889  (57600/60000)
Best accuracy = 71.35% at epoch 10
```



```
Final test accuracies:
  LR=1       : 19.97%
  LR=0.1     : 87.53%
  LR=0.01    : 83.56%
  LR=0.001   : 71.35%
```

Q1.3

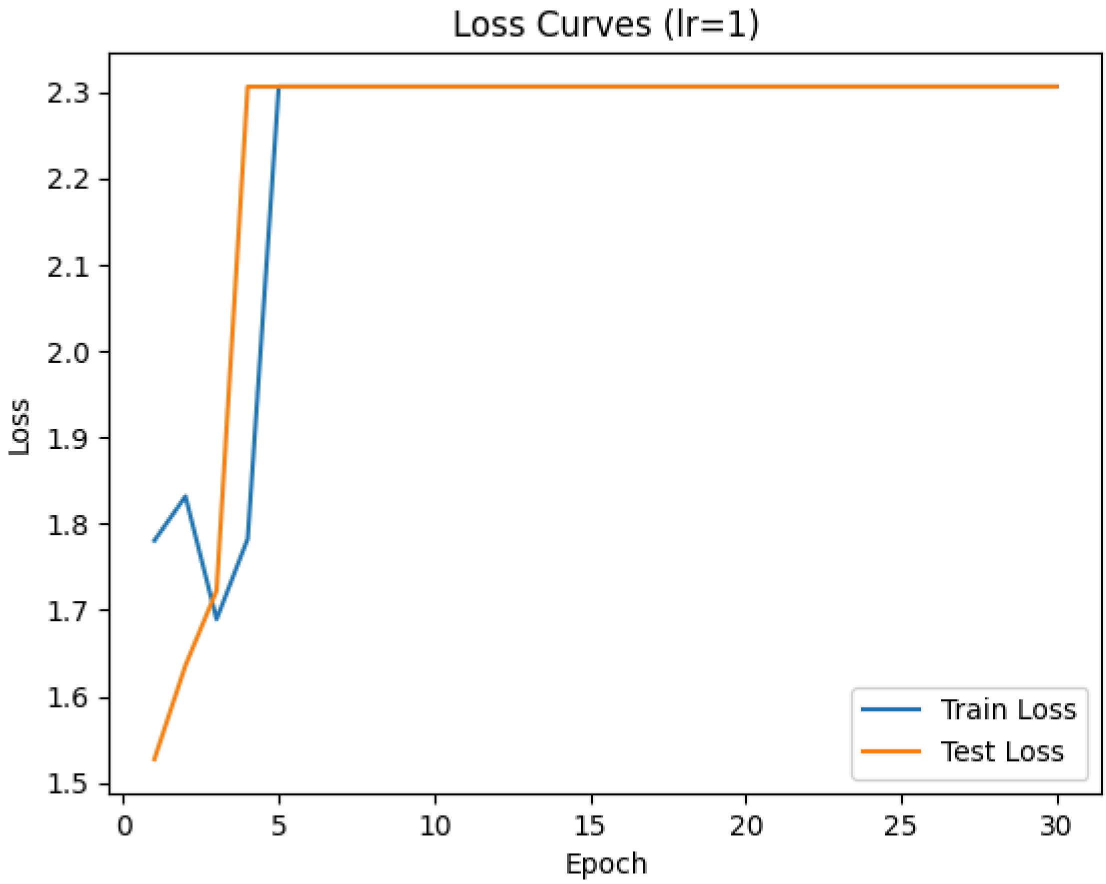
In [5]:

```
# Parameters
loss_fn = nn.CrossEntropyLoss()
learning_rate = 1
epochs = 30

# Experiment for learning rate = 1
print(f"Learning rate: {learning_rate} \n")
model = NeuralNetwork().to(device)
optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)

train_losses, test_losses, test_accs = [], [], []
for epoch in range(1, epochs + 1):
    print(f"Epoch {epoch}/{epochs}")
    train_loss = train(train_dataloader, model, loss_fn, optimizer)
    acc, test_loss = test(test_dataloader, model, loss_fn)

    train_losses.append(train_loss)
    test_losses.append(test_loss)
    test_accs.append(acc)

plot_train_test_loss(train_losses, test_losses, test_accs, learning_rate)
result = test_accs[-1]
```

```python
# Final result for learning rate = 1
print(f"Final test accuracy for LR={learning_rate}: {result*100:.2f}%")
```

```
Learning rate: 1

Epoch 1/30
  [Batch 000] loss: 2.301673  (0/60000)
  [Batch 100] loss: 2.180399  (6400/60000)
  [Batch 200] loss: 1.906144  (12800/60000)
  [Batch 300] loss: 1.751939  (19200/60000)
  [Batch 400] loss: 1.592444  (25600/60000)
  [Batch 500] loss: 1.535240  (32000/60000)
  [Batch 600] loss: 1.259818  (38400/60000)
  [Batch 700] loss: 2.105487  (44800/60000)
  [Batch 800] loss: 1.609622  (51200/60000)
  [Batch 900] loss: 1.257706  (57600/60000)
Epoch 2/30
  [Batch 000] loss: 1.567218  (0/60000)
  [Batch 100] loss: 1.402401  (6400/60000)
  [Batch 200] loss: 1.450479  (12800/60000)
  [Batch 300] loss: 1.292953  (19200/60000)
  [Batch 400] loss: 1.169132  (25600/60000)
  [Batch 500] loss: 1.325012  (32000/60000)
  [Batch 600] loss: 1.571929  (38400/60000)
  [Batch 700] loss: 1.448727  (44800/60000)
  [Batch 800] loss: 1.522715  (51200/60000)
  [Batch 900] loss: 1.710152  (57600/60000)
Epoch 3/30
  [Batch 000] loss: 1.495112  (0/60000)
  [Batch 100] loss: 1.435577  (6400/60000)
  [Batch 200] loss: 1.591795  (12800/60000)
  [Batch 300] loss: 1.641858  (19200/60000)
  [Batch 400] loss: 2.756652  (25600/60000)
  [Batch 500] loss: 1.721997  (32000/60000)
  [Batch 600] loss: 1.674409  (38400/60000)
  [Batch 700] loss: 1.685281  (44800/60000)
  [Batch 800] loss: 1.688752  (51200/60000)
  [Batch 900] loss: 1.685812  (57600/60000)
Epoch 4/30
  [Batch 000] loss: 1.775116  (0/60000)
  [Batch 100] loss: 1.352238  (6400/60000)
  [Batch 200] loss: 1.534785  (12800/60000)
  [Batch 300] loss: 1.759572  (19200/60000)
  [Batch 400] loss: 1.536888  (25600/60000)
  [Batch 500] loss: 1.601738  (32000/60000)
  [Batch 600] loss: 1.597994  (38400/60000)
  [Batch 700] loss: 1.702103  (44800/60000)
  [Batch 800] loss: 1.729876  (51200/60000)
  [Batch 900] loss: 2.328488  (57600/60000)
Epoch 5/30
  [Batch 000] loss: 2.313051  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 6/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 7/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 8/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 9/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 10/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 11/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 12/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 13/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 14/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 15/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 16/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 17/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 18/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 19/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 20/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 21/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 22/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 23/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 24/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 25/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 26/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 27/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 28/30
```

```
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 29/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Epoch 30/30
  [Batch 000] loss: 2.313046  (0/60000)
  [Batch 100] loss: 2.306795  (6400/60000)
  [Batch 200] loss: 2.312776  (12800/60000)
  [Batch 300] loss: 2.304621  (19200/60000)
  [Batch 400] loss: 2.281378  (25600/60000)
  [Batch 500] loss: 2.304950  (32000/60000)
  [Batch 600] loss: 2.305945  (38400/60000)
  [Batch 700] loss: 2.288610  (44800/60000)
  [Batch 800] loss: 2.300421  (51200/60000)
  [Batch 900] loss: 2.330204  (57600/60000)
Best accuracy = 35.80% at epoch 1
```



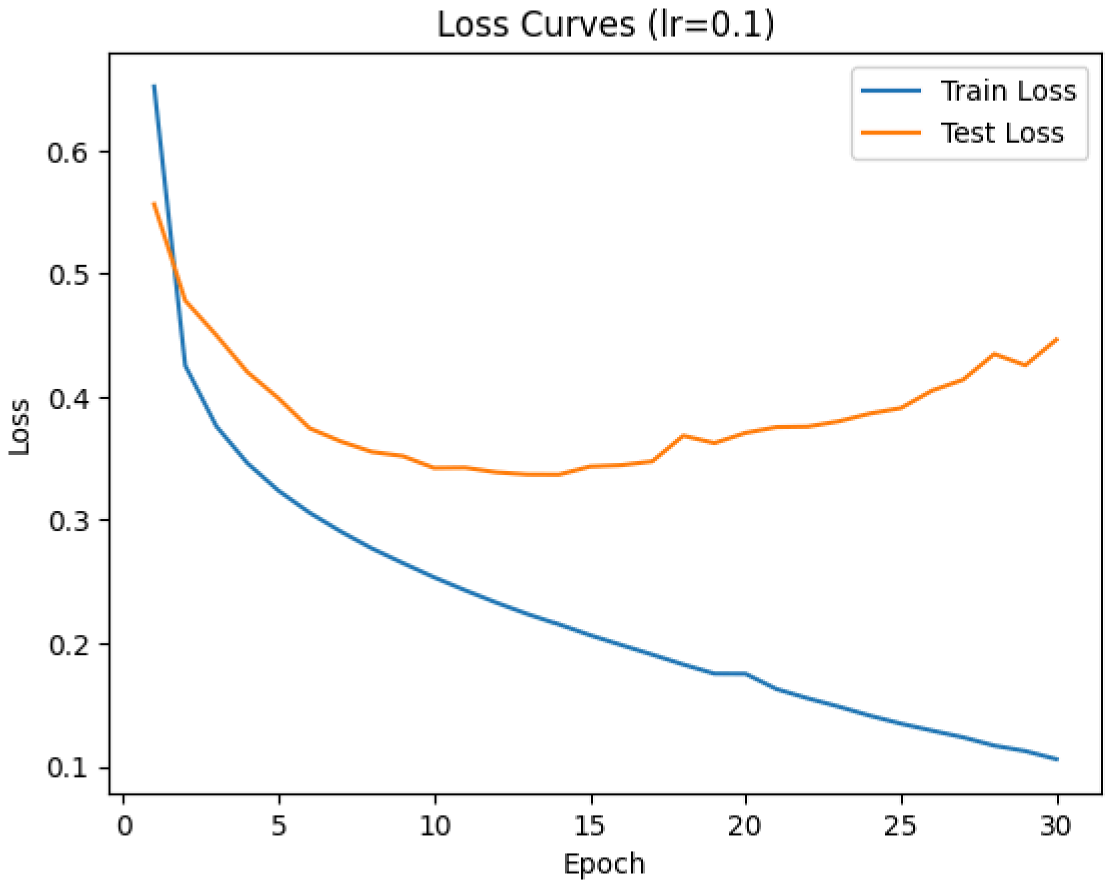
```
Final test accuracy for LR=1: 10.00%
```

In [6]:

```python
# Parameters
loss_fn = nn.CrossEntropyLoss()
learning_rate = 1e-1
epochs = 30

# Experiment for learning rate = 1
print(f"Learning rate: {learning_rate} \n")
model = NeuralNetwork().to(device)
optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)

train_losses, test_losses, test_accs = [], [], []
for epoch in range(1, epochs + 1):
    print(f"Epoch {epoch}/{epochs}")
    train_loss = train(train_dataloader, model, loss_fn, optimizer)
    acc, test_loss = test(test_dataloader, model, loss_fn)

    train_losses.append(train_loss)
    test_losses.append(test_loss)
    test_accs.append(acc)

plot_train_test_loss(train_losses, test_losses, test_accs, learning_rate)
result = test_accs[-1]

# Final result for learning rate = 1
print(f"Final test accuracy for LR={learning_rate}: {result*100:.2f}%")
```

```
Learning rate: 0.1

Epoch 1/30
  [Batch 000] loss: 2.304804  (0/60000)
  [Batch 100] loss: 0.895236  (6400/60000)
  [Batch 200] loss: 0.573996  (12800/60000)
  [Batch 300] loss: 0.709345  (19200/60000)
  [Batch 400] loss: 0.597106  (25600/60000)
  [Batch 500] loss: 0.516124  (32000/60000)
  [Batch 600] loss: 0.545407  (38400/60000)
  [Batch 700] loss: 0.605605  (44800/60000)
  [Batch 800] loss: 0.616153  (51200/60000)
  [Batch 900] loss: 0.457095  (57600/60000)
Epoch 2/30
  [Batch 000] loss: 0.436648  (0/60000)
  [Batch 100] loss: 0.438896  (6400/60000)
  [Batch 200] loss: 0.366696  (12800/60000)
  [Batch 300] loss: 0.441557  (19200/60000)
  [Batch 400] loss: 0.427104  (25600/60000)
  [Batch 500] loss: 0.449212  (32000/60000)
  [Batch 600] loss: 0.418601  (38400/60000)
  [Batch 700] loss: 0.506757  (44800/60000)
  [Batch 800] loss: 0.512055  (51200/60000)
  [Batch 900] loss: 0.420511  (57600/60000)
Epoch 3/30
  [Batch 000] loss: 0.336738  (0/60000)
  [Batch 100] loss: 0.360954  (6400/60000)
  [Batch 200] loss: 0.310589  (12800/60000)
  [Batch 300] loss: 0.371099  (19200/60000)
  [Batch 400] loss: 0.357700  (25600/60000)
  [Batch 500] loss: 0.415752  (32000/60000)
  [Batch 600] loss: 0.357215  (38400/60000)
  [Batch 700] loss: 0.457122  (44800/60000)
  [Batch 800] loss: 0.458615  (51200/60000)
  [Batch 900] loss: 0.396482  (57600/60000)
Epoch 4/30
  [Batch 000] loss: 0.297435  (0/60000)
  [Batch 100] loss: 0.320123  (6400/60000)
  [Batch 200] loss: 0.253365  (12800/60000)
  [Batch 300] loss: 0.323071  (19200/60000)
  [Batch 400] loss: 0.321577  (25600/60000)
  [Batch 500] loss: 0.398656  (32000/60000)
  [Batch 600] loss: 0.321518  (38400/60000)
  [Batch 700] loss: 0.421656  (44800/60000)
  [Batch 800] loss: 0.418483  (51200/60000)
  [Batch 900] loss: 0.392937  (57600/60000)
Epoch 5/30
  [Batch 000] loss: 0.267860  (0/60000)
  [Batch 100] loss: 0.300204  (6400/60000)
  [Batch 200] loss: 0.219344  (12800/60000)
  [Batch 300] loss: 0.291915  (19200/60000)
  [Batch 400] loss: 0.315374  (25600/60000)
  [Batch 500] loss: 0.387783  (32000/60000)
  [Batch 600] loss: 0.303520  (38400/60000)
  [Batch 700] loss: 0.384178  (44800/60000)
  [Batch 800] loss: 0.383739  (51200/60000)
  [Batch 900] loss: 0.378493  (57600/60000)
Epoch 6/30
  [Batch 000] loss: 0.235974  (0/60000)
  [Batch 100] loss: 0.286809  (6400/60000)
  [Batch 200] loss: 0.201473  (12800/60000)
  [Batch 300] loss: 0.269018  (19200/60000)
  [Batch 400] loss: 0.313954  (25600/60000)
  [Batch 500] loss: 0.373747  (32000/60000)
  [Batch 600] loss: 0.279501  (38400/60000)
  [Batch 700] loss: 0.356502  (44800/60000)
  [Batch 800] loss: 0.366660  (51200/60000)
  [Batch 900] loss: 0.366931  (57600/60000)
Epoch 7/30
  [Batch 000] loss: 0.218722  (0/60000)
  [Batch 100] loss: 0.269413  (6400/60000)
  [Batch 200] loss: 0.188366  (12800/60000)
  [Batch 300] loss: 0.252746  (19200/60000)
  [Batch 400] loss: 0.303089  (25600/60000)
  [Batch 500] loss: 0.367052  (32000/60000)
  [Batch 600] loss: 0.260226  (38400/60000)
  [Batch 700] loss: 0.328331  (44800/60000)
  [Batch 800] loss: 0.348781  (51200/60000)
  [Batch 900] loss: 0.351994  (57600/60000)
Epoch 8/30
  [Batch 000] loss: 0.199769  (0/60000)
  [Batch 100] loss: 0.261744  (6400/60000)
  [Batch 200] loss: 0.172947  (12800/60000)
  [Batch 300] loss: 0.236014  (19200/60000)
  [Batch 400] loss: 0.309375  (25600/60000)
  [Batch 500] loss: 0.362288  (32000/60000)
  [Batch 600] loss: 0.247291  (38400/60000)
  [Batch 700] loss: 0.317672  (44800/60000)
  [Batch 800] loss: 0.322527  (51200/60000)
  [Batch 900] loss: 0.340993  (57600/60000)
Epoch 9/30
  [Batch 000] loss: 0.190596  (0/60000)
  [Batch 100] loss: 0.247718  (6400/60000)
  [Batch 200] loss: 0.167679  (12800/60000)
  [Batch 300] loss: 0.230954  (19200/60000)
  [Batch 400] loss: 0.305756  (25600/60000)
  [Batch 500] loss: 0.351744  (32000/60000)
  [Batch 600] loss: 0.240909  (38400/60000)
  [Batch 700] loss: 0.305458  (44800/60000)
  [Batch 800] loss: 0.301378  (51200/60000)
  [Batch 900] loss: 0.326493  (57600/60000)
Epoch 10/30
  [Batch 000] loss: 0.181582  (0/60000)
  [Batch 100] loss: 0.242863  (6400/60000)
  [Batch 200] loss: 0.160285  (12800/60000)
  [Batch 300] loss: 0.222758  (19200/60000)
  [Batch 400] loss: 0.293809  (25600/60000)
  [Batch 500] loss: 0.342762  (32000/60000)
  [Batch 600] loss: 0.220564  (38400/60000)
  [Batch 700] loss: 0.280702  (44800/60000)
  [Batch 800] loss: 0.290114  (51200/60000)
  [Batch 900] loss: 0.319244  (57600/60000)
Epoch 11/30
  [Batch 000] loss: 0.183301  (0/60000)
  [Batch 100] loss: 0.217512  (6400/60000)
  [Batch 200] loss: 0.152263  (12800/60000)
  [Batch 300] loss: 0.211685  (19200/60000)
  [Batch 400] loss: 0.285622  (25600/60000)
  [Batch 500] loss: 0.325996  (32000/60000)
  [Batch 600] loss: 0.215599  (38400/60000)
  [Batch 700] loss: 0.279719  (44800/60000)
  [Batch 800] loss: 0.271976  (51200/60000)
  [Batch 900] loss: 0.299343  (57600/60000)
Epoch 12/30
  [Batch 000] loss: 0.187060  (0/60000)
  [Batch 100] loss: 0.201003  (6400/60000)
  [Batch 200] loss: 0.154254  (12800/60000)
  [Batch 300] loss: 0.203234  (19200/60000)
  [Batch 400] loss: 0.281978  (25600/60000)
  [Batch 500] loss: 0.320328  (32000/60000)
  [Batch 600] loss: 0.213640  (38400/60000)
  [Batch 700] loss: 0.273048  (44800/60000)
  [Batch 800] loss: 0.267146  (51200/60000)
  [Batch 900] loss: 0.286171  (57600/60000)
Epoch 13/30
  [Batch 000] loss: 0.180294  (0/60000)
  [Batch 100] loss: 0.195170  (6400/60000)
  [Batch 200] loss: 0.148643  (12800/60000)
  [Batch 300] loss: 0.193625  (19200/60000)
  [Batch 400] loss: 0.278037  (25600/60000)
  [Batch 500] loss: 0.298819  (32000/60000)
  [Batch 600] loss: 0.211892  (38400/60000)
  [Batch 700] loss: 0.270168  (44800/60000)
  [Batch 800] loss: 0.249636  (51200/60000)
  [Batch 900] loss: 0.256294  (57600/60000)
Epoch 14/30
  [Batch 000] loss: 0.170697  (0/60000)
  [Batch 100] loss: 0.177168  (6400/60000)
  [Batch 200] loss: 0.139469  (12800/60000)
  [Batch 300] loss: 0.191078  (19200/60000)
  [Batch 400] loss: 0.258844  (25600/60000)
  [Batch 500] loss: 0.278155  (32000/60000)
  [Batch 600] loss: 0.200971  (38400/60000)
  [Batch 700] loss: 0.255171  (44800/60000)
  [Batch 800] loss: 0.235082  (51200/60000)
  [Batch 900] loss: 0.241967  (57600/60000)
Epoch 15/30
  [Batch 000] loss: 0.160994  (0/60000)
  [Batch 100] loss: 0.169010  (6400/60000)
  [Batch 200] loss: 0.134192  (12800/60000)
  [Batch 300] loss: 0.187007  (19200/60000)
  [Batch 400] loss: 0.256512  (25600/60000)
  [Batch 500] loss: 0.254175  (32000/60000)
  [Batch 600] loss: 0.209498  (38400/60000)
  [Batch 700] loss: 0.254823  (44800/60000)
  [Batch 800] loss: 0.226103  (51200/60000)
  [Batch 900] loss: 0.240090  (57600/60000)
Epoch 16/30
  [Batch 000] loss: 0.162729  (0/60000)
  [Batch 100] loss: 0.152655  (6400/60000)
  [Batch 200] loss: 0.123352  (12800/60000)
  [Batch 300] loss: 0.175907  (19200/60000)
  [Batch 400] loss: 0.244933  (25600/60000)
  [Batch 500] loss: 0.248179  (32000/60000)
  [Batch 600] loss: 0.189604  (38400/60000)
  [Batch 700] loss: 0.232708  (44800/60000)
  [Batch 800] loss: 0.224589  (51200/60000)
  [Batch 900] loss: 0.232238  (57600/60000)
Epoch 17/30
  [Batch 000] loss: 0.158021  (0/60000)
  [Batch 100] loss: 0.152275  (6400/60000)
  [Batch 200] loss: 0.113993  (12800/60000)
  [Batch 300] loss: 0.157422  (19200/60000)
  [Batch 400] loss: 0.242315  (25600/60000)
  [Batch 500] loss: 0.246735  (32000/60000)
  [Batch 600] loss: 0.186291  (38400/60000)
  [Batch 700] loss: 0.218757  (44800/60000)
  [Batch 800] loss: 0.208648  (51200/60000)
  [Batch 900] loss: 0.225521  (57600/60000)
Epoch 18/30
  [Batch 000] loss: 0.157887  (0/60000)
  [Batch 100] loss: 0.139320  (6400/60000)
  [Batch 200] loss: 0.109191  (12800/60000)
  [Batch 300] loss: 0.156173  (19200/60000)
  [Batch 400] loss: 0.248191  (25600/60000)
  [Batch 500] loss: 0.221902  (32000/60000)
  [Batch 600] loss: 0.183075  (38400/60000)
  [Batch 700] loss: 0.207792  (44800/60000)
  [Batch 800] loss: 0.195380  (51200/60000)
  [Batch 900] loss: 0.205618  (57600/60000)
Epoch 19/30
  [Batch 000] loss: 0.171632  (0/60000)
  [Batch 100] loss: 0.139819  (6400/60000)
  [Batch 200] loss: 0.105827  (12800/60000)
  [Batch 300] loss: 0.150030  (19200/60000)
  [Batch 400] loss: 0.230271  (25600/60000)
  [Batch 500] loss: 0.211308  (32000/60000)
  [Batch 600] loss: 0.172763  (38400/60000)
  [Batch 700] loss: 0.195058  (44800/60000)
  [Batch 800] loss: 0.183614  (51200/60000)
  [Batch 900] loss: 0.202671  (57600/60000)
Epoch 20/30
  [Batch 000] loss: 0.147657  (0/60000)
  [Batch 100] loss: 0.133308  (6400/60000)
  [Batch 200] loss: 0.096258  (12800/60000)
  [Batch 300] loss: 0.139302  (19200/60000)
  [Batch 400] loss: 0.254815  (25600/60000)
  [Batch 500] loss: 0.211551  (32000/60000)
  [Batch 600] loss: 0.172787  (38400/60000)
  [Batch 700] loss: 0.197464  (44800/60000)
  [Batch 800] loss: 0.195931  (51200/60000)
  [Batch 900] loss: 0.192947  (57600/60000)
Epoch 21/30
  [Batch 000] loss: 0.147739  (0/60000)
  [Batch 100] loss: 0.122773  (6400/60000)
  [Batch 200] loss: 0.096658  (12800/60000)
  [Batch 300] loss: 0.141154  (19200/60000)
  [Batch 400] loss: 0.212491  (25600/60000)
  [Batch 500] loss: 0.198326  (32000/60000)
  [Batch 600] loss: 0.176873  (38400/60000)
  [Batch 700] loss: 0.185472  (44800/60000)
  [Batch 800] loss: 0.168675  (51200/60000)
  [Batch 900] loss: 0.162077  (57600/60000)
Epoch 22/30
  [Batch 000] loss: 0.147666  (0/60000)
  [Batch 100] loss: 0.117704  (6400/60000)
  [Batch 200] loss: 0.090335  (12800/60000)
  [Batch 300] loss: 0.140757  (19200/60000)
  [Batch 400] loss: 0.238429  (25600/60000)
  [Batch 500] loss: 0.176119  (32000/60000)
  [Batch 600] loss: 0.162346  (38400/60000)
  [Batch 700] loss: 0.171297  (44800/60000)
  [Batch 800] loss: 0.166761  (51200/60000)
  [Batch 900] loss: 0.159775  (57600/60000)
Epoch 23/30
  [Batch 000] loss: 0.127852  (0/60000)
  [Batch 100] loss: 0.109497  (6400/60000)
  [Batch 200] loss: 0.084232  (12800/60000)
  [Batch 300] loss: 0.123048  (19200/60000)
  [Batch 400] loss: 0.207736  (25600/60000)
  [Batch 500] loss: 0.170687  (32000/60000)
  [Batch 600] loss: 0.160453  (38400/60000)
  [Batch 700] loss: 0.166409  (44800/60000)
  [Batch 800] loss: 0.153274  (51200/60000)
  [Batch 900] loss: 0.155722  (57600/60000)
Epoch 24/30
  [Batch 000] loss: 0.139706  (0/60000)
  [Batch 100] loss: 0.094445  (6400/60000)
  [Batch 200] loss: 0.066577  (12800/60000)
  [Batch 300] loss: 0.103106  (19200/60000)
  [Batch 400] loss: 0.206607  (25600/60000)
  [Batch 500] loss: 0.173171  (32000/60000)
  [Batch 600] loss: 0.153869  (38400/60000)
  [Batch 700] loss: 0.164364  (44800/60000)
  [Batch 800] loss: 0.156813  (51200/60000)
  [Batch 900] loss: 0.199034  (57600/60000)
Epoch 25/30
  [Batch 000] loss: 0.124724  (0/60000)
  [Batch 100] loss: 0.097367  (6400/60000)
  [Batch 200] loss: 0.069158  (12800/60000)
  [Batch 300] loss: 0.092649  (19200/60000)
  [Batch 400] loss: 0.189718  (25600/60000)
  [Batch 500] loss: 0.168768  (32000/60000)
  [Batch 600] loss: 0.151006  (38400/60000)
  [Batch 700] loss: 0.144484  (44800/60000)
  [Batch 800] loss: 0.149385  (51200/60000)
  [Batch 900] loss: 0.154997  (57600/60000)
Epoch 26/30
  [Batch 000] loss: 0.126715  (0/60000)
  [Batch 100] loss: 0.091287  (6400/60000)
  [Batch 200] loss: 0.070703  (12800/60000)
  [Batch 300] loss: 0.111177  (19200/60000)
  [Batch 400] loss: 0.184350  (25600/60000)
  [Batch 500] loss: 0.146959  (32000/60000)
  [Batch 600] loss: 0.145940  (38400/60000)
  [Batch 700] loss: 0.145265  (44800/60000)
  [Batch 800] loss: 0.148012  (51200/60000)
  [Batch 900] loss: 0.175648  (57600/60000)
Epoch 27/30
  [Batch 000] loss: 0.144030  (0/60000)
  [Batch 100] loss: 0.095848  (6400/60000)
  [Batch 200] loss: 0.076548  (12800/60000)
  [Batch 300] loss: 0.089755  (19200/60000)
  [Batch 400] loss: 0.173557  (25600/60000)
  [Batch 500] loss: 0.158651  (32000/60000)
  [Batch 600] loss: 0.141098  (38400/60000)
  [Batch 700] loss: 0.131263  (44800/60000)
  [Batch 800] loss: 0.133289  (51200/60000)
  [Batch 900] loss: 0.139711  (57600/60000)
Epoch 28/30
```

```
  [Batch 000] loss: 0.117706  (0/60000)
  [Batch 100] loss: 0.078146  (6400/60000)
  [Batch 200] loss: 0.063647  (12800/60000)
  [Batch 300] loss: 0.092889  (19200/60000)
  [Batch 400] loss: 0.181643  (25600/60000)
  [Batch 500] loss: 0.138865  (32000/60000)
  [Batch 600] loss: 0.135319  (38400/60000)
  [Batch 700] loss: 0.129540  (44800/60000)
  [Batch 800] loss: 0.134562  (51200/60000)
  [Batch 900] loss: 0.111496  (57600/60000)
Epoch 29/30
  [Batch 000] loss: 0.106764  (0/60000)
  [Batch 100] loss: 0.079182  (6400/60000)
  [Batch 200] loss: 0.086343  (12800/60000)
  [Batch 300] loss: 0.082987  (19200/60000)
  [Batch 400] loss: 0.212712  (25600/60000)
  [Batch 500] loss: 0.144837  (32000/60000)
  [Batch 600] loss: 0.119036  (38400/60000)
  [Batch 700] loss: 0.104292  (44800/60000)
  [Batch 800] loss: 0.113989  (51200/60000)
  [Batch 900] loss: 0.164794  (57600/60000)
Epoch 30/30
  [Batch 000] loss: 0.105254  (0/60000)
  [Batch 100] loss: 0.074792  (6400/60000)
  [Batch 200] loss: 0.085564  (12800/60000)
  [Batch 300] loss: 0.080546  (19200/60000)
  [Batch 400] loss: 0.156498  (25600/60000)
  [Batch 500] loss: 0.130137  (32000/60000)
  [Batch 600] loss: 0.101242  (38400/60000)
  [Batch 700] loss: 0.113958  (44800/60000)
  [Batch 800] loss: 0.142786  (51200/60000)
  [Batch 900] loss: 0.150328  (57600/60000)
Best accuracy = 88.07% at epoch 13
```



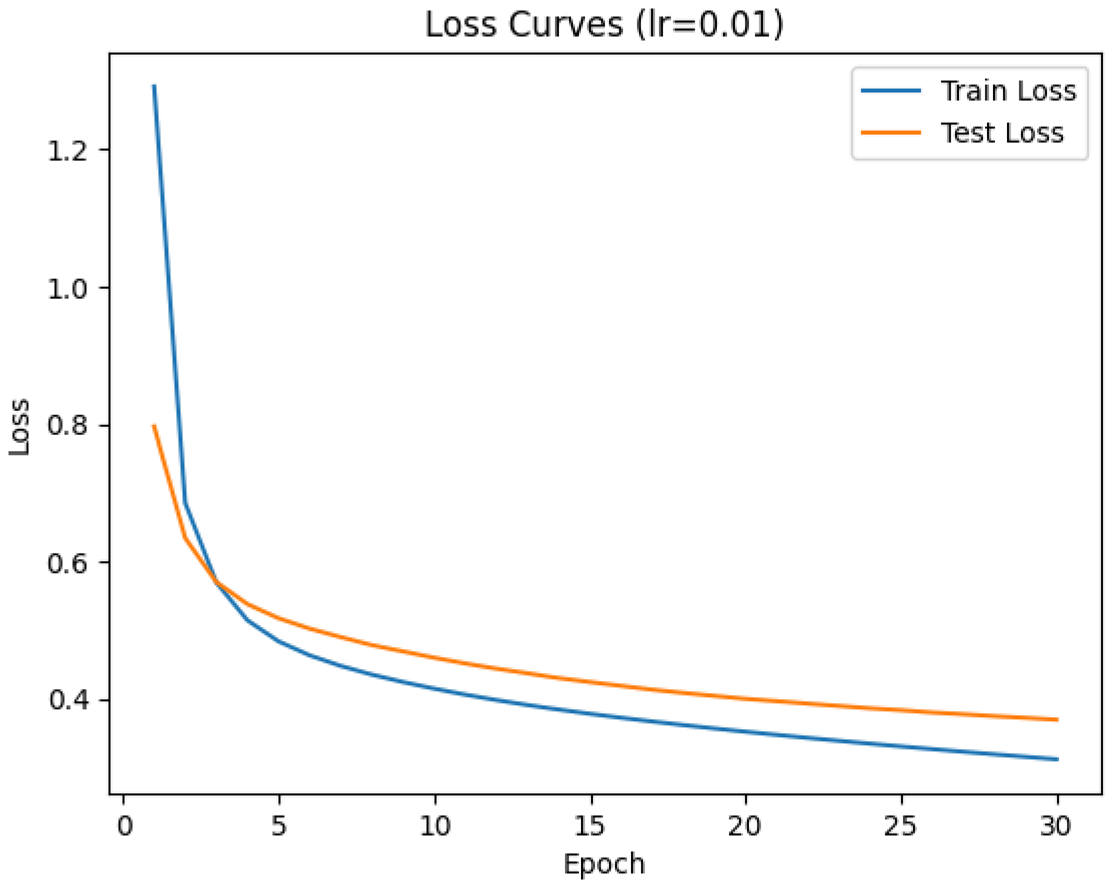
```
Final test accuracy for LR=0.1: 86.85%
```

In [7]:

```
# Parameters
loss_fn = nn.CrossEntropyLoss()
learning_rate = 1e-2
epochs = 30

# Experiment for learning rate = 1
print(f"Learning rate: {learning_rate} \n")
model = NeuralNetwork().to(device)
optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)

train_losses, test_losses, test_accs = [], [], []
for epoch in range(1, epochs + 1):
    print(f"Epoch {epoch}/{epochs}")
    train_loss = train(train_dataloader, model, loss_fn, optimizer)
    acc, test_loss = test(test_dataloader, model, loss_fn)

    train_losses.append(train_loss)
    test_losses.append(test_loss)
    test_accs.append(acc)

plot_train_test_loss(train_losses, test_losses, test_accs, learning_rate)
result = test_accs[-1]

# Final result for learning rate = 1
print(f"Final test accuracy for LR={learning_rate}: {result*100:.2f}%")
```

```
Learning rate: 0.01

Epoch 1/30
  [Batch 000] loss: 2.297597  (0/60000)
  [Batch 100] loss: 2.145239  (6400/60000)
  [Batch 200] loss: 1.779472  (12800/60000)
  [Batch 300] loss: 1.490282  (19200/60000)
  [Batch 400] loss: 1.140645  (25600/60000)
  [Batch 500] loss: 1.041433  (32000/60000)
  [Batch 600] loss: 1.005681  (38400/60000)
  [Batch 700] loss: 0.874601  (44800/60000)
  [Batch 800] loss: 0.868524  (51200/60000)
  [Batch 900] loss: 0.806097  (57600/60000)
Epoch 2/30
  [Batch 000] loss: 0.799003  (0/60000)
  [Batch 100] loss: 0.852552  (6400/60000)
  [Batch 200] loss: 0.588857  (12800/60000)
  [Batch 300] loss: 0.783328  (19200/60000)
  [Batch 400] loss: 0.665234  (25600/60000)
  [Batch 500] loss: 0.639718  (32000/60000)
  [Batch 600] loss: 0.719117  (38400/60000)
  [Batch 700] loss: 0.690038  (44800/60000)
  [Batch 800] loss: 0.708284  (51200/60000)
  [Batch 900] loss: 0.628871  (57600/60000)
Epoch 3/30
  [Batch 000] loss: 0.560225  (0/60000)
  [Batch 100] loss: 0.652057  (6400/60000)
  [Batch 200] loss: 0.436228  (12800/60000)
  [Batch 300] loss: 0.665195  (19200/60000)
  [Batch 400] loss: 0.574570  (25600/60000)
  [Batch 500] loss: 0.559656  (32000/60000)
  [Batch 600] loss: 0.593891  (38400/60000)
  [Batch 700] loss: 0.643736  (44800/60000)
  [Batch 800] loss: 0.672878  (51200/60000)
  [Batch 900] loss: 0.537938  (57600/60000)
Epoch 4/30
  [Batch 000] loss: 0.462047  (0/60000)
  [Batch 100] loss: 0.557796  (6400/60000)
  [Batch 200] loss: 0.375339  (12800/60000)
  [Batch 300] loss: 0.600955  (19200/60000)
  [Batch 400] loss: 0.521480  (25600/60000)
  [Batch 500] loss: 0.518245  (32000/60000)
  [Batch 600] loss: 0.537572  (38400/60000)
  [Batch 700] loss: 0.640210  (44800/60000)
  [Batch 800] loss: 0.652847  (51200/60000)
  [Batch 900] loss: 0.478346  (57600/60000)
Epoch 5/30
  [Batch 000] loss: 0.406357  (0/60000)
  [Batch 100] loss: 0.509682  (6400/60000)
  [Batch 200] loss: 0.341202  (12800/60000)
  [Batch 300] loss: 0.558573  (19200/60000)
  [Batch 400] loss: 0.482436  (25600/60000)
  [Batch 500] loss: 0.488985  (32000/60000)
  [Batch 600] loss: 0.506594  (38400/60000)
  [Batch 700] loss: 0.634504  (44800/60000)
  [Batch 800] loss: 0.629777  (51200/60000)
  [Batch 900] loss: 0.444871  (57600/60000)
Epoch 6/30
  [Batch 000] loss: 0.364580  (0/60000)
  [Batch 100] loss: 0.481469  (6400/60000)
  [Batch 200] loss: 0.318602  (12800/60000)
  [Batch 300] loss: 0.529349  (19200/60000)
  [Batch 400] loss: 0.454426  (25600/60000)
  [Batch 500] loss: 0.468523  (32000/60000)
  [Batch 600] loss: 0.483336  (38400/60000)
  [Batch 700] loss: 0.621878  (44800/60000)
  [Batch 800] loss: 0.607064  (51200/60000)
  [Batch 900] loss: 0.426916  (57600/60000)
Epoch 7/30
  [Batch 000] loss: 0.333105  (0/60000)
  [Batch 100] loss: 0.462092  (6400/60000)
  [Batch 200] loss: 0.301348  (12800/60000)
  [Batch 300] loss: 0.510007  (19200/60000)
  [Batch 400] loss: 0.432850  (25600/60000)
  [Batch 500] loss: 0.454344  (32000/60000)
  [Batch 600] loss: 0.464636  (38400/60000)
  [Batch 700] loss: 0.607035  (44800/60000)
  [Batch 800] loss: 0.585409  (51200/60000)
  [Batch 900] loss: 0.416604  (57600/60000)
Epoch 8/30
  [Batch 000] loss: 0.310190  (0/60000)
  [Batch 100] loss: 0.446139  (6400/60000)
  [Batch 200] loss: 0.287306  (12800/60000)
  [Batch 300] loss: 0.495181  (19200/60000)
  [Batch 400] loss: 0.414667  (25600/60000)
  [Batch 500] loss: 0.444063  (32000/60000)
  [Batch 600] loss: 0.448116  (38400/60000)
  [Batch 700] loss: 0.592468  (44800/60000)
  [Batch 800] loss: 0.566288  (51200/60000)
  [Batch 900] loss: 0.409123  (57600/60000)
Epoch 9/30
  [Batch 000] loss: 0.292656  (0/60000)
  [Batch 100] loss: 0.432430  (6400/60000)
  [Batch 200] loss: 0.276593  (12800/60000)
  [Batch 300] loss: 0.483064  (19200/60000)
  [Batch 400] loss: 0.399757  (25600/60000)
  [Batch 500] loss: 0.435298  (32000/60000)
  [Batch 600] loss: 0.437599  (38400/60000)
  [Batch 700] loss: 0.580449  (44800/60000)
  [Batch 800] loss: 0.551067  (51200/60000)
  [Batch 900] loss: 0.403976  (57600/60000)
Epoch 10/30
  [Batch 000] loss: 0.279295  (0/60000)
  [Batch 100] loss: 0.420121  (6400/60000)
  [Batch 200] loss: 0.266584  (12800/60000)
  [Batch 300] loss: 0.471924  (19200/60000)
  [Batch 400] loss: 0.383534  (25600/60000)
  [Batch 500] loss: 0.427809  (32000/60000)
  [Batch 600] loss: 0.427010  (38400/60000)
  [Batch 700] loss: 0.569058  (44800/60000)
  [Batch 800] loss: 0.538354  (51200/60000)
  [Batch 900] loss: 0.399800  (57600/60000)
Epoch 11/30
  [Batch 000] loss: 0.269689  (0/60000)
  [Batch 100] loss: 0.407943  (6400/60000)
  [Batch 200] loss: 0.258925  (12800/60000)
  [Batch 300] loss: 0.461359  (19200/60000)
  [Batch 400] loss: 0.370619  (25600/60000)
  [Batch 500] loss: 0.421615  (32000/60000)
  [Batch 600] loss: 0.416850  (38400/60000)
  [Batch 700] loss: 0.558628  (44800/60000)
  [Batch 800] loss: 0.526706  (51200/60000)
  [Batch 900] loss: 0.396804  (57600/60000)
Epoch 12/30
  [Batch 000] loss: 0.260804  (0/60000)
  [Batch 100] loss: 0.398110  (6400/60000)
  [Batch 200] loss: 0.252341  (12800/60000)
  [Batch 300] loss: 0.452065  (19200/60000)
  [Batch 400] loss: 0.359434  (25600/60000)
  [Batch 500] loss: 0.414424  (32000/60000)
  [Batch 600] loss: 0.407397  (38400/60000)
  [Batch 700] loss: 0.548599  (44800/60000)
  [Batch 800] loss: 0.517297  (51200/60000)
  [Batch 900] loss: 0.393840  (57600/60000)
Epoch 13/30
  [Batch 000] loss: 0.253597  (0/60000)
  [Batch 100] loss: 0.389398  (6400/60000)
  [Batch 200] loss: 0.246657  (12800/60000)
  [Batch 300] loss: 0.442607  (19200/60000)
  [Batch 400] loss: 0.348924  (25600/60000)
  [Batch 500] loss: 0.408072  (32000/60000)
  [Batch 600] loss: 0.396639  (38400/60000)
  [Batch 700] loss: 0.538543  (44800/60000)
  [Batch 800] loss: 0.508711  (51200/60000)
  [Batch 900] loss: 0.391080  (57600/60000)
Epoch 14/30
  [Batch 000] loss: 0.248414  (0/60000)
  [Batch 100] loss: 0.382254  (6400/60000)
  [Batch 200] loss: 0.240985  (12800/60000)
  [Batch 300] loss: 0.433764  (19200/60000)
  [Batch 400] loss: 0.340201  (25600/60000)
  [Batch 500] loss: 0.403168  (32000/60000)
  [Batch 600] loss: 0.386461  (38400/60000)
  [Batch 700] loss: 0.529644  (44800/60000)
  [Batch 800] loss: 0.500384  (51200/60000)
  [Batch 900] loss: 0.388953  (57600/60000)
Epoch 15/30
  [Batch 000] loss: 0.243264  (0/60000)
  [Batch 100] loss: 0.377102  (6400/60000)
  [Batch 200] loss: 0.236973  (12800/60000)
  [Batch 300] loss: 0.425228  (19200/60000)
  [Batch 400] loss: 0.333074  (25600/60000)
  [Batch 500] loss: 0.396663  (32000/60000)
  [Batch 600] loss: 0.377657  (38400/60000)
  [Batch 700] loss: 0.520959  (44800/60000)
  [Batch 800] loss: 0.491937  (51200/60000)
  [Batch 900] loss: 0.386544  (57600/60000)
Epoch 16/30
  [Batch 000] loss: 0.238198  (0/60000)
  [Batch 100] loss: 0.371188  (6400/60000)
  [Batch 200] loss: 0.233625  (12800/60000)
  [Batch 300] loss: 0.418030  (19200/60000)
  [Batch 400] loss: 0.327714  (25600/60000)
  [Batch 500] loss: 0.391668  (32000/60000)
  [Batch 600] loss: 0.371105  (38400/60000)
  [Batch 700] loss: 0.513748  (44800/60000)
  [Batch 800] loss: 0.483393  (51200/60000)
  [Batch 900] loss: 0.385123  (57600/60000)
Epoch 17/30
  [Batch 000] loss: 0.233238  (0/60000)
  [Batch 100] loss: 0.366876  (6400/60000)
  [Batch 200] loss: 0.230181  (12800/60000)
  [Batch 300] loss: 0.411154  (19200/60000)
  [Batch 400] loss: 0.321979  (25600/60000)
  [Batch 500] loss: 0.386986  (32000/60000)
  [Batch 600] loss: 0.363846  (38400/60000)
  [Batch 700] loss: 0.506928  (44800/60000)
  [Batch 800] loss: 0.475719  (51200/60000)
  [Batch 900] loss: 0.382134  (57600/60000)
Epoch 18/30
  [Batch 000] loss: 0.227990  (0/60000)
  [Batch 100] loss: 0.362035  (6400/60000)
  [Batch 200] loss: 0.227421  (12800/60000)
  [Batch 300] loss: 0.404554  (19200/60000)
  [Batch 400] loss: 0.317210  (25600/60000)
  [Batch 500] loss: 0.382205  (32000/60000)
  [Batch 600] loss: 0.359189  (38400/60000)
  [Batch 700] loss: 0.500123  (44800/60000)
  [Batch 800] loss: 0.469954  (51200/60000)
  [Batch 900] loss: 0.380585  (57600/60000)
Epoch 19/30
  [Batch 000] loss: 0.223318  (0/60000)
  [Batch 100] loss: 0.356932  (6400/60000)
  [Batch 200] loss: 0.224415  (12800/60000)
  [Batch 300] loss: 0.398227  (19200/60000)
  [Batch 400] loss: 0.313409  (25600/60000)
  [Batch 500] loss: 0.379055  (32000/60000)
  [Batch 600] loss: 0.351500  (38400/60000)
  [Batch 700] loss: 0.493117  (44800/60000)
  [Batch 800] loss: 0.464684  (51200/60000)
  [Batch 900] loss: 0.378651  (57600/60000)
Epoch 20/30
  [Batch 000] loss: 0.220221  (0/60000)
  [Batch 100] loss: 0.353089  (6400/60000)
  [Batch 200] loss: 0.221179  (12800/60000)
  [Batch 300] loss: 0.391830  (19200/60000)
  [Batch 400] loss: 0.310293  (25600/60000)
  [Batch 500] loss: 0.375351  (32000/60000)
  [Batch 600] loss: 0.346178  (38400/60000)
  [Batch 700] loss: 0.486746  (44800/60000)
  [Batch 800] loss: 0.459461  (51200/60000)
  [Batch 900] loss: 0.375798  (57600/60000)
Epoch 21/30
  [Batch 000] loss: 0.217091  (0/60000)
  [Batch 100] loss: 0.349274  (6400/60000)
  [Batch 200] loss: 0.218514  (12800/60000)
  [Batch 300] loss: 0.385337  (19200/60000)
  [Batch 400] loss: 0.307185  (25600/60000)
  [Batch 500] loss: 0.371372  (32000/60000)
  [Batch 600] loss: 0.341217  (38400/60000)
  [Batch 700] loss: 0.479783  (44800/60000)
  [Batch 800] loss: 0.455141  (51200/60000)
  [Batch 900] loss: 0.373525  (57600/60000)
Epoch 22/30
  [Batch 000] loss: 0.214700  (0/60000)
  [Batch 100] loss: 0.343899  (6400/60000)
  [Batch 200] loss: 0.215722  (12800/60000)
  [Batch 300] loss: 0.379155  (19200/60000)
  [Batch 400] loss: 0.304236  (25600/60000)
  [Batch 500] loss: 0.366802  (32000/60000)
  [Batch 600] loss: 0.335905  (38400/60000)
  [Batch 700] loss: 0.473781  (44800/60000)
  [Batch 800] loss: 0.450134  (51200/60000)
  [Batch 900] loss: 0.372013  (57600/60000)
Epoch 23/30
  [Batch 000] loss: 0.212068  (0/60000)
  [Batch 100] loss: 0.340003  (6400/60000)
  [Batch 200] loss: 0.213311  (12800/60000)
  [Batch 300] loss: 0.374064  (19200/60000)
  [Batch 400] loss: 0.301538  (25600/60000)
  [Batch 500] loss: 0.362536  (32000/60000)
  [Batch 600] loss: 0.331584  (38400/60000)
  [Batch 700] loss: 0.467610  (44800/60000)
  [Batch 800] loss: 0.445298  (51200/60000)
  [Batch 900] loss: 0.370164  (57600/60000)
Epoch 24/30
  [Batch 000] loss: 0.209147  (0/60000)
  [Batch 100] loss: 0.335752  (6400/60000)
  [Batch 200] loss: 0.210975  (12800/60000)
  [Batch 300] loss: 0.368908  (19200/60000)
  [Batch 400] loss: 0.298476  (25600/60000)
  [Batch 500] loss: 0.358229  (32000/60000)
  [Batch 600] loss: 0.328922  (38400/60000)
  [Batch 700] loss: 0.461994  (44800/60000)
  [Batch 800] loss: 0.439396  (51200/60000)
  [Batch 900] loss: 0.367833  (57600/60000)
Epoch 25/30
  [Batch 000] loss: 0.207176  (0/60000)
  [Batch 100] loss: 0.331164  (6400/60000)
  [Batch 200] loss: 0.209471  (12800/60000)
  [Batch 300] loss: 0.363497  (19200/60000)
  [Batch 400] loss: 0.295503  (25600/60000)
  [Batch 500] loss: 0.354328  (32000/60000)
  [Batch 600] loss: 0.324493  (38400/60000)
  [Batch 700] loss: 0.454668  (44800/60000)
  [Batch 800] loss: 0.433117  (51200/60000)
  [Batch 900] loss: 0.364787  (57600/60000)
Epoch 26/30
  [Batch 000] loss: 0.205298  (0/60000)
  [Batch 100] loss: 0.327925  (6400/60000)
  [Batch 200] loss: 0.206799  (12800/60000)
  [Batch 300] loss: 0.357894  (19200/60000)
  [Batch 400] loss: 0.293235  (25600/60000)
  [Batch 500] loss: 0.350974  (32000/60000)
  [Batch 600] loss: 0.323158  (38400/60000)
  [Batch 700] loss: 0.450028  (44800/60000)
  [Batch 800] loss: 0.428212  (51200/60000)
  [Batch 900] loss: 0.363223  (57600/60000)
Epoch 27/30
  [Batch 000] loss: 0.203529  (0/60000)
  [Batch 100] loss: 0.324919  (6400/60000)
  [Batch 200] loss: 0.204187  (12800/60000)
  [Batch 300] loss: 0.352448  (19200/60000)
  [Batch 400] loss: 0.291214  (25600/60000)
  [Batch 500] loss: 0.346723  (32000/60000)
  [Batch 600] loss: 0.319924  (38400/60000)
  [Batch 700] loss: 0.445385  (44800/60000)
  [Batch 800] loss: 0.424046  (51200/60000)
  [Batch 900] loss: 0.360469  (57600/60000)
Epoch 28/30
```

```
  [Batch 000] loss: 0.201758  (0/60000)
  [Batch 100] loss: 0.320160  (6400/60000)
  [Batch 200] loss: 0.203266  (12800/60000)
  [Batch 300] loss: 0.348552  (19200/60000)
  [Batch 400] loss: 0.289605  (25600/60000)
  [Batch 500] loss: 0.344026  (32000/60000)
  [Batch 600] loss: 0.317359  (38400/60000)
  [Batch 700] loss: 0.439096  (44800/60000)
  [Batch 800] loss: 0.418926  (51200/60000)
  [Batch 900] loss: 0.358580  (57600/60000)
Epoch 29/30
  [Batch 000] loss: 0.199397  (0/60000)
  [Batch 100] loss: 0.316069  (6400/60000)
  [Batch 200] loss: 0.201307  (12800/60000)
  [Batch 300] loss: 0.343710  (19200/60000)
  [Batch 400] loss: 0.288007  (25600/60000)
  [Batch 500] loss: 0.340290  (32000/60000)
  [Batch 600] loss: 0.316213  (38400/60000)
  [Batch 700] loss: 0.435394  (44800/60000)
  [Batch 800] loss: 0.413641  (51200/60000)
  [Batch 900] loss: 0.357002  (57600/60000)
Epoch 30/30
  [Batch 000] loss: 0.196839  (0/60000)
  [Batch 100] loss: 0.312713  (6400/60000)
  [Batch 200] loss: 0.199656  (12800/60000)
  [Batch 300] loss: 0.338813  (19200/60000)
  [Batch 400] loss: 0.286923  (25600/60000)
  [Batch 500] loss: 0.338375  (32000/60000)
  [Batch 600] loss: 0.313621  (38400/60000)
  [Batch 700] loss: 0.431885  (44800/60000)
  [Batch 800] loss: 0.409366  (51200/60000)
  [Batch 900] loss: 0.355182  (57600/60000)
Best accuracy = 86.88% at epoch 30
```



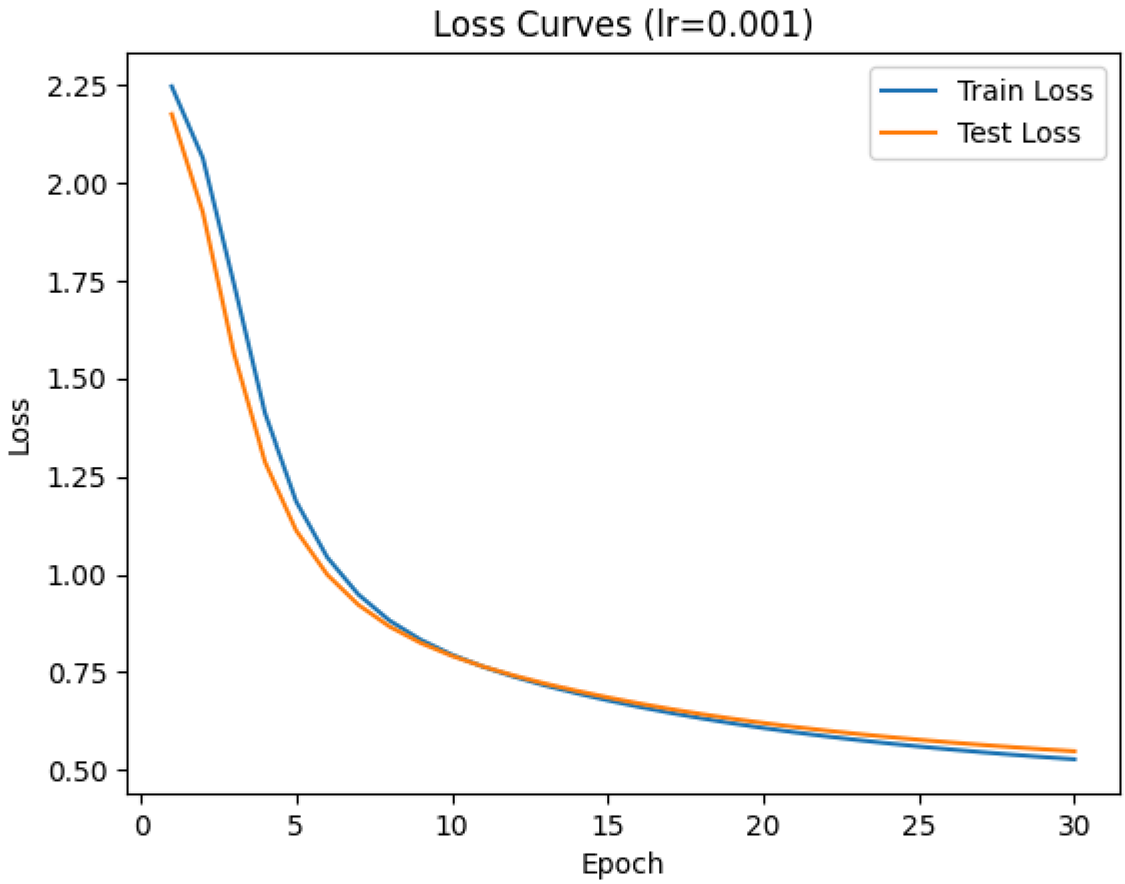
```
Final test accuracy for LR=0.01: 86.88%
```

In [8]:

```
# Parameters
loss_fn = nn.CrossEntropyLoss()
learning_rate = 1e-3
epochs = 30

# Experiment for learning rate = 1
print(f"Learning rate: {learning_rate} \n")
model = NeuralNetwork().to(device)
optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)

train_losses, test_losses, test_accs = [], [], []
for epoch in range(1, epochs + 1):
    print(f"Epoch {epoch}/{epochs}")
    train_loss = train(train_dataloader, model, loss_fn, optimizer)
    acc, test_loss = test(test_dataloader, model, loss_fn)

    train_losses.append(train_loss)
    test_losses.append(test_loss)
    test_accs.append(acc)

plot_train_test_loss(train_losses, test_losses, test_accs, learning_rate)
result = test_accs[-1]

# Final result for learning rate = 1
print(f"Final test accuracy for LR={learning_rate}: {result*100:.2f}%")
```

```
Learning rate: 0.001

Epoch 1/30
  [Batch 000] loss: 2.303828  (0/60000)
  [Batch 100] loss: 2.294583  (6400/60000)
  [Batch 200] loss: 2.276409  (12800/60000)
  [Batch 300] loss: 2.283519  (19200/60000)
  [Batch 400] loss: 2.256564  (25600/60000)
  [Batch 500] loss: 2.231722  (32000/60000)
  [Batch 600] loss: 2.242526  (38400/60000)
  [Batch 700] loss: 2.208930  (44800/60000)
  [Batch 800] loss: 2.205115  (51200/60000)
  [Batch 900] loss: 2.176370  (57600/60000)
Epoch 2/30
  [Batch 000] loss: 2.183001  (0/60000)
  [Batch 100] loss: 2.172985  (6400/60000)
  [Batch 200] loss: 2.119371  (12800/60000)
  [Batch 300] loss: 2.146827  (19200/60000)
  [Batch 400] loss: 2.089874  (25600/60000)
  [Batch 500] loss: 2.027265  (32000/60000)
  [Batch 600] loss: 2.064234  (38400/60000)
  [Batch 700] loss: 1.980165  (44800/60000)
  [Batch 800] loss: 1.986772  (51200/60000)
  [Batch 900] loss: 1.924709  (57600/60000)
Epoch 3/30
  [Batch 000] loss: 1.950836  (0/60000)
  [Batch 100] loss: 1.924830  (6400/60000)
  [Batch 200] loss: 1.812185  (12800/60000)
  [Batch 300] loss: 1.868949  (19200/60000)
  [Batch 400] loss: 1.753796  (25600/60000)
  [Batch 500] loss: 1.686423  (32000/60000)
  [Batch 600] loss: 1.726975  (38400/60000)
  [Batch 700] loss: 1.615046  (44800/60000)
  [Batch 800] loss: 1.645683  (51200/60000)
  [Batch 900] loss: 1.546779  (57600/60000)
Epoch 4/30
  [Batch 000] loss: 1.625989  (0/60000)
  [Batch 100] loss: 1.593464  (6400/60000)
  [Batch 200] loss: 1.443544  (12800/60000)
  [Batch 300] loss: 1.526096  (19200/60000)
  [Batch 400] loss: 1.403372  (25600/60000)
  [Batch 500] loss: 1.383578  (32000/60000)
  [Batch 600] loss: 1.404219  (38400/60000)
  [Batch 700] loss: 1.321659  (44800/60000)
  [Batch 800] loss: 1.358658  (51200/60000)
  [Batch 900] loss: 1.257736  (57600/60000)
Epoch 5/30
  [Batch 000] loss: 1.365765  (0/60000)
  [Batch 100] loss: 1.347140  (6400/60000)
  [Batch 200] loss: 1.180880  (12800/60000)
  [Batch 300] loss: 1.289942  (19200/60000)
  [Batch 400] loss: 1.167366  (25600/60000)
  [Batch 500] loss: 1.182463  (32000/60000)
  [Batch 600] loss: 1.199882  (38400/60000)
  [Batch 700] loss: 1.135370  (44800/60000)
  [Batch 800] loss: 1.173741  (51200/60000)
  [Batch 900] loss: 1.087108  (57600/60000)
Epoch 6/30
  [Batch 000] loss: 1.186573  (0/60000)
  [Batch 100] loss: 1.187922  (6400/60000)
  [Batch 200] loss: 1.005452  (12800/60000)
  [Batch 300] loss: 1.141508  (19200/60000)
  [Batch 400] loss: 1.020257  (25600/60000)
  [Batch 500] loss: 1.044880  (32000/60000)
  [Batch 600] loss: 1.072933  (38400/60000)
  [Batch 700] loss: 1.015414  (44800/60000)
  [Batch 800] loss: 1.052434  (51200/60000)
  [Batch 900] loss: 0.980494  (57600/60000)
Epoch 7/30
  [Batch 000] loss: 1.060672  (0/60000)
  [Batch 100] loss: 1.084226  (6400/60000)
  [Batch 200] loss: 0.884960  (12800/60000)
  [Batch 300] loss: 1.042576  (19200/60000)
  [Batch 400] loss: 0.927745  (25600/60000)
  [Batch 500] loss: 0.946810  (32000/60000)
  [Batch 600] loss: 0.989755  (38400/60000)
  [Batch 700] loss: 0.936694  (44800/60000)
  [Batch 800] loss: 0.968083  (51200/60000)
  [Batch 900] loss: 0.909808  (57600/60000)
Epoch 8/30
  [Batch 000] loss: 0.968224  (0/60000)
  [Batch 100] loss: 1.012559  (6400/60000)
  [Batch 200] loss: 0.798588  (12800/60000)
  [Batch 300] loss: 0.972811  (19200/60000)
  [Batch 400] loss: 0.866301  (25600/60000)
  [Batch 500] loss: 0.874238  (32000/60000)
  [Batch 600] loss: 0.931585  (38400/60000)
  [Batch 700] loss: 0.883664  (44800/60000)
  [Batch 800] loss: 0.906659  (51200/60000)
  [Batch 900] loss: 0.859272  (57600/60000)
Epoch 9/30
  [Batch 000] loss: 0.897086  (0/60000)
  [Batch 100] loss: 0.959436  (6400/60000)
  [Batch 200] loss: 0.734227  (12800/60000)
  [Batch 300] loss: 0.920661  (19200/60000)
  [Batch 400] loss: 0.822767  (25600/60000)
  [Batch 500] loss: 0.818962  (32000/60000)
  [Batch 600] loss: 0.888017  (38400/60000)
  [Batch 700] loss: 0.846666  (44800/60000)
  [Batch 800] loss: 0.860403  (51200/60000)
  [Batch 900] loss: 0.820918  (57600/60000)
Epoch 10/30
  [Batch 000] loss: 0.840391  (0/60000)
  [Batch 100] loss: 0.917360  (6400/60000)
  [Batch 200] loss: 0.684296  (12800/60000)
  [Batch 300] loss: 0.880141  (19200/60000)
  [Batch 400] loss: 0.789799  (25600/60000)
  [Batch 500] loss: 0.775983  (32000/60000)
  [Batch 600] loss: 0.853190  (38400/60000)
  [Batch 700] loss: 0.819529  (44800/60000)
  [Batch 800] loss: 0.824495  (51200/60000)
  [Batch 900] loss: 0.790276  (57600/60000)
Epoch 11/30
  [Batch 000] loss: 0.793604  (0/60000)
  [Batch 100] loss: 0.882190  (6400/60000)
  [Batch 200] loss: 0.644301  (12800/60000)
  [Batch 300] loss: 0.847613  (19200/60000)
  [Batch 400] loss: 0.763741  (25600/60000)
  [Batch 500] loss: 0.742269  (32000/60000)
  [Batch 600] loss: 0.823611  (38400/60000)
  [Batch 700] loss: 0.798187  (44800/60000)
  [Batch 800] loss: 0.795551  (51200/60000)
  [Batch 900] loss: 0.764730  (57600/60000)
Epoch 12/30
  [Batch 000] loss: 0.753954  (0/60000)
  [Batch 100] loss: 0.851470  (6400/60000)
  [Batch 200] loss: 0.611192  (12800/60000)
  [Batch 300] loss: 0.820959  (19200/60000)
  [Batch 400] loss: 0.742021  (25600/60000)
  [Batch 500] loss: 0.715250  (32000/60000)
  [Batch 600] loss: 0.797613  (38400/60000)
  [Batch 700] loss: 0.780365  (44800/60000)
  [Batch 800] loss: 0.771503  (51200/60000)
  [Batch 900] loss: 0.742787  (57600/60000)
Epoch 13/30
  [Batch 000] loss: 0.719558  (0/60000)
  [Batch 100] loss: 0.823890  (6400/60000)
  [Batch 200] loss: 0.583001  (12800/60000)
  [Batch 300] loss: 0.798571  (19200/60000)
  [Batch 400] loss: 0.723364  (25600/60000)
  [Batch 500] loss: 0.693125  (32000/60000)
  [Batch 600] loss: 0.773971  (38400/60000)
  [Batch 700] loss: 0.764705  (44800/60000)
  [Batch 800] loss: 0.751068  (51200/60000)
  [Batch 900] loss: 0.723374  (57600/60000)
Epoch 14/30
  [Batch 000] loss: 0.689118  (0/60000)
  [Batch 100] loss: 0.798727  (6400/60000)
  [Batch 200] loss: 0.558424  (12800/60000)
  [Batch 300] loss: 0.779274  (19200/60000)
  [Batch 400] loss: 0.706993  (25600/60000)
  [Batch 500] loss: 0.674626  (32000/60000)
  [Batch 600] loss: 0.752109  (38400/60000)
  [Batch 700] loss: 0.750658  (44800/60000)
  [Batch 800] loss: 0.733283  (51200/60000)
  [Batch 900] loss: 0.705733  (57600/60000)
Epoch 15/30
  [Batch 000] loss: 0.662079  (0/60000)
  [Batch 100] loss: 0.775399  (6400/60000)
  [Batch 200] loss: 0.536745  (12800/60000)
  [Batch 300] loss: 0.762192  (19200/60000)
  [Batch 400] loss: 0.692596  (25600/60000)
  [Batch 500] loss: 0.658787  (32000/60000)
  [Batch 600] loss: 0.731725  (38400/60000)
  [Batch 700] loss: 0.737813  (44800/60000)
  [Batch 800] loss: 0.717625  (51200/60000)
  [Batch 900] loss: 0.689597  (57600/60000)
Epoch 16/30
  [Batch 000] loss: 0.637909  (0/60000)
  [Batch 100] loss: 0.753695  (6400/60000)
  [Batch 200] loss: 0.517611  (12800/60000)
  [Batch 300] loss: 0.746848  (19200/60000)
  [Batch 400] loss: 0.679704  (25600/60000)
  [Batch 500] loss: 0.645157  (32000/60000)
  [Batch 600] loss: 0.712672  (38400/60000)
  [Batch 700] loss: 0.726052  (44800/60000)
  [Batch 800] loss: 0.703751  (51200/60000)
  [Batch 900] loss: 0.674642  (57600/60000)
Epoch 17/30
  [Batch 000] loss: 0.616357  (0/60000)
  [Batch 100] loss: 0.733584  (6400/60000)
  [Batch 200] loss: 0.500528  (12800/60000)
  [Batch 300] loss: 0.732867  (19200/60000)
  [Batch 400] loss: 0.668189  (25600/60000)
  [Batch 500] loss: 0.633395  (32000/60000)
  [Batch 600] loss: 0.694878  (38400/60000)
  [Batch 700] loss: 0.715239  (44800/60000)
  [Batch 800] loss: 0.691168  (51200/60000)
  [Batch 900] loss: 0.660824  (57600/60000)
Epoch 18/30
  [Batch 000] loss: 0.596907  (0/60000)
  [Batch 100] loss: 0.714965  (6400/60000)
  [Batch 200] loss: 0.485146  (12800/60000)
  [Batch 300] loss: 0.720283  (19200/60000)
  [Batch 400] loss: 0.657588  (25600/60000)
  [Batch 500] loss: 0.623058  (32000/60000)
  [Batch 600] loss: 0.678141  (38400/60000)
  [Batch 700] loss: 0.705538  (44800/60000)
  [Batch 800] loss: 0.680322  (51200/60000)
  [Batch 900] loss: 0.647814  (57600/60000)
Epoch 19/30
  [Batch 000] loss: 0.579261  (0/60000)
  [Batch 100] loss: 0.697758  (6400/60000)
  [Batch 200] loss: 0.471065  (12800/60000)
  [Batch 300] loss: 0.708816  (19200/60000)
  [Batch 400] loss: 0.647917  (25600/60000)
  [Batch 500] loss: 0.613724  (32000/60000)
  [Batch 600] loss: 0.662677  (38400/60000)
  [Batch 700] loss: 0.696815  (44800/60000)
  [Batch 800] loss: 0.670919  (51200/60000)
  [Batch 900] loss: 0.635654  (57600/60000)
Epoch 20/30
  [Batch 000] loss: 0.563216  (0/60000)
  [Batch 100] loss: 0.681991  (6400/60000)
  [Batch 200] loss: 0.458231  (12800/60000)
  [Batch 300] loss: 0.698135  (19200/60000)
  [Batch 400] loss: 0.639045  (25600/60000)
  [Batch 500] loss: 0.605454  (32000/60000)
  [Batch 600] loss: 0.648380  (38400/60000)
  [Batch 700] loss: 0.689160  (44800/60000)
  [Batch 800] loss: 0.662812  (51200/60000)
  [Batch 900] loss: 0.624248  (57600/60000)
Epoch 21/30
  [Batch 000] loss: 0.548592  (0/60000)
  [Batch 100] loss: 0.667491  (6400/60000)
  [Batch 200] loss: 0.446559  (12800/60000)
  [Batch 300] loss: 0.688192  (19200/60000)
  [Batch 400] loss: 0.630889  (25600/60000)
  [Batch 500] loss: 0.597982  (32000/60000)
  [Batch 600] loss: 0.635078  (38400/60000)
  [Batch 700] loss: 0.682485  (44800/60000)
  [Batch 800] loss: 0.655763  (51200/60000)
  [Batch 900] loss: 0.613533  (57600/60000)
Epoch 22/30
  [Batch 000] loss: 0.535174  (0/60000)
  [Batch 100] loss: 0.654143  (6400/60000)
  [Batch 200] loss: 0.435875  (12800/60000)
  [Batch 300] loss: 0.678929  (19200/60000)
  [Batch 400] loss: 0.623222  (25600/60000)
  [Batch 500] loss: 0.591159  (32000/60000)
  [Batch 600] loss: 0.622732  (38400/60000)
  [Batch 700] loss: 0.676770  (44800/60000)
  [Batch 800] loss: 0.649685  (51200/60000)
  [Batch 900] loss: 0.603374  (57600/60000)
Epoch 23/30
  [Batch 000] loss: 0.522843  (0/60000)
  [Batch 100] loss: 0.641830  (6400/60000)
  [Batch 200] loss: 0.426014  (12800/60000)
  [Batch 300] loss: 0.670207  (19200/60000)
  [Batch 400] loss: 0.615987  (25600/60000)
  [Batch 500] loss: 0.584903  (32000/60000)
  [Batch 600] loss: 0.611262  (38400/60000)
  [Batch 700] loss: 0.671978  (44800/60000)
  [Batch 800] loss: 0.644356  (51200/60000)
  [Batch 900] loss: 0.593674  (57600/60000)
Epoch 24/30
  [Batch 000] loss: 0.511525  (0/60000)
  [Batch 100] loss: 0.630470  (6400/60000)
  [Batch 200] loss: 0.416927  (12800/60000)
  [Batch 300] loss: 0.661929  (19200/60000)
  [Batch 400] loss: 0.609058  (25600/60000)
  [Batch 500] loss: 0.578955  (32000/60000)
  [Batch 600] loss: 0.600613  (38400/60000)
  [Batch 700] loss: 0.668027  (44800/60000)
  [Batch 800] loss: 0.639778  (51200/60000)
  [Batch 900] loss: 0.584402  (57600/60000)
Epoch 25/30
  [Batch 000] loss: 0.501037  (0/60000)
  [Batch 100] loss: 0.619993  (6400/60000)
  [Batch 200] loss: 0.408534  (12800/60000)
  [Batch 300] loss: 0.654118  (19200/60000)
  [Batch 400] loss: 0.602319  (25600/60000)
  [Batch 500] loss: 0.573257  (32000/60000)
  [Batch 600] loss: 0.590776  (38400/60000)
  [Batch 700] loss: 0.664859  (44800/60000)
  [Batch 800] loss: 0.635787  (51200/60000)
  [Batch 900] loss: 0.575495  (57600/60000)
Epoch 26/30
  [Batch 000] loss: 0.491285  (0/60000)
  [Batch 100] loss: 0.610301  (6400/60000)
  [Batch 200] loss: 0.400757  (12800/60000)
  [Batch 300] loss: 0.646703  (19200/60000)
  [Batch 400] loss: 0.595716  (25600/60000)
  [Batch 500] loss: 0.567787  (32000/60000)
  [Batch 600] loss: 0.581683  (38400/60000)
  [Batch 700] loss: 0.662401  (44800/60000)
  [Batch 800] loss: 0.632231  (51200/60000)
  [Batch 900] loss: 0.566941  (57600/60000)
Epoch 27/30
  [Batch 000] loss: 0.482137  (0/60000)
  [Batch 100] loss: 0.601342  (6400/60000)
  [Batch 200] loss: 0.393518  (12800/60000)
  [Batch 300] loss: 0.639669  (19200/60000)
  [Batch 400] loss: 0.589234  (25600/60000)
  [Batch 500] loss: 0.562435  (32000/60000)
  [Batch 600] loss: 0.573248  (38400/60000)
  [Batch 700] loss: 0.660576  (44800/60000)
  [Batch 800] loss: 0.629120  (51200/60000)
  [Batch 900] loss: 0.558634  (57600/60000)
Epoch 28/30
```

```
  [Batch 000] loss: 0.473504  (0/60000)
  [Batch 100] loss: 0.593025  (6400/60000)
  [Batch 200] loss: 0.386805  (12800/60000)
  [Batch 300] loss: 0.632911  (19200/60000)
  [Batch 400] loss: 0.582811  (25600/60000)
  [Batch 500] loss: 0.557173  (32000/60000)
  [Batch 600] loss: 0.565380  (38400/60000)
  [Batch 700] loss: 0.659258  (44800/60000)
  [Batch 800] loss: 0.626315  (51200/60000)
  [Batch 900] loss: 0.550554  (57600/60000)
Epoch 29/30
  [Batch 000] loss: 0.465373  (0/60000)
  [Batch 100] loss: 0.585320  (6400/60000)
  [Batch 200] loss: 0.380557  (12800/60000)
  [Batch 300] loss: 0.626448  (19200/60000)
  [Batch 400] loss: 0.576541  (25600/60000)
  [Batch 500] loss: 0.551986  (32000/60000)
  [Batch 600] loss: 0.558004  (38400/60000)
  [Batch 700] loss: 0.658406  (44800/60000)
  [Batch 800] loss: 0.623770  (51200/60000)
  [Batch 900] loss: 0.542706  (57600/60000)
Epoch 30/30
  [Batch 000] loss: 0.457661  (0/60000)
  [Batch 100] loss: 0.578198  (6400/60000)
  [Batch 200] loss: 0.374724  (12800/60000)
  [Batch 300] loss: 0.620286  (19200/60000)
  [Batch 400] loss: 0.570387  (25600/60000)
  [Batch 500] loss: 0.546915  (32000/60000)
  [Batch 600] loss: 0.551148  (38400/60000)
  [Batch 700] loss: 0.657875  (44800/60000)
  [Batch 800] loss: 0.621470  (51200/60000)
  [Batch 900] loss: 0.535014  (57600/60000)
Best accuracy = 80.85% at epoch 30
```



```
Final test accuracy for LR=0.001: 80.85%
```

Q1.5

In [19]:

```
class FlexibleNet(nn.Module):
    def __init__(self, hidden_units):
        super().__init__()
        layers = [nn.Flatten()]
        input_size = 28 * 28
        for h in hidden_units:
            layers += [nn.Linear(input_size, h), nn.ReLU()]
            input_size = h
        layers.append(nn.Linear(input_size, 10))
        self.net = nn.Sequential(*layers)

    def forward(self, x):
        return self.net(x)

def count_params(model):
    return sum(p.numel() for p in model.parameters() if p.requires_grad)
```

In [14]:

```
wide_net_layers = [1024, 1024]
deep_net_layers = [512, 512, 512, 512]

base_net = NeuralNetwork().to(device)
wide_net = FlexibleNet(wide_net_layers).to(device)
deep_net = FlexibleNet(deep_net_layers).to(device)


print(f"Base net parameters: {count_params(base_net)}")
print(f"Wide net parameters: {count_params(wide_net)}")
print(f"Deep net parameters: {count_params(deep_net)}")
```

```
Base net parameters: 669706
Wide net parameters: 1863690
Deep net parameters: 1195018
```

In [15]:

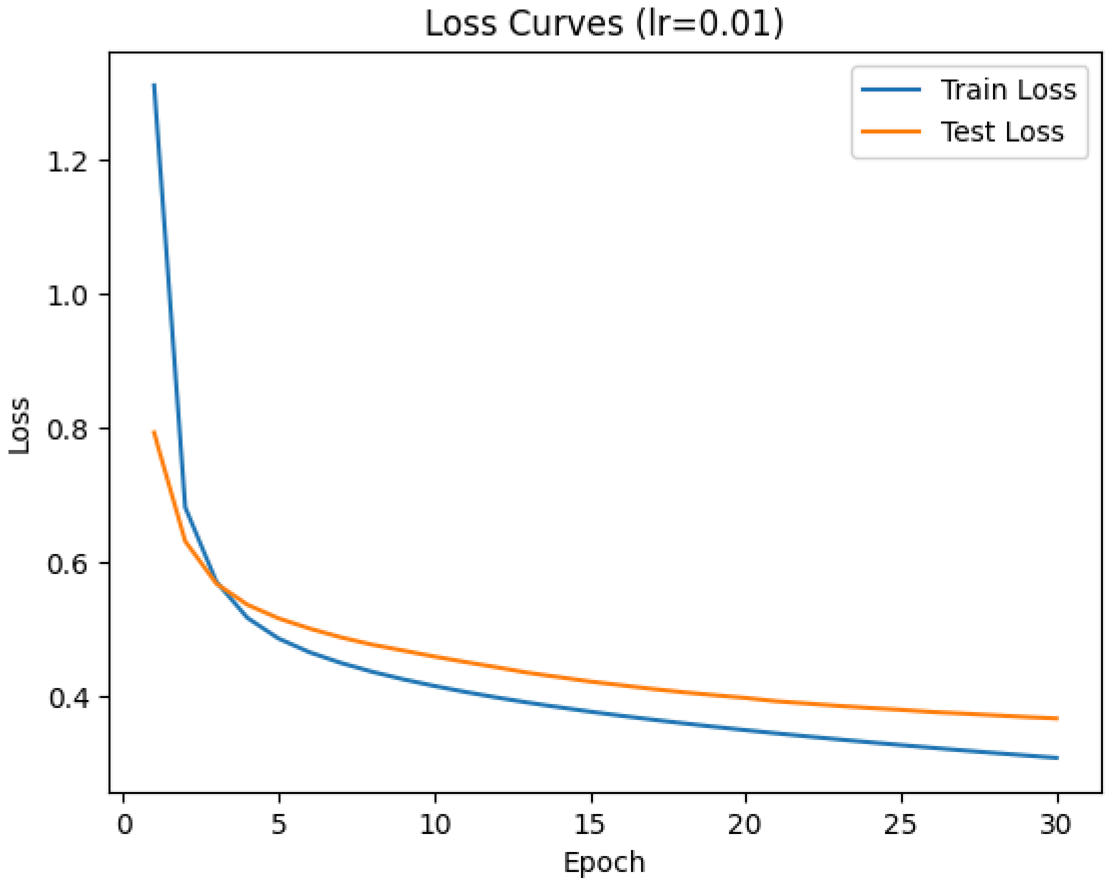
```
# Experiment
learning_rate = 1e-2
epochs = 30

# Base net
base_net = NeuralNetwork().to(device)
wide_net = FlexibleNet(wide_net_layers).to(device)
deep_net = FlexibleNet(deep_net_layers).to(device)

# Optimizers
base_optimizer = torch.optim.SGD(base_net.parameters(), lr=learning_rate)
wide_optimizer = torch.optim.SGD(wide_net.parameters(), lr=learning_rate)
deep_optimizer = torch.optim.SGD(deep_net.parameters(), lr=learning_rate)

# Training loop
base_train_losses, base_test_losses, base_test_accs = [], [], []
wide_train_losses, wide_test_losses, wide_test_accs = [], [], []
deep_train_losses, deep_test_losses, deep_test_accs = [], [], []

for epoch in range(1, epochs + 1):
    print(f"Epoch {epoch}/{epochs}")

    # Base net
    base_train_loss = train(train_dataloader, base_net, loss_fn, base_opt

    # Wide net
    wide_train_loss = train(train_dataloader, wide_net, loss_fn, wide_opt

    # Deep net
    deep_train_loss = train(train_dataloader, deep_net, loss_fn, deep_opt

    # Test loop
    base_acc, base_test_loss = test(test_dataloader, base_net, loss_fn)
    wide_acc, wide_test_loss = test(test_dataloader, wide_net, loss_fn)
    deep_acc, deep_test_loss = test(test_dataloader, deep_net, loss_fn)

    # Record results
    base_train_losses.append(base_train_loss)
    base_test_losses.append(base_test_loss)
    base_test_accs.append(base_acc)

    wide_train_losses.append(wide_train_loss)
    wide_test_losses.append(wide_test_loss)
    wide_test_accs.append(wide_acc)

    deep_train_losses.append(deep_train_loss)
    deep_test_losses.append(deep_test_loss)
    deep_test_accs.append(deep_acc)

# Plot results
print("Base net")
plot_train_test_loss(base_train_losses, base_test_losses, base_test_accs,
print("Wide net")
```

```
plot_train_test_loss(wide_train_losses, wide_test_losses, wide_test_accs,
print("Deep net")
plot_train_test_loss(deep_train_losses, deep_test_losses, deep_test_accs,

# Final results
print("Final test accuracies:")
print(f"Base net: {base_test_accs[-1]*100:.2f}%")
print(f"Wide net: {wide_test_accs[-1]*100:.2f}%")
print(f"Deep net: {deep_test_accs[-1]*100:.2f}%")
```

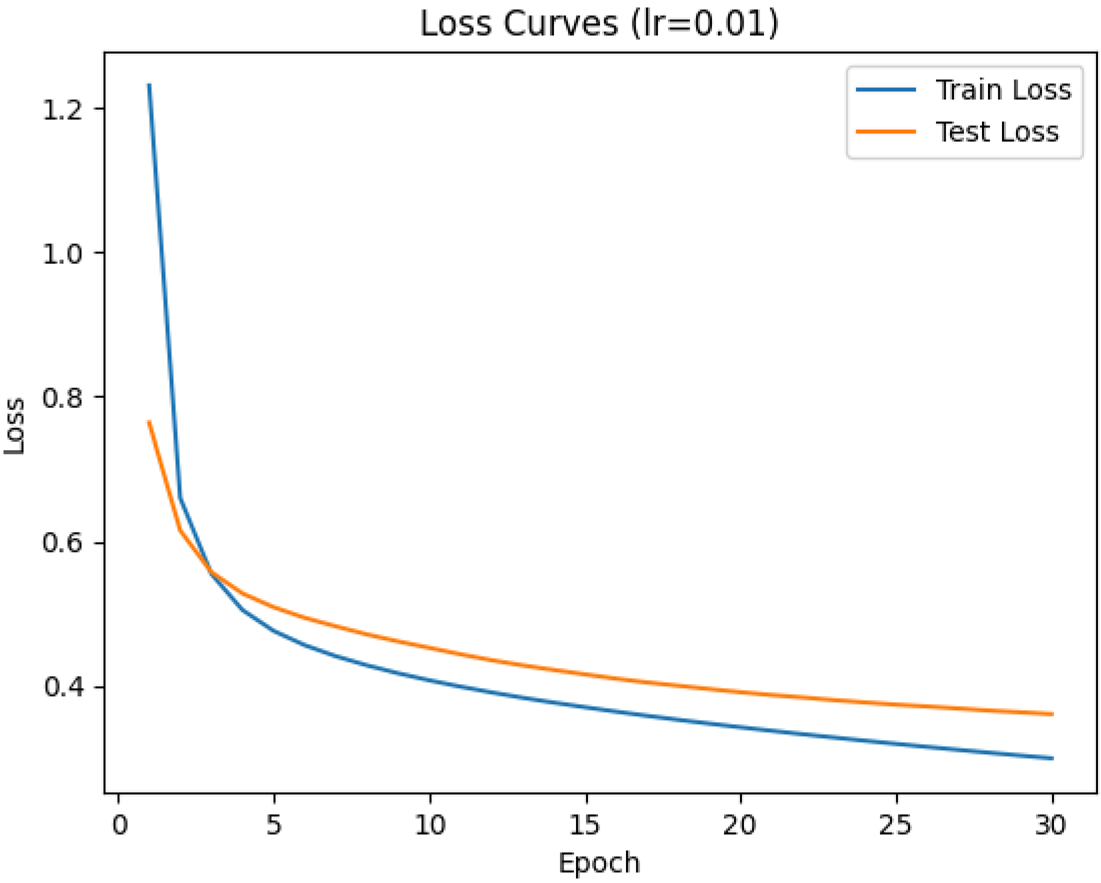
```
Epoch 1/30
  [Batch 000] loss: 2.303660  (0/60000)
  [Batch 100] loss: 2.176378  (6400/60000)
  [Batch 200] loss: 1.831413  (12800/60000)
  [Batch 300] loss: 1.540735  (19200/60000)
  [Batch 400] loss: 1.173211  (25600/60000)
  [Batch 500] loss: 1.068853  (32000/60000)
  [Batch 600] loss: 1.009820  (38400/60000)
  [Batch 700] loss: 0.884349  (44800/60000)
  [Batch 800] loss: 0.877845  (51200/60000)
  [Batch 900] loss: 0.806081  (57600/60000)
  [Batch 000] loss: 2.294289  (0/60000)
  [Batch 100] loss: 2.108594  (6400/60000)
  [Batch 200] loss: 1.667768  (12800/60000)
  [Batch 300] loss: 1.397994  (19200/60000)
  [Batch 400] loss: 1.054158  (25600/60000)
  [Batch 500] loss: 0.971647  (32000/60000)
  [Batch 600] loss: 0.953923  (38400/60000)
  [Batch 700] loss: 0.826362  (44800/60000)
  [Batch 800] loss: 0.835680  (51200/60000)
  [Batch 900] loss: 0.786859  (57600/60000)
  [Batch 000] loss: 2.300710  (0/60000)
  [Batch 100] loss: 2.294351  (6400/60000)
  [Batch 200] loss: 2.283535  (12800/60000)
  [Batch 300] loss: 2.280654  (19200/60000)
  [Batch 400] loss: 2.256174  (25600/60000)
  [Batch 500] loss: 2.189263  (32000/60000)
  [Batch 600] loss: 2.146876  (38400/60000)
  [Batch 700] loss: 1.926384  (44800/60000)
  [Batch 800] loss: 1.664787  (51200/60000)
  [Batch 900] loss: 1.332405  (57600/60000)
Epoch 2/30
  [Batch 000] loss: 0.800691  (0/60000)
  [Batch 100] loss: 0.843801  (6400/60000)
  [Batch 200] loss: 0.593920  (12800/60000)
  [Batch 300] loss: 0.794218  (19200/60000)
  [Batch 400] loss: 0.663038  (25600/60000)
  [Batch 500] loss: 0.642618  (32000/60000)
  [Batch 600] loss: 0.702158  (38400/60000)
  [Batch 700] loss: 0.693678  (44800/60000)
  [Batch 800] loss: 0.693785  (51200/60000)
  [Batch 900] loss: 0.630974  (57600/60000)
  [Batch 000] loss: 0.747122  (0/60000)
  [Batch 100] loss: 0.827825  (6400/60000)
  [Batch 200] loss: 0.560677  (12800/60000)
  [Batch 300] loss: 0.761986  (19200/60000)
  [Batch 400] loss: 0.638837  (25600/60000)
  [Batch 500] loss: 0.623569  (32000/60000)
  [Batch 600] loss: 0.686314  (38400/60000)
  [Batch 700] loss: 0.663960  (44800/60000)
  [Batch 800] loss: 0.684812  (51200/60000)
  [Batch 900] loss: 0.623728  (57600/60000)
  [Batch 000] loss: 1.360003  (0/60000)
  [Batch 100] loss: 1.223379  (6400/60000)
  [Batch 200] loss: 0.974823  (12800/60000)
  [Batch 300] loss: 1.035741  (19200/60000)
  [Batch 400] loss: 0.863341  (25600/60000)
  [Batch 500] loss: 0.840484  (32000/60000)
  [Batch 600] loss: 0.940701  (38400/60000)
  [Batch 700] loss: 0.855377  (44800/60000)
  [Batch 800] loss: 0.812440  (51200/60000)
  [Batch 900] loss: 0.838719  (57600/60000)
Epoch 3/30
  [Batch 000] loss: 0.560952  (0/60000)
  [Batch 100] loss: 0.654364  (6400/60000)
  [Batch 200] loss: 0.444231  (12800/60000)
  [Batch 300] loss: 0.679643  (19200/60000)
  [Batch 400] loss: 0.581939  (25600/60000)
  [Batch 500] loss: 0.558913  (32000/60000)
  [Batch 600] loss: 0.590268  (38400/60000)
  [Batch 700] loss: 0.648587  (44800/60000)
  [Batch 800] loss: 0.661696  (51200/60000)
  [Batch 900] loss: 0.545465  (57600/60000)
  [Batch 000] loss: 0.540681  (0/60000)
  [Batch 100] loss: 0.636915  (6400/60000)
  [Batch 200] loss: 0.428830  (12800/60000)
  [Batch 300] loss: 0.651970  (19200/60000)
  [Batch 400] loss: 0.556481  (25600/60000)
  [Batch 500] loss: 0.548499  (32000/60000)
  [Batch 600] loss: 0.573539  (38400/60000)
  [Batch 700] loss: 0.632697  (44800/60000)
  [Batch 800] loss: 0.651960  (51200/60000)
  [Batch 900] loss: 0.533578  (57600/60000)
  [Batch 000] loss: 0.791263  (0/60000)
  [Batch 100] loss: 0.851716  (6400/60000)
  [Batch 200] loss: 0.623054  (12800/60000)
  [Batch 300] loss: 0.765514  (19200/60000)
  [Batch 400] loss: 0.677022  (25600/60000)
  [Batch 500] loss: 0.563869  (32000/60000)
  [Batch 600] loss: 0.743666  (38400/60000)
  [Batch 700] loss: 0.704555  (44800/60000)
  [Batch 800] loss: 0.651758  (51200/60000)
  [Batch 900] loss: 0.664636  (57600/60000)
Epoch 4/30
  [Batch 000] loss: 0.467872  (0/60000)
  [Batch 100] loss: 0.566793  (6400/60000)
  [Batch 200] loss: 0.386309  (12800/60000)
  [Batch 300] loss: 0.611020  (19200/60000)
  [Batch 400] loss: 0.526651  (25600/60000)
  [Batch 500] loss: 0.518357  (32000/60000)
  [Batch 600] loss: 0.537046  (38400/60000)
  [Batch 700] loss: 0.646580  (44800/60000)
  [Batch 800] loss: 0.645819  (51200/60000)
  [Batch 900] loss: 0.488098  (57600/60000)
  [Batch 000] loss: 0.455949  (0/60000)
  [Batch 100] loss: 0.550238  (6400/60000)
  [Batch 200] loss: 0.372105  (12800/60000)
  [Batch 300] loss: 0.590763  (19200/60000)
  [Batch 400] loss: 0.503814  (25600/60000)
  [Batch 500] loss: 0.506485  (32000/60000)
  [Batch 600] loss: 0.522790  (38400/60000)
  [Batch 700] loss: 0.633594  (44800/60000)
  [Batch 800] loss: 0.630736  (51200/60000)
  [Batch 900] loss: 0.475119  (57600/60000)
  [Batch 000] loss: 0.613578  (0/60000)
  [Batch 100] loss: 0.703891  (6400/60000)
  [Batch 200] loss: 0.494123  (12800/60000)
  [Batch 300] loss: 0.651072  (19200/60000)
  [Batch 400] loss: 0.624162  (25600/60000)
  [Batch 500] loss: 0.478404  (32000/60000)
```

```
    [Batch 600] loss: 0.631008  (38400/60000)
    [Batch 700] loss: 0.648785  (44800/60000)
    [Batch 800] loss: 0.605905  (51200/60000)
    [Batch 900] loss: 0.590089  (57600/60000)
  Epoch 5/30
    [Batch 000] loss: 0.410724  (0/60000)
    [Batch 100] loss: 0.521108  (6400/60000)
    [Batch 200] loss: 0.354405  (12800/60000)
    [Batch 300] loss: 0.566074  (19200/60000)
    [Batch 400] loss: 0.482708  (25600/60000)
    [Batch 500] loss: 0.492035  (32000/60000)
    [Batch 600] loss: 0.505559  (38400/60000)
    [Batch 700] loss: 0.642154  (44800/60000)
    [Batch 800] loss: 0.625001  (51200/60000)
    [Batch 900] loss: 0.453567  (57600/60000)
    [Batch 000] loss: 0.403756  (0/60000)
    [Batch 100] loss: 0.507669  (6400/60000)
    [Batch 200] loss: 0.338477  (12800/60000)
    [Batch 300] loss: 0.550700  (19200/60000)
    [Batch 400] loss: 0.468374  (25600/60000)
    [Batch 500] loss: 0.475941  (32000/60000)
    [Batch 600] loss: 0.492566  (38400/60000)
    [Batch 700] loss: 0.627217  (44800/60000)
    [Batch 800] loss: 0.609615  (51200/60000)
    [Batch 900] loss: 0.442164  (57600/60000)
    [Batch 000] loss: 0.531213  (0/60000)
    [Batch 100] loss: 0.630232  (6400/60000)
    [Batch 200] loss: 0.455367  (12800/60000)
    [Batch 300] loss: 0.578939  (19200/60000)
    [Batch 400] loss: 0.592237  (25600/60000)
    [Batch 500] loss: 0.452802  (32000/60000)
    [Batch 600] loss: 0.548498  (38400/60000)
    [Batch 700] loss: 0.633696  (44800/60000)
    [Batch 800] loss: 0.585563  (51200/60000)
    [Batch 900] loss: 0.543404  (57600/60000)
  Epoch 6/30
    [Batch 000] loss: 0.368802  (0/60000)
    [Batch 100] loss: 0.494291  (6400/60000)
    [Batch 200] loss: 0.332494  (12800/60000)
    [Batch 300] loss: 0.535592  (19200/60000)
    [Batch 400] loss: 0.453121  (25600/60000)
    [Batch 500] loss: 0.474874  (32000/60000)
    [Batch 600] loss: 0.481837  (38400/60000)
    [Batch 700] loss: 0.628293  (44800/60000)
    [Batch 800] loss: 0.604514  (51200/60000)
    [Batch 900] loss: 0.435473  (57600/60000)
    [Batch 000] loss: 0.365311  (0/60000)
    [Batch 100] loss: 0.482488  (6400/60000)
    [Batch 200] loss: 0.315867  (12800/60000)
    [Batch 300] loss: 0.523342  (19200/60000)
    [Batch 400] loss: 0.445744  (25600/60000)
    [Batch 500] loss: 0.456055  (32000/60000)
    [Batch 600] loss: 0.470355  (38400/60000)
    [Batch 700] loss: 0.613988  (44800/60000)
    [Batch 800] loss: 0.589703  (51200/60000)
    [Batch 900] loss: 0.424251  (57600/60000)
    [Batch 000] loss: 0.457735  (0/60000)
    [Batch 100] loss: 0.592134  (6400/60000)
    [Batch 200] loss: 0.435578  (12800/60000)
    [Batch 300] loss: 0.534019  (19200/60000)
    [Batch 400] loss: 0.542013  (25600/60000)
    [Batch 500] loss: 0.440266  (32000/60000)
    [Batch 600] loss: 0.491544  (38400/60000)
    [Batch 700] loss: 0.625184  (44800/60000)
    [Batch 800] loss: 0.575539  (51200/60000)
    [Batch 900] loss: 0.512049  (57600/60000)
  Epoch 7/30
    [Batch 000] loss: 0.337375  (0/60000)
    [Batch 100] loss: 0.474504  (6400/60000)
    [Batch 200] loss: 0.315841  (12800/60000)
    [Batch 300] loss: 0.514734  (19200/60000)
    [Batch 400] loss: 0.430024  (25600/60000)
    [Batch 500] loss: 0.463148  (32000/60000)
    [Batch 600] loss: 0.462436  (38400/60000)
    [Batch 700] loss: 0.612063  (44800/60000)
    [Batch 800] loss: 0.583404  (51200/60000)
    [Batch 900] loss: 0.422937  (57600/60000)
    [Batch 000] loss: 0.337044  (0/60000)
    [Batch 100] loss: 0.464252  (6400/60000)
    [Batch 200] loss: 0.298569  (12800/60000)
    [Batch 300] loss: 0.503994  (19200/60000)
    [Batch 400] loss: 0.427455  (25600/60000)
    [Batch 500] loss: 0.443029  (32000/60000)
    [Batch 600] loss: 0.453057  (38400/60000)
    [Batch 700] loss: 0.599678  (44800/60000)
    [Batch 800] loss: 0.571919  (51200/60000)
    [Batch 900] loss: 0.414348  (57600/60000)
    [Batch 000] loss: 0.396363  (0/60000)
    [Batch 100] loss: 0.566446  (6400/60000)
    [Batch 200] loss: 0.417047  (12800/60000)
    [Batch 300] loss: 0.503244  (19200/60000)
    [Batch 400] loss: 0.485771  (25600/60000)
    [Batch 500] loss: 0.432308  (32000/60000)
    [Batch 600] loss: 0.456621  (38400/60000)
    [Batch 700] loss: 0.614510  (44800/60000)
    [Batch 800] loss: 0.562884  (51200/60000)
    [Batch 900] loss: 0.487526  (57600/60000)
  Epoch 8/30
    [Batch 000] loss: 0.314005  (0/60000)
    [Batch 100] loss: 0.458184  (6400/60000)
    [Batch 200] loss: 0.301783  (12800/60000)
    [Batch 300] loss: 0.497960  (19200/60000)
    [Batch 400] loss: 0.411494  (25600/60000)
    [Batch 500] loss: 0.454038  (32000/60000)
    [Batch 600] loss: 0.446423  (38400/60000)
    [Batch 700] loss: 0.596694  (44800/60000)
    [Batch 800] loss: 0.567050  (51200/60000)
    [Batch 900] loss: 0.416423  (57600/60000)
    [Batch 000] loss: 0.316747  (0/60000)
    [Batch 100] loss: 0.449444  (6400/60000)
    [Batch 200] loss: 0.284865  (12800/60000)
    [Batch 300] loss: 0.488169  (19200/60000)
    [Batch 400] loss: 0.410839  (25600/60000)
    [Batch 500] loss: 0.432527  (32000/60000)
    [Batch 600] loss: 0.438673  (38400/60000)
    [Batch 700] loss: 0.585315  (44800/60000)
    [Batch 800] loss: 0.557336  (51200/60000)
    [Batch 900] loss: 0.408510  (57600/60000)
    [Batch 000] loss: 0.350010  (0/60000)
    [Batch 100] loss: 0.539460  (6400/60000)
    [Batch 200] loss: 0.393846  (12800/60000)
    [Batch 300] loss: 0.476324  (19200/60000)
    [Batch 400] loss: 0.435656  (25600/60000)
    [Batch 500] loss: 0.423994  (32000/60000)
    [Batch 600] loss: 0.433979  (38400/60000)
    [Batch 700] loss: 0.602827  (44800/60000)
    [Batch 800] loss: 0.546599  (51200/60000)
    [Batch 900] loss: 0.465918  (57600/60000)
  Epoch 9/30
    [Batch 000] loss: 0.297060  (0/60000)
    [Batch 100] loss: 0.443253  (6400/60000)
    [Batch 200] loss: 0.290752  (12800/60000)
    [Batch 300] loss: 0.483926  (19200/60000)
    [Batch 400] loss: 0.395123  (25600/60000)
    [Batch 500] loss: 0.445518  (32000/60000)
    [Batch 600] loss: 0.432836  (38400/60000)
    [Batch 700] loss: 0.582581  (44800/60000)
    [Batch 800] loss: 0.550732  (51200/60000)
    [Batch 900] loss: 0.412054  (57600/60000)
    [Batch 000] loss: 0.301361  (0/60000)
    [Batch 100] loss: 0.436677  (6400/60000)
    [Batch 200] loss: 0.273918  (12800/60000)
    [Batch 300] loss: 0.475068  (19200/60000)
    [Batch 400] loss: 0.394420  (25600/60000)
    [Batch 500] loss: 0.423195  (32000/60000)
    [Batch 600] loss: 0.425655  (38400/60000)
    [Batch 700] loss: 0.571121  (44800/60000)
    [Batch 800] loss: 0.543163  (51200/60000)
    [Batch 900] loss: 0.403782  (57600/60000)
    [Batch 000] loss: 0.316280  (0/60000)
    [Batch 100] loss: 0.515803  (6400/60000)
    [Batch 200] loss: 0.368464  (12800/60000)
    [Batch 300] loss: 0.456710  (19200/60000)
    [Batch 400] loss: 0.401133  (25600/60000)
    [Batch 500] loss: 0.416039  (32000/60000)
    [Batch 600] loss: 0.421593  (38400/60000)
    [Batch 700] loss: 0.592362  (44800/60000)
    [Batch 800] loss: 0.529151  (51200/60000)
    [Batch 900] loss: 0.449626  (57600/60000)
  Epoch 10/30
    [Batch 000] loss: 0.284807  (0/60000)
    [Batch 100] loss: 0.429789  (6400/60000)
    [Batch 200] loss: 0.282770  (12800/60000)
    [Batch 300] loss: 0.470240  (19200/60000)
    [Batch 400] loss: 0.380014  (25600/60000)
    [Batch 500] loss: 0.437327  (32000/60000)
    [Batch 600] loss: 0.421181  (38400/60000)
    [Batch 700] loss: 0.569191  (44800/60000)
    [Batch 800] loss: 0.537645  (51200/60000)
    [Batch 900] loss: 0.408997  (57600/60000)
    [Batch 000] loss: 0.290007  (0/60000)
    [Batch 100] loss: 0.424360  (6400/60000)
    [Batch 200] loss: 0.264777  (12800/60000)
    [Batch 300] loss: 0.463160  (19200/60000)
    [Batch 400] loss: 0.381171  (25600/60000)
    [Batch 500] loss: 0.415756  (32000/60000)
    [Batch 600] loss: 0.413938  (38400/60000)
    [Batch 700] loss: 0.558284  (44800/60000)
    [Batch 800] loss: 0.529737  (51200/60000)
    [Batch 900] loss: 0.400025  (57600/60000)
    [Batch 000] loss: 0.294829  (0/60000)
    [Batch 100] loss: 0.492023  (6400/60000)
    [Batch 200] loss: 0.347787  (12800/60000)
    [Batch 300] loss: 0.441011  (19200/60000)
    [Batch 400] loss: 0.377398  (25600/60000)
    [Batch 500] loss: 0.408612  (32000/60000)
    [Batch 600] loss: 0.412473  (38400/60000)
    [Batch 700] loss: 0.582855  (44800/60000)
    [Batch 800] loss: 0.511934  (51200/60000)
    [Batch 900] loss: 0.435906  (57600/60000)
  Epoch 11/30
    [Batch 000] loss: 0.274709  (0/60000)
    [Batch 100] loss: 0.417422  (6400/60000)
    [Batch 200] loss: 0.275203  (12800/60000)
    [Batch 300] loss: 0.457970  (19200/60000)
    [Batch 400] loss: 0.366769  (25600/60000)
    [Batch 500] loss: 0.430802  (32000/60000)
    [Batch 600] loss: 0.410804  (38400/60000)
    [Batch 700] loss: 0.556922  (44800/60000)
    [Batch 800] loss: 0.526071  (51200/60000)
    [Batch 900] loss: 0.405236  (57600/60000)
    [Batch 000] loss: 0.280528  (0/60000)
    [Batch 100] loss: 0.412982  (6400/60000)
    [Batch 200] loss: 0.257511  (12800/60000)
    [Batch 300] loss: 0.451153  (19200/60000)
    [Batch 400] loss: 0.369354  (25600/60000)
    [Batch 500] loss: 0.407550  (32000/60000)
    [Batch 600] loss: 0.403006  (38400/60000)
    [Batch 700] loss: 0.546720  (44800/60000)
    [Batch 800] loss: 0.517691  (51200/60000)
    [Batch 900] loss: 0.396597  (57600/60000)
    [Batch 000] loss: 0.275530  (0/60000)
    [Batch 100] loss: 0.472064  (6400/60000)
    [Batch 200] loss: 0.329550  (12800/60000)
    [Batch 300] loss: 0.429229  (19200/60000)
    [Batch 400] loss: 0.359678  (25600/60000)
    [Batch 500] loss: 0.401958  (32000/60000)
    [Batch 600] loss: 0.405899  (38400/60000)
    [Batch 700] loss: 0.570417  (44800/60000)
    [Batch 800] loss: 0.496699  (51200/60000)
    [Batch 900] loss: 0.422468  (57600/60000)
  Epoch 12/30
    [Batch 000] loss: 0.265763  (0/60000)
    [Batch 100] loss: 0.405483  (6400/60000)
    [Batch 200] loss: 0.269909  (12800/60000)
    [Batch 300] loss: 0.446450  (19200/60000)
    [Batch 400] loss: 0.354795  (25600/60000)
    [Batch 500] loss: 0.422961  (32000/60000)
    [Batch 600] loss: 0.401942  (38400/60000)
    [Batch 700] loss: 0.546280  (44800/60000)
    [Batch 800] loss: 0.514771  (51200/60000)
    [Batch 900] loss: 0.403323  (57600/60000)
    [Batch 000] loss: 0.272744  (0/60000)
    [Batch 100] loss: 0.403383  (6400/60000)
    [Batch 200] loss: 0.251254  (12800/60000)
    [Batch 300] loss: 0.439926  (19200/60000)
    [Batch 400] loss: 0.357850  (25600/60000)
    [Batch 500] loss: 0.400047  (32000/60000)
    [Batch 600] loss: 0.392445  (38400/60000)
    [Batch 700] loss: 0.535926  (44800/60000)
    [Batch 800] loss: 0.507206  (51200/60000)
    [Batch 900] loss: 0.394402  (57600/60000)
    [Batch 000] loss: 0.257934  (0/60000)
    [Batch 100] loss: 0.454136  (6400/60000)
    [Batch 200] loss: 0.315634  (12800/60000)
    [Batch 300] loss: 0.415812  (19200/60000)
    [Batch 400] loss: 0.349269  (25600/60000)
    [Batch 500] loss: 0.397468  (32000/60000)
    [Batch 600] loss: 0.397953  (38400/60000)
    [Batch 700] loss: 0.559332  (44800/60000)
    [Batch 800] loss: 0.484184  (51200/60000)
    [Batch 900] loss: 0.408309  (57600/60000)
  Epoch 13/30
    [Batch 000] loss: 0.258730  (0/60000)
    [Batch 100] loss: 0.394514  (6400/60000)
    [Batch 200] loss: 0.264961  (12800/60000)
    [Batch 300] loss: 0.436139  (19200/60000)
    [Batch 400] loss: 0.345704  (25600/60000)
    [Batch 500] loss: 0.414879  (32000/60000)
    [Batch 600] loss: 0.394285  (38400/60000)
    [Batch 700] loss: 0.534961  (44800/60000)
    [Batch 800] loss: 0.505546  (51200/60000)
    [Batch 900] loss: 0.400435  (57600/60000)
    [Batch 000] loss: 0.265241  (0/60000)
    [Batch 100] loss: 0.394180  (6400/60000)
    [Batch 200] loss: 0.245222  (12800/60000)
    [Batch 300] loss: 0.428971  (19200/60000)
    [Batch 400] loss: 0.348831  (25600/60000)
    [Batch 500] loss: 0.393369  (32000/60000)
    [Batch 600] loss: 0.382777  (38400/60000)
    [Batch 700] loss: 0.525451  (44800/60000)
    [Batch 800] loss: 0.498197  (51200/60000)
    [Batch 900] loss: 0.392309  (57600/60000)
    [Batch 000] loss: 0.243799  (0/60000)
    [Batch 100] loss: 0.435390  (6400/60000)
    [Batch 200] loss: 0.303578  (12800/60000)
    [Batch 300] loss: 0.404346  (19200/60000)
    [Batch 400] loss: 0.340459  (25600/60000)
    [Batch 500] loss: 0.393119  (32000/60000)
    [Batch 600] loss: 0.390378  (38400/60000)
    [Batch 700] loss: 0.547604  (44800/60000)
    [Batch 800] loss: 0.475502  (51200/60000)
    [Batch 900] loss: 0.397546  (57600/60000)
  Epoch 14/30
    [Batch 000] loss: 0.251790  (0/60000)
    [Batch 100] loss: 0.385864  (6400/60000)
    [Batch 200] loss: 0.259369  (12800/60000)
    [Batch 300] loss: 0.426157  (19200/60000)
    [Batch 400] loss: 0.336615  (25600/60000)
    [Batch 500] loss: 0.406228  (32000/60000)
    [Batch 600] loss: 0.386333  (38400/60000)
    [Batch 700] loss: 0.525194  (44800/60000)
    [Batch 800] loss: 0.496041  (51200/60000)
    [Batch 900] loss: 0.397267  (57600/60000)
    [Batch 000] loss: 0.258933  (0/60000)
    [Batch 100] loss: 0.386103  (6400/60000)
    [Batch 200] loss: 0.239503  (12800/60000)
    [Batch 300] loss: 0.419503  (19200/60000)
    [Batch 400] loss: 0.340210  (25600/60000)
    [Batch 500] loss: 0.387750  (32000/60000)
    [Batch 600] loss: 0.373899  (38400/60000)
    [Batch 700] loss: 0.516445  (44800/60000)
    [Batch 800] loss: 0.489557  (51200/60000)
    [Batch 900] loss: 0.389292  (57600/60000)
    [Batch 000] loss: 0.230960  (0/60000)
    [Batch 100] loss: 0.417780  (6400/60000)
    [Batch 200] loss: 0.293272  (12800/60000)
    [Batch 300] loss: 0.394745  (19200/60000)
    [Batch 400] loss: 0.329051  (25600/60000)
    [Batch 500] loss: 0.388391  (32000/60000)
    [Batch 600] loss: 0.381464  (38400/60000)
    [Batch 700] loss: 0.529992  (44800/60000)
    [Batch 800] loss: 0.463324  (51200/60000)
    [Batch 900] loss: 0.387921  (57600/60000)
  Epoch 15/30
    [Batch 000] loss: 0.246792  (0/60000)
    [Batch 100] loss: 0.378958  (6400/60000)
    [Batch 200] loss: 0.254803  (12800/60000)
    [Batch 300] loss: 0.417276  (19200/60000)
    [Batch 400] loss: 0.328878  (25600/60000)
    [Batch 500] loss: 0.398793  (32000/60000)
    [Batch 600] loss: 0.378741  (38400/60000)
    [Batch 700] loss: 0.515254  (44800/60000)
    [Batch 800] loss: 0.488656  (51200/60000)
    [Batch 900] loss: 0.395151  (57600/60000)
    [Batch 000] loss: 0.253961  (0/60000)
    [Batch 100] loss: 0.378832  (6400/60000)
    [Batch 200] loss: 0.234100  (12800/60000)
    [Batch 300] loss: 0.410175  (19200/60000)
    [Batch 400] loss: 0.332270  (25600/60000)
    [Batch 500] loss: 0.382102  (32000/60000)
    [Batch 600] loss: 0.366032  (38400/60000)
    [Batch 700] loss: 0.508436  (44800/60000)
    [Batch 800] loss: 0.481147  (51200/60000)
    [Batch 900] loss: 0.385196  (57600/60000)
    [Batch 000] loss: 0.222476  (0/60000)
    [Batch 100] loss: 0.402480  (6400/60000)
    [Batch 200] loss: 0.282887  (12800/60000)
    [Batch 300] loss: 0.383797  (19200/60000)
    [Batch 400] loss: 0.321690  (25600/60000)
    [Batch 500] loss: 0.384402  (32000/60000)
    [Batch 600] loss: 0.374431  (38400/60000)
    [Batch 700] loss: 0.514307  (44800/60000)
    [Batch 800] loss: 0.454118  (51200/60000)
    [Batch 900] loss: 0.381879  (57600/60000)
  Epoch 16/30
    [Batch 000] loss: 0.241671  (0/60000)
    [Batch 100] loss: 0.370885  (6400/60000)
    [Batch 200] loss: 0.250524  (12800/60000)
    [Batch 300] loss: 0.408977  (19200/60000)
    [Batch 400] loss: 0.322115  (25600/60000)
    [Batch 500] loss: 0.392691  (32000/60000)
    [Batch 600] loss: 0.372223  (38400/60000)
    [Batch 700] loss: 0.507250  (44800/60000)
    [Batch 800] loss: 0.480861  (51200/60000)
    [Batch 900] loss: 0.392523  (57600/60000)
    [Batch 000] loss: 0.248614  (0/60000)
    [Batch 100] loss: 0.371908  (6400/60000)
    [Batch 200] loss: 0.229961  (12800/60000)
    [Batch 300] loss: 0.402096  (19200/60000)
    [Batch 400] loss: 0.325169  (25600/60000)
    [Batch 500] loss: 0.376142  (32000/60000)
    [Batch 600] loss: 0.358054  (38400/60000)
    [Batch 700] loss: 0.500485  (44800/60000)
    [Batch 800] loss: 0.471465  (51200/60000)
    [Batch 900] loss: 0.381821  (57600/60000)
    [Batch 000] loss: 0.212765  (0/60000)
    [Batch 100] loss: 0.388698  (6400/60000)
    [Batch 200] loss: 0.273186  (12800/60000)
    [Batch 300] loss: 0.372748  (19200/60000)
    [Batch 400] loss: 0.316826  (25600/60000)
    [Batch 500] loss: 0.380068  (32000/60000)
    [Batch 600] loss: 0.365403  (38400/60000)
    [Batch 700] loss: 0.497912  (44800/60000)
    [Batch 800] loss: 0.448924  (51200/60000)
    [Batch 900] loss: 0.379295  (57600/60000)
  Epoch 17/30
    [Batch 000] loss: 0.237446  (0/60000)
    [Batch 100] loss: 0.363169  (6400/60000)
    [Batch 200] loss: 0.246852  (12800/60000)
    [Batch 300] loss: 0.401536  (19200/60000)
    [Batch 400] loss: 0.315789  (25600/60000)
    [Batch 500] loss: 0.387236  (32000/60000)
    [Batch 600] loss: 0.364108  (38400/60000)
    [Batch 700] loss: 0.498949  (44800/60000)
    [Batch 800] loss: 0.474084  (51200/60000)
    [Batch 900] loss: 0.390375  (57600/60000)
    [Batch 000] loss: 0.244125  (0/60000)
    [Batch 100] loss: 0.365175  (6400/60000)
    [Batch 200] loss: 0.226835  (12800/60000)
    [Batch 300] loss: 0.394233  (19200/60000)
    [Batch 400] loss: 0.318997  (25600/60000)
    [Batch 500] loss: 0.370798  (32000/60000)
    [Batch 600] loss: 0.351395  (38400/60000)
    [Batch 700] loss: 0.493012  (44800/60000)
    [Batch 800] loss: 0.463292  (51200/60000)
    [Batch 900] loss: 0.377890  (57600/60000)
    [Batch 000] loss: 0.206155  (0/60000)
    [Batch 100] loss: 0.376768  (6400/60000)
    [Batch 200] loss: 0.264545  (12800/60000)
    [Batch 300] loss: 0.359251  (19200/60000)
    [Batch 400] loss: 0.315255  (25600/60000)
    [Batch 500] loss: 0.377366  (32000/60000)
    [Batch 600] loss: 0.358532  (38400/60000)
    [Batch 700] loss: 0.482512  (44800/60000)
    [Batch 800] loss: 0.441966  (51200/60000)
    [Batch 900] loss: 0.373391  (57600/60000)
  Epoch 18/30
    [Batch 000] loss: 0.231839  (0/60000)
    [Batch 100] loss: 0.356817  (6400/60000)
    [Batch 200] loss: 0.242853  (12800/60000)
    [Batch 300] loss: 0.394472  (19200/60000)
    [Batch 400] loss: 0.309758  (25600/60000)
    [Batch 500] loss: 0.381065  (32000/60000)
    [Batch 600] loss: 0.357559  (38400/60000)
    [Batch 700] loss: 0.490965  (44800/60000)
    [Batch 800] loss: 0.466892  (51200/60000)
    [Batch 900] loss: 0.387861  (57600/60000)
    [Batch 000] loss: 0.239383  (0/60000)
    [Batch 100] loss: 0.358671  (6400/60000)
    [Batch 200] loss: 0.223504  (12800/60000)
    [Batch 300] loss: 0.386226  (19200/60000)
    [Batch 400] loss: 0.313218  (25600/60000)
    [Batch 500] loss: 0.365993  (32000/60000)
    [Batch 600] loss: 0.345127  (38400/60000)
    [Batch 700] loss: 0.484996  (44800/60000)
    [Batch 800] loss: 0.454993  (51200/60000)
    [Batch 900] loss: 0.374117  (57600/60000)
    [Batch 000] loss: 0.203331  (0/60000)
    [Batch 100] loss: 0.369762  (6400/60000)
    [Batch 200] loss: 0.254632  (12800/60000)
    [Batch 300] loss: 0.346494  (19200/60000)
    [Batch 400] loss: 0.309122  (25600/60000)
    [Batch 500] loss: 0.373122  (32000/60000)
    [Batch 600] loss: 0.351184  (38400/60000)
    [Batch 700] loss: 0.471710  (44800/60000)
    [Batch 800] loss: 0.437386  (51200/60000)
    [Batch 900] loss: 0.368835  (57600/60000)
  Epoch 19/30
    [Batch 000] loss: 0.227881  (0/60000)
    [Batch 100] loss: 0.350006  (6400/60000)
    [Batch 200] loss: 0.240550  (12800/60000)
    [Batch 300] loss: 0.387439  (19200/60000)
    [Batch 400] loss: 0.303774  (25600/60000)
    [Batch 500] loss: 0.374878  (32000/60000)
    [Batch 600] loss: 0.350608  (38400/60000)
    [Batch 700] loss: 0.482055  (44800/60000)
    [Batch 800] loss: 0.460563  (51200/60000)
    [Batch 900] loss: 0.385296  (57600/60000)
    [Batch 000] loss: 0.235014  (0/60000)
    [Batch 100] loss: 0.352739  (6400/60000)
    [Batch 200] loss: 0.220255  (12800/60000)
    [Batch 300] loss: 0.378653  (19200/60000)
    [Batch 400] loss: 0.309570  (25600/60000)
    [Batch 500] loss: 0.360252  (32000/60000)
    [Batch 600] loss: 0.339794  (38400/60000)
    [Batch 700] loss: 0.476750  (44800/60000)
    [Batch 800] loss: 0.447477  (51200/60000)
    [Batch 900] loss: 0.372033  (57600/60000)
    [Batch 000] loss: 0.198291  (0/60000)
    [Batch 100] loss: 0.360382  (6400/60000)
    [Batch 200] loss: 0.246952  (12800/60000)
    [Batch 300] loss: 0.338822  (19200/60000)
    [Batch 400] loss: 0.307305  (25600/60000)
    [Batch 500] loss: 0.368955  (32000/60000)
    [Batch 600] loss: 0.343159  (38400/60000)
    [Batch 700] loss: 0.462857  (44800/60000)
    [Batch 800] loss: 0.430092  (51200/60000)
    [Batch 900] loss: 0.364301  (57600/60000)
  Epoch 20/30
    [Batch 000] loss: 0.224251  (0/60000)
    [Batch 100] loss: 0.344384  (6400/60000)
    [Batch 200] loss: 0.237018  (12800/60000)
    [Batch 300] loss: 0.379859  (19200/60000)
    [Batch 400] loss: 0.298264  (25600/60000)
    [Batch 500] loss: 0.369587  (32000/60000)
    [Batch 600] loss: 0.345217  (38400/60000)
    [Batch 700] loss: 0.475010  (44800/60000)
    [Batch 800] loss: 0.454296  (51200/60000)
    [Batch 900] loss: 0.381200  (57600/60000)
    [Batch 000] loss: 0.232205  (0/60000)
    [Batch 100] loss: 0.347566  (6400/60000)
    [Batch 200] loss: 0.217738  (12800/60000)
    [Batch 300] loss: 0.371291  (19200/60000)
    [Batch 400] loss: 0.306699  (25600/60000)
    [Batch 500] loss: 0.355576  (32000/60000)
    [Batch 600] loss: 0.334537  (38400/60000)
    [Batch 700] loss: 0.467725  (44800/60000)
    [Batch 800] loss: 0.438962  (51200/60000)
    [Batch 900] loss: 0.368718  (57600/60000)
    [Batch 000] loss: 0.194095  (0/60000)
    [Batch 100] loss: 0.354282  (6400/60000)
    [Batch 200] loss: 0.235605  (12800/60000)
    [Batch 300] loss: 0.327707  (19200/60000)
    [Batch 400] loss: 0.307845  (25600/60000)
    [Batch 500] loss: 0.364978  (32000/60000)
    [Batch 600] loss: 0.339614  (38400/60000)
    [Batch 700] loss: 0.449399  (44800/60000)
    [Batch 800] loss: 0.427145  (51200/60000)
    [Batch 900] loss: 0.356140  (57600/60000)
  Epoch 21/30
    [Batch 000] loss: 0.220966  (0/60000)
    [Batch 100] loss: 0.339180  (6400/60000)
    [Batch 200] loss: 0.234624  (12800/60000)
    [Batch 300] loss: 0.373764  (19200/60000)
    [Batch 400] loss: 0.293952  (25600/60000)
    [Batch 500] loss: 0.364481  (32000/60000)
    [Batch 600] loss: 0.340861  (38400/60000)
    [Batch 700] loss: 0.467652  (44800/60000)
    [Batch 800] loss: 0.444332  (51200/60000)
    [Batch 900] loss: 0.378368  (57600/60000)
    [Batch 000] loss: 0.228741  (0/60000)
    [Batch 100] loss: 0.341532  (6400/60000)
    [Batch 200] loss: 0.214457  (12800/60000)
    [Batch 300] loss: 0.363502  (19200/60000)
    [Batch 400] loss: 0.304211  (25600/60000)
    [Batch 500] loss: 0.350934  (32000/60000)
    [Batch 600] loss: 0.329097  (38400/60000)
    [Batch 700] loss: 0.460092  (44800/60000)
    [Batch 800] loss: 0.430890  (51200/60000)
    [Batch 900] loss: 0.364498  (57600/60000)
    [Batch 000] loss: 0.191153  (0/60000)
    [Batch 100] loss: 0.346494  (6400/60000)
    [Batch 200] loss: 0.229873  (12800/60000)
    [Batch 300] loss: 0.317043  (19200/60000)
    [Batch 400] loss: 0.309145  (25600/60000)
    [Batch 500] loss: 0.362552  (32000/60000)
    [Batch 600] loss: 0.330304  (38400/60000)
    [Batch 700] loss: 0.436304  (44800/60000)
    [Batch 800] loss: 0.421669  (51200/60000)
    [Batch 900] loss: 0.353336  (57600/60000)
  Epoch 22/30
    [Batch 000] loss: 0.216622  (0/60000)
    [Batch 100] loss: 0.333148  (6400/60000)
    [Batch 200] loss: 0.231333  (12800/60000)
    [Batch 300] loss: 0.368171  (19200/60000)
    [Batch 400] loss: 0.289738  (25600/60000)
    [Batch 500] loss: 0.360260  (32000/60000)
    [Batch 600] loss: 0.335772  (38400/60000)
    [Batch 700] loss: 0.459905  (44800/60000)
    [Batch 800] loss: 0.436925  (51200/60000)
    [Batch 900] loss: 0.375184  (57600/60000)
    [Batch 000] loss: 0.226124  (0/60000)
    [Batch 100] loss: 0.335874  (6400/60000)
    [Batch 200] loss: 0.211172  (12800/60000)
    [Batch 300] loss: 0.356274  (19200/60000)
    [Batch 400] loss: 0.301353  (25600/60000)
    [Batch 500] loss: 0.346811  (32000/60000)
    [Batch 600] loss: 0.324410  (38400/60000)
    [Batch 700] loss: 0.452963  (44800/60000)
    [Batch 800] loss: 0.422781  (51200/60000)
    [Batch 900] loss: 0.360588  (57600/60000)
    [Batch 000] loss: 0.189589  (0/60000)
    [Batch 100] loss: 0.339370  (6400/60000)
    [Batch 200] loss: 0.222137  (12800/60000)
    [Batch 300] loss: 0.305536  (19200/60000)
    [Batch 400] loss: 0.308866  (25600/60000)
    [Batch 500] loss: 0.355544  (32000/60000)
    [Batch 600] loss: 0.325927  (38400/60000)
    [Batch 700] loss: 0.423145  (44800/60000)
    [Batch 800] loss: 0.416750  (51200/60000)
    [Batch 900] loss: 0.344521  (57600/60000)
  Epoch 23/30
    [Batch 000] loss: 0.213971  (0/60000)
    [Batch 100] loss: 0.328633  (6400/60000)
    [Batch 200] loss: 0.228318  (12800/60000)
    [Batch 300] loss: 0.363092  (19200/60000)
    [Batch 400] loss: 0.287277  (25600/60000)
    [Batch 500] loss: 0.356401  (32000/60000)
    [Batch 600] loss: 0.331522  (38400/60000)
    [Batch 700] loss: 0.452768  (44800/60000)
    [Batch 800] loss: 0.429878  (51200/60000)
    [Batch 900] loss: 0.372934  (57600/60000)
    [Batch 000] loss: 0.223037  (0/60000)
    [Batch 100] loss: 0.330683  (6400/60000)
    [Batch 200] loss: 0.208758  (12800/60000)
    [Batch 300] loss: 0.350164  (19200/60000)
    [Batch 400] loss: 0.299521  (25600/60000)
    [Batch 500] loss: 0.344355  (32000/60000)
    [Batch 600] loss: 0.320864  (38400/60000)
    [Batch 700] loss: 0.445319  (44800/60000)
    [Batch 800] loss: 0.413692  (51200/60000)
    [Batch 900] loss: 0.356833  (57600/60000)
    [Batch 000] loss: 0.190184  (0/60000)
    [Batch 100] loss: 0.333189  (6400/60000)
    [Batch 200] loss: 0.218318  (12800/60000)
    [Batch 300] loss: 0.296867  (19200/60000)
    [Batch 400] loss: 0.307127  (25600/60000)
    [Batch 500] loss: 0.352413  (32000/60000)
    [Batch 600] loss: 0.319598  (38400/60000)
    [Batch 700] loss: 0.410412  (44800/60000)
    [Batch 800] loss: 0.409589  (51200/60000)
    [Batch 900] loss: 0.340442  (57600/60000)
  Epoch 24/30
    [Batch 000] loss: 0.210583  (0/60000)
    [Batch 100] loss: 0.325450  (6400/60000)
    [Batch 200] loss: 0.225828  (12800/60000)
    [Batch 300] loss: 0.358572  (19200/60000)
    [Batch 400] loss: 0.283780  (25600/60000)
    [Batch 500] loss: 0.351790  (32000/60000)
    [Batch 600] loss: 0.329195  (38400/60000)
    [Batch 700] loss: 0.445259  (44800/60000)
    [Batch 800] loss: 0.421460  (51200/60000)
    [Batch 900] loss: 0.367853  (57600/60000)
    [Batch 000] loss: 0.219444  (0/60000)
    [Batch 100] loss: 0.325988  (6400/60000)
    [Batch 200] loss: 0.205869  (12800/60000)
    [Batch 300] loss: 0.343588  (19200/60000)
    [Batch 400] loss: 0.296785  (25600/60000)
    [Batch 500] loss: 0.341293  (32000/60000)
    [Batch 600] loss: 0.317897  (38400/60000)
    [Batch 700] loss: 0.437556  (44800/60000)
    [Batch 800] loss: 0.403789  (51200/60000)
    [Batch 900] loss: 0.353512  (57600/60000)
    [Batch 000] loss: 0.187532  (0/60000)
    [Batch 100] loss: 0.324868  (6400/60000)
    [Batch 200] loss: 0.211461  (12800/60000)
    [Batch 300] loss: 0.288531  (19200/60000)
    [Batch 400] loss: 0.306588  (25600/60000)
    [Batch 500] loss: 0.344261  (32000/60000)
    [Batch 600] loss: 0.307868  (38400/60000)
    [Batch 700] loss: 0.394668  (44800/60000)
    [Batch 800] loss: 0.402819  (51200/60000)
    [Batch 900] loss: 0.332711  (57600/60000)
  Epoch 25/30
    [Batch 000] loss: 0.206751  (0/60000)
    [Batch 100] loss: 0.320503  (6400/60000)
    [Batch 200] loss: 0.223800  (12800/60000)
    [Batch 300] loss: 0.353375  (19200/60000)
    [Batch 400] loss: 0.280779  (25600/60000)
    [Batch 500] loss: 0.346400  (32000/60000)
    [Batch 600] loss: 0.325915  (38400/60000)
    [Batch 700] loss: 0.438420  (44800/60000)
    [Batch 800] loss: 0.415959  (51200/60000)
    [Batch 900] loss: 0.365714  (57600/60000)
    [Batch 000] loss: 0.217307  (0/60000)
    [Batch 100] loss: 0.320902  (6400/60000)
    [Batch 200] loss: 0.202450  (12800/60000)
    [Batch 300] loss: 0.336983  (19200/60000)
    [Batch 400] loss: 0.294409  (25600/60000)
    [Batch 500] loss: 0.338067  (32000/60000)
    [Batch 600] loss: 0.313755  (38400/60000)
    [Batch 700] loss: 0.429227  (44800/60000)
    [Batch 800] loss: 0.395720  (51200/60000)
    [Batch 900] loss: 0.349621  (57600/60000)
    [Batch 000] loss: 0.184444  (0/60000)
    [Batch 100] loss: 0.317771  (6400/60000)
    [Batch 200] loss: 0.207196  (12800/60000)
    [Batch 300] loss: 0.280179  (19200/60000)
    [Batch 400] loss: 0.304012  (25600/60000)
    [Batch 500] loss: 0.341377  (32000/60000)
    [Batch 600] loss: 0.300014  (38400/60000)
    [Batch 700] loss: 0.381965  (44800/60000)
    [Batch 800] loss: 0.398766  (51200/60000)
    [Batch 900] loss: 0.321937  (57600/60000)
  Epoch 26/30
    [Batch 000] loss: 0.203791  (0/60000)
    [Batch 100] loss: 0.317371  (6400/60000)
    [Batch 200] loss: 0.222508  (12800/60000)
    [Batch 300] loss: 0.347494  (19200/60000)
    [Batch 400] loss: 0.279112  (25600/60000)
    [Batch 500] loss: 0.342247  (32000/60000)
    [Batch 600] loss: 0.320793  (38400/60000)
    [Batch 700] loss: 0.432712  (44800/60000)
    [Batch 800] loss: 0.409044  (51200/60000)
    [Batch 900] loss: 0.363120  (57600/60000)
    [Batch 000] loss: 0.215063  (0/60000)
    [Batch 100] loss: 0.317010  (6400/60000)
    [Batch 200] loss: 0.199054  (12800/60000)
    [Batch 300] loss: 0.330374  (19200/60000)
    [Batch 400] loss: 0.292398  (25600/60000)
    [Batch 500] loss: 0.334191  (32000/60000)
    [Batch 600] loss: 0.310670  (38400/60000)
    [Batch 700] loss: 0.422418  (44800/60000)
    [Batch 800] loss: 0.388994  (51200/60000)
    [Batch 900] loss: 0.346974  (57600/60000)
    [Batch 000] loss: 0.182784  (0/60000)
    [Batch 100] loss: 0.313222  (6400/60000)
    [Batch 200] loss: 0.205843  (12800/60000)
    [Batch 300] loss: 0.271697  (19200/60000)
    [Batch 400] loss: 0.301903  (25600/60000)
    [Batch 500] loss: 0.334777  (32000/60000)
    [Batch 600] loss: 0.295912  (38400/60000)
    [Batch 700] loss: 0.370287  (44800/60000)
    [Batch 800] loss: 0.390214  (51200/60000)
    [Batch 900] loss: 0.312501  (57600/60000)
  Epoch 27/30
    [Batch 000] loss: 0.200433  (0/60000)
    [Batch 100] loss: 0.314548  (6400/60000)
    [Batch 200] loss: 0.220303  (12800/60000)
    [Batch 300] loss: 0.341830  (19200/60000)
    [Batch 400] loss: 0.276551  (25600/60000)
    [Batch 500] loss: 0.338486  (32000/60000)
    [Batch 600] loss: 0.318069  (38400/60000)
    [Batch 700] loss: 0.426905  (44800/60000)
    [Batch 800] loss: 0.402979  (51200/60000)
    [Batch 900] loss: 0.359902  (57600/60000)
    [Batch 000] loss: 0.213556  (0/60000)
    [Batch 100] loss: 0.313072  (6400/60000)
    [Batch 200] loss: 0.196728  (12800/60000)
    [Batch 300] loss: 0.324680  (19200/60000)
    [Batch 400] loss: 0.290272  (25600/60000)
    [Batch 500] loss: 0.331877  (32000/60000)
    [Batch 600] loss: 0.307179  (38400/60000)
    [Batch 700] loss: 0.413393  (44800/60000)
    [Batch 800] loss: 0.381755  (51200/60000)
    [Batch 900] loss: 0.343521  (57600/60000)
    [Batch 000] loss: 0.179273  (0/60000)
    [Batch 100] loss: 0.304797  (6400/60000)
    [Batch 200] loss: 0.199669  (12800/60000)
    [Batch 300] loss: 0.264351  (19200/60000)
    [Batch 400] loss: 0.296730  (25600/60000)
    [Batch 500] loss: 0.331421  (32000/60000)
    [Batch 600] loss: 0.288423  (38400/60000)
    [Batch 700] loss: 0.357458  (44800/60000)
    [Batch 800] loss: 0.382226  (51200/60000)
    [Batch 900] loss: 0.307577  (57600/60000)
  Epoch 28/30
    [Batch 000] loss: 0.199421  (0/60000)
    [Batch 100] loss: 0.311158  (6400/60000)
    [Batch 200] loss: 0.218740  (12800/60000)
    [Batch 300] loss: 0.336579  (19200/60000)
    [Batch 400] loss: 0.274323  (25600/60000)
    [Batch 500] loss: 0.334923  (32000/60000)
    [Batch 600] loss: 0.314382  (38400/60000)
    [Batch 700] loss: 0.419785  (44800/60000)
    [Batch 800] loss: 0.396195  (51200/60000)
    [Batch 900] loss: 0.357717  (57600/60000)
    [Batch 000] loss: 0.211488  (0/60000)
    [Batch 100] loss: 0.309261  (6400/60000)
    [Batch 200] loss: 0.193447  (12800/60000)
    [Batch 300] loss: 0.318996  (19200/60000)
    [Batch 400] loss: 0.288461  (25600/60000)
    [Batch 500] loss: 0.329003  (32000/60000)
    [Batch 600] loss: 0.304736  (38400/60000)
    [Batch 700] loss: 0.406126  (44800/60000)
    [Batch 800] loss: 0.374846  (51200/60000)
    [Batch 900] loss: 0.340475  (57600/60000)
    [Batch 000] loss: 0.180265  (0/60000)
    [Batch 100] loss: 0.296190  (6400/60000)
    [Batch 200] loss: 0.198631  (12800/60000)
    [Batch 300] loss: 0.253714  (19200/60000)
    [Batch 400] loss: 0.299999  (25600/60000)
    [Batch 500] loss: 0.329096  (32000/60000)
    [Batch 600] loss: 0.281578  (38400/60000)
    [Batch 700] loss: 0.343718  (44800/60000)
    [Batch 800] loss: 0.376446  (51200/60000)
    [Batch 900] loss: 0.297859  (57600/60000)
  Epoch 29/30
    [Batch 000] loss: 0.198328  (0/60000)
    [Batch 100] loss: 0.309532  (6400/60000)
    [Batch 200] loss: 0.216126  (12800/60000)
    [Batch 300] loss: 0.331542  (19200/60000)
    [Batch 400] loss: 0.271712  (25600/60000)
    [Batch 500] loss: 0.331752  (32000/60000)
    [Batch 600] loss: 0.310407  (38400/60000)
    [Batch 700] loss: 0.414896  (44800/60000)
    [Batch 800] loss: 0.390860  (51200/60000)
    [Batch 900] loss: 0.352945  (57600/60000)
    [Batch 000] loss: 0.209034  (0/60000)
    [Batch 100] loss: 0.305093  (6400/60000)
    [Batch 200] loss: 0.190233  (12800/60000)
    [Batch 300] loss: 0.313034  (19200/60000)
    [Batch 400] loss: 0.287236  (25600/60000)
    [Batch 500] loss: 0.325966  (32000/60000)
    [Batch 600] loss: 0.300108  (38400/60000)
    [Batch 700] loss: 0.397039  (44800/60000)
    [Batch 800] loss: 0.367362  (51200/60000)
    [Batch 900] loss: 0.337198  (57600/60000)
    [Batch 000] loss: 0.175998  (0/60000)
    [Batch 100] loss: 0.289461  (6400/60000)
    [Batch 200] loss: 0.196046  (12800/60000)
    [Batch 300] loss: 0.247812  (19200/60000)
    [Batch 400] loss: 0.299479  (25600/60000)
    [Batch 500] loss: 0.324082  (32000/60000)
    [Batch 600] loss: 0.275913  (38400/60000)
    [Batch 700] loss: 0.330215  (44800/60000)
    [Batch 800] loss: 0.368330  (51200/60000)
    [Batch 900] loss: 0.288209  (57600/60000)
  Epoch 30/30
```

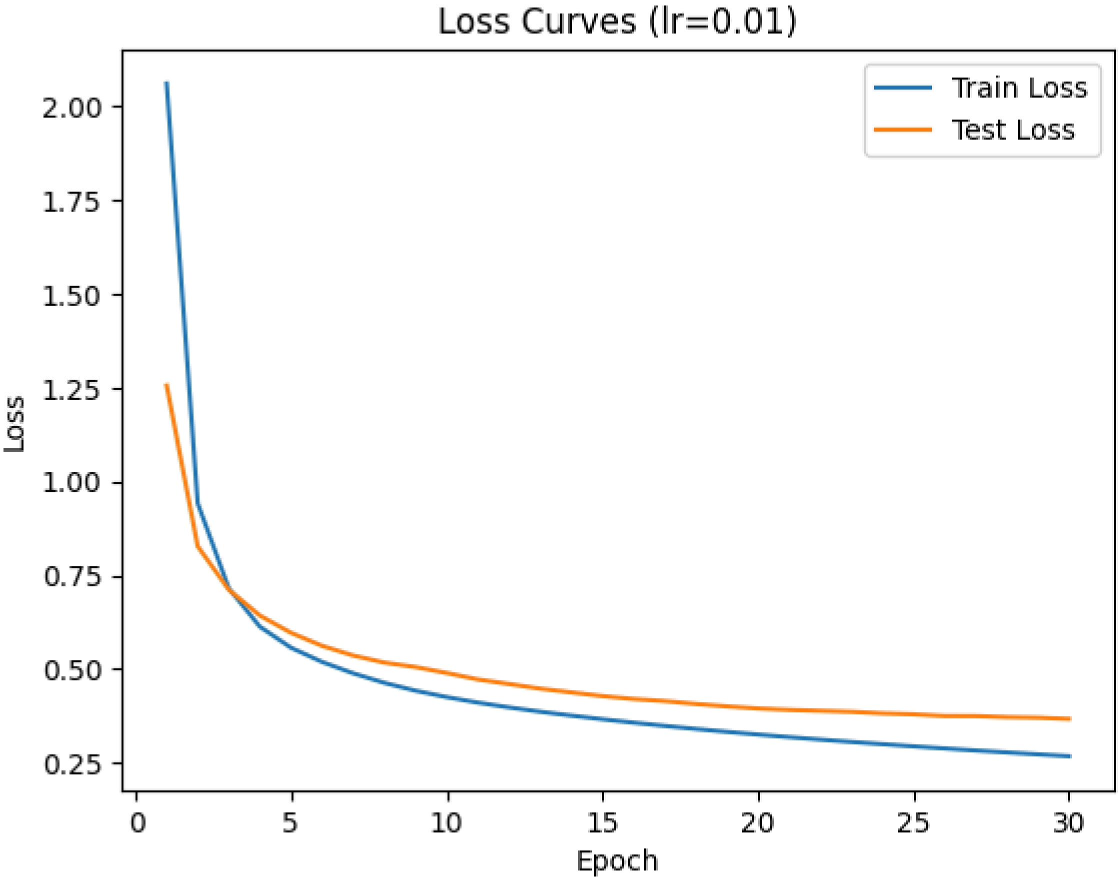
```
  [Batch 000] loss: 0.195287  (0/60000)
  [Batch 100] loss: 0.307835  (6400/60000)
  [Batch 200] loss: 0.213825  (12800/60000)
  [Batch 300] loss: 0.327355  (19200/60000)
  [Batch 400] loss: 0.270185  (25600/60000)
  [Batch 500] loss: 0.327925  (32000/60000)
  [Batch 600] loss: 0.306585  (38400/60000)
  [Batch 700] loss: 0.410113  (44800/60000)
  [Batch 800] loss: 0.385182  (51200/60000)
  [Batch 900] loss: 0.350782  (57600/60000)
  [Batch 000] loss: 0.207048  (0/60000)
  [Batch 100] loss: 0.300026  (6400/60000)
  [Batch 200] loss: 0.187261  (12800/60000)
  [Batch 300] loss: 0.306284  (19200/60000)
  [Batch 400] loss: 0.287327  (25600/60000)
  [Batch 500] loss: 0.323053  (32000/60000)
  [Batch 600] loss: 0.295005  (38400/60000)
  [Batch 700] loss: 0.388972  (44800/60000)
  [Batch 800] loss: 0.361345  (51200/60000)
  [Batch 900] loss: 0.333708  (57600/60000)
  [Batch 000] loss: 0.177100  (0/60000)
  [Batch 100] loss: 0.280042  (6400/60000)
  [Batch 200] loss: 0.191660  (12800/60000)
  [Batch 300] loss: 0.240354  (19200/60000)
  [Batch 400] loss: 0.294749  (25600/60000)
  [Batch 500] loss: 0.320323  (32000/60000)
  [Batch 600] loss: 0.267108  (38400/60000)
  [Batch 700] loss: 0.320850  (44800/60000)
  [Batch 800] loss: 0.356857  (51200/60000)
  [Batch 900] loss: 0.288631  (57600/60000)
Base net
Best accuracy = 86.87% at epoch 30
```



```
Wide net
Best accuracy = 86.98% at epoch 30
```



```
Deep net
Best accuracy = 86.60% at epoch 30
```



```
Final test accuracies:
Base net: 86.87%
Wide net: 86.98%
Deep net: 86.60%
```

Q1.6

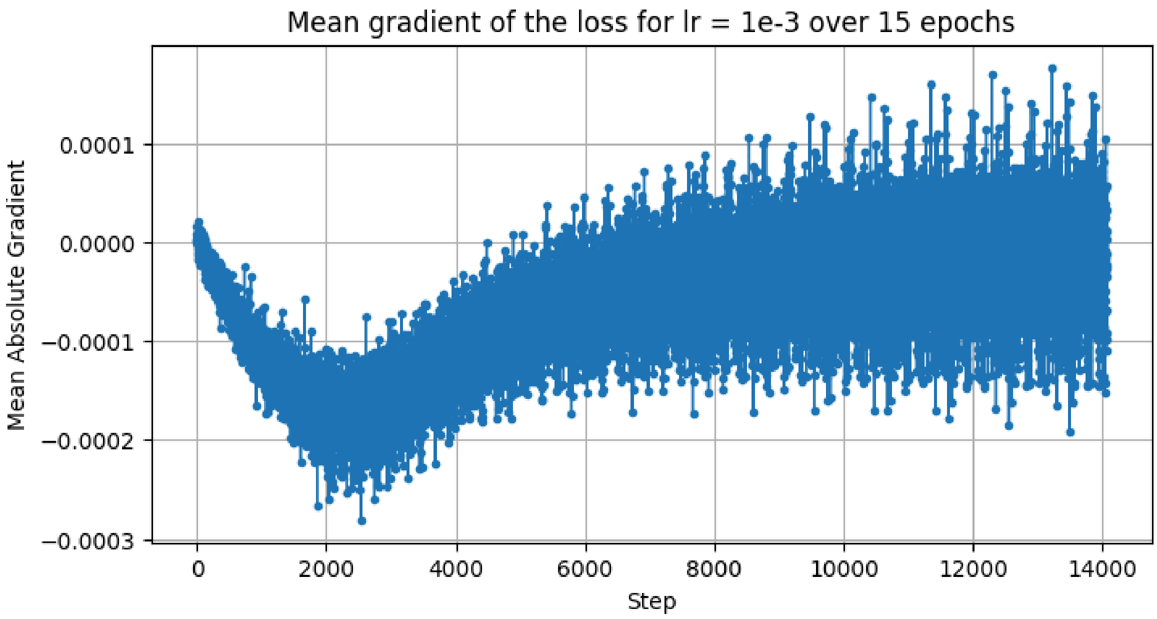
In [20]:

```
from numpy import mean

def train_with_grads(dataloader, model, loss_fn, optimizer):
    model.train()
    mean_gradients = []
    total = len(dataloader.dataset)

    for batch_idx, (X, y) in enumerate(dataloader):

        X, y = X.to(device), y.to(device)

        optimizer.zero_grad()
        preds = model(X)
        loss = loss_fn(preds, y)
        loss.backward()
        optimizer.step()

        # collect all grads into one vector, compute its mean
        grads = torch.cat([p.grad.view(-1)
                           for p in model.parameters() if p.grad is not N
        mean_gradients.append(grads.mean().item())

        if batch_idx % 500 == 0:
            seen = batch_idx * dataloader.batch_size
            print(f"  [Batch {batch_idx:03d}] mean grad: {mean_gradients[

    return mean_gradients
```

In [22]:

```
model = NeuralNetwork().to(device)
optimizer = torch.optim.SGD(model.parameters(), lr=1e-3)
loss_fn = nn.CrossEntropyLoss()

mean_grads = []
for epoch in range(15):
    print(f"Epoch {epoch+1}/15")
    epoch_mean_grads = train_with_grads(train_dataloader, model, loss_fn,
    mean_grads.extend(epoch_mean_grads)

plt.figure(figsize=(8,4))
plt.plot(mean_grads, marker='.', linewidth=1)
plt.title("Mean gradient of the loss for lr = 1e-3 over 15 epochs")
plt.xlabel("Step")
plt.ylabel("Mean Absolute Gradient")
plt.grid(True)
plt.show()
```

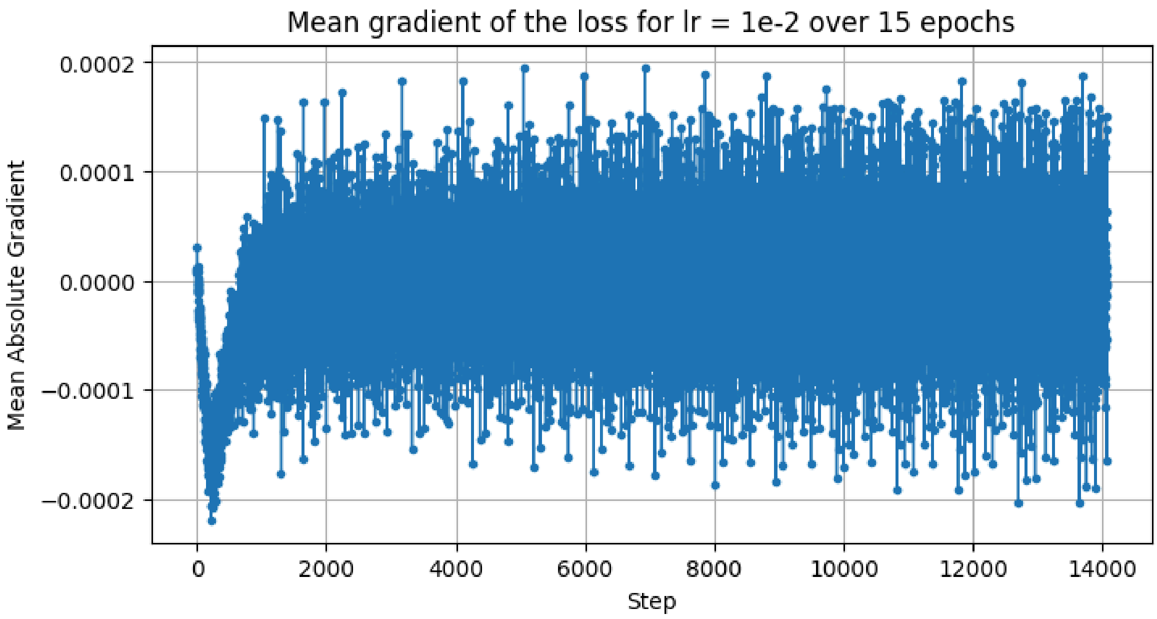
```
Epoch 1/15
  [Batch 000] mean grad: -0.000001  (0/60000)
  [Batch 500] mean grad: -0.000088  (32000/60000)
Epoch 2/15
  [Batch 000] mean grad: -0.000103  (0/60000)
  [Batch 500] mean grad: -0.000182  (32000/60000)
Epoch 3/15
  [Batch 000] mean grad: -0.000130  (0/60000)
  [Batch 500] mean grad: -0.000197  (32000/60000)
Epoch 4/15
  [Batch 000] mean grad: -0.000113  (0/60000)
  [Batch 500] mean grad: -0.000143  (32000/60000)
Epoch 5/15
  [Batch 000] mean grad: -0.000093  (0/60000)
  [Batch 500] mean grad: -0.000098  (32000/60000)
Epoch 6/15
  [Batch 000] mean grad: -0.000081  (0/60000)
  [Batch 500] mean grad: -0.000077  (32000/60000)
Epoch 7/15
  [Batch 000] mean grad: -0.000071  (0/60000)
  [Batch 500] mean grad: -0.000063  (32000/60000)
Epoch 8/15
  [Batch 000] mean grad: -0.000065  (0/60000)
  [Batch 500] mean grad: -0.000050  (32000/60000)
Epoch 9/15
  [Batch 000] mean grad: -0.000059  (0/60000)
  [Batch 500] mean grad: -0.000041  (32000/60000)
Epoch 10/15
  [Batch 000] mean grad: -0.000056  (0/60000)
  [Batch 500] mean grad: -0.000036  (32000/60000)
Epoch 11/15
  [Batch 000] mean grad: -0.000053  (0/60000)
  [Batch 500] mean grad: -0.000029  (32000/60000)
Epoch 12/15
  [Batch 000] mean grad: -0.000051  (0/60000)
  [Batch 500] mean grad: -0.000025  (32000/60000)
Epoch 13/15
  [Batch 000] mean grad: -0.000051  (0/60000)
  [Batch 500] mean grad: -0.000022  (32000/60000)
Epoch 14/15
  [Batch 000] mean grad: -0.000050  (0/60000)
  [Batch 500] mean grad: -0.000016  (32000/60000)
Epoch 15/15
  [Batch 000] mean grad: -0.000047  (0/60000)
  [Batch 500] mean grad: -0.000010  (32000/60000)
```



In [23]:

```
model = NeuralNetwork().to(device)
optimizer = torch.optim.SGD(model.parameters(), lr=1e-2)
loss_fn = nn.CrossEntropyLoss()

mean_grads = []
for epoch in range(15):
    print(f"Epoch {epoch+1}/15")
    epoch_mean_grads = train_with_grads(train_dataloader, model, loss_fn,
    mean_grads.extend(epoch_mean_grads)

plt.figure(figsize=(8,4))
plt.plot(mean_grads, marker='.', linewidth=1)
plt.title("Mean gradient of the loss for lr = 1e-2 over 15 epochs")
plt.xlabel("Step")
plt.ylabel("Mean Absolute Gradient")
plt.grid(True)
plt.show()
```

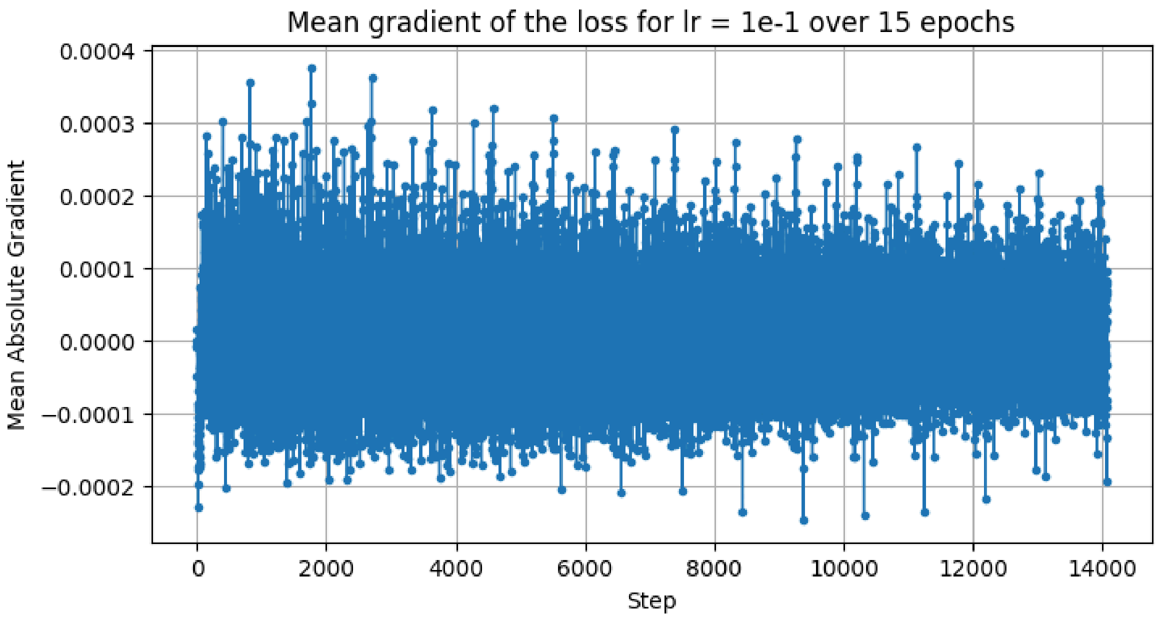
```
Epoch 1/15
  [Batch 000] mean grad: 0.000011  (0/60000)
  [Batch 500] mean grad: -0.000060  (32000/60000)
Epoch 2/15
  [Batch 000] mean grad: -0.000036  (0/60000)
  [Batch 500] mean grad: 0.000034  (32000/60000)
Epoch 3/15
  [Batch 000] mean grad: -0.000000  (0/60000)
  [Batch 500] mean grad: 0.000056  (32000/60000)
Epoch 4/15
  [Batch 000] mean grad: 0.000023  (0/60000)
  [Batch 500] mean grad: 0.000069  (32000/60000)
Epoch 5/15
  [Batch 000] mean grad: 0.000024  (0/60000)
  [Batch 500] mean grad: 0.000074  (32000/60000)
Epoch 6/15
  [Batch 000] mean grad: 0.000017  (0/60000)
  [Batch 500] mean grad: 0.000084  (32000/60000)
Epoch 7/15
  [Batch 000] mean grad: 0.000011  (0/60000)
  [Batch 500] mean grad: 0.000101  (32000/60000)
Epoch 8/15
  [Batch 000] mean grad: 0.000005  (0/60000)
  [Batch 500] mean grad: 0.000105  (32000/60000)
Epoch 9/15
  [Batch 000] mean grad: -0.000004  (0/60000)
  [Batch 500] mean grad: 0.000119  (32000/60000)
Epoch 10/15
  [Batch 000] mean grad: -0.000010  (0/60000)
  [Batch 500] mean grad: 0.000114  (32000/60000)
Epoch 11/15
  [Batch 000] mean grad: -0.000021  (0/60000)
  [Batch 500] mean grad: 0.000112  (32000/60000)
Epoch 12/15
  [Batch 000] mean grad: -0.000030  (0/60000)
  [Batch 500] mean grad: 0.000111  (32000/60000)
Epoch 13/15
  [Batch 000] mean grad: -0.000029  (0/60000)
  [Batch 500] mean grad: 0.000111  (32000/60000)
Epoch 14/15
  [Batch 000] mean grad: -0.000033  (0/60000)
  [Batch 500] mean grad: 0.000102  (32000/60000)
Epoch 15/15
  [Batch 000] mean grad: -0.000042  (0/60000)
  [Batch 500] mean grad: 0.000111  (32000/60000)
```



In [24]:

```
model = NeuralNetwork().to(device)
optimizer = torch.optim.SGD(model.parameters(), lr=1e-1)
loss_fn = nn.CrossEntropyLoss()

mean_grads = []
for epoch in range(15):
    print(f"Epoch {epoch+1}/15")
    epoch_mean_grads = train_with_grads(train_dataloader, model, loss_fn,
    mean_grads.extend(epoch_mean_grads)

# Plot first 100 steps
plt.figure(figsize=(8,4))
plt.plot(mean_grads, marker='.', linewidth=1)
plt.title("Mean gradient of the loss for lr = 1e-1 over 15 epochs")
plt.xlabel("Step")
plt.ylabel("Mean Absolute Gradient")
plt.grid(True)
plt.show()
```

```
Epoch 1/15
  [Batch 000] mean grad: 0.000016  (0/60000)
  [Batch 500] mean grad: -0.000004  (32000/60000)
Epoch 2/15
  [Batch 000] mean grad: -0.000119  (0/60000)
  [Batch 500] mean grad: 0.000070  (32000/60000)
Epoch 3/15
  [Batch 000] mean grad: -0.000100  (0/60000)
  [Batch 500] mean grad: 0.000086  (32000/60000)
Epoch 4/15
  [Batch 000] mean grad: -0.000083  (0/60000)
  [Batch 500] mean grad: 0.000104  (32000/60000)
Epoch 5/15
  [Batch 000] mean grad: -0.000079  (0/60000)
  [Batch 500] mean grad: 0.000103  (32000/60000)
Epoch 6/15
  [Batch 000] mean grad: -0.000063  (0/60000)
  [Batch 500] mean grad: 0.000114  (32000/60000)
Epoch 7/15
  [Batch 000] mean grad: -0.000071  (0/60000)
  [Batch 500] mean grad: 0.000120  (32000/60000)
Epoch 8/15
  [Batch 000] mean grad: -0.000079  (0/60000)
  [Batch 500] mean grad: 0.000096  (32000/60000)
Epoch 9/15
  [Batch 000] mean grad: -0.000073  (0/60000)
  [Batch 500] mean grad: 0.000093  (32000/60000)
Epoch 10/15
  [Batch 000] mean grad: -0.000073  (0/60000)
  [Batch 500] mean grad: 0.000086  (32000/60000)
Epoch 11/15
  [Batch 000] mean grad: -0.000066  (0/60000)
  [Batch 500] mean grad: 0.000078  (32000/60000)
Epoch 12/15
  [Batch 000] mean grad: -0.000073  (0/60000)
  [Batch 500] mean grad: 0.000086  (32000/60000)
Epoch 13/15
  [Batch 000] mean grad: -0.000053  (0/60000)
  [Batch 500] mean grad: 0.000053  (32000/60000)
Epoch 14/15
  [Batch 000] mean grad: -0.000040  (0/60000)
  [Batch 500] mean grad: 0.000071  (32000/60000)
Epoch 15/15
  [Batch 000] mean grad: -0.000045  (0/60000)
  [Batch 500] mean grad: 0.000046  (32000/60000)
```



Q1.7

In [11]:

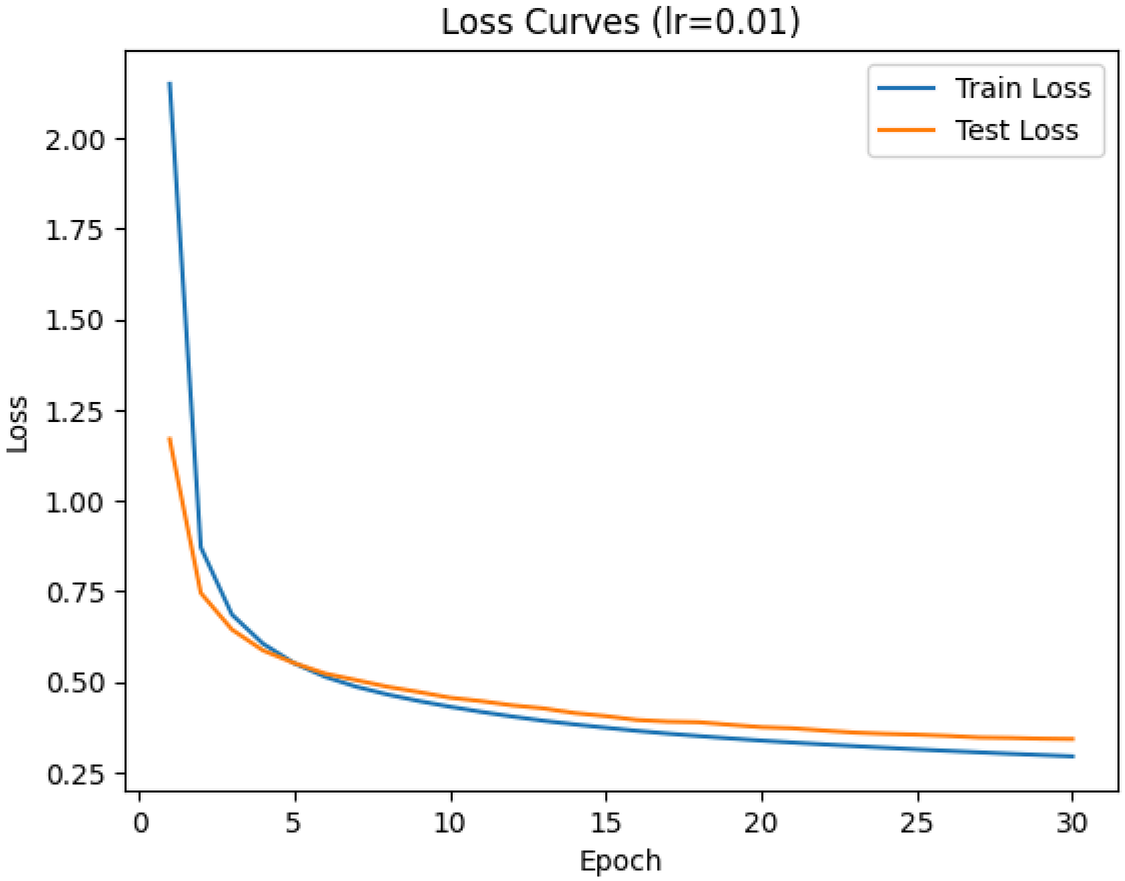
```
class CNN(nn.Module):
    def __init__(self):
        super().__init__()
        self.convolution_stack = nn.Sequential(
            nn.Conv2d(1, 6, kernel_size=5),
            nn.ReLU(),
            nn.MaxPool2d(2,2),
            nn.Conv2d(6, 16, kernel_size=5),
            nn.ReLU(),
            nn.MaxPool2d(2,2),
        )
        self.flatten = nn.Flatten()
        self.linear_stack = nn.Sequential(
            nn.Linear(256, 128),
            nn.ReLU(),
            nn.Linear(128,64),
            nn.ReLU(),
            nn.Linear(64, 10)
        )

    def forward(self, x):
        x = self.convolution_stack(x)
        x = self.flatten(x)
        x = self.linear_stack(x)
        return x


cnn_model = CNN().to(device)
print("Total parameters: ", count_params(cnn_model))
```

```
Total parameters:  44374
```

In [104…

```
# training set up
epochs = 30
lr = 1e-2
optimizer = torch.optim.SGD(params=cnn_model.parameters(), lr=lr)
criterion = nn.CrossEntropyLoss()
```

```python
# training loop
train_losses, test_losses, test_accs = [], [], []
for epoch in range(1, epochs + 1):
    print(f"Epoch {epoch}/{epochs}")
    train_loss = train(train_dataloader, cnn_model, loss_fn, optimizer)
    acc, test_loss = test(test_dataloader, cnn_model, loss_fn)

    train_losses.append(train_loss)
    test_losses.append(test_loss)
    test_accs.append(acc)

# plot results
plot_train_test_loss(train_losses, test_losses, test_accs, learning_rate)
result = test_accs[-1]

print(f"Final test accuracy for CNN model at LR={learning_rate}: {result*
Epoch 1/30
  [Batch 000] loss: 2.319222  (0/60000)
  [Batch 100] loss: 2.320369  (6400/60000)
  [Batch 200] loss: 2.308376  (12800/60000)
  [Batch 300] loss: 2.290546  (19200/60000)
  [Batch 400] loss: 2.297048  (25600/60000)
  [Batch 500] loss: 2.277369  (32000/60000)
  [Batch 600] loss: 2.258766  (38400/60000)
  [Batch 700] loss: 2.205157  (44800/60000)
  [Batch 800] loss: 1.993607  (51200/60000)
  [Batch 900] loss: 1.307252  (57600/60000)
Epoch 2/30
  [Batch 000] loss: 1.247804  (0/60000)
  [Batch 100] loss: 1.097389  (6400/60000)
  [Batch 200] loss: 0.823539  (12800/60000)
  [Batch 300] loss: 0.989792  (19200/60000)
  [Batch 400] loss: 0.814363  (25600/60000)
  [Batch 500] loss: 0.846072  (32000/60000)
  [Batch 600] loss: 0.830532  (38400/60000)
  [Batch 700] loss: 0.811034  (44800/60000)
  [Batch 800] loss: 0.703085  (51200/60000)
  [Batch 900] loss: 0.782062  (57600/60000)
Epoch 3/30
  [Batch 000] loss: 0.710415  (0/60000)
  [Batch 100] loss: 0.775468  (6400/60000)
  [Batch 200] loss: 0.528587  (12800/60000)
  [Batch 300] loss: 0.762045  (19200/60000)
  [Batch 400] loss: 0.630107  (25600/60000)
  [Batch 500] loss: 0.728308  (32000/60000)
  [Batch 600] loss: 0.721376  (38400/60000)
  [Batch 700] loss: 0.652401  (44800/60000)
  [Batch 800] loss: 0.614842  (51200/60000)
  [Batch 900] loss: 0.680980  (57600/60000)
Epoch 4/30
  [Batch 000] loss: 0.587980  (0/60000)
  [Batch 100] loss: 0.656794  (6400/60000)
  [Batch 200] loss: 0.442876  (12800/60000)
  [Batch 300] loss: 0.656523  (19200/60000)
  [Batch 400] loss: 0.573771  (25600/60000)
  [Batch 500] loss: 0.642893  (32000/60000)
  [Batch 600] loss: 0.635159  (38400/60000)
  [Batch 700] loss: 0.575739  (44800/60000)
  [Batch 800] loss: 0.576712  (51200/60000)
  [Batch 900] loss: 0.600152  (57600/60000)
Epoch 5/30
  [Batch 000] loss: 0.528091  (0/60000)
  [Batch 100] loss: 0.608629  (6400/60000)
  [Batch 200] loss: 0.396014  (12800/60000)
  [Batch 300] loss: 0.570760  (19200/60000)
  [Batch 400] loss: 0.552719  (25600/60000)
  [Batch 500] loss: 0.577094  (32000/60000)
  [Batch 600] loss: 0.576011  (38400/60000)
  [Batch 700] loss: 0.544210  (44800/60000)
  [Batch 800] loss: 0.546069  (51200/60000)
  [Batch 900] loss: 0.539704  (57600/60000)
Epoch 6/30
  [Batch 000] loss: 0.492761  (0/60000)
  [Batch 100] loss: 0.586027  (6400/60000)
  [Batch 200] loss: 0.355123  (12800/60000)
  [Batch 300] loss: 0.515460  (19200/60000)
  [Batch 400] loss: 0.516774  (25600/60000)
  [Batch 500] loss: 0.538527  (32000/60000)
  [Batch 600] loss: 0.527394  (38400/60000)
  [Batch 700] loss: 0.530946  (44800/60000)
  [Batch 800] loss: 0.530039  (51200/60000)
  [Batch 900] loss: 0.490035  (57600/60000)
Epoch 7/30
  [Batch 000] loss: 0.453584  (0/60000)
  [Batch 100] loss: 0.562093  (6400/60000)
  [Batch 200] loss: 0.327724  (12800/60000)
  [Batch 300] loss: 0.479121  (19200/60000)
  [Batch 400] loss: 0.505713  (25600/60000)
  [Batch 500] loss: 0.513929  (32000/60000)
  [Batch 600] loss: 0.493995  (38400/60000)
  [Batch 700] loss: 0.537878  (44800/60000)
  [Batch 800] loss: 0.522351  (51200/60000)
  [Batch 900] loss: 0.453483  (57600/60000)
Epoch 8/30
  [Batch 000] loss: 0.424394  (0/60000)
  [Batch 100] loss: 0.541648  (6400/60000)
  [Batch 200] loss: 0.310947  (12800/60000)
  [Batch 300] loss: 0.455334  (19200/60000)
  [Batch 400] loss: 0.489063  (25600/60000)
  [Batch 500] loss: 0.503141  (32000/60000)
  [Batch 600] loss: 0.464992  (38400/60000)
  [Batch 700] loss: 0.544187  (44800/60000)
  [Batch 800] loss: 0.524266  (51200/60000)
  [Batch 900] loss: 0.429380  (57600/60000)
Epoch 9/30
  [Batch 000] loss: 0.401083  (0/60000)
  [Batch 100] loss: 0.512739  (6400/60000)
  [Batch 200] loss: 0.297962  (12800/60000)
  [Batch 300] loss: 0.432300  (19200/60000)
  [Batch 400] loss: 0.468625  (25600/60000)
  [Batch 500] loss: 0.490959  (32000/60000)
  [Batch 600] loss: 0.445620  (38400/60000)
  [Batch 700] loss: 0.548143  (44800/60000)
  [Batch 800] loss: 0.527113  (51200/60000)
  [Batch 900] loss: 0.415424  (57600/60000)
Epoch 10/30
  [Batch 000] loss: 0.381062  (0/60000)
  [Batch 100] loss: 0.486434  (6400/60000)
  [Batch 200] loss: 0.284131  (12800/60000)
  [Batch 300] loss: 0.412992  (19200/60000)
  [Batch 400] loss: 0.459123  (25600/60000)
  [Batch 500] loss: 0.475113  (32000/60000)
  [Batch 600] loss: 0.423050  (38400/60000)
  [Batch 700] loss: 0.550685  (44800/60000)
  [Batch 800] loss: 0.529984  (51200/60000)
  [Batch 900] loss: 0.405610  (57600/60000)
Epoch 11/30
  [Batch 000] loss: 0.358179  (0/60000)
  [Batch 100] loss: 0.464353  (6400/60000)
  [Batch 200] loss: 0.268642  (12800/60000)
  [Batch 300] loss: 0.399266  (19200/60000)
  [Batch 400] loss: 0.447958  (25600/60000)
  [Batch 500] loss: 0.454218  (32000/60000)
  [Batch 600] loss: 0.400921  (38400/60000)
  [Batch 700] loss: 0.551483  (44800/60000)
  [Batch 800] loss: 0.531531  (51200/60000)
  [Batch 900] loss: 0.403281  (57600/60000)
Epoch 12/30
  [Batch 000] loss: 0.343810  (0/60000)
  [Batch 100] loss: 0.445155  (6400/60000)
  [Batch 200] loss: 0.251689  (12800/60000)
  [Batch 300] loss: 0.389077  (19200/60000)
  [Batch 400] loss: 0.439276  (25600/60000)
  [Batch 500] loss: 0.438760  (32000/60000)
  [Batch 600] loss: 0.380109  (38400/60000)
  [Batch 700] loss: 0.534504  (44800/60000)
  [Batch 800] loss: 0.543729  (51200/60000)
  [Batch 900] loss: 0.403333  (57600/60000)
Epoch 13/30
  [Batch 000] loss: 0.326146  (0/60000)
  [Batch 100] loss: 0.424069  (6400/60000)
  [Batch 200] loss: 0.244144  (12800/60000)
  [Batch 300] loss: 0.382074  (19200/60000)
  [Batch 400] loss: 0.430648  (25600/60000)
  [Batch 500] loss: 0.416456  (32000/60000)
  [Batch 600] loss: 0.367318  (38400/60000)
  [Batch 700] loss: 0.543571  (44800/60000)
  [Batch 800] loss: 0.554549  (51200/60000)
  [Batch 900] loss: 0.399432  (57600/60000)
Epoch 14/30
  [Batch 000] loss: 0.311803  (0/60000)
  [Batch 100] loss: 0.408274  (6400/60000)
  [Batch 200] loss: 0.241687  (12800/60000)
  [Batch 300] loss: 0.374899  (19200/60000)
  [Batch 400] loss: 0.409446  (25600/60000)
  [Batch 500] loss: 0.406656  (32000/60000)
  [Batch 600] loss: 0.346551  (38400/60000)
  [Batch 700] loss: 0.534602  (44800/60000)
  [Batch 800] loss: 0.556179  (51200/60000)
  [Batch 900] loss: 0.397943  (57600/60000)
Epoch 15/30
  [Batch 000] loss: 0.296221  (0/60000)
  [Batch 100] loss: 0.391719  (6400/60000)
  [Batch 200] loss: 0.239603  (12800/60000)
  [Batch 300] loss: 0.370896  (19200/60000)
  [Batch 400] loss: 0.403158  (25600/60000)
  [Batch 500] loss: 0.397036  (32000/60000)
  [Batch 600] loss: 0.335745  (38400/60000)
  [Batch 700] loss: 0.532091  (44800/60000)
  [Batch 800] loss: 0.567159  (51200/60000)
  [Batch 900] loss: 0.402577  (57600/60000)
Epoch 16/30
  [Batch 000] loss: 0.292036  (0/60000)
  [Batch 100] loss: 0.384559  (6400/60000)
  [Batch 200] loss: 0.235244  (12800/60000)
  [Batch 300] loss: 0.373531  (19200/60000)
  [Batch 400] loss: 0.405045  (25600/60000)
  [Batch 500] loss: 0.392910  (32000/60000)
  [Batch 600] loss: 0.321157  (38400/60000)
  [Batch 700] loss: 0.524166  (44800/60000)
  [Batch 800] loss: 0.568631  (51200/60000)
  [Batch 900] loss: 0.401264  (57600/60000)
Epoch 17/30
  [Batch 000] loss: 0.278507  (0/60000)
  [Batch 100] loss: 0.373271  (6400/60000)
  [Batch 200] loss: 0.233270  (12800/60000)
  [Batch 300] loss: 0.372483  (19200/60000)
  [Batch 400] loss: 0.413072  (25600/60000)
  [Batch 500] loss: 0.387517  (32000/60000)
  [Batch 600] loss: 0.310293  (38400/60000)
  [Batch 700] loss: 0.517802  (44800/60000)
  [Batch 800] loss: 0.567771  (51200/60000)
  [Batch 900] loss: 0.399536  (57600/60000)
Epoch 18/30
  [Batch 000] loss: 0.272545  (0/60000)
  [Batch 100] loss: 0.360583  (6400/60000)
  [Batch 200] loss: 0.233138  (12800/60000)
  [Batch 300] loss: 0.369502  (19200/60000)
  [Batch 400] loss: 0.373831  (25600/60000)
  [Batch 500] loss: 0.383820  (32000/60000)
  [Batch 600] loss: 0.299411  (38400/60000)
  [Batch 700] loss: 0.509262  (44800/60000)
  [Batch 800] loss: 0.561757  (51200/60000)
  [Batch 900] loss: 0.406090  (57600/60000)
Epoch 19/30
  [Batch 000] loss: 0.268625  (0/60000)
  [Batch 100] loss: 0.359025  (6400/60000)
  [Batch 200] loss: 0.239913  (12800/60000)
  [Batch 300] loss: 0.362961  (19200/60000)
  [Batch 400] loss: 0.378334  (25600/60000)
  [Batch 500] loss: 0.382208  (32000/60000)
  [Batch 600] loss: 0.293880  (38400/60000)
  [Batch 700] loss: 0.505597  (44800/60000)
  [Batch 800] loss: 0.556000  (51200/60000)
  [Batch 900] loss: 0.398336  (57600/60000)
Epoch 20/30
  [Batch 000] loss: 0.259312  (0/60000)
  [Batch 100] loss: 0.340185  (6400/60000)
  [Batch 200] loss: 0.227159  (12800/60000)
  [Batch 300] loss: 0.359809  (19200/60000)
  [Batch 400] loss: 0.368325  (25600/60000)
  [Batch 500] loss: 0.380791  (32000/60000)
  [Batch 600] loss: 0.287028  (38400/60000)
  [Batch 700] loss: 0.488432  (44800/60000)
  [Batch 800] loss: 0.550328  (51200/60000)
  [Batch 900] loss: 0.397184  (57600/60000)
Epoch 21/30
  [Batch 000] loss: 0.256088  (0/60000)
  [Batch 100] loss: 0.331409  (6400/60000)
  [Batch 200] loss: 0.228831  (12800/60000)
  [Batch 300] loss: 0.353396  (19200/60000)
  [Batch 400] loss: 0.361078  (25600/60000)
  [Batch 500] loss: 0.380689  (32000/60000)
  [Batch 600] loss: 0.286759  (38400/60000)
  [Batch 700] loss: 0.481254  (44800/60000)
  [Batch 800] loss: 0.549990  (51200/60000)
  [Batch 900] loss: 0.396965  (57600/60000)
Epoch 22/30
  [Batch 000] loss: 0.252295  (0/60000)
  [Batch 100] loss: 0.326514  (6400/60000)
  [Batch 200] loss: 0.222769  (12800/60000)
  [Batch 300] loss: 0.342288  (19200/60000)
  [Batch 400] loss: 0.348831  (25600/60000)
  [Batch 500] loss: 0.374597  (32000/60000)
  [Batch 600] loss: 0.274735  (38400/60000)
  [Batch 700] loss: 0.472798  (44800/60000)
  [Batch 800] loss: 0.540620  (51200/60000)
  [Batch 900] loss: 0.394164  (57600/60000)
Epoch 23/30
  [Batch 000] loss: 0.243192  (0/60000)
  [Batch 100] loss: 0.317266  (6400/60000)
  [Batch 200] loss: 0.221263  (12800/60000)
  [Batch 300] loss: 0.339551  (19200/60000)
  [Batch 400] loss: 0.346962  (25600/60000)
  [Batch 500] loss: 0.374917  (32000/60000)
  [Batch 600] loss: 0.275654  (38400/60000)
  [Batch 700] loss: 0.460218  (44800/60000)
  [Batch 800] loss: 0.538109  (51200/60000)
  [Batch 900] loss: 0.390615  (57600/60000)
Epoch 24/30
  [Batch 000] loss: 0.236268  (0/60000)
  [Batch 100] loss: 0.304293  (6400/60000)
  [Batch 200] loss: 0.218523  (12800/60000)
  [Batch 300] loss: 0.336096  (19200/60000)
  [Batch 400] loss: 0.335458  (25600/60000)
  [Batch 500] loss: 0.380496  (32000/60000)
  [Batch 600] loss: 0.273133  (38400/60000)
  [Batch 700] loss: 0.450091  (44800/60000)
  [Batch 800] loss: 0.528670  (51200/60000)
  [Batch 900] loss: 0.387175  (57600/60000)
Epoch 25/30
  [Batch 000] loss: 0.226462  (0/60000)
  [Batch 100] loss: 0.300792  (6400/60000)
  [Batch 200] loss: 0.220624  (12800/60000)
  [Batch 300] loss: 0.329989  (19200/60000)
  [Batch 400] loss: 0.325055  (25600/60000)
  [Batch 500] loss: 0.382454  (32000/60000)
  [Batch 600] loss: 0.269731  (38400/60000)
  [Batch 700] loss: 0.436095  (44800/60000)
  [Batch 800] loss: 0.522496  (51200/60000)
  [Batch 900] loss: 0.387705  (57600/60000)
Epoch 26/30
  [Batch 000] loss: 0.221331  (0/60000)
  [Batch 100] loss: 0.291635  (6400/60000)
  [Batch 200] loss: 0.217896  (12800/60000)
  [Batch 300] loss: 0.331855  (19200/60000)
  [Batch 400] loss: 0.321873  (25600/60000)
  [Batch 500] loss: 0.380834  (32000/60000)
  [Batch 600] loss: 0.261753  (38400/60000)
  [Batch 700] loss: 0.430151  (44800/60000)
  [Batch 800] loss: 0.510805  (51200/60000)
  [Batch 900] loss: 0.385109  (57600/60000)
Epoch 27/30
  [Batch 000] loss: 0.212754  (0/60000)
  [Batch 100] loss: 0.282731  (6400/60000)
  [Batch 200] loss: 0.214980  (12800/60000)
  [Batch 300] loss: 0.321798  (19200/60000)
  [Batch 400] loss: 0.316920  (25600/60000)
  [Batch 500] loss: 0.382389  (32000/60000)
  [Batch 600] loss: 0.261003  (38400/60000)
  [Batch 700] loss: 0.424499  (44800/60000)
  [Batch 800] loss: 0.498055  (51200/60000)
  [Batch 900] loss: 0.385663  (57600/60000)
Epoch 28/30
  [Batch 000] loss: 0.209263  (0/60000)
  [Batch 100] loss: 0.276226  (6400/60000)
  [Batch 200] loss: 0.204996  (12800/60000)
  [Batch 300] loss: 0.314777  (19200/60000)
  [Batch 400] loss: 0.313495  (25600/60000)
  [Batch 500] loss: 0.380727  (32000/60000)
  [Batch 600] loss: 0.262593  (38400/60000)
  [Batch 700] loss: 0.418126  (44800/60000)
  [Batch 800] loss: 0.496337  (51200/60000)
  [Batch 900] loss: 0.380829  (57600/60000)
Epoch 29/30
  [Batch 000] loss: 0.204128  (0/60000)
  [Batch 100] loss: 0.271669  (6400/60000)
  [Batch 200] loss: 0.203760  (12800/60000)
  [Batch 300] loss: 0.312172  (19200/60000)
  [Batch 400] loss: 0.308574  (25600/60000)
  [Batch 500] loss: 0.384639  (32000/60000)
  [Batch 600] loss: 0.256890  (38400/60000)
  [Batch 700] loss: 0.399532  (44800/60000)
  [Batch 800] loss: 0.477033  (51200/60000)
  [Batch 900] loss: 0.381044  (57600/60000)
Epoch 30/30
  [Batch 000] loss: 0.199272  (0/60000)
  [Batch 100] loss: 0.266004  (6400/60000)
  [Batch 200] loss: 0.201317  (12800/60000)
  [Batch 300] loss: 0.304960  (19200/60000)
  [Batch 400] loss: 0.305259  (25600/60000)
  [Batch 500] loss: 0.384783  (32000/60000)
  [Batch 600] loss: 0.252054  (38400/60000)
  [Batch 700] loss: 0.397029  (44800/60000)
  [Batch 800] loss: 0.473770  (51200/60000)
  [Batch 900] loss: 0.377472  (57600/60000)
Best accuracy = 87.79% at epoch 30
```



```
Final test accuracy for CNN model at LR=0.01: 87.79%
```

Question 2

In [13]:

```
class FlexibleNetActivation(nn.Module):
    def __init__(self, hidden_units, activation_fn=nn.ReLU):
        super().__init__()
        layers = [nn.Flatten()]
        input_size = 28 * 28
        for h in hidden_units:
            layers += [nn.Linear(input_size, h), activation_fn()]
            input_size = h
        layers.append(nn.Linear(input_size, 10))
        self.net = nn.Sequential(*layers)

    def forward(self, x):
        return self.net(x)

def train_with_loss_and_grads(dataloader, model, loss_fn, optimizer, log_
    model.train()
    batch_losses = []
    mean_gradients = []
    total = len(dataloader.dataset)

    for batch_idx, (X, y) in enumerate(dataloader):
        X, y = X.to(device), y.to(device)

        optimizer.zero_grad()
        preds = model(X)
        loss = loss_fn(preds, y)
        loss.backward()
        optimizer.step()

        # Record loss and gradients
        batch_losses.append(loss.item())
        grads = torch.cat([p.grad.view(-1)
                           for p in model.parameters() if p.grad is not N
        mean_gradients.append(grads.mean().item())

        # Log both metrics at intervals
        if batch_idx % log_interval == 0:
            seen = batch_idx * dataloader.batch_size
            print(f"[Batch {batch_idx:03d}] loss: {loss:.6f}, mean grad:

    avg_loss = np.mean(batch_losses)
    return avg_loss, mean_gradients
```

In [14]:

```
# Compare ReLU, Sigmoid and Tanh without Xavier initialization
lr = 0.01
epochs = 30
hidden_lay = [256, 512, 64]
criterion = nn.CrossEntropyLoss()

model_relu = FlexibleNetActivation(hidden_units=hidden_lay, activation_fn
model_sigmoid = FlexibleNetActivation(hidden_units=hidden_lay, activation
model_tanh = FlexibleNetActivation(hidden_units=hidden_lay, activation_fn

optimizer_relu = torch.optim.SGD(params=model_relu.parameters(), lr=lr)
optimizer_sigmoid = torch.optim.SGD(params=model_sigmoid.parameters(), lr
```

```python
optimizer_tanh = torch.optim.SGD(params=model_tanh.parameters(), lr=lr)

relu_train_losses, relu_test_losses, relu_test_accs = [], [], []
sigmoid_train_losses, sigmoid_test_losses, sigmoid_test_accs = [], [], []
tanh_train_losses, tanh_test_losses, tanh_test_accs = [], [], []
gradients_relu, gradients_sigmoid, gradients_tanh = [], [], []

for epoch in range(1, epochs + 1):
    print(f"Epoch {epoch}/{epochs}")
    # ReLU
    train_loss, train_grads = train_with_loss_and_grads(train_dataloader,
    acc, test_loss = test(test_dataloader, model_relu, loss_fn)

    relu_train_losses.append(train_loss)
    relu_test_losses.append(test_loss)
    relu_test_accs.append(acc)
    gradients_relu.append(train_grads)

    # Sigmoid
    train_loss, train_grads = train_with_loss_and_grads(train_dataloader,
    acc, test_loss = test(test_dataloader, model_sigmoid, loss_fn)

    sigmoid_train_losses.append(train_loss)
    sigmoid_test_losses.append(test_loss)
    sigmoid_test_accs.append(acc)
    gradients_sigmoid.append(train_grads)

    # Tanh
    train_loss, train_grads = train_with_loss_and_grads(train_dataloader,
    acc, test_loss = test(test_dataloader, model_tanh, loss_fn)

    tanh_train_losses.append(train_loss)
    tanh_test_losses.append(test_loss)
    tanh_test_accs.append(acc)
    gradients_tanh.append(train_grads)

# Plot results
print("ReLU without Xavier initialization")
plot_train_test_loss(relu_train_losses, relu_test_losses, relu_test_accs,
print("Sigmoid without Xavier initialization")
plot_train_test_loss(sigmoid_train_losses, sigmoid_test_losses, sigmoid_t
print("Tanh without Xavier initialization")
plot_train_test_loss(tanh_train_losses, tanh_test_losses, tanh_test_accs,

# plot gradients
# --- Flatten per-epoch lists into one long list per activation ---
grads_relu_flat    = [g for epoch in gradients_relu    for g in epoch]
grads_sigmoid_flat = [g for epoch in gradients_sigmoid for g in epoch]
grads_tanh_flat    = [g for epoch in gradients_tanh    for g in epoch]

# --- Plot ---
# Plot ReLU gradients
plt.figure()
plt.plot(grads_relu_flat, label='ReLU')
plt.xlabel('Training Step')
plt.ylabel('Mean Gradient')
plt.title('Mean Gradient for ReLU without Xavier initialization')
plt.legend()
plt.grid(True)
plt.show()
```

```
# Plot Sigmoid gradients
plt.figure()
plt.plot(grads_sigmoid_flat, label='Sigmoid')
plt.xlabel('Training Step')
plt.ylabel('Mean Gradient')
plt.title('Mean Gradient for Sigmoid without Xavier initialization')
plt.legend()
plt.grid(True)
plt.show()

# Plot Tanh gradients
plt.figure()
plt.plot(grads_tanh_flat, label='Tanh')
plt.xlabel('Training Step')
plt.ylabel('Mean Gradient')
plt.title('Mean Gradient for Tanh without Xavier initialization')
plt.legend()
plt.grid(True)
plt.show()

# Final results
print("Final test accuracies without Xavier initialization:")
print(f"ReLU: {relu_test_accs[-1]*100:.2f}%")
print(f"Sigmoid: {sigmoid_test_accs[-1]*100:.2f}%")
print(f"Tanh: {tanh_test_accs[-1]*100:.2f}%")
```

```
Epoch 1/30
```

```
---------------------------------------------------------------------------
-
RuntimeError                              Traceback (most recent call las
t)
Cell In[14], line 23
     21 print(f"Epoch {epoch}/{epochs}")
     22 # ReLU
---> 23 train_loss, train_grads = train_with_loss_and_grads(train_dataload
er, model_relu, loss_fn, optimizer_relu)
     24 acc, test_loss = test(test_dataloader, model_relu, loss_fn)
     26 relu_train_losses.append(train_loss)

Cell In[13], line 25, in train_with_loss_and_grads(dataloader, model, loss
_fn, optimizer, log_interval)
     22 X, y = X.to(device), y.to(device)
     24 optimizer.zero_grad()
---> 25 preds = model(X)
     26 loss = loss_fn(preds, y)
     27 loss.backward()

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/torch/nn/modules/module.py:1736, in Module._wrapped_call_impl(s
elf, *args, **kwargs)
   1734     return self._compiled_call_impl(*args, **kwargs)  # type: igno
re[misc]
   1735 else:
-> 1736     return self._call_impl(*args, **kwargs)

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/torch/nn/modules/module.py:1747, in Module._call_impl(self, *ar
gs, **kwargs)
   1742 # If we don't have any hooks, we want to skip the rest of the logi
c in
   1743 # this function, and just call forward.
   1744 if not (self._backward_hooks or self._backward_pre_hooks or self._
forward_hooks or self._forward_pre_hooks
   1745         or _global_backward_pre_hooks or _global_backward_hooks
   1746         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1747     return forward_call(*args, **kwargs)
   1749 result = None
   1750 called_always_called_hooks = set()

Cell In[13], line 13, in FlexibleNetActivation.forward(self, x)
     12 def forward(self, x):
---> 13     return self.net(x)

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/torch/nn/modules/module.py:1736, in Module._wrapped_call_impl(s
elf, *args, **kwargs)
   1734     return self._compiled_call_impl(*args, **kwargs)  # type: igno
re[misc]
   1735 else:
-> 1736     return self._call_impl(*args, **kwargs)

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/torch/nn/modules/module.py:1747, in Module._call_impl(self, *ar
gs, **kwargs)
   1742 # If we don't have any hooks, we want to skip the rest of the logi
c in
   1743 # this function, and just call forward.
   1744 if not (self._backward_hooks or self._backward_pre_hooks or self._
forward_hooks or self._forward_pre_hooks
   1745         or _global_backward_pre_hooks or _global_backward_hooks
   1746         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1747     return forward_call(*args, **kwargs)
   1749 result = None
   1750 called_always_called_hooks = set()

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/torch/nn/modules/container.py:250, in Sequential.forward(self,
input)
    248 def forward(self, input):
    249     for module in self:
--> 250         input = module(input)
    251     return input

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/torch/nn/modules/module.py:1736, in Module._wrapped_call_impl(s
elf, *args, **kwargs)
   1734     return self._compiled_call_impl(*args, **kwargs)  # type: igno
re[misc]
   1735 else:
-> 1736     return self._call_impl(*args, **kwargs)

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/torch/nn/modules/module.py:1747, in Module._call_impl(self, *ar
gs, **kwargs)
   1742 # If we don't have any hooks, we want to skip the rest of the logi
c in
   1743 # this function, and just call forward.
   1744 if not (self._backward_hooks or self._backward_pre_hooks or self._
forward_hooks or self._forward_pre_hooks
   1745         or _global_backward_pre_hooks or _global_backward_hooks
   1746         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1747     return forward_call(*args, **kwargs)
   1749 result = None
   1750 called_always_called_hooks = set()

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/torch/nn/modules/linear.py:125, in Linear.forward(self, input)
    124 def forward(self, input: Tensor) -> Tensor:
--> 125     return F.linear(input, self.weight, self.bias)

RuntimeError: Tensor for argument weight is on cpu but expected on mps
```

In [120…

```
# Compare ReLU, Sigmoid and Tanh with Xavier initialization
lr = 0.01
epochs = 30
hidden_lay = [256, 512, 64]
criterion = nn.CrossEntropyLoss()

model_relu = FlexibleNetActivation(hidden_units=hidden_lay, activation_fn
model_relu.apply(init_weights)
model_sigmoid = FlexibleNetActivation(hidden_units=hidden_lay, activation
model_sigmoid.apply(init_weights)
model_tanh = FlexibleNetActivation(hidden_units=hidden_lay, activation_fn
model_tanh.apply(init_weights)

optimizer_relu = torch.optim.SGD(params=model_relu.parameters(), lr=lr)
optimizer_sigmoid = torch.optim.SGD(params=model_sigmoid.parameters(), lr
optimizer_tanh = torch.optim.SGD(params=model_tanh.parameters(), lr=lr)

relu_train_losses, relu_test_losses, relu_test_accs = [], [], []
sigmoid_train_losses, sigmoid_test_losses, sigmoid_test_accs = [], [], []
tanh_train_losses, tanh_test_losses, tanh_test_accs = [], [], []
gradients_relu, gradients_sigmoid, gradients_tanh = [], [], []

for epoch in range(1, epochs + 1):
    print(f"Epoch {epoch}/{epochs}")
    # ReLU
    train_loss, train_grads = train_with_loss_and_grads(train_dataloader,
    acc, test_loss = test(test_dataloader, model_relu, loss_fn)

    relu_train_losses.append(train_loss)
    relu_test_losses.append(test_loss)
    relu_test_accs.append(acc)
    gradients_relu.append(train_grads)

    # Sigmoid
    train_loss, train_grads = train_with_loss_and_grads(train_dataloader,
    acc, test_loss = test(test_dataloader, model_sigmoid, loss_fn)

    sigmoid_train_losses.append(train_loss)
    sigmoid_test_losses.append(test_loss)
    sigmoid_test_accs.append(acc)
    gradients_sigmoid.append(train_grads)

    # Tanh
    train_loss, train_grads = train_with_loss_and_grads(train_dataloader,
    acc, test_loss = test(test_dataloader, model_tanh, loss_fn)

    tanh_train_losses.append(train_loss)
    tanh_test_losses.append(test_loss)
    tanh_test_accs.append(acc)
    gradients_tanh.append(train_grads)

# Plot results
print("ReLU with Xavier initialization")
plot_train_test_loss(relu_train_losses, relu_test_losses, relu_test_accs,
print("Sigmoid with Xavier initialization")
plot_train_test_loss(sigmoid_train_losses, sigmoid_test_losses, sigmoid_t
print("Tanh with Xavier initialization")
plot_train_test_loss(tanh_train_losses, tanh_test_losses, tanh_test_accs,

# plot gradients
# --- Flatten per-epoch lists into one long list per activation ---
grads_relu_flat    = [g for epoch in gradients_relu    for g in epoch]
grads_sigmoid_flat = [g for epoch in gradients_sigmoid for g in epoch]
grads_tanh_flat    = [g for epoch in gradients_tanh    for g in epoch]

# --- Plot ---
# Plot ReLU gradients
plt.figure()
plt.plot(grads_relu_flat, label='ReLU')
plt.xlabel('Training Step')
plt.ylabel('Mean Gradient')
plt.title('Mean Gradient for ReLU with Xavier')
plt.legend()
plt.grid(True)
plt.show()
```

```python

# Plot Sigmoid gradients
plt.figure()
plt.plot(grads_sigmoid_flat, label='Sigmoid')
plt.xlabel('Training Step')
plt.ylabel('Mean Gradient')
plt.title('Mean Gradient for Sigmoid with Xavier')
plt.legend()
plt.grid(True)
plt.show()

# Plot Tanh gradients
plt.figure()
plt.plot(grads_tanh_flat, label='Tanh')
plt.xlabel('Training Step')
plt.ylabel('Mean Gradient')
plt.title('Mean Gradient for Tanh with Xavier')
plt.legend()
plt.grid(True)
plt.show()

# Final results
print("Final test accuracies with Xavier initialization:")
print(f"ReLU: {relu_test_accs[-1]*100:.2f}%")
print(f"Sigmoid: {sigmoid_test_accs[-1]*100:.2f}%")
print(f"Tanh: {tanh_test_accs[-1]*100:.2f}%")
```

```
Epoch 1/30
[Batch 000] loss: 2.326460, mean grad: 0.000085  (0/60000)
[Batch 100] loss: 1.476230, mean grad: -0.000370  (6400/60000)
[Batch 200] loss: 0.875288, mean grad: -0.000210  (12800/60000)
[Batch 300] loss: 0.928026, mean grad: -0.000084  (19200/60000)
[Batch 400] loss: 0.715097, mean grad: -0.000065  (25600/60000)
[Batch 500] loss: 0.705201, mean grad: 0.000065  (32000/60000)
[Batch 600] loss: 0.729516, mean grad: -0.000068  (38400/60000)
[Batch 700] loss: 0.651017, mean grad: 0.000081  (44800/60000)
[Batch 800] loss: 0.708798, mean grad: -0.000019  (51200/60000)
[Batch 900] loss: 0.566258, mean grad: 0.000058  (57600/60000)
[Batch 000] loss: 2.320356, mean grad: 0.000030  (0/60000)
[Batch 100] loss: 2.302383, mean grad: -0.000005  (6400/60000)
[Batch 200] loss: 2.297068, mean grad: -0.000000  (12800/60000)
[Batch 300] loss: 2.300693, mean grad: 0.000006  (19200/60000)
[Batch 400] loss: 2.290179, mean grad: -0.000011  (25600/60000)
[Batch 500] loss: 2.297604, mean grad: -0.000002  (32000/60000)
[Batch 600] loss: 2.294482, mean grad: -0.000001  (38400/60000)
[Batch 700] loss: 2.284480, mean grad: -0.000004  (44800/60000)
[Batch 800] loss: 2.288481, mean grad: -0.000000  (51200/60000)
[Batch 900] loss: 2.292617, mean grad: -0.000001  (57600/60000)
[Batch 000] loss: 2.487321, mean grad: 0.000230  (0/60000)
[Batch 100] loss: 1.130605, mean grad: 0.000108  (6400/60000)
[Batch 200] loss: 0.720960, mean grad: 0.000103  (12800/60000)
[Batch 300] loss: 0.862806, mean grad: -0.000101  (19200/60000)
[Batch 400] loss: 0.670656, mean grad: 0.000067  (25600/60000)
[Batch 500] loss: 0.635599, mean grad: -0.000166  (32000/60000)
[Batch 600] loss: 0.668676, mean grad: -0.000100  (38400/60000)
[Batch 700] loss: 0.637685, mean grad: 0.000004  (44800/60000)
[Batch 800] loss: 0.697231, mean grad: -0.000124  (51200/60000)
[Batch 900] loss: 0.545585, mean grad: -0.000163  (57600/60000)
Epoch 2/30
[Batch 000] loss: 0.557332, mean grad: -0.000005  (0/60000)
[Batch 100] loss: 0.587695, mean grad: -0.000186  (6400/60000)
[Batch 200] loss: 0.472122, mean grad: 0.000072  (12800/60000)
[Batch 300] loss: 0.639300, mean grad: 0.000010  (19200/60000)
[Batch 400] loss: 0.546015, mean grad: 0.000085  (25600/60000)
[Batch 500] loss: 0.535093, mean grad: 0.000170  (32000/60000)
[Batch 600] loss: 0.557333, mean grad: -0.000078  (38400/60000)
[Batch 700] loss: 0.615267, mean grad: 0.000090  (44800/60000)
[Batch 800] loss: 0.637711, mean grad: 0.000128  (51200/60000)
[Batch 900] loss: 0.447407, mean grad: 0.000160  (57600/60000)
[Batch 000] loss: 2.286940, mean grad: -0.000003  (0/60000)
[Batch 100] loss: 2.286488, mean grad: -0.000004  (6400/60000)
[Batch 200] loss: 2.273681, mean grad: -0.000000  (12800/60000)
[Batch 300] loss: 2.280525, mean grad: 0.000006  (19200/60000)
[Batch 400] loss: 2.264158, mean grad: -0.000009  (25600/60000)
[Batch 500] loss: 2.272158, mean grad: 0.000001  (32000/60000)
[Batch 600] loss: 2.266753, mean grad: -0.000001  (38400/60000)
[Batch 700] loss: 2.254229, mean grad: -0.000002  (44800/60000)
[Batch 800] loss: 2.255371, mean grad: -0.000002  (51200/60000)
[Batch 900] loss: 2.254779, mean grad: 0.000001  (57600/60000)
[Batch 000] loss: 0.488449, mean grad: 0.000116  (0/60000)
[Batch 100] loss: 0.545538, mean grad: 0.000072  (6400/60000)
[Batch 200] loss: 0.354671, mean grad: 0.000058  (12800/60000)
[Batch 300] loss: 0.606309, mean grad: -0.000079  (19200/60000)
[Batch 400] loss: 0.495657, mean grad: 0.000072  (25600/60000)
[Batch 500] loss: 0.474496, mean grad: -0.000158  (32000/60000)
[Batch 600] loss: 0.523506, mean grad: -0.000147  (38400/60000)
[Batch 700] loss: 0.590560, mean grad: 0.000019  (44800/60000)
```

```
[Batch 800] loss: 0.658859, mean grad: -0.000199  (51200/60000)
[Batch 900] loss: 0.439362, mean grad: -0.000194  (57600/60000)
Epoch 3/30
[Batch 000] loss: 0.450441, mean grad: 0.000008  (0/60000)
[Batch 100] loss: 0.485221, mean grad: -0.000186  (6400/60000)
[Batch 200] loss: 0.400682, mean grad: 0.000057  (12800/60000)
[Batch 300] loss: 0.553073, mean grad: 0.000017  (19200/60000)
[Batch 400] loss: 0.489203, mean grad: 0.000077  (25600/60000)
[Batch 500] loss: 0.486387, mean grad: 0.000177  (32000/60000)
[Batch 600] loss: 0.504552, mean grad: -0.000085  (38400/60000)
[Batch 700] loss: 0.599590, mean grad: 0.000086  (44800/60000)
[Batch 800] loss: 0.590980, mean grad: 0.000171  (51200/60000)
[Batch 900] loss: 0.419820, mean grad: 0.000226  (57600/60000)
[Batch 000] loss: 2.251704, mean grad: -0.000003  (0/60000)
[Batch 100] loss: 2.250492, mean grad: -0.000004  (6400/60000)
[Batch 200] loss: 2.227663, mean grad: -0.000001  (12800/60000)
[Batch 300] loss: 2.237710, mean grad: 0.000005  (19200/60000)
[Batch 400] loss: 2.207591, mean grad: -0.000010  (25600/60000)
[Batch 500] loss: 2.214213, mean grad: 0.000005  (32000/60000)
[Batch 600] loss: 2.199316, mean grad: -0.000002  (38400/60000)
[Batch 700] loss: 2.177125, mean grad: -0.000000  (44800/60000)
[Batch 800] loss: 2.165233, mean grad: -0.000001  (51200/60000)
[Batch 900] loss: 2.148456, mean grad: 0.000000  (57600/60000)
[Batch 000] loss: 0.375763, mean grad: 0.000090  (0/60000)
[Batch 100] loss: 0.480909, mean grad: 0.000092  (6400/60000)
[Batch 200] loss: 0.297610, mean grad: 0.000040  (12800/60000)
[Batch 300] loss: 0.533007, mean grad: -0.000057  (19200/60000)
[Batch 400] loss: 0.432044, mean grad: 0.000045  (25600/60000)
[Batch 500] loss: 0.434413, mean grad: -0.000122  (32000/60000)
[Batch 600] loss: 0.469175, mean grad: -0.000164  (38400/60000)
[Batch 700] loss: 0.564922, mean grad: 0.000025  (44800/60000)
[Batch 800] loss: 0.622509, mean grad: -0.000223  (51200/60000)
[Batch 900] loss: 0.405545, mean grad: -0.000221  (57600/60000)
Epoch 4/30
[Batch 000] loss: 0.400781, mean grad: -0.000054  (0/60000)
[Batch 100] loss: 0.447223, mean grad: -0.000198  (6400/60000)
[Batch 200] loss: 0.352857, mean grad: 0.000038  (12800/60000)
[Batch 300] loss: 0.504713, mean grad: -0.000026  (19200/60000)
[Batch 400] loss: 0.449293, mean grad: 0.000096  (25600/60000)
[Batch 500] loss: 0.461430, mean grad: 0.000142  (32000/60000)
[Batch 600] loss: 0.465950, mean grad: -0.000049  (38400/60000)
[Batch 700] loss: 0.580056, mean grad: 0.000055  (44800/60000)
[Batch 800] loss: 0.561344, mean grad: 0.000218  (51200/60000)
[Batch 900] loss: 0.414686, mean grad: 0.000270  (57600/60000)
[Batch 000] loss: 2.153934, mean grad: -0.000002  (0/60000)
[Batch 100] loss: 2.141097, mean grad: -0.000002  (6400/60000)
[Batch 200] loss: 2.089690, mean grad: 0.000000  (12800/60000)
[Batch 300] loss: 2.097500, mean grad: 0.000004  (19200/60000)
[Batch 400] loss: 2.023932, mean grad: -0.000013  (25600/60000)
[Batch 500] loss: 2.022944, mean grad: 0.000008  (32000/60000)
[Batch 600] loss: 1.972576, mean grad: -0.000004  (38400/60000)
[Batch 700] loss: 1.925059, mean grad: 0.000002  (44800/60000)
[Batch 800] loss: 1.881554, mean grad: 0.000002  (51200/60000)
[Batch 900] loss: 1.832172, mean grad: 0.000004  (57600/60000)
[Batch 000] loss: 0.325696, mean grad: 0.000072  (0/60000)
[Batch 100] loss: 0.456279, mean grad: 0.000097  (6400/60000)
[Batch 200] loss: 0.271461, mean grad: 0.000032  (12800/60000)
[Batch 300] loss: 0.500685, mean grad: -0.000042  (19200/60000)
[Batch 400] loss: 0.392004, mean grad: 0.000021  (25600/60000)
[Batch 500] loss: 0.412542, mean grad: -0.000090  (32000/60000)
```

```
[Batch 600] loss: 0.436357, mean grad: -0.000168  (38400/60000)
[Batch 700] loss: 0.542725, mean grad: 0.000029  (44800/60000)
[Batch 800] loss: 0.591074, mean grad: -0.000229  (51200/60000)
[Batch 900] loss: 0.390811, mean grad: -0.000241  (57600/60000)
Epoch 5/30
[Batch 000] loss: 0.359567, mean grad: -0.000072  (0/60000)
[Batch 100] loss: 0.427261, mean grad: -0.000224  (6400/60000)
[Batch 200] loss: 0.321198, mean grad: -0.000024  (12800/60000)
[Batch 300] loss: 0.474484, mean grad: 0.000013  (19200/60000)
[Batch 400] loss: 0.409861, mean grad: 0.000083  (25600/60000)
[Batch 500] loss: 0.445743, mean grad: 0.000164  (32000/60000)
[Batch 600] loss: 0.438152, mean grad: -0.000034  (38400/60000)
[Batch 700] loss: 0.559709, mean grad: 0.000042  (44800/60000)
[Batch 800] loss: 0.534308, mean grad: 0.000204  (51200/60000)
[Batch 900] loss: 0.414180, mean grad: 0.000256  (57600/60000)
[Batch 000] loss: 1.871247, mean grad: -0.000001  (0/60000)
[Batch 100] loss: 1.836112, mean grad: -0.000001  (6400/60000)
[Batch 200] loss: 1.760198, mean grad: 0.000004  (12800/60000)
[Batch 300] loss: 1.774645, mean grad: 0.000006  (19200/60000)
[Batch 400] loss: 1.686372, mean grad: -0.000012  (25600/60000)
[Batch 500] loss: 1.713276, mean grad: 0.000007  (32000/60000)
[Batch 600] loss: 1.654039, mean grad: -0.000004  (38400/60000)
[Batch 700] loss: 1.623444, mean grad: 0.000004  (44800/60000)
[Batch 800] loss: 1.601539, mean grad: 0.000005  (51200/60000)
[Batch 900] loss: 1.562348, mean grad: 0.000013  (57600/60000)
[Batch 000] loss: 0.298167, mean grad: 0.000064  (0/60000)
[Batch 100] loss: 0.439556, mean grad: 0.000098  (6400/60000)
[Batch 200] loss: 0.255590, mean grad: 0.000031  (12800/60000)
[Batch 300] loss: 0.480079, mean grad: -0.000035  (19200/60000)
[Batch 400] loss: 0.364582, mean grad: 0.000009  (25600/60000)
[Batch 500] loss: 0.396706, mean grad: -0.000065  (32000/60000)
[Batch 600] loss: 0.413798, mean grad: -0.000166  (38400/60000)
[Batch 700] loss: 0.524168, mean grad: 0.000033  (44800/60000)
[Batch 800] loss: 0.565561, mean grad: -0.000226  (51200/60000)
[Batch 900] loss: 0.381666, mean grad: -0.000255  (57600/60000)
Epoch 6/30
[Batch 000] loss: 0.333850, mean grad: -0.000079  (0/60000)
[Batch 100] loss: 0.408974, mean grad: -0.000220  (6400/60000)
[Batch 200] loss: 0.297845, mean grad: -0.000024  (12800/60000)
[Batch 300] loss: 0.454993, mean grad: -0.000002  (19200/60000)
[Batch 400] loss: 0.382728, mean grad: 0.000071  (25600/60000)
[Batch 500] loss: 0.434831, mean grad: 0.000132  (32000/60000)
[Batch 600] loss: 0.417423, mean grad: -0.000055  (38400/60000)
[Batch 700] loss: 0.538503, mean grad: 0.000044  (44800/60000)
[Batch 800] loss: 0.511593, mean grad: 0.000169  (51200/60000)
[Batch 900] loss: 0.413404, mean grad: 0.000248  (57600/60000)
[Batch 000] loss: 1.621312, mean grad: -0.000004  (0/60000)
[Batch 100] loss: 1.595998, mean grad: -0.000002  (6400/60000)
[Batch 200] loss: 1.522354, mean grad: 0.000007  (12800/60000)
[Batch 300] loss: 1.546333, mean grad: 0.000006  (19200/60000)
[Batch 400] loss: 1.493846, mean grad: -0.000007  (25600/60000)
[Batch 500] loss: 1.517279, mean grad: 0.000011  (32000/60000)
[Batch 600] loss: 1.465402, mean grad: -0.000007  (38400/60000)
[Batch 700] loss: 1.437755, mean grad: 0.000005  (44800/60000)
[Batch 800] loss: 1.438951, mean grad: 0.000001  (51200/60000)
[Batch 900] loss: 1.396894, mean grad: 0.000018  (57600/60000)
[Batch 000] loss: 0.281503, mean grad: 0.000065  (0/60000)
[Batch 100] loss: 0.425229, mean grad: 0.000101  (6400/60000)
[Batch 200] loss: 0.244609, mean grad: 0.000034  (12800/60000)
[Batch 300] loss: 0.463824, mean grad: -0.000032  (19200/60000)
```

```
[Batch 400] loss: 0.344965, mean grad: 0.000004  (25600/60000)
[Batch 500] loss: 0.384051, mean grad: -0.000047  (32000/60000)
[Batch 600] loss: 0.397182, mean grad: -0.000164  (38400/60000)
[Batch 700] loss: 0.508074, mean grad: 0.000037  (44800/60000)
[Batch 800] loss: 0.544933, mean grad: -0.000220  (51200/60000)
[Batch 900] loss: 0.374567, mean grad: -0.000266  (57600/60000)
Epoch 7/30
[Batch 000] loss: 0.312461, mean grad: -0.000083  (0/60000)
[Batch 100] loss: 0.396717, mean grad: -0.000203  (6400/60000)
[Batch 200] loss: 0.284246, mean grad: -0.000033  (12800/60000)
[Batch 300] loss: 0.437777, mean grad: -0.000025  (19200/60000)
[Batch 400] loss: 0.361982, mean grad: 0.000039  (25600/60000)
[Batch 500] loss: 0.425039, mean grad: 0.000147  (32000/60000)
[Batch 600] loss: 0.399701, mean grad: -0.000052  (38400/60000)
[Batch 700] loss: 0.514738, mean grad: 0.000041  (44800/60000)
[Batch 800] loss: 0.495506, mean grad: 0.000138  (51200/60000)
[Batch 900] loss: 0.412021, mean grad: 0.000273  (57600/60000)
[Batch 000] loss: 1.451634, mean grad: -0.000011  (0/60000)
[Batch 100] loss: 1.441508, mean grad: -0.000005  (6400/60000)
[Batch 200] loss: 1.342032, mean grad: 0.000007  (12800/60000)
[Batch 300] loss: 1.394064, mean grad: 0.000009  (19200/60000)
[Batch 400] loss: 1.350740, mean grad: -0.000004  (25600/60000)
[Batch 500] loss: 1.365719, mean grad: 0.000021  (32000/60000)
[Batch 600] loss: 1.332927, mean grad: -0.000010  (38400/60000)
[Batch 700] loss: 1.299797, mean grad: 0.000008  (44800/60000)
[Batch 800] loss: 1.321727, mean grad: -0.000006  (51200/60000)
[Batch 900] loss: 1.276204, mean grad: 0.000024  (57600/60000)
[Batch 000] loss: 0.270691, mean grad: 0.000070  (0/60000)
[Batch 100] loss: 0.412077, mean grad: 0.000105  (6400/60000)
[Batch 200] loss: 0.236460, mean grad: 0.000038  (12800/60000)
[Batch 300] loss: 0.449898, mean grad: -0.000032  (19200/60000)
[Batch 400] loss: 0.330504, mean grad: 0.000004  (25600/60000)
[Batch 500] loss: 0.373509, mean grad: -0.000033  (32000/60000)
[Batch 600] loss: 0.384493, mean grad: -0.000162  (38400/60000)
[Batch 700] loss: 0.493811, mean grad: 0.000041  (44800/60000)
[Batch 800] loss: 0.528007, mean grad: -0.000210  (51200/60000)
[Batch 900] loss: 0.368599, mean grad: -0.000274  (57600/60000)
Epoch 8/30
[Batch 000] loss: 0.291580, mean grad: -0.000087  (0/60000)
[Batch 100] loss: 0.383755, mean grad: -0.000195  (6400/60000)
[Batch 200] loss: 0.274435, mean grad: -0.000028  (12800/60000)
[Batch 300] loss: 0.422763, mean grad: -0.000020  (19200/60000)
[Batch 400] loss: 0.344205, mean grad: 0.000032  (25600/60000)
[Batch 500] loss: 0.413943, mean grad: 0.000136  (32000/60000)
[Batch 600] loss: 0.383961, mean grad: -0.000039  (38400/60000)
[Batch 700] loss: 0.495986, mean grad: 0.000016  (44800/60000)
[Batch 800] loss: 0.481475, mean grad: 0.000163  (51200/60000)
[Batch 900] loss: 0.408911, mean grad: 0.000266  (57600/60000)
[Batch 000] loss: 1.322532, mean grad: -0.000019  (0/60000)
[Batch 100] loss: 1.319756, mean grad: -0.000008  (6400/60000)
[Batch 200] loss: 1.195430, mean grad: 0.000005  (12800/60000)
[Batch 300] loss: 1.279429, mean grad: 0.000012  (19200/60000)
[Batch 400] loss: 1.220603, mean grad: -0.000001  (25600/60000)
[Batch 500] loss: 1.247820, mean grad: 0.000030  (32000/60000)
[Batch 600] loss: 1.235788, mean grad: -0.000009  (38400/60000)
[Batch 700] loss: 1.192835, mean grad: 0.000011  (44800/60000)
[Batch 800] loss: 1.232039, mean grad: -0.000013  (51200/60000)
[Batch 900] loss: 1.184631, mean grad: 0.000028  (57600/60000)
[Batch 000] loss: 0.263260, mean grad: 0.000076  (0/60000)
[Batch 100] loss: 0.399849, mean grad: 0.000108  (6400/60000)
```

```
[Batch 200] loss: 0.230198, mean grad: 0.000041  (12800/60000)
[Batch 300] loss: 0.437601, mean grad: -0.000033  (19200/60000)
[Batch 400] loss: 0.319548, mean grad: 0.000005  (25600/60000)
[Batch 500] loss: 0.364529, mean grad: -0.000023  (32000/60000)
[Batch 600] loss: 0.374560, mean grad: -0.000161  (38400/60000)
[Batch 700] loss: 0.481151, mean grad: 0.000043  (44800/60000)
[Batch 800] loss: 0.513807, mean grad: -0.000198  (51200/60000)
[Batch 900] loss: 0.363495, mean grad: -0.000282  (57600/60000)
Epoch 9/30
[Batch 000] loss: 0.275914, mean grad: -0.000105  (0/60000)
[Batch 100] loss: 0.372025, mean grad: -0.000182  (6400/60000)
[Batch 200] loss: 0.264839, mean grad: -0.000019  (12800/60000)
[Batch 300] loss: 0.407897, mean grad: -0.000044  (19200/60000)
[Batch 400] loss: 0.332229, mean grad: 0.000048  (25600/60000)
[Batch 500] loss: 0.405989, mean grad: 0.000145  (32000/60000)
[Batch 600] loss: 0.367210, mean grad: -0.000024  (38400/60000)
[Batch 700] loss: 0.478965, mean grad: 0.000051  (44800/60000)
[Batch 800] loss: 0.466557, mean grad: 0.000176  (51200/60000)
[Batch 900] loss: 0.404994, mean grad: 0.000288  (57600/60000)
[Batch 000] loss: 1.224287, mean grad: -0.000025  (0/60000)
[Batch 100] loss: 1.223060, mean grad: -0.000010  (6400/60000)
[Batch 200] loss: 1.081263, mean grad: 0.000000  (12800/60000)
[Batch 300] loss: 1.188982, mean grad: 0.000015  (19200/60000)
[Batch 400] loss: 1.107982, mean grad: 0.000003  (25600/60000)
[Batch 500] loss: 1.156145, mean grad: 0.000034  (32000/60000)
[Batch 600] loss: 1.159935, mean grad: -0.000005  (38400/60000)
[Batch 700] loss: 1.109574, mean grad: 0.000012  (44800/60000)
[Batch 800] loss: 1.160981, mean grad: -0.000019  (51200/60000)
[Batch 900] loss: 1.113363, mean grad: 0.000031  (57600/60000)
[Batch 000] loss: 0.257854, mean grad: 0.000081  (0/60000)
[Batch 100] loss: 0.388469, mean grad: 0.000111  (6400/60000)
[Batch 200] loss: 0.225315, mean grad: 0.000044  (12800/60000)
[Batch 300] loss: 0.426592, mean grad: -0.000035  (19200/60000)
[Batch 400] loss: 0.310964, mean grad: 0.000006  (25600/60000)
[Batch 500] loss: 0.356742, mean grad: -0.000016  (32000/60000)
[Batch 600] loss: 0.366555, mean grad: -0.000161  (38400/60000)
[Batch 700] loss: 0.469914, mean grad: 0.000045  (44800/60000)
[Batch 800] loss: 0.501590, mean grad: -0.000186  (51200/60000)
[Batch 900] loss: 0.359122, mean grad: -0.000289  (57600/60000)
Epoch 10/30
[Batch 000] loss: 0.263113, mean grad: -0.000095  (0/60000)
[Batch 100] loss: 0.362387, mean grad: -0.000176  (6400/60000)
[Batch 200] loss: 0.257745, mean grad: -0.000055  (12800/60000)
[Batch 300] loss: 0.395692, mean grad: -0.000014  (19200/60000)
[Batch 400] loss: 0.320966, mean grad: 0.000031  (25600/60000)
[Batch 500] loss: 0.397663, mean grad: 0.000123  (32000/60000)
[Batch 600] loss: 0.356141, mean grad: -0.000007  (38400/60000)
[Batch 700] loss: 0.462046, mean grad: 0.000030  (44800/60000)
[Batch 800] loss: 0.455388, mean grad: 0.000186  (51200/60000)
[Batch 900] loss: 0.398204, mean grad: 0.000276  (57600/60000)
[Batch 000] loss: 1.148705, mean grad: -0.000028  (0/60000)
[Batch 100] loss: 1.149653, mean grad: -0.000012  (6400/60000)
[Batch 200] loss: 0.996846, mean grad: -0.000005  (12800/60000)
[Batch 300] loss: 1.118736, mean grad: 0.000016  (19200/60000)
[Batch 400] loss: 1.018406, mean grad: 0.000008  (25600/60000)
[Batch 500] loss: 1.085186, mean grad: 0.000035  (32000/60000)
[Batch 600] loss: 1.098793, mean grad: -0.000001  (38400/60000)
[Batch 700] loss: 1.046727, mean grad: 0.000011  (44800/60000)
[Batch 800] loss: 1.103801, mean grad: -0.000024  (51200/60000)
[Batch 900] loss: 1.057141, mean grad: 0.000033  (57600/60000)
```

```
[Batch 000] loss: 0.253675, mean grad: 0.000085  (0/60000)
[Batch 100] loss: 0.377884, mean grad: 0.000113  (6400/60000)
[Batch 200] loss: 0.221477, mean grad: 0.000045  (12800/60000)
[Batch 300] loss: 0.416636, mean grad: -0.000037  (19200/60000)
[Batch 400] loss: 0.303978, mean grad: 0.000006  (25600/60000)
[Batch 500] loss: 0.349889, mean grad: -0.000010  (32000/60000)
[Batch 600] loss: 0.359866, mean grad: -0.000163  (38400/60000)
[Batch 700] loss: 0.459864, mean grad: 0.000045  (44800/60000)
[Batch 800] loss: 0.490803, mean grad: -0.000175  (51200/60000)
[Batch 900] loss: 0.355363, mean grad: -0.000296  (57600/60000)
Epoch 11/30
[Batch 000] loss: 0.250396, mean grad: -0.000107  (0/60000)
[Batch 100] loss: 0.355419, mean grad: -0.000173  (6400/60000)
[Batch 200] loss: 0.251416, mean grad: -0.000047  (12800/60000)
[Batch 300] loss: 0.382562, mean grad: -0.000002  (19200/60000)
[Batch 400] loss: 0.311462, mean grad: -0.000011  (25600/60000)
[Batch 500] loss: 0.389473, mean grad: 0.000128  (32000/60000)
[Batch 600] loss: 0.345205, mean grad: -0.000010  (38400/60000)
[Batch 700] loss: 0.446469, mean grad: 0.000067  (44800/60000)
[Batch 800] loss: 0.445168, mean grad: 0.000209  (51200/60000)
[Batch 900] loss: 0.392396, mean grad: 0.000272  (57600/60000)
[Batch 000] loss: 1.087611, mean grad: -0.000029  (0/60000)
[Batch 100] loss: 1.093305, mean grad: -0.000015  (6400/60000)
[Batch 200] loss: 0.933153, mean grad: -0.000009  (12800/60000)
[Batch 300] loss: 1.063826, mean grad: 0.000019  (19200/60000)
[Batch 400] loss: 0.949469, mean grad: 0.000013  (25600/60000)
[Batch 500] loss: 1.028521, mean grad: 0.000035  (32000/60000)
[Batch 600] loss: 1.048391, mean grad: 0.000002  (38400/60000)
[Batch 700] loss: 0.999061, mean grad: 0.000010  (44800/60000)
[Batch 800] loss: 1.056025, mean grad: -0.000028  (51200/60000)
[Batch 900] loss: 1.010884, mean grad: 0.000034  (57600/60000)
[Batch 000] loss: 0.250252, mean grad: 0.000087  (0/60000)
[Batch 100] loss: 0.368042, mean grad: 0.000114  (6400/60000)
[Batch 200] loss: 0.218442, mean grad: 0.000046  (12800/60000)
[Batch 300] loss: 0.407551, mean grad: -0.000039  (19200/60000)
[Batch 400] loss: 0.298074, mean grad: 0.000006  (25600/60000)
[Batch 500] loss: 0.343782, mean grad: -0.000006  (32000/60000)
[Batch 600] loss: 0.354058, mean grad: -0.000165  (38400/60000)
[Batch 700] loss: 0.450732, mean grad: 0.000045  (44800/60000)
[Batch 800] loss: 0.481047, mean grad: -0.000165  (51200/60000)
[Batch 900] loss: 0.352105, mean grad: -0.000303  (57600/60000)
Epoch 12/30
[Batch 000] loss: 0.237010, mean grad: -0.000091  (0/60000)
[Batch 100] loss: 0.350213, mean grad: -0.000160  (6400/60000)
[Batch 200] loss: 0.245762, mean grad: -0.000033  (12800/60000)
[Batch 300] loss: 0.373573, mean grad: -0.000013  (19200/60000)
[Batch 400] loss: 0.305243, mean grad: 0.000035  (25600/60000)
[Batch 500] loss: 0.379080, mean grad: 0.000106  (32000/60000)
[Batch 600] loss: 0.332769, mean grad: -0.000010  (38400/60000)
[Batch 700] loss: 0.435089, mean grad: 0.000073  (44800/60000)
[Batch 800] loss: 0.433570, mean grad: 0.000187  (51200/60000)
[Batch 900] loss: 0.385269, mean grad: 0.000249  (57600/60000)
[Batch 000] loss: 1.034901, mean grad: -0.000030  (0/60000)
[Batch 100] loss: 1.046907, mean grad: -0.000017  (6400/60000)
[Batch 200] loss: 0.881771, mean grad: -0.000012  (12800/60000)
[Batch 300] loss: 1.019036, mean grad: 0.000022  (19200/60000)
[Batch 400] loss: 0.896336, mean grad: 0.000017  (25600/60000)
[Batch 500] loss: 0.980788, mean grad: 0.000035  (32000/60000)
[Batch 600] loss: 1.005630, mean grad: 0.000004  (38400/60000)
[Batch 700] loss: 0.961874, mean grad: 0.000010  (44800/60000)
```

```
[Batch 800] loss: 1.014259, mean grad: -0.000030  (51200/60000)
[Batch 900] loss: 0.971081, mean grad: 0.000035  (57600/60000)
[Batch 000] loss: 0.247319, mean grad: 0.000089  (0/60000)
[Batch 100] loss: 0.358893, mean grad: 0.000115  (6400/60000)
[Batch 200] loss: 0.216026, mean grad: 0.000046  (12800/60000)
[Batch 300] loss: 0.399199, mean grad: -0.000042  (19200/60000)
[Batch 400] loss: 0.292927, mean grad: 0.000006  (25600/60000)
[Batch 500] loss: 0.338283, mean grad: -0.000003  (32000/60000)
[Batch 600] loss: 0.348835, mean grad: -0.000167  (38400/60000)
[Batch 700] loss: 0.442261, mean grad: 0.000045  (44800/60000)
[Batch 800] loss: 0.472050, mean grad: -0.000156  (51200/60000)
[Batch 900] loss: 0.349249, mean grad: -0.000310  (57600/60000)
Epoch 13/30
[Batch 000] loss: 0.224919, mean grad: -0.000115  (0/60000)
[Batch 100] loss: 0.343111, mean grad: -0.000172  (6400/60000)
[Batch 200] loss: 0.237817, mean grad: -0.000024  (12800/60000)
[Batch 300] loss: 0.360911, mean grad: 0.000017  (19200/60000)
[Batch 400] loss: 0.302991, mean grad: 0.000054  (25600/60000)
[Batch 500] loss: 0.373727, mean grad: 0.000099  (32000/60000)
[Batch 600] loss: 0.323015, mean grad: -0.000005  (38400/60000)
[Batch 700] loss: 0.422651, mean grad: 0.000092  (44800/60000)
[Batch 800] loss: 0.423277, mean grad: 0.000211  (51200/60000)
[Batch 900] loss: 0.379095, mean grad: 0.000272  (57600/60000)
[Batch 000] loss: 0.987891, mean grad: -0.000031  (0/60000)
[Batch 100] loss: 1.006365, mean grad: -0.000019  (6400/60000)
[Batch 200] loss: 0.838091, mean grad: -0.000014  (12800/60000)
[Batch 300] loss: 0.981113, mean grad: 0.000025  (19200/60000)
[Batch 400] loss: 0.855308, mean grad: 0.000020  (25600/60000)
[Batch 500] loss: 0.939013, mean grad: 0.000035  (32000/60000)
[Batch 600] loss: 0.968538, mean grad: 0.000005  (38400/60000)
[Batch 700] loss: 0.932031, mean grad: 0.000009  (44800/60000)
[Batch 800] loss: 0.976630, mean grad: -0.000033  (51200/60000)
[Batch 900] loss: 0.936184, mean grad: 0.000036  (57600/60000)
[Batch 000] loss: 0.244729, mean grad: 0.000089  (0/60000)
[Batch 100] loss: 0.350379, mean grad: 0.000116  (6400/60000)
[Batch 200] loss: 0.214085, mean grad: 0.000047  (12800/60000)
[Batch 300] loss: 0.391476, mean grad: -0.000044  (19200/60000)
[Batch 400] loss: 0.288336, mean grad: 0.000005  (25600/60000)
[Batch 500] loss: 0.333291, mean grad: 0.000000  (32000/60000)
[Batch 600] loss: 0.344011, mean grad: -0.000170  (38400/60000)
[Batch 700] loss: 0.434245, mean grad: 0.000045  (44800/60000)
[Batch 800] loss: 0.463639, mean grad: -0.000149  (51200/60000)
[Batch 900] loss: 0.346726, mean grad: -0.000316  (57600/60000)
Epoch 14/30
[Batch 000] loss: 0.218804, mean grad: -0.000105  (0/60000)
[Batch 100] loss: 0.337703, mean grad: -0.000177  (6400/60000)
[Batch 200] loss: 0.230805, mean grad: -0.000036  (12800/60000)
[Batch 300] loss: 0.350938, mean grad: 0.000024  (19200/60000)
[Batch 400] loss: 0.298569, mean grad: 0.000062  (25600/60000)
[Batch 500] loss: 0.365145, mean grad: 0.000090  (32000/60000)
[Batch 600] loss: 0.314891, mean grad: -0.000006  (38400/60000)
[Batch 700] loss: 0.409746, mean grad: 0.000114  (44800/60000)
[Batch 800] loss: 0.409743, mean grad: 0.000250  (51200/60000)
[Batch 900] loss: 0.373724, mean grad: 0.000245  (57600/60000)
[Batch 000] loss: 0.945696, mean grad: -0.000032  (0/60000)
[Batch 100] loss: 0.970029, mean grad: -0.000021  (6400/60000)
[Batch 200] loss: 0.799955, mean grad: -0.000015  (12800/60000)
[Batch 300] loss: 0.948238, mean grad: 0.000028  (19200/60000)
[Batch 400] loss: 0.823767, mean grad: 0.000022  (25600/60000)
[Batch 500] loss: 0.901750, mean grad: 0.000035  (32000/60000)
```

```
[Batch 600] loss: 0.935906, mean grad: 0.000007  (38400/60000)
[Batch 700] loss: 0.907115, mean grad: 0.000009  (44800/60000)
[Batch 800] loss: 0.942136, mean grad: -0.000035  (51200/60000)
[Batch 900] loss: 0.905574, mean grad: 0.000036  (57600/60000)
[Batch 000] loss: 0.242403, mean grad: 0.000088  (0/60000)
[Batch 100] loss: 0.342445, mean grad: 0.000117  (6400/60000)
[Batch 200] loss: 0.212511, mean grad: 0.000047  (12800/60000)
[Batch 300] loss: 0.384303, mean grad: -0.000046  (19200/60000)
[Batch 400] loss: 0.284176, mean grad: 0.000005  (25600/60000)
[Batch 500] loss: 0.328732, mean grad: 0.000002  (32000/60000)
[Batch 600] loss: 0.339472, mean grad: -0.000173  (38400/60000)
[Batch 700] loss: 0.426532, mean grad: 0.000044  (44800/60000)
[Batch 800] loss: 0.455707, mean grad: -0.000144  (51200/60000)
[Batch 900] loss: 0.344486, mean grad: -0.000322  (57600/60000)
Epoch 15/30
[Batch 000] loss: 0.213320, mean grad: -0.000125  (0/60000)
[Batch 100] loss: 0.332127, mean grad: -0.000156  (6400/60000)
[Batch 200] loss: 0.223546, mean grad: -0.000029  (12800/60000)
[Batch 300] loss: 0.341496, mean grad: 0.000003  (19200/60000)
[Batch 400] loss: 0.294872, mean grad: 0.000063  (25600/60000)
[Batch 500] loss: 0.359156, mean grad: 0.000075  (32000/60000)
[Batch 600] loss: 0.307697, mean grad: -0.000006  (38400/60000)
[Batch 700] loss: 0.393911, mean grad: 0.000110  (44800/60000)
[Batch 800] loss: 0.399354, mean grad: 0.000237  (51200/60000)
[Batch 900] loss: 0.368647, mean grad: 0.000240  (57600/60000)
[Batch 000] loss: 0.907635, mean grad: -0.000033  (0/60000)
[Batch 100] loss: 0.937177, mean grad: -0.000022  (6400/60000)
[Batch 200] loss: 0.765917, mean grad: -0.000016  (12800/60000)
[Batch 300] loss: 0.919261, mean grad: 0.000030  (19200/60000)
[Batch 400] loss: 0.799490, mean grad: 0.000023  (25600/60000)
[Batch 500] loss: 0.868011, mean grad: 0.000035  (32000/60000)
[Batch 600] loss: 0.906731, mean grad: 0.000008  (38400/60000)
[Batch 700] loss: 0.885071, mean grad: 0.000009  (44800/60000)
[Batch 800] loss: 0.910060, mean grad: -0.000037  (51200/60000)
[Batch 900] loss: 0.878674, mean grad: 0.000035  (57600/60000)
[Batch 000] loss: 0.240296, mean grad: 0.000087  (0/60000)
[Batch 100] loss: 0.335036, mean grad: 0.000117  (6400/60000)
[Batch 200] loss: 0.211214, mean grad: 0.000047  (12800/60000)
[Batch 300] loss: 0.377614, mean grad: -0.000048  (19200/60000)
[Batch 400] loss: 0.280370, mean grad: 0.000004  (25600/60000)
[Batch 500] loss: 0.324551, mean grad: 0.000004  (32000/60000)
[Batch 600] loss: 0.335158, mean grad: -0.000175  (38400/60000)
[Batch 700] loss: 0.419026, mean grad: 0.000043  (44800/60000)
[Batch 800] loss: 0.448187, mean grad: -0.000140  (51200/60000)
[Batch 900] loss: 0.342499, mean grad: -0.000328  (57600/60000)
Epoch 16/30
[Batch 000] loss: 0.205979, mean grad: -0.000130  (0/60000)
[Batch 100] loss: 0.327879, mean grad: -0.000142  (6400/60000)
[Batch 200] loss: 0.217867, mean grad: -0.000002  (12800/60000)
[Batch 300] loss: 0.335296, mean grad: -0.000001  (19200/60000)
[Batch 400] loss: 0.291149, mean grad: 0.000072  (25600/60000)
[Batch 500] loss: 0.350666, mean grad: 0.000092  (32000/60000)
[Batch 600] loss: 0.297853, mean grad: -0.000048  (38400/60000)
[Batch 700] loss: 0.380978, mean grad: 0.000088  (44800/60000)
[Batch 800] loss: 0.389846, mean grad: 0.000222  (51200/60000)
[Batch 900] loss: 0.361556, mean grad: 0.000258  (57600/60000)
[Batch 000] loss: 0.872716, mean grad: -0.000034  (0/60000)
[Batch 100] loss: 0.907236, mean grad: -0.000023  (6400/60000)
[Batch 200] loss: 0.734544, mean grad: -0.000017  (12800/60000)
[Batch 300] loss: 0.893314, mean grad: 0.000032  (19200/60000)
```

```
[Batch 400] loss: 0.780354, mean grad: 0.000023  (25600/60000)
[Batch 500] loss: 0.836727, mean grad: 0.000035  (32000/60000)
[Batch 600] loss: 0.880056, mean grad: 0.000010  (38400/60000)
[Batch 700] loss: 0.864321, mean grad: 0.000009  (44800/60000)
[Batch 800] loss: 0.879773, mean grad: -0.000038  (51200/60000)
[Batch 900] loss: 0.854720, mean grad: 0.000034  (57600/60000)
[Batch 000] loss: 0.238379, mean grad: 0.000085  (0/60000)
[Batch 100] loss: 0.328107, mean grad: 0.000118  (6400/60000)
[Batch 200] loss: 0.210127, mean grad: 0.000046  (12800/60000)
[Batch 300] loss: 0.371356, mean grad: -0.000051  (19200/60000)
[Batch 400] loss: 0.276865, mean grad: 0.000005  (25600/60000)
[Batch 500] loss: 0.320705, mean grad: 0.000006  (32000/60000)
[Batch 600] loss: 0.331039, mean grad: -0.000178  (38400/60000)
[Batch 700] loss: 0.411668, mean grad: 0.000042  (44800/60000)
[Batch 800] loss: 0.441032, mean grad: -0.000138  (51200/60000)
[Batch 900] loss: 0.340743, mean grad: -0.000334  (57600/60000)
Epoch 17/30
[Batch 000] loss: 0.200174, mean grad: -0.000119  (0/60000)
[Batch 100] loss: 0.324934, mean grad: -0.000122  (6400/60000)
[Batch 200] loss: 0.211903, mean grad: -0.000033  (12800/60000)
[Batch 300] loss: 0.325864, mean grad: 0.000010  (19200/60000)
[Batch 400] loss: 0.289360, mean grad: 0.000049  (25600/60000)
[Batch 500] loss: 0.344324, mean grad: 0.000059  (32000/60000)
[Batch 600] loss: 0.290301, mean grad: -0.000054  (38400/60000)
[Batch 700] loss: 0.371204, mean grad: 0.000087  (44800/60000)
[Batch 800] loss: 0.380738, mean grad: 0.000218  (51200/60000)
[Batch 900] loss: 0.359148, mean grad: 0.000248  (57600/60000)
[Batch 000] loss: 0.839852, mean grad: -0.000034  (0/60000)
[Batch 100] loss: 0.879645, mean grad: -0.000024  (6400/60000)
[Batch 200] loss: 0.704511, mean grad: -0.000017  (12800/60000)
[Batch 300] loss: 0.869612, mean grad: 0.000032  (19200/60000)
[Batch 400] loss: 0.764399, mean grad: 0.000023  (25600/60000)
[Batch 500] loss: 0.806709, mean grad: 0.000035  (32000/60000)
[Batch 600] loss: 0.855074, mean grad: 0.000011  (38400/60000)
[Batch 700] loss: 0.843843, mean grad: 0.000009  (44800/60000)
[Batch 800] loss: 0.850734, mean grad: -0.000038  (51200/60000)
[Batch 900] loss: 0.832897, mean grad: 0.000032  (57600/60000)
[Batch 000] loss: 0.236634, mean grad: 0.000082  (0/60000)
[Batch 100] loss: 0.321619, mean grad: 0.000118  (6400/60000)
[Batch 200] loss: 0.209194, mean grad: 0.000045  (12800/60000)
[Batch 300] loss: 0.365482, mean grad: -0.000053  (19200/60000)
[Batch 400] loss: 0.273621, mean grad: 0.000005  (25600/60000)
[Batch 500] loss: 0.317160, mean grad: 0.000008  (32000/60000)
[Batch 600] loss: 0.327100, mean grad: -0.000180  (38400/60000)
[Batch 700] loss: 0.404439, mean grad: 0.000041  (44800/60000)
[Batch 800] loss: 0.434202, mean grad: -0.000136  (51200/60000)
[Batch 900] loss: 0.339199, mean grad: -0.000339  (57600/60000)
Epoch 18/30
[Batch 000] loss: 0.196242, mean grad: -0.000134  (0/60000)
[Batch 100] loss: 0.321365, mean grad: -0.000121  (6400/60000)
[Batch 200] loss: 0.203920, mean grad: 0.000000  (12800/60000)
[Batch 300] loss: 0.315916, mean grad: 0.000032  (19200/60000)
[Batch 400] loss: 0.289079, mean grad: 0.000062  (25600/60000)
[Batch 500] loss: 0.337562, mean grad: 0.000042  (32000/60000)
[Batch 600] loss: 0.285471, mean grad: -0.000054  (38400/60000)
[Batch 700] loss: 0.359874, mean grad: 0.000082  (44800/60000)
[Batch 800] loss: 0.371937, mean grad: 0.000243  (51200/60000)
[Batch 900] loss: 0.354962, mean grad: 0.000201  (57600/60000)
[Batch 000] loss: 0.808173, mean grad: -0.000034  (0/60000)
[Batch 100] loss: 0.853908, mean grad: -0.000025  (6400/60000)
```

```
[Batch 200] loss: 0.674816, mean grad: -0.000017  (12800/60000)
[Batch 300] loss: 0.847477, mean grad: 0.000033  (19200/60000)
[Batch 400] loss: 0.749972, mean grad: 0.000022  (25600/60000)
[Batch 500] loss: 0.776914, mean grad: 0.000036  (32000/60000)
[Batch 600] loss: 0.831236, mean grad: 0.000012  (38400/60000)
[Batch 700] loss: 0.823179, mean grad: 0.000009  (44800/60000)
[Batch 800] loss: 0.822614, mean grad: -0.000038  (51200/60000)
[Batch 900] loss: 0.812516, mean grad: 0.000030  (57600/60000)
[Batch 000] loss: 0.235043, mean grad: 0.000079  (0/60000)
[Batch 100] loss: 0.315542, mean grad: 0.000119  (6400/60000)
[Batch 200] loss: 0.208371, mean grad: 0.000044  (12800/60000)
[Batch 300] loss: 0.359950, mean grad: -0.000054  (19200/60000)
[Batch 400] loss: 0.270607, mean grad: 0.000006  (25600/60000)
[Batch 500] loss: 0.313886, mean grad: 0.000009  (32000/60000)
[Batch 600] loss: 0.323327, mean grad: -0.000182  (38400/60000)
[Batch 700] loss: 0.397345, mean grad: 0.000040  (44800/60000)
[Batch 800] loss: 0.427653, mean grad: -0.000136  (51200/60000)
[Batch 900] loss: 0.337850, mean grad: -0.000344  (57600/60000)
Epoch 19/30
[Batch 000] loss: 0.192213, mean grad: -0.000131  (0/60000)
[Batch 100] loss: 0.318031, mean grad: -0.000120  (6400/60000)
[Batch 200] loss: 0.198084, mean grad: -0.000004  (12800/60000)
[Batch 300] loss: 0.310188, mean grad: 0.000025  (19200/60000)
[Batch 400] loss: 0.285625, mean grad: 0.000076  (25600/60000)
[Batch 500] loss: 0.331721, mean grad: 0.000033  (32000/60000)
[Batch 600] loss: 0.279339, mean grad: -0.000070  (38400/60000)
[Batch 700] loss: 0.352039, mean grad: 0.000082  (44800/60000)
[Batch 800] loss: 0.365974, mean grad: 0.000232  (51200/60000)
[Batch 900] loss: 0.346876, mean grad: 0.000199  (57600/60000)
[Batch 000] loss: 0.777181, mean grad: -0.000033  (0/60000)
[Batch 100] loss: 0.829699, mean grad: -0.000027  (6400/60000)
[Batch 200] loss: 0.644915, mean grad: -0.000016  (12800/60000)
[Batch 300] loss: 0.826500, mean grad: 0.000033  (19200/60000)
[Batch 400] loss: 0.735845, mean grad: 0.000022  (25600/60000)
[Batch 500] loss: 0.746765, mean grad: 0.000036  (32000/60000)
[Batch 600] loss: 0.808335, mean grad: 0.000012  (38400/60000)
[Batch 700] loss: 0.802423, mean grad: 0.000009  (44800/60000)
[Batch 800] loss: 0.795410, mean grad: -0.000037  (51200/60000)
[Batch 900] loss: 0.793135, mean grad: 0.000028  (57600/60000)
[Batch 000] loss: 0.233585, mean grad: 0.000077  (0/60000)
[Batch 100] loss: 0.309851, mean grad: 0.000119  (6400/60000)
[Batch 200] loss: 0.207624, mean grad: 0.000042  (12800/60000)
[Batch 300] loss: 0.354728, mean grad: -0.000056  (19200/60000)
[Batch 400] loss: 0.267793, mean grad: 0.000008  (25600/60000)
[Batch 500] loss: 0.310857, mean grad: 0.000011  (32000/60000)
[Batch 600] loss: 0.319707, mean grad: -0.000184  (38400/60000)
[Batch 700] loss: 0.390416, mean grad: 0.000038  (44800/60000)
[Batch 800] loss: 0.421339, mean grad: -0.000137  (51200/60000)
[Batch 900] loss: 0.336673, mean grad: -0.000349  (57600/60000)
Epoch 20/30
[Batch 000] loss: 0.187562, mean grad: -0.000143  (0/60000)
[Batch 100] loss: 0.314921, mean grad: -0.000113  (6400/60000)
[Batch 200] loss: 0.191478, mean grad: -0.000021  (12800/60000)
[Batch 300] loss: 0.301883, mean grad: 0.000002  (19200/60000)
[Batch 400] loss: 0.282841, mean grad: 0.000044  (25600/60000)
[Batch 500] loss: 0.325722, mean grad: 0.000006  (32000/60000)
[Batch 600] loss: 0.272696, mean grad: -0.000095  (38400/60000)
[Batch 700] loss: 0.341605, mean grad: 0.000112  (44800/60000)
[Batch 800] loss: 0.356276, mean grad: 0.000201  (51200/60000)
[Batch 900] loss: 0.343028, mean grad: 0.000202  (57600/60000)
```

```
[Batch 000] loss: 0.746802, mean grad: -0.000032  (0/60000)
[Batch 100] loss: 0.806907, mean grad: -0.000028  (6400/60000)
[Batch 200] loss: 0.614834, mean grad: -0.000014  (12800/60000)
[Batch 300] loss: 0.806628, mean grad: 0.000034  (19200/60000)
[Batch 400] loss: 0.721248, mean grad: 0.000020  (25600/60000)
[Batch 500] loss: 0.716346, mean grad: 0.000037  (32000/60000)
[Batch 600] loss: 0.786508, mean grad: 0.000012  (38400/60000)
[Batch 700] loss: 0.782134, mean grad: 0.000009  (44800/60000)
[Batch 800] loss: 0.769462, mean grad: -0.000036  (51200/60000)
[Batch 900] loss: 0.774614, mean grad: 0.000026  (57600/60000)
[Batch 000] loss: 0.232235, mean grad: 0.000074  (0/60000)
[Batch 100] loss: 0.304529, mean grad: 0.000119  (6400/60000)
[Batch 200] loss: 0.206927, mean grad: 0.000040  (12800/60000)
[Batch 300] loss: 0.349784, mean grad: -0.000057  (19200/60000)
[Batch 400] loss: 0.265153, mean grad: 0.000010  (25600/60000)
[Batch 500] loss: 0.308049, mean grad: 0.000012  (32000/60000)
[Batch 600] loss: 0.316227, mean grad: -0.000186  (38400/60000)
[Batch 700] loss: 0.383688, mean grad: 0.000037  (44800/60000)
[Batch 800] loss: 0.415215, mean grad: -0.000139  (51200/60000)
[Batch 900] loss: 0.335649, mean grad: -0.000353  (57600/60000)
Epoch 21/30
[Batch 000] loss: 0.184291, mean grad: -0.000146  (0/60000)
[Batch 100] loss: 0.311259, mean grad: -0.000123  (6400/60000)
[Batch 200] loss: 0.188334, mean grad: -0.000030  (12800/60000)
[Batch 300] loss: 0.293124, mean grad: 0.000003  (19200/60000)
[Batch 400] loss: 0.280616, mean grad: 0.000068  (25600/60000)
[Batch 500] loss: 0.322591, mean grad: 0.000016  (32000/60000)
[Batch 600] loss: 0.267083, mean grad: -0.000091  (38400/60000)
[Batch 700] loss: 0.330392, mean grad: 0.000124  (44800/60000)
[Batch 800] loss: 0.348693, mean grad: 0.000193  (51200/60000)
[Batch 900] loss: 0.337302, mean grad: 0.000145  (57600/60000)
[Batch 000] loss: 0.717340, mean grad: -0.000031  (0/60000)
[Batch 100] loss: 0.785624, mean grad: -0.000029  (6400/60000)
[Batch 200] loss: 0.585183, mean grad: -0.000013  (12800/60000)
[Batch 300] loss: 0.788105, mean grad: 0.000034  (19200/60000)
[Batch 400] loss: 0.705827, mean grad: 0.000019  (25600/60000)
[Batch 500] loss: 0.686368, mean grad: 0.000038  (32000/60000)
[Batch 600] loss: 0.766107, mean grad: 0.000011  (38400/60000)
[Batch 700] loss: 0.763121, mean grad: 0.000009  (44800/60000)
[Batch 800] loss: 0.745375, mean grad: -0.000034  (51200/60000)
[Batch 900] loss: 0.757071, mean grad: 0.000025  (57600/60000)
[Batch 000] loss: 0.230964, mean grad: 0.000071  (0/60000)
[Batch 100] loss: 0.299565, mean grad: 0.000118  (6400/60000)
[Batch 200] loss: 0.206263, mean grad: 0.000038  (12800/60000)
[Batch 300] loss: 0.345094, mean grad: -0.000059  (19200/60000)
[Batch 400] loss: 0.262661, mean grad: 0.000012  (25600/60000)
[Batch 500] loss: 0.305437, mean grad: 0.000013  (32000/60000)
[Batch 600] loss: 0.312877, mean grad: -0.000188  (38400/60000)
[Batch 700] loss: 0.377201, mean grad: 0.000035  (44800/60000)
[Batch 800] loss: 0.409239, mean grad: -0.000140  (51200/60000)
[Batch 900] loss: 0.334759, mean grad: -0.000357  (57600/60000)
Epoch 22/30
[Batch 000] loss: 0.181905, mean grad: -0.000136  (0/60000)
[Batch 100] loss: 0.308584, mean grad: -0.000117  (6400/60000)
[Batch 200] loss: 0.182839, mean grad: -0.000061  (12800/60000)
[Batch 300] loss: 0.288758, mean grad: 0.000011  (19200/60000)
[Batch 400] loss: 0.278367, mean grad: 0.000082  (25600/60000)
[Batch 500] loss: 0.316868, mean grad: 0.000012  (32000/60000)
[Batch 600] loss: 0.260434, mean grad: -0.000082  (38400/60000)
[Batch 700] loss: 0.324367, mean grad: 0.000136  (44800/60000)
```

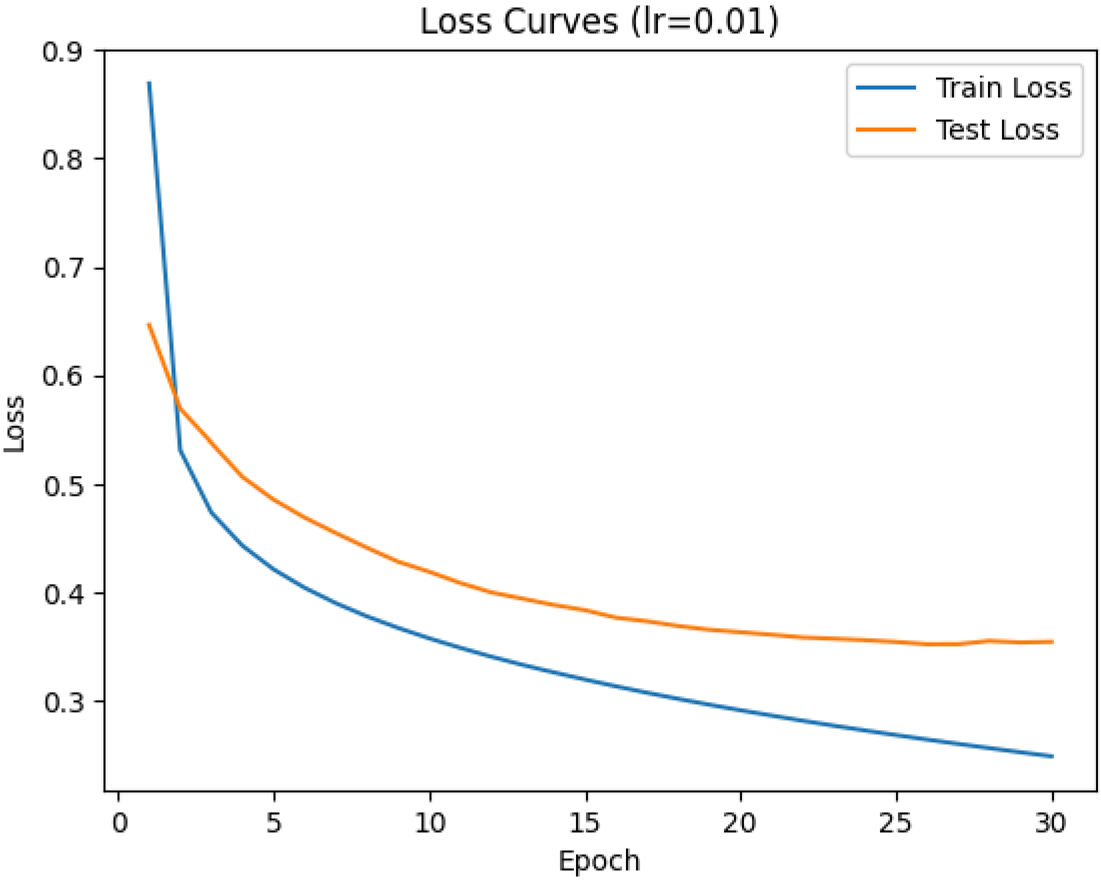
```
[Batch 800] loss: 0.342382, mean grad: 0.000201  (51200/60000)
[Batch 900] loss: 0.331234, mean grad: 0.000172  (57600/60000)
[Batch 000] loss: 0.689359, mean grad: -0.000031  (0/60000)
[Batch 100] loss: 0.766056, mean grad: -0.000030  (6400/60000)
[Batch 200] loss: 0.556976, mean grad: -0.000011  (12800/60000)
[Batch 300] loss: 0.771248, mean grad: 0.000035  (19200/60000)
[Batch 400] loss: 0.689543, mean grad: 0.000017  (25600/60000)
[Batch 500] loss: 0.657867, mean grad: 0.000039  (32000/60000)
[Batch 600] loss: 0.747449, mean grad: 0.000010  (38400/60000)
[Batch 700] loss: 0.746122, mean grad: 0.000008  (44800/60000)
[Batch 800] loss: 0.723797, mean grad: -0.000032  (51200/60000)
[Batch 900] loss: 0.740725, mean grad: 0.000023  (57600/60000)
[Batch 000] loss: 0.229745, mean grad: 0.000068  (0/60000)
[Batch 100] loss: 0.294946, mean grad: 0.000118  (6400/60000)
[Batch 200] loss: 0.205621, mean grad: 0.000036  (12800/60000)
[Batch 300] loss: 0.340638, mean grad: -0.000060  (19200/60000)
[Batch 400] loss: 0.260295, mean grad: 0.000014  (25600/60000)
[Batch 500] loss: 0.303001, mean grad: 0.000014  (32000/60000)
[Batch 600] loss: 0.309649, mean grad: -0.000190  (38400/60000)
[Batch 700] loss: 0.370989, mean grad: 0.000032  (44800/60000)
[Batch 800] loss: 0.403375, mean grad: -0.000143  (51200/60000)
[Batch 900] loss: 0.333989, mean grad: -0.000361  (57600/60000)
Epoch 23/30
[Batch 000] loss: 0.178640, mean grad: -0.000153  (0/60000)
[Batch 100] loss: 0.305883, mean grad: -0.000082  (6400/60000)
[Batch 200] loss: 0.178787, mean grad: -0.000044  (12800/60000)
[Batch 300] loss: 0.282751, mean grad: -0.000009  (19200/60000)
[Batch 400] loss: 0.275668, mean grad: 0.000062  (25600/60000)
[Batch 500] loss: 0.311727, mean grad: -0.000035  (32000/60000)
[Batch 600] loss: 0.254281, mean grad: -0.000072  (38400/60000)
[Batch 700] loss: 0.320356, mean grad: 0.000095  (44800/60000)
[Batch 800] loss: 0.336160, mean grad: 0.000163  (51200/60000)
[Batch 900] loss: 0.326672, mean grad: 0.000140  (57600/60000)
[Batch 000] loss: 0.663475, mean grad: -0.000030  (0/60000)
[Batch 100] loss: 0.748392, mean grad: -0.000030  (6400/60000)
[Batch 200] loss: 0.531231, mean grad: -0.000009  (12800/60000)
[Batch 300] loss: 0.756201, mean grad: 0.000035  (19200/60000)
[Batch 400] loss: 0.672595, mean grad: 0.000015  (25600/60000)
[Batch 500] loss: 0.631761, mean grad: 0.000040  (32000/60000)
[Batch 600] loss: 0.730606, mean grad: 0.000010  (38400/60000)
[Batch 700] loss: 0.731522, mean grad: 0.000008  (44800/60000)
[Batch 800] loss: 0.705155, mean grad: -0.000030  (51200/60000)
[Batch 900] loss: 0.725712, mean grad: 0.000022  (57600/60000)
[Batch 000] loss: 0.228556, mean grad: 0.000066  (0/60000)
[Batch 100] loss: 0.290663, mean grad: 0.000117  (6400/60000)
[Batch 200] loss: 0.204988, mean grad: 0.000034  (12800/60000)
[Batch 300] loss: 0.336398, mean grad: -0.000061  (19200/60000)
[Batch 400] loss: 0.258036, mean grad: 0.000017  (25600/60000)
[Batch 500] loss: 0.300722, mean grad: 0.000014  (32000/60000)
[Batch 600] loss: 0.306540, mean grad: -0.000193  (38400/60000)
[Batch 700] loss: 0.365079, mean grad: 0.000030  (44800/60000)
[Batch 800] loss: 0.397593, mean grad: -0.000145  (51200/60000)
[Batch 900] loss: 0.333324, mean grad: -0.000365  (57600/60000)
Epoch 24/30
[Batch 000] loss: 0.179955, mean grad: -0.000165  (0/60000)
[Batch 100] loss: 0.302268, mean grad: -0.000095  (6400/60000)
[Batch 200] loss: 0.174713, mean grad: -0.000070  (12800/60000)
[Batch 300] loss: 0.278193, mean grad: 0.000014  (19200/60000)
[Batch 400] loss: 0.275443, mean grad: 0.000064  (25600/60000)
[Batch 500] loss: 0.308093, mean grad: -0.000037  (32000/60000)
```

```
[Batch 600] loss: 0.247580, mean grad: -0.000068  (38400/60000)
[Batch 700] loss: 0.314770, mean grad: 0.000110  (44800/60000)
[Batch 800] loss: 0.329091, mean grad: 0.000135  (51200/60000)
[Batch 900] loss: 0.323159, mean grad: 0.000129  (57600/60000)
[Batch 000] loss: 0.640128, mean grad: -0.000029  (0/60000)
[Batch 100] loss: 0.732694, mean grad: -0.000030  (6400/60000)
[Batch 200] loss: 0.508568, mean grad: -0.000007  (12800/60000)
[Batch 300] loss: 0.742819, mean grad: 0.000036  (19200/60000)
[Batch 400] loss: 0.655344, mean grad: 0.000013  (25600/60000)
[Batch 500] loss: 0.608535, mean grad: 0.000042  (32000/60000)
[Batch 600] loss: 0.715358, mean grad: 0.000009  (38400/60000)
[Batch 700] loss: 0.719285, mean grad: 0.000007  (44800/60000)
[Batch 800] loss: 0.689511, mean grad: -0.000028  (51200/60000)
[Batch 900] loss: 0.711995, mean grad: 0.000022  (57600/60000)
[Batch 000] loss: 0.227380, mean grad: 0.000064  (0/60000)
[Batch 100] loss: 0.286702, mean grad: 0.000116  (6400/60000)
[Batch 200] loss: 0.204355, mean grad: 0.000033  (12800/60000)
[Batch 300] loss: 0.332356, mean grad: -0.000061  (19200/60000)
[Batch 400] loss: 0.255867, mean grad: 0.000020  (25600/60000)
[Batch 500] loss: 0.298583, mean grad: 0.000014  (32000/60000)
[Batch 600] loss: 0.303545, mean grad: -0.000195  (38400/60000)
[Batch 700] loss: 0.359488, mean grad: 0.000027  (44800/60000)
[Batch 800] loss: 0.391870, mean grad: -0.000148  (51200/60000)
[Batch 900] loss: 0.332755, mean grad: -0.000369  (57600/60000)
Epoch 25/30
[Batch 000] loss: 0.175226, mean grad: -0.000161  (0/60000)
[Batch 100] loss: 0.298811, mean grad: -0.000074  (6400/60000)
[Batch 200] loss: 0.169153, mean grad: -0.000070  (12800/60000)
[Batch 300] loss: 0.271968, mean grad: 0.000029  (19200/60000)
[Batch 400] loss: 0.272026, mean grad: 0.000081  (25600/60000)
[Batch 500] loss: 0.302157, mean grad: -0.000057  (32000/60000)
[Batch 600] loss: 0.244455, mean grad: -0.000073  (38400/60000)
[Batch 700] loss: 0.306609, mean grad: 0.000108  (44800/60000)
[Batch 800] loss: 0.327087, mean grad: 0.000148  (51200/60000)
[Batch 900] loss: 0.315785, mean grad: 0.000133  (57600/60000)
[Batch 000] loss: 0.619462, mean grad: -0.000028  (0/60000)
[Batch 100] loss: 0.718868, mean grad: -0.000030  (6400/60000)
[Batch 200] loss: 0.489078, mean grad: -0.000005  (12800/60000)
[Batch 300] loss: 0.730771, mean grad: 0.000036  (19200/60000)
[Batch 400] loss: 0.638260, mean grad: 0.000011  (25600/60000)
[Batch 500] loss: 0.588229, mean grad: 0.000044  (32000/60000)
[Batch 600] loss: 0.701332, mean grad: 0.000008  (38400/60000)
[Batch 700] loss: 0.709091, mean grad: 0.000006  (44800/60000)
[Batch 800] loss: 0.676616, mean grad: -0.000027  (51200/60000)
[Batch 900] loss: 0.699410, mean grad: 0.000021  (57600/60000)
[Batch 000] loss: 0.226211, mean grad: 0.000062  (0/60000)
[Batch 100] loss: 0.283047, mean grad: 0.000115  (6400/60000)
[Batch 200] loss: 0.203709, mean grad: 0.000031  (12800/60000)
[Batch 300] loss: 0.328496, mean grad: -0.000062  (19200/60000)
[Batch 400] loss: 0.253776, mean grad: 0.000023  (25600/60000)
[Batch 500] loss: 0.296570, mean grad: 0.000014  (32000/60000)
[Batch 600] loss: 0.300662, mean grad: -0.000197  (38400/60000)
[Batch 700] loss: 0.354225, mean grad: 0.000025  (44800/60000)
[Batch 800] loss: 0.386190, mean grad: -0.000151  (51200/60000)
[Batch 900] loss: 0.332274, mean grad: -0.000373  (57600/60000)
Epoch 26/30
[Batch 000] loss: 0.173200, mean grad: -0.000166  (0/60000)
[Batch 100] loss: 0.294823, mean grad: -0.000091  (6400/60000)
[Batch 200] loss: 0.164860, mean grad: -0.000064  (12800/60000)
[Batch 300] loss: 0.269673, mean grad: 0.000039  (19200/60000)
```

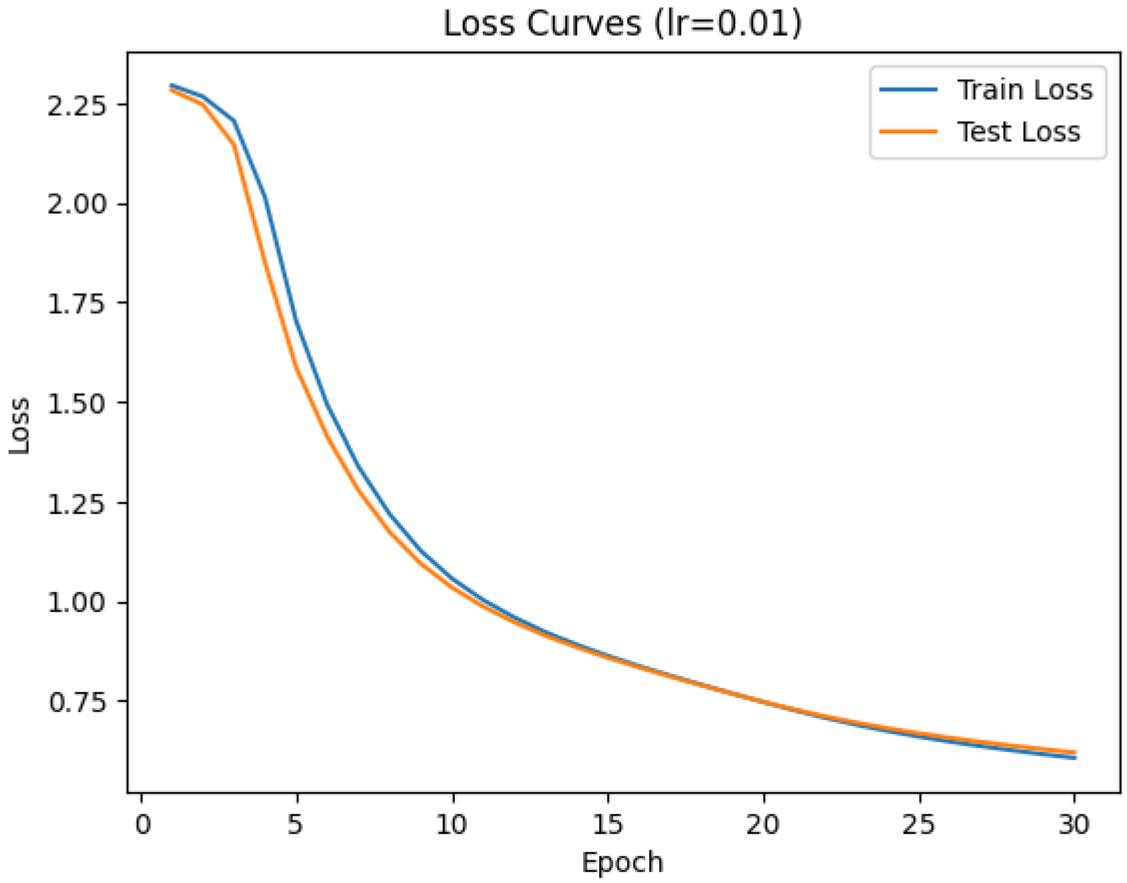
```
[Batch 400] loss: 0.271688, mean grad: 0.000108  (25600/60000)
[Batch 500] loss: 0.300663, mean grad: -0.000070  (32000/60000)
[Batch 600] loss: 0.239613, mean grad: -0.000074  (38400/60000)
[Batch 700] loss: 0.298815, mean grad: 0.000127  (44800/60000)
[Batch 800] loss: 0.323524, mean grad: 0.000152  (51200/60000)
[Batch 900] loss: 0.311877, mean grad: 0.000095  (57600/60000)
[Batch 000] loss: 0.601362, mean grad: -0.000027  (0/60000)
[Batch 100] loss: 0.706723, mean grad: -0.000029  (6400/60000)
[Batch 200] loss: 0.472464, mean grad: -0.000003  (12800/60000)
[Batch 300] loss: 0.719694, mean grad: 0.000036  (19200/60000)
[Batch 400] loss: 0.621823, mean grad: 0.000008  (25600/60000)
[Batch 500] loss: 0.570603, mean grad: 0.000046  (32000/60000)
[Batch 600] loss: 0.688173, mean grad: 0.000008  (38400/60000)
[Batch 700] loss: 0.700539, mean grad: 0.000005  (44800/60000)
[Batch 800] loss: 0.666049, mean grad: -0.000025  (51200/60000)
[Batch 900] loss: 0.687761, mean grad: 0.000021  (57600/60000)
[Batch 000] loss: 0.225050, mean grad: 0.000061  (0/60000)
[Batch 100] loss: 0.279679, mean grad: 0.000114  (6400/60000)
[Batch 200] loss: 0.203036, mean grad: 0.000029  (12800/60000)
[Batch 300] loss: 0.324803, mean grad: -0.000062  (19200/60000)
[Batch 400] loss: 0.251752, mean grad: 0.000026  (25600/60000)
[Batch 500] loss: 0.294671, mean grad: 0.000013  (32000/60000)
[Batch 600] loss: 0.297890, mean grad: -0.000200  (38400/60000)
[Batch 700] loss: 0.349288, mean grad: 0.000022  (44800/60000)
[Batch 800] loss: 0.380541, mean grad: -0.000154  (51200/60000)
[Batch 900] loss: 0.331874, mean grad: -0.000376  (57600/60000)
Epoch 27/30
[Batch 000] loss: 0.171420, mean grad: -0.000152  (0/60000)
[Batch 100] loss: 0.291266, mean grad: -0.000063  (6400/60000)
[Batch 200] loss: 0.162135, mean grad: -0.000049  (12800/60000)
[Batch 300] loss: 0.266581, mean grad: 0.000038  (19200/60000)
[Batch 400] loss: 0.268973, mean grad: 0.000101  (25600/60000)
[Batch 500] loss: 0.295570, mean grad: -0.000069  (32000/60000)
[Batch 600] loss: 0.237122, mean grad: -0.000058  (38400/60000)
[Batch 700] loss: 0.290383, mean grad: 0.000149  (44800/60000)
[Batch 800] loss: 0.320181, mean grad: 0.000174  (51200/60000)
[Batch 900] loss: 0.308138, mean grad: 0.000118  (57600/60000)
[Batch 000] loss: 0.585577, mean grad: -0.000026  (0/60000)
[Batch 100] loss: 0.696026, mean grad: -0.000029  (6400/60000)
[Batch 200] loss: 0.458260, mean grad: -0.000002  (12800/60000)
[Batch 300] loss: 0.709303, mean grad: 0.000036  (19200/60000)
[Batch 400] loss: 0.606437, mean grad: 0.000006  (25600/60000)
[Batch 500] loss: 0.555312, mean grad: 0.000047  (32000/60000)
[Batch 600] loss: 0.675626, mean grad: 0.000007  (38400/60000)
[Batch 700] loss: 0.693269, mean grad: 0.000004  (44800/60000)
[Batch 800] loss: 0.657347, mean grad: -0.000023  (51200/60000)
[Batch 900] loss: 0.676876, mean grad: 0.000022  (57600/60000)
[Batch 000] loss: 0.223905, mean grad: 0.000059  (0/60000)
[Batch 100] loss: 0.276574, mean grad: 0.000113  (6400/60000)
[Batch 200] loss: 0.202321, mean grad: 0.000027  (12800/60000)
[Batch 300] loss: 0.321263, mean grad: -0.000062  (19200/60000)
[Batch 400] loss: 0.249789, mean grad: 0.000029  (25600/60000)
[Batch 500] loss: 0.292876, mean grad: 0.000013  (32000/60000)
[Batch 600] loss: 0.295229, mean grad: -0.000202  (38400/60000)
[Batch 700] loss: 0.344671, mean grad: 0.000019  (44800/60000)
[Batch 800] loss: 0.374918, mean grad: -0.000157  (51200/60000)
[Batch 900] loss: 0.331545, mean grad: -0.000380  (57600/60000)
Epoch 28/30
[Batch 000] loss: 0.172712, mean grad: -0.000160  (0/60000)
[Batch 100] loss: 0.286936, mean grad: -0.000062  (6400/60000)
```

```
[Batch 200] loss: 0.159408, mean grad: -0.000019  (12800/60000)
[Batch 300] loss: 0.261673, mean grad: 0.000053  (19200/60000)
[Batch 400] loss: 0.265920, mean grad: 0.000114  (25600/60000)
[Batch 500] loss: 0.291385, mean grad: -0.000089  (32000/60000)
[Batch 600] loss: 0.229037, mean grad: -0.000046  (38400/60000)
[Batch 700] loss: 0.285039, mean grad: 0.000146  (44800/60000)
[Batch 800] loss: 0.315450, mean grad: 0.000162  (51200/60000)
[Batch 900] loss: 0.300271, mean grad: 0.000099  (57600/60000)
[Batch 000] loss: 0.571817, mean grad: -0.000025  (0/60000)
[Batch 100] loss: 0.686541, mean grad: -0.000028  (6400/60000)
[Batch 200] loss: 0.445991, mean grad: -0.000000  (12800/60000)
[Batch 300] loss: 0.699414, mean grad: 0.000036  (19200/60000)
[Batch 400] loss: 0.592369, mean grad: 0.000004  (25600/60000)
[Batch 500] loss: 0.542001, mean grad: 0.000049  (32000/60000)
[Batch 600] loss: 0.663540, mean grad: 0.000007  (38400/60000)
[Batch 700] loss: 0.686997, mean grad: 0.000002  (44800/60000)
[Batch 800] loss: 0.650085, mean grad: -0.000021  (51200/60000)
[Batch 900] loss: 0.666621, mean grad: 0.000022  (57600/60000)
[Batch 000] loss: 0.222785, mean grad: 0.000058  (0/60000)
[Batch 100] loss: 0.273709, mean grad: 0.000112  (6400/60000)
[Batch 200] loss: 0.201548, mean grad: 0.000026  (12800/60000)
[Batch 300] loss: 0.317862, mean grad: -0.000062  (19200/60000)
[Batch 400] loss: 0.247882, mean grad: 0.000032  (25600/60000)
[Batch 500] loss: 0.291175, mean grad: 0.000012  (32000/60000)
[Batch 600] loss: 0.292678, mean grad: -0.000204  (38400/60000)
[Batch 700] loss: 0.340358, mean grad: 0.000016  (44800/60000)
[Batch 800] loss: 0.369319, mean grad: -0.000160  (51200/60000)
[Batch 900] loss: 0.331280, mean grad: -0.000383  (57600/60000)
Epoch 29/30
[Batch 000] loss: 0.179313, mean grad: -0.000160  (0/60000)
[Batch 100] loss: 0.280746, mean grad: -0.000085  (6400/60000)
[Batch 200] loss: 0.156469, mean grad: -0.000045  (12800/60000)
[Batch 300] loss: 0.256844, mean grad: 0.000036  (19200/60000)
[Batch 400] loss: 0.263104, mean grad: 0.000109  (25600/60000)
[Batch 500] loss: 0.291213, mean grad: -0.000080  (32000/60000)
[Batch 600] loss: 0.228055, mean grad: -0.000051  (38400/60000)
[Batch 700] loss: 0.278510, mean grad: 0.000096  (44800/60000)
[Batch 800] loss: 0.312978, mean grad: 0.000167  (51200/60000)
[Batch 900] loss: 0.295945, mean grad: 0.000065  (57600/60000)
[Batch 000] loss: 0.559806, mean grad: -0.000024  (0/60000)
[Batch 100] loss: 0.678046, mean grad: -0.000027  (6400/60000)
[Batch 200] loss: 0.435245, mean grad: 0.000001  (12800/60000)
[Batch 300] loss: 0.689924, mean grad: 0.000035  (19200/60000)
[Batch 400] loss: 0.579738, mean grad: 0.000002  (25600/60000)
[Batch 500] loss: 0.530356, mean grad: 0.000051  (32000/60000)
[Batch 600] loss: 0.651840, mean grad: 0.000006  (38400/60000)
[Batch 700] loss: 0.681516, mean grad: 0.000001  (44800/60000)
[Batch 800] loss: 0.643911, mean grad: -0.000019  (51200/60000)
[Batch 900] loss: 0.656885, mean grad: 0.000023  (57600/60000)
[Batch 000] loss: 0.221700, mean grad: 0.000056  (0/60000)
[Batch 100] loss: 0.271060, mean grad: 0.000111  (6400/60000)
[Batch 200] loss: 0.200702, mean grad: 0.000024  (12800/60000)
[Batch 300] loss: 0.314586, mean grad: -0.000062  (19200/60000)
[Batch 400] loss: 0.246028, mean grad: 0.000035  (25600/60000)
[Batch 500] loss: 0.289558, mean grad: 0.000011  (32000/60000)
[Batch 600] loss: 0.290237, mean grad: -0.000205  (38400/60000)
[Batch 700] loss: 0.336331, mean grad: 0.000012  (44800/60000)
[Batch 800] loss: 0.363748, mean grad: -0.000164  (51200/60000)
[Batch 900] loss: 0.331069, mean grad: -0.000386  (57600/60000)
Epoch 30/30
```

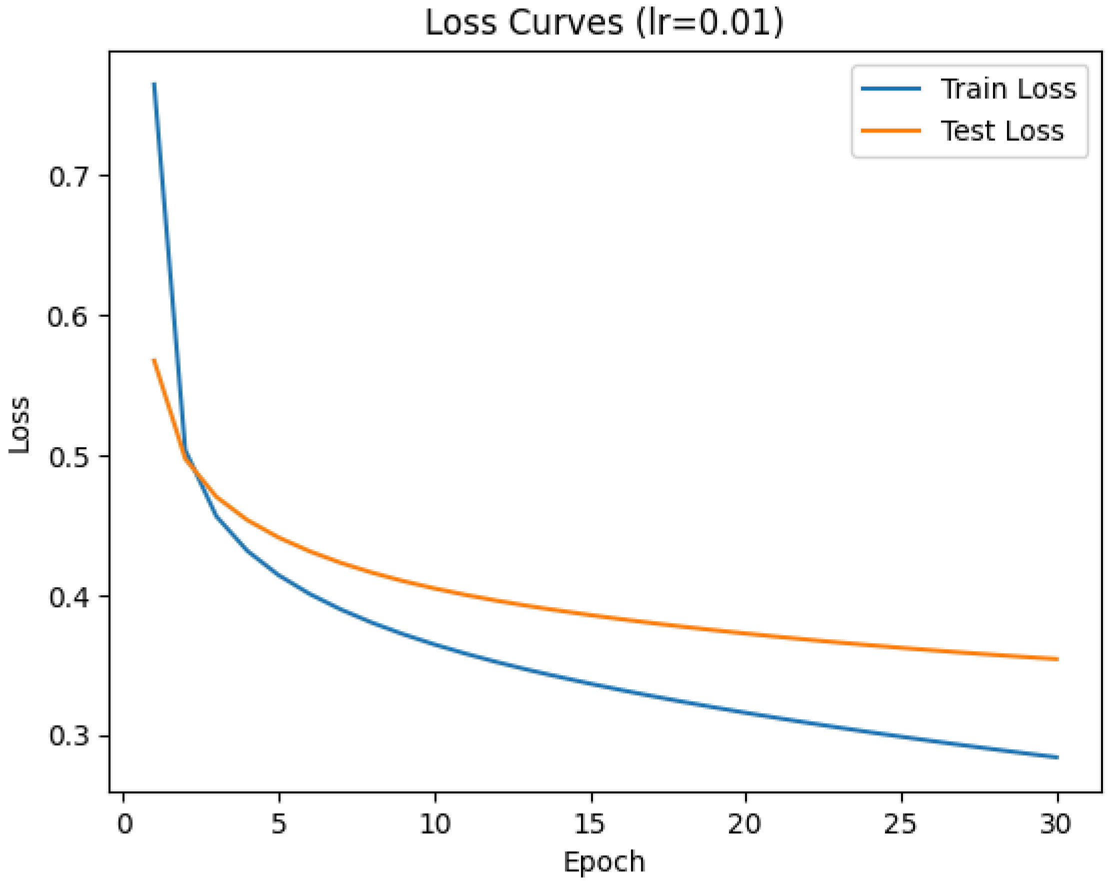
```
[Batch 000] loss: 0.176607, mean grad: -0.000147  (0/60000)
[Batch 100] loss: 0.278806, mean grad: -0.000067  (6400/60000)
[Batch 200] loss: 0.154330, mean grad: -0.000028  (12800/60000)
[Batch 300] loss: 0.252865, mean grad: 0.000033  (19200/60000)
[Batch 400] loss: 0.263106, mean grad: 0.000100  (25600/60000)
[Batch 500] loss: 0.287578, mean grad: -0.000072  (32000/60000)
[Batch 600] loss: 0.223638, mean grad: -0.000057  (38400/60000)
[Batch 700] loss: 0.273674, mean grad: 0.000105  (44800/60000)
[Batch 800] loss: 0.306466, mean grad: 0.000136  (51200/60000)
[Batch 900] loss: 0.289375, mean grad: 0.000065  (57600/60000)
[Batch 000] loss: 0.549302, mean grad: -0.000023  (0/60000)
[Batch 100] loss: 0.670339, mean grad: -0.000026  (6400/60000)
[Batch 200] loss: 0.425687, mean grad: 0.000002  (12800/60000)
[Batch 300] loss: 0.680785, mean grad: 0.000035  (19200/60000)
[Batch 400] loss: 0.568540, mean grad: 0.000000  (25600/60000)
[Batch 500] loss: 0.520109, mean grad: 0.000053  (32000/60000)
[Batch 600] loss: 0.640496, mean grad: 0.000006  (38400/60000)
[Batch 700] loss: 0.676685, mean grad: -0.000001  (44800/60000)
[Batch 800] loss: 0.638551, mean grad: -0.000017  (51200/60000)
[Batch 900] loss: 0.647568, mean grad: 0.000024  (57600/60000)
[Batch 000] loss: 0.220657, mean grad: 0.000055  (0/60000)
[Batch 100] loss: 0.268605, mean grad: 0.000110  (6400/60000)
[Batch 200] loss: 0.199771, mean grad: 0.000022  (12800/60000)
[Batch 300] loss: 0.311423, mean grad: -0.000062  (19200/60000)
[Batch 400] loss: 0.244226, mean grad: 0.000038  (25600/60000)
[Batch 500] loss: 0.288017, mean grad: 0.000010  (32000/60000)
[Batch 600] loss: 0.287902, mean grad: -0.000207  (38400/60000)
[Batch 700] loss: 0.332566, mean grad: 0.000009  (44800/60000)
[Batch 800] loss: 0.358209, mean grad: -0.000167  (51200/60000)
[Batch 900] loss: 0.330900, mean grad: -0.000389  (57600/60000)
ReLU with Xavier initialization
Best accuracy = 87.41% at epoch 29
```

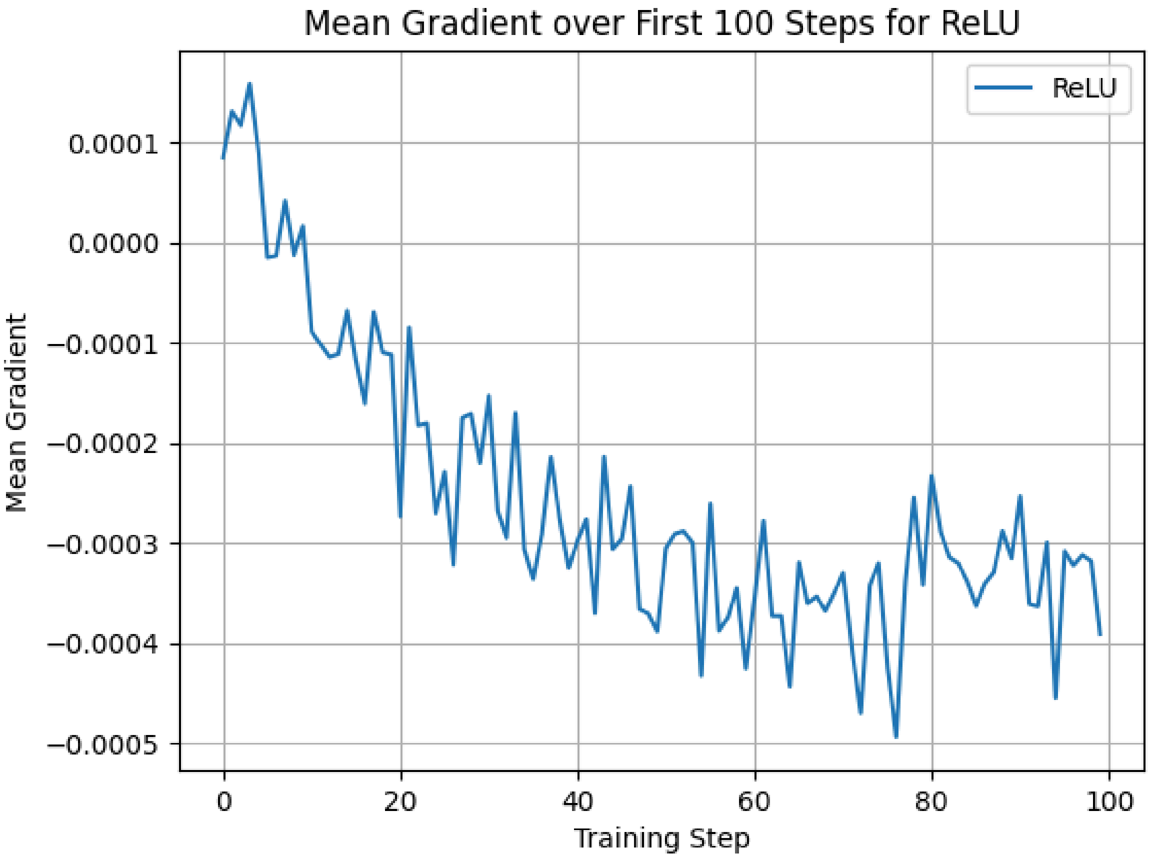


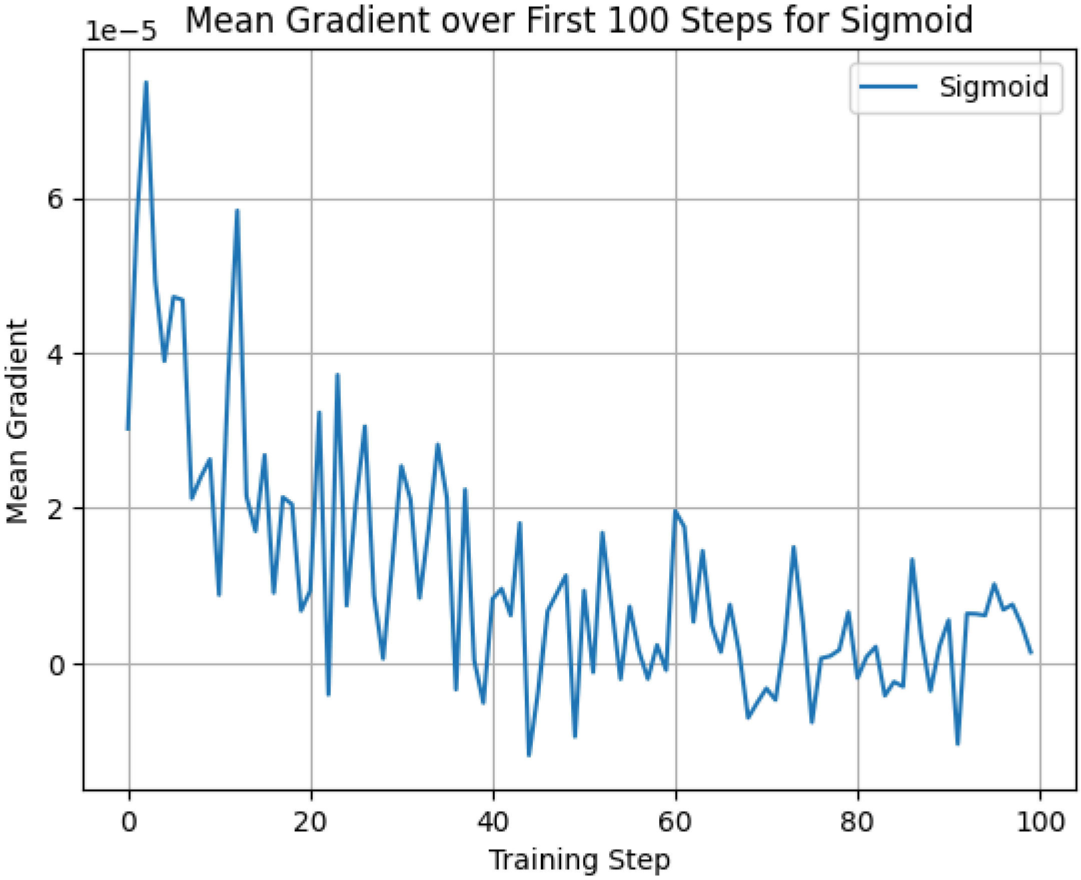
```
Sigmoid with Xavier initialization
Best accuracy = 76.66% at epoch 30
```

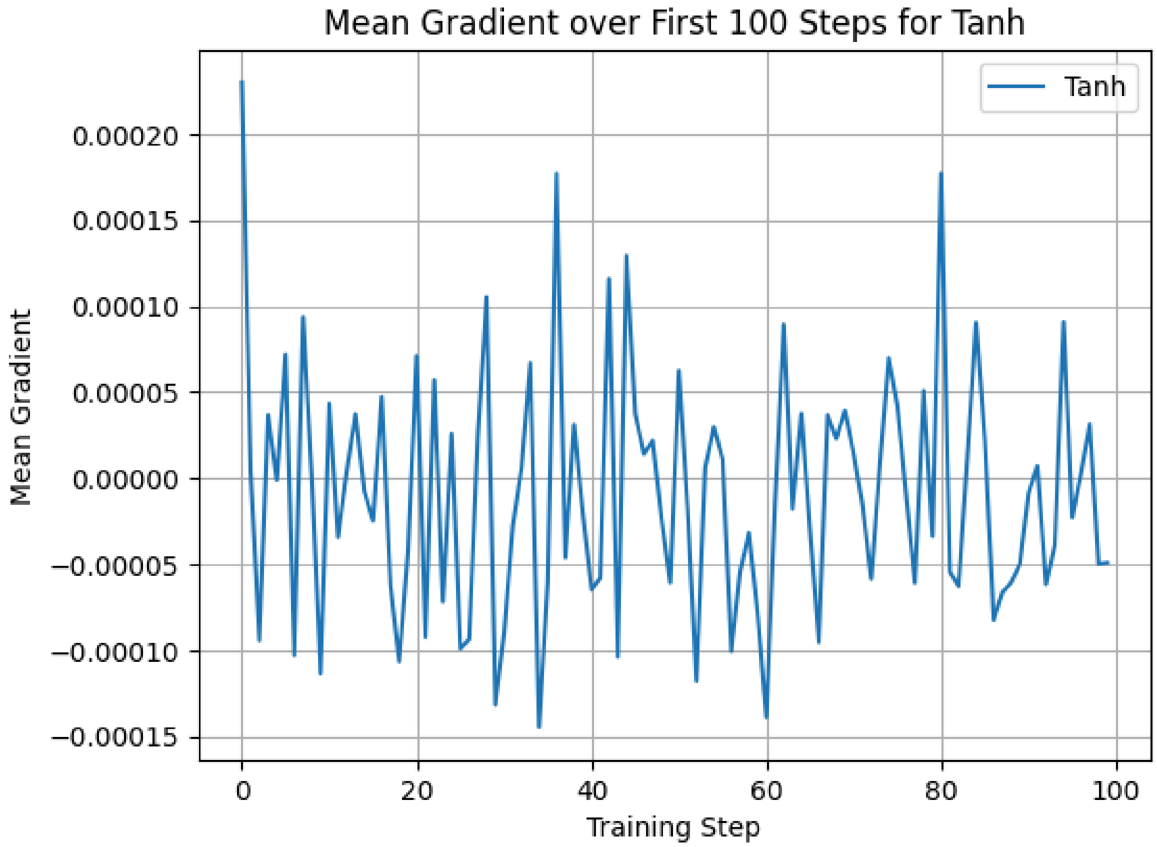


```
Tanh with Xavier initialization
Best accuracy = 87.07% at epoch 30
```




Mean Gradient over First 100 Steps for ReLU
ReLU
0.0001
0.0000
−0.0001
−0.0002
−0.0003
−0.0004
−0.0005
Mean Gradient
0
20
40
60
80
100
Training Step


Mean Gradient over First 100 Steps for Sigmoid
1e−5
Sigmoid
6
4
2
0
Mean Gradient
0
20
40
60
80
100
Training Step



```
Final test accuracies with Xavier initialization:
ReLU: 87.40%
Sigmoid: 76.66%
Tanh: 87.07%
```