

Mansoor Nabawi, 309498

Exercise 06, 17,06,2022

Exercise 1: PyTorch Network Analysis

I am using Kaggle because i can use GPU for 33 hours, but i can't run tensorboard there. As tensorboard does not function in Kaggle i need to re run the code in Colab to get visualization. in Colab I can use GPU for a limited time so i do coding in Kaggle and for the final run i do it in Colab.

References:

- <https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html>
- <https://towardsdatascience.com/implementing-yann-lecuns-lenet-5-in-pytorch-5e05a0911320>
- https://github.com/bentrevett/pytorch-image-classification/blob/master/2_lenet.ipynb
- <https://github.com/Bingmang/pytorch-cifar10-notebook/blob/master/LeNet.ipynb>
- <https://www.javatpoint.com/pytorch-testing-of-lenet-model-for-cifar-10-dataset>
- https://pytorch.org/tutorials/intermediate/tensorboard_tutorial.html
- <https://www.kaggle.com/code/xcwang21/cifar10-with-mlp-lenet-resnet-for-beginners/notebook>

importing libraries

```
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
import torch.utils.data as data

import torchvision.transforms as transforms
import torchvision.datasets as datasets

from sklearn import decomposition
from sklearn import manifold
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
from tqdm.notebook import tqdm, trange
import matplotlib.pyplot as plt
import numpy as np

import copy
import random
import time

from torch.utils.tensorboard import SummaryWriter

%load_ext tensorboard
```

```
The tensorboard extension is already loaded. To reload it, use:
%reload_ext tensorboard
```

set the random seed for reproducability

```
SEED = 1234

random.seed(SEED)
np.random.seed(SEED)
torch.manual_seed(SEED)
torch.cuda.manual_seed(SEED)
torch.backends.cudnn.deterministic = True
```

Loading data and normalizing them using mean and std of it.

```
ROOT = '.data'

train_data = datasets.MNIST(root=ROOT,
                             train=True,
                             download=True)

mean = train_data.data.float().mean() / 255
std = train_data.data.float().std() / 255

print(f'Calculated mean: {mean}')
print(f'Calculated std: {std}')

Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz to .data/MNIST/raw/train-images-idx3-ubyte.gz
9913344/? [00:00<00:00, 25209581.75it/s]

Extracting .data/MNIST/raw/train-images-idx3-ubyte.gz to .data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz to .data/MNIST/raw/train-labels-idx1-ubyte.gz
29696/? [00:00<00:00, 697106.17it/s]

Extracting .data/MNIST/raw/train-labels-idx1-ubyte.gz to .data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz to .data/MNIST/raw/t10k-images-idx3-ubyte.gz
1649664/? [00:00<00:00, 18014601.64it/s]

Extracting .data/MNIST/raw/t10k-images-idx3-ubyte.gz to .data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz to .data/MNIST/raw/t10k-labels-idx1-ubyte.gz
5120/? [00:00<00:00, 183305.90it/s]

Extracting .data/MNIST/raw/t10k-labels-idx1-ubyte.gz to .data/MNIST/raw

Calculated mean: 0.13066047430038452
Calculated std: 0.30810779333114624
```

defining transformations

```
train_transforms = transforms.Compose([
    transforms.RandomRotation(5, fill=(0,)),
    transforms.RandomCrop(28, padding=2),
    transforms.ToTensor(),
    transforms.Normalize(mean=[mean], std=[std])
])

test_transforms = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize(mean=[mean], std=[std])
])
```

```
train_data = datasets.MNIST(root=ROOT,
                             train=True,
                             download=True,
                             transform=train_transforms)

test_data = datasets.MNIST(root=ROOT,
                            train=False,
                            download=True,
                            transform=test_transforms)
```

Data Loader definition

```
BATCH_SIZE = 128

train_iterator = data.DataLoader(train_data,
                                 shuffle=True,
                                 num_workers=2,
                                 batch_size=BATCH_SIZE)

test_iterator = data.DataLoader(test_data,
                                num_workers=2,
                                batch_size=BATCH_SIZE,
                                shuffle = False)
```

Defining our LeNet model

- I used an architecture provided in one of the references.

```
class LeNet(nn.Module):
    def __init__(self, output_dim):
        super().__init__()

        self.conv1 = nn.Conv2d(in_channels=1,
                                out_channels=6,
                                kernel_size=5)

        self.conv2 = nn.Conv2d(in_channels=6,
                                out_channels=16,
                                kernel_size=5)

        self.fc_1 = nn.Linear(16 * 4 * 4, 120)
        self.fc_2 = nn.Linear(120, 84)
        self.fc_3 = nn.Linear(84, output_dim)

    def forward(self, x):

        # x = [batch size, 1, 28, 28]

        x = self.conv1(x)

        # x = [batch size, 6, 24, 24]

        x = F.max_pool2d(x, kernel_size=2)

        # x = [batch size, 6, 12, 12]

        x = F.relu(x)

        x = self.conv2(x)

        # x = [batch size, 16, 8, 8]

        x = F.max_pool2d(x, kernel_size=2)

        # x = [batch size, 16, 4, 4]

        x = F.relu(x)

        x = x.view(x.shape[0], -1)

        # x = [batch size, 16*4*4 = 256]

        h = x

        x = self.fc_1(x)

        # x = [batch size, 120]

        x = F.relu(x)

        x = self.fc_2(x)

        # x = batch size, 84]

        x = F.relu(x)

        x = self.fc_3(x)

        # x = [batch size, output dim]

        return x, h
```

```
OUTPUT_DIM = 10

model = LeNet(OUTPUT_DIM)

def count_parameters(model):
```

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

print(f'The model has {count_parameters(model):,} trainable parameters')

optimizer = optim.Adam(model.parameters())

criterion = nn.CrossEntropyLoss()

#Checking if we can use GPU

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

model = model.to(device)
criterion = criterion.to(device)
```

The model has 44,426 trainable parameters

```
def calculate_accuracy(y_pred, y):
    top_pred = y_pred.argmax(1, keepdim=True)
    correct = top_pred.eq(y.view_as(top_pred)).sum()
    acc = correct.float() / y.shape[0]
    return acc
```

train and test loop

```
def train(model, iterator, optimizer, criterion, device):

    epoch_loss = 0
    epoch_acc = 0

    model.train()

    for (x, y) in tqdm(iterator, desc="Training", leave=False):

        x = x.to(device)
        y = y.to(device)

        optimizer.zero_grad()

        y_pred, _ = model(x)

        loss = criterion(y_pred, y)

        acc = calculate_accuracy(y_pred, y)

        loss.backward()

        optimizer.step()

        epoch_loss += loss.item()
        epoch_acc += acc.item()

    return epoch_loss / len(iterator), epoch_acc / len(iterator)
```

```
def test(model, iterator, criterion, device):

    epoch_loss = 0
    epoch_acc = 0

    model.eval()

    with torch.no_grad():

        for (x, y) in tqdm(iterator, desc="Testing", leave=False):

            x = x.to(device)
            y = y.to(device)

            y_pred, _ = model(x)

            loss = criterion(y_pred, y)

            acc = calculate_accuracy(y_pred, y)

            epoch_loss += loss.item()
            epoch_acc += acc.item()

    return epoch_loss / len(iterator), epoch_acc / len(iterator)
```

```
def epoch_time(start_time, end_time):
    elapsed_time = end_time - start_time
    elapsed_mins = int(elapsed_time / 60)
    elapsed_secs = int(elapsed_time - (elapsed_mins * 60))
    return elapsed_mins, elapsed_secs
```

using Tensorboard for visualization

```
logs = 'runs/mnist_LeNet'
tb = SummaryWriter(logs)
```

```
print("Optimizer:", optimizer)
```

This was just for one fixed optimizer and learning rate

```
EPOCHS = 20

# best_valid_loss = float('inf')

learning_rate = 0.001
optimizer = optim.Adam(model.parameters(), lr=learning_rate)

print("Optimizer:", optimizer)
for epoch in trange(EPOCHS, desc="Epochs"):

    start_time = time.monotonic()
```

```
train_loss, train_acc = train(model, train_iterator, optimizer, criterion, device)
test_loss, test_acc = test(model, test_iterator, criterion, device)

end_time = time.monotonic()

epoch_mins, epoch_secs = epoch_time(start_time, end_time)

print(f'Epoch: {epoch+1:02} | Epoch Time: {epoch_mins}m {epoch_secs}s')
print(f'\tTrain Loss: {train_loss:.3f} | Train Acc: {train_acc*100:.2f}%')
print(f'\t Test. Loss: {test_loss:.3f} | Val. Acc: {test_acc*100:.2f}%')

if tb is not None:
    tb.add_scalars(f'Loss_{learning_rate}', {"Train":train_loss,
                                             "Test":test_loss}, epoch)
    tb.add_scalars(f'Accuracy_{learning_rate}', {"Train":train_acc,
                                                "Test":test_acc}, epoch)

    tb.add_scalars(f'Train_Accuracy', {f"{learning_rate}":train_acc}, epoch)
    tb.add_scalars(f'Test_Accuracy', {f"{learning_rate}":test_acc}, epoch)

    tb.add_scalars(f'Train_loss', {f"{learning_rate}":train_loss}, epoch)
    tb.add_scalars(f'Test_loss', {f"{learning_rate}":test_loss}, epoch)
```

A loop over different learning rates with chosen optimizer, first we use Adam.

- for every learning rate, we initialize a LeNet model and train it for 20 epochs, in every epoch we show the loss and accuracy.
- we also use tensorboard for later visualization.

```
selected_optimizer='Adam'
for learning_rate in [ 0.01, 0.001, 0.0001]:

    OUTPUT_DIM = 10

    model = LeNet(OUTPUT_DIM)

    # def count_parameters(model):
    #     return sum(p.numel() for p in model.parameters() if p.requires_grad)

    # print(f'The model has {count_parameters(model):,} trainable parameters')

    optimizer = optim.Adam(model.parameters())

    criterion = nn.CrossEntropyLoss()

    #Checking if we can use GPU

    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

    model = model.to(device)
    criterion = criterion.to(device)

    avail_optimizers = {'Adam':torch.optim.Adam(model.parameters(), lr=learning_rate, betas=(0.9, 0.999), eps=1e-08, weight_decay=0, amsgrad=False),
                        'RMS': torch.optim.RMSprop(model.parameters(), lr=learning_rate, alpha=0.99, eps=1e-08, weight_decay=0, momentum=0, centered=False),
                        'SGD': torch.optim.SGD(model.parameters(), lr=learning_rate, momentum=0, dampening=0, weight_decay=0, nesterov=False)}

    optimizer = avail_optimizers[selected_optimizer]

    epochs = 20
    print(f"selected optimizer and learning rate: {optimizer}")
    for n_iter in tqdm(range(epochs)):
        print(f"Epoch {n_iter+1}/{epochs}")

        start_time = time.monotonic()

        train_loss, train_acc = train(model, train_iterator, optimizer, criterion, device)
        test_loss, test_acc = test(model, test_iterator, criterion, device)

        end_time = time.monotonic()

        epoch_mins, epoch_secs = epoch_time(start_time, end_time)

        print(f'Epoch: {n_iter+1:02} | Epoch Time: {epoch_mins}m {epoch_secs}s')
        print(f'\tTrain Loss: {train_loss:.3f} | Train Acc: {train_acc*100:.2f}%')
        print(f'\t Test. Loss: {test_loss:.3f} | Test. Acc: {test_acc*100:.2f}%')

        if tb is not None:
            tb.add_scalars(f'Loss_{learning_rate}', {"Train":train_loss,
                                                     "Test":test_loss}, n_iter)
            tb.add_scalars(f'Accuracy_{learning_rate}', {"Train":train_acc,
                                                         "Test":test_acc}, n_iter)

            tb.add_scalars(f'Train_Accuracy', {f"{learning_rate}":train_acc}, n_iter)
            tb.add_scalars(f'Test_Accuracy', {f"{learning_rate}":test_acc}, n_iter)

            tb.add_scalars(f'Train_loss', {f"{learning_rate}":train_loss}, n_iter)
            tb.add_scalars(f'Test_loss', {f"{learning_rate}":test_loss}, n_iter)
```

```
selected optimizer and learning rate: Adam (
Parameter Group 0
  amsgrad: False
  betas: (0.9, 0.999)
  eps: 1e-08
  lr: 0.01
  maximize: False
  weight_decay: 0
)
100% 20/20 [06:56<00:00, 20.61s/it]

Epoch 1/20
Training: 100% 469/469 [00:19<00:00, 27.40it/s]

Testing: 99% 78/79 [00:01<00:00, 50.38it/s]
Epoch: 01 | Epoch Time: 0m 21s
      Train Loss: 0.368 | Train Acc: 88.01%
      Test. Loss: 0.086 | Test. Acc: 97.27%
Epoch 2/20
Training: 99% 466/469 [00:18<00:00, 25.99it/s]

Testing: 95% 75/79 [00:01<00:00, 48.34it/s]
Epoch: 02 | Epoch Time: 0m 20s
      Train Loss: 0.153 | Train Acc: 95.40%
      Test. Loss: 0.077 | Test. Acc: 97.68%
Epoch 3/20
Training: 100% 468/469 [00:18<00:00, 27.06it/s]

Testing: 99% 78/79 [00:01<00:00, 49.83it/s]
Epoch: 03 | Epoch Time: 0m 20s
      Train Loss: 0.138 | Train Acc: 95.91%
      Test. Loss: 0.085 | Test. Acc: 97.54%
Epoch 4/20
Training: 99% 466/469 [00:18<00:00, 25.81it/s]

Testing: 94% 74/79 [00:02<00:00, 46.41it/s]
Epoch: 04 | Epoch Time: 0m 21s
      Train Loss: 0.129 | Train Acc: 96.29%
      Test. Loss: 0.070 | Test. Acc: 98.02%
Epoch 5/20
Training: 100% 468/469 [00:19<00:00, 25.96it/s]

Testing: 95% 75/79 [00:01<00:00, 46.36it/s]
Epoch: 05 | Epoch Time: 0m 21s
      Train Loss: 0.125 | Train Acc: 96.52%
      Test. Loss: 0.075 | Test. Acc: 97.94%
Epoch 6/20
Training: 99% 466/469 [00:18<00:00, 25.90it/s]

Testing: 94% 74/79 [00:01<00:00, 48.63it/s]
Epoch: 06 | Epoch Time: 0m 20s
      Train Loss: 0.128 | Train Acc: 96.39%
      Test. Loss: 0.069 | Test. Acc: 97.88%
Epoch 7/20
Training: 100% 469/469 [00:18<00:00, 27.29it/s]

Testing: 94% 74/79 [00:01<00:00, 46.69it/s]
Epoch: 07 | Epoch Time: 0m 20s
      Train Loss: 0.119 | Train Acc: 96.68%
      Test. Loss: 0.077 | Test. Acc: 97.91%
Epoch 8/20
Training: 100% 468/469 [00:18<00:00, 27.01it/s]

Testing: 99% 78/79 [00:01<00:00, 48.06it/s]
Epoch: 08 | Epoch Time: 0m 20s
      Train Loss: 0.119 | Train Acc: 96.65%
      Test. Loss: 0.070 | Test. Acc: 98.08%
Epoch 9/20
Training: 100% 468/469 [00:18<00:00, 25.64it/s]

Testing: 94% 74/79 [00:01<00:00, 45.78it/s]
Epoch: 09 | Epoch Time: 0m 20s
      Train Loss: 0.123 | Train Acc: 96.64%
      Test. Loss: 0.076 | Test. Acc: 97.97%
Epoch 10/20
Training: 100% 468/469 [00:18<00:00, 26.81it/s]

Testing: 92% 73/79 [00:01<00:00, 46.89it/s]
Epoch: 10 | Epoch Time: 0m 20s
      Train Loss: 0.120 | Train Acc: 96.73%
      Test. Loss: 0.075 | Test. Acc: 97.79%
Epoch 11/20
Training: 100% 468/469 [00:19<00:00, 26.44it/s]

Testing: 96% 76/79 [00:01<00:00, 49.53it/s]
Epoch: 11 | Epoch Time: 0m 21s
      Train Loss: 0.120 | Train Acc: 96.79%
      Test. Loss: 0.063 | Test. Acc: 98.19%
Epoch 12/20
Training: 100% 467/469 [00:18<00:00, 23.02it/s]

Testing: 97% 77/79 [00:01<00:00, 47.51it/s]
Epoch: 12 | Epoch Time: 0m 20s
      Train Loss: 0.114 | Train Acc: 96.86%
      Test. Loss: 0.062 | Test. Acc: 98.29%
Epoch 13/20
Training: 99% 466/469 [00:18<00:00, 25.08it/s]

Testing: 92% 73/79 [00:01<00:00, 44.05it/s]
Epoch: 13 | Epoch Time: 0m 21s
      Train Loss: 0.112 | Train Acc: 97.05%
      Test. Loss: 0.072 | Test. Acc: 98.09%
Epoch 14/20
Training: 99% 466/469 [00:18<00:00, 23.82it/s]

Testing: 96% 76/79 [00:01<00:00, 46.92it/s]
Epoch: 14 | Epoch Time: 0m 21s
      Train Loss: 0.111 | Train Acc: 97.11%
      Test. Loss: 0.055 | Test. Acc: 98.44%
Epoch 15/20
Training: 100% 468/469 [00:18<00:00, 26.63it/s]

Testing: 94% 74/79 [00:01<00:00, 45.96it/s]
Epoch: 15 | Epoch Time: 0m 20s
      Train Loss: 0.105 | Train Acc: 97.27%
```

```
      train Loss: 0.105 | Train Acc: 97.27%
      Test. Loss: 0.054 | Test. Acc: 98.47%
Epoch 16/20
Training: 100%                                467/469 [00:18<00:00, 25.11it/s]

Testing: 95%                                75/79 [00:01<00:00, 43.47it/s]
Epoch: 16 | Epoch Time: 0m 20s
      Train Loss: 0.113 | Train Acc: 97.03%
      Test. Loss: 0.086 | Test. Acc: 97.76%
Epoch 17/20
Training: 100%                                467/469 [00:18<00:00, 26.48it/s]

Testing: 94%                                74/79 [00:01<00:00, 45.14it/s]
Epoch: 17 | Epoch Time: 0m 20s
      Train Loss: 0.112 | Train Acc: 97.11%
      Test. Loss: 0.083 | Test. Acc: 97.77%
Epoch 18/20
Training: 100%                                468/469 [00:18<00:00, 26.30it/s]

Testing: 95%                                75/79 [00:01<00:00, 48.49it/s]
Epoch: 18 | Epoch Time: 0m 20s
      Train Loss: 0.108 | Train Acc: 97.16%
      Test. Loss: 0.086 | Test. Acc: 97.83%
Epoch 19/20
Training: 99%                                465/469 [00:18<00:00, 23.58it/s]

Testing: 94%                                74/79 [00:01<00:00, 48.35it/s]
Epoch: 19 | Epoch Time: 0m 20s
      Train Loss: 0.108 | Train Acc: 97.18%
      Test. Loss: 0.061 | Test. Acc: 98.38%
Epoch 20/20
Training: 100%                                467/469 [00:18<00:00, 26.85it/s]

Testing: 92%                                73/79 [00:01<00:00, 48.55it/s]
Epoch: 20 | Epoch Time: 0m 20s
      Train Loss: 0.107 | Train Acc: 97.30%
      Test. Loss: 0.076 | Test. Acc: 98.24%
selected optimizer and learning rate: Adam (
Parameter Group 0
  amsgrad: False
  betas: (0.9, 0.999)
  eps: 1e-08
  lr: 0.001
  maximize: False
  weight_decay: 0
)
100%                                20/20 [07:00<00:00, 21.17s/it]
Epoch 1/20
Training: 99%                                466/469 [00:18<00:00, 26.20it/s]

Testing: 96%                                76/79 [00:01<00:00, 46.97it/s]
Epoch: 01 | Epoch Time: 0m 20s
      Train Loss: 0.481 | Train Acc: 84.89%
      Test. Loss: 0.142 | Test. Acc: 95.53%
Epoch 2/20
Training: 100%                                468/469 [00:18<00:00, 25.53it/s]

Testing: 96%                                76/79 [00:01<00:00, 46.38it/s]
Epoch: 02 | Epoch Time: 0m 20s
      Train Loss: 0.147 | Train Acc: 95.44%
      Test. Loss: 0.066 | Test. Acc: 97.94%
Epoch 3/20
Training: 100%                                467/469 [00:18<00:00, 25.99it/s]

Testing: 96%                                76/79 [00:01<00:00, 49.10it/s]
Epoch: 03 | Epoch Time: 0m 20s
      Train Loss: 0.111 | Train Acc: 96.65%
      Test. Loss: 0.062 | Test. Acc: 98.10%
Epoch 4/20
Training: 100%                                467/469 [00:18<00:00, 24.55it/s]

Testing: 95%                                75/79 [00:01<00:00, 47.17it/s]
Epoch: 04 | Epoch Time: 0m 21s
      Train Loss: 0.090 | Train Acc: 97.22%
      Test. Loss: 0.046 | Test. Acc: 98.55%
Epoch 5/20
Training: 100%                                469/469 [00:19<00:00, 25.89it/s]

Testing: 99%                                78/79 [00:01<00:00, 46.28it/s]
Epoch: 05 | Epoch Time: 0m 21s
      Train Loss: 0.078 | Train Acc: 97.51%
      Test. Loss: 0.040 | Test. Acc: 98.70%
Epoch 6/20
Training: 99%                                466/469 [00:19<00:00, 25.51it/s]

Testing: 95%                                75/79 [00:01<00:00, 48.22it/s]
Epoch: 06 | Epoch Time: 0m 21s
      Train Loss: 0.069 | Train Acc: 97.85%
      Test. Loss: 0.033 | Test. Acc: 98.91%
Epoch 7/20
Training: 100%                                467/469 [00:18<00:00, 24.21it/s]

Testing: 92%                                73/79 [00:01<00:00, 45.66it/s]
Epoch: 07 | Epoch Time: 0m 20s
      Train Loss: 0.063 | Train Acc: 98.08%
      Test. Loss: 0.038 | Test. Acc: 98.73%
Epoch 8/20
Training: 99%                                466/469 [00:19<00:00, 23.83it/s]

Testing: 91%                                72/79 [00:01<00:00, 45.50it/s]
Epoch: 08 | Epoch Time: 0m 21s
      Train Loss: 0.058 | Train Acc: 98.27%
      Test. Loss: 0.034 | Test. Acc: 98.95%
Epoch 9/20
Training: 100%                                467/469 [00:19<00:00, 25.71it/s]

Testing: 100%                                79/79 [00:01<00:00, 51.60it/s]
Epoch: 09 | Epoch Time: 0m 21s
      Train Loss: 0.053 | Train Acc: 98.40%
      Test. Loss: 0.048 | Test. Acc: 98.44%
Epoch 10/20
Training: 100%                                467/469 [00:18<00:00, 23.90it/s]

Testing: 92%                                73/79 [00:01<00:00, 45.73it/s]
```

Epoch: 10 Epoch Time: 0m 20s Train Loss: 0.053 Train Acc: 98.37% Test. Loss: 0.030 Test. Acc: 98.97%	
Epoch 11/20	
Training: 100%	467/469 [00:18<00:00, 24.60it/s]
Testing: 95%	75/79 [00:01<00:00, 44.87it/s]
Epoch: 11 Epoch Time: 0m 20s Train Loss: 0.045 Train Acc: 98.60% Test. Loss: 0.029 Test. Acc: 99.04%	
Epoch 12/20	
Training: 100%	467/469 [00:18<00:00, 26.51it/s]
Testing: 92%	73/79 [00:01<00:00, 45.99it/s]
Epoch: 12 Epoch Time: 0m 20s Train Loss: 0.045 Train Acc: 98.58% Test. Loss: 0.029 Test. Acc: 99.03%	
Epoch 13/20	
Training: 100%	467/469 [00:18<00:00, 26.01it/s]
Testing: 96%	76/79 [00:01<00:00, 46.17it/s]
Epoch: 13 Epoch Time: 0m 20s Train Loss: 0.043 Train Acc: 98.66% Test. Loss: 0.028 Test. Acc: 99.13%	
Epoch 14/20	
Training: 99%	465/469 [00:19<00:00, 23.38it/s]
Testing: 97%	77/79 [00:01<00:00, 46.09it/s]
Epoch: 14 Epoch Time: 0m 21s Train Loss: 0.040 Train Acc: 98.78% Test. Loss: 0.029 Test. Acc: 99.08%	
Epoch 15/20	
Training: 100%	468/469 [00:20<00:00, 24.96it/s]
Testing: 99%	78/79 [00:01<00:00, 46.91it/s]
Epoch: 15 Epoch Time: 0m 22s Train Loss: 0.038 Train Acc: 98.80% Test. Loss: 0.025 Test. Acc: 99.19%	
Epoch 16/20	
Training: 100%	468/469 [00:18<00:00, 25.30it/s]
Testing: 96%	76/79 [00:01<00:00, 45.01it/s]
Epoch: 16 Epoch Time: 0m 20s Train Loss: 0.038 Train Acc: 98.81% Test. Loss: 0.028 Test. Acc: 99.11%	
Epoch 17/20	
Training: 100%	468/469 [00:18<00:00, 23.97it/s]
Testing: 96%	76/79 [00:01<00:00, 44.98it/s]
Epoch: 17 Epoch Time: 0m 20s Train Loss: 0.036 Train Acc: 98.88% Test. Loss: 0.028 Test. Acc: 99.16%	
Epoch 18/20	
Training: 100%	467/469 [00:18<00:00, 24.45it/s]
Testing: 92%	73/79 [00:01<00:00, 45.23it/s]
Epoch: 18 Epoch Time: 0m 20s Train Loss: 0.035 Train Acc: 98.85% Test. Loss: 0.028 Test. Acc: 99.09%	
Epoch 19/20	
Training: 100%	468/469 [00:19<00:00, 25.37it/s]
Testing: 94%	74/79 [00:01<00:00, 45.25it/s]
Epoch: 19 Epoch Time: 0m 21s Train Loss: 0.034 Train Acc: 98.90% Test. Loss: 0.031 Test. Acc: 98.94%	
Epoch 20/20	
Training: 100%	467/469 [00:19<00:00, 23.46it/s]
Testing: 95%	75/79 [00:01<00:00, 43.19it/s]
Epoch: 20 Epoch Time: 0m 21s Train Loss: 0.032 Train Acc: 98.98% Test. Loss: 0.025 Test. Acc: 99.21%	
selected optimizer and learning rate: Adam (
Parameter Group 0	
amsgrad: False	
betas: (0.9, 0.999)	
eps: 1e-08	
lr: 0.0001	
maximize: False	
weight_decay: 0	
)	
100%	20/20 [07:01<00:00, 21.22s/it]
Epoch 1/20	
Training: 99%	466/469 [00:19<00:00, 25.24it/s]
Testing: 96%	76/79 [00:01<00:00, 46.87it/s]
Epoch: 01 Epoch Time: 0m 21s Train Loss: 1.361 Train Acc: 57.69% Test. Loss: 0.403 Test. Acc: 89.44%	
Epoch 2/20	
Training: 99%	466/469 [00:19<00:00, 23.33it/s]
Testing: 97%	77/79 [00:01<00:00, 46.20it/s]
Epoch: 02 Epoch Time: 0m 21s Train Loss: 0.495 Train Acc: 85.37% Test. Loss: 0.206 Test. Acc: 94.31%	
Epoch 3/20	
Training: 100%	467/469 [00:18<00:00, 25.11it/s]
Testing: 95%	75/79 [00:01<00:00, 46.14it/s]
Epoch: 03 Epoch Time: 0m 21s Train Loss: 0.336 Train Acc: 89.90% Test. Loss: 0.151 Test. Acc: 95.57%	
Epoch 4/20	
Training: 100%	468/469 [00:18<00:00, 27.37it/s]
Testing: 97%	77/79 [00:01<00:00, 45.82it/s]
Epoch: 04 Epoch Time: 0m 20s Train Loss: 0.268 Train Acc: 92.15% Test. Loss: 0.125 Test. Acc: 96.24%	
Epoch 5/20	
Training: 100%	467/469 [00:18<00:00, 26.38it/s]
Testing: 97%	77/79 [00:01<00:00, 47.27it/s]

```
Testing: 97%
Epoch: 05 | Epoch Time: 0m 20s
      Train Loss: 0.226 | Train Acc: 93.17%
      Test. Loss: 0.112 | Test. Acc: 96.45%
Epoch 6/20
Training: 100%
467/469 [00:18<00:00, 25.71it/s]

Testing: 97%
77/79 [00:01<00:00, 47.40it/s]
Epoch: 06 | Epoch Time: 0m 20s
      Train Loss: 0.199 | Train Acc: 94.07%
      Test. Loss: 0.095 | Test. Acc: 96.83%
Epoch 7/20
Training: 100%
468/469 [00:18<00:00, 24.40it/s]

Testing: 94%
74/79 [00:01<00:00, 42.93it/s]
Epoch: 07 | Epoch Time: 0m 20s
      Train Loss: 0.178 | Train Acc: 94.65%
      Test. Loss: 0.089 | Test. Acc: 97.14%
Epoch 8/20
Training: 99%
466/469 [00:18<00:00, 23.38it/s]

Testing: 97%
77/79 [00:01<00:00, 46.87it/s]
Epoch: 08 | Epoch Time: 0m 21s
      Train Loss: 0.162 | Train Acc: 95.12%
      Test. Loss: 0.084 | Test. Acc: 97.43%
Epoch 9/20
Training: 99%
466/469 [00:19<00:00, 26.26it/s]

Testing: 96%
76/79 [00:01<00:00, 46.28it/s]
Epoch: 09 | Epoch Time: 0m 21s
      Train Loss: 0.149 | Train Acc: 95.49%
      Test. Loss: 0.074 | Test. Acc: 97.77%
Epoch 10/20
Training: 100%
467/469 [00:18<00:00, 23.92it/s]

Testing: 92%
73/79 [00:01<00:00, 46.16it/s]
Epoch: 10 | Epoch Time: 0m 20s
      Train Loss: 0.139 | Train Acc: 95.80%
      Test. Loss: 0.069 | Test. Acc: 97.87%
Epoch 11/20
Training: 100%
467/469 [00:18<00:00, 25.39it/s]

Testing: 95%
75/79 [00:01<00:00, 46.53it/s]
Epoch: 11 | Epoch Time: 0m 20s
      Train Loss: 0.133 | Train Acc: 95.94%
      Test. Loss: 0.069 | Test. Acc: 97.71%
Epoch 12/20
Training: 100%
468/469 [00:18<00:00, 26.40it/s]

Testing: 92%
73/79 [00:01<00:00, 45.39it/s]
Epoch: 12 | Epoch Time: 0m 20s
      Train Loss: 0.125 | Train Acc: 96.17%
      Test. Loss: 0.062 | Test. Acc: 98.00%
Epoch 13/20
Training: 100%
468/469 [00:19<00:00, 24.86it/s]

Testing: 96%
76/79 [00:01<00:00, 47.47it/s]
Epoch: 13 | Epoch Time: 0m 21s
      Train Loss: 0.118 | Train Acc: 96.37%
      Test. Loss: 0.059 | Test. Acc: 98.13%
Epoch 14/20
Training: 100%
467/469 [00:19<00:00, 24.59it/s]

Testing: 97%
77/79 [00:01<00:00, 46.76it/s]
Epoch: 14 | Epoch Time: 0m 21s
      Train Loss: 0.114 | Train Acc: 96.49%
      Test. Loss: 0.057 | Test. Acc: 98.22%
Epoch 15/20
Training: 99%
466/469 [00:19<00:00, 23.16it/s]

Testing: 94%
74/79 [00:01<00:00, 46.13it/s]
Epoch: 15 | Epoch Time: 0m 21s
      Train Loss: 0.111 | Train Acc: 96.59%
      Test. Loss: 0.055 | Test. Acc: 98.22%
Epoch 16/20
Training: 100%
468/469 [00:19<00:00, 25.81it/s]

Testing: 92%
73/79 [00:01<00:00, 44.02it/s]
Epoch: 16 | Epoch Time: 0m 21s
      Train Loss: 0.105 | Train Acc: 96.83%
      Test. Loss: 0.054 | Test. Acc: 98.22%
Epoch 17/20
Training: 100%
468/469 [00:19<00:00, 25.18it/s]

Testing: 96%
76/79 [00:01<00:00, 47.15it/s]
Epoch: 17 | Epoch Time: 0m 21s
      Train Loss: 0.100 | Train Acc: 96.90%
      Test. Loss: 0.052 | Test. Acc: 98.24%
Epoch 18/20
Training: 99%
466/469 [00:19<00:00, 24.96it/s]

Testing: 94%
74/79 [00:01<00:00, 44.81it/s]
Epoch: 18 | Epoch Time: 0m 21s
```

%tensorboard --logdir runs/mnist_LeNet

Selecting TensorBoard with logdir runs/mnist_LeNet (started 0:01:21 ago; port 6006, pid 1625).

TensorBoard
SCALARS TIME SERIES
INACTIVE

☐ Show data download links
☐ Ignore outliers in chart scaling

Tooltip sorting method: default ▼

Smoothing

○
0.6

Horizontal Axis

STEP
RELATIVE
WALL

Runs

Write a regex to filter runs

Write a regex to filter runs

☒

 .

☒ Loss_0.01_Test

☒ Loss_0.01_Train

☒ Accuracy_0.01_Test

☒ Accuracy_0.01_Train

☒ Train_Accuracy_0.01

☒ Test_Accuracy_0.01

☒ Train_loss_0.01

☒ Test_loss_0.01

☒ Loss_0.001_Test

☒ Loss_0.001_Train

☒ Accuracy_0.001_Test

☒ Accuracy_0.001_Train

☒ Train_Accuracy_0.001

☒ Test_Accuracy_0.001

☒ Train_loss_0.001

TOGGLE ALL RUNS

runs/mnist_LeNet

🔍
Filter tags (regular expressions supported)

Accuracy_0.0001
⌵

Accuracy_0.0001

tag: Accuracy_0.0001

📐
☰
🔄

Accuracy_0.001
⌵

Accuracy_0.001

tag: Accuracy_0.001

📐
☰
🔄

Accuracy_0.01
⌵

Loss_0.0001
⌵

Loss_0.001
⌵

```
logs_RMS = 'runs/mnist_LeNet_RMS'
tb_rms = SummaryWriter(logs_RMS)
```

https://colab.research.google.com/drive/1oLb7F3x1oAZiKTC-iX3mUVw-rACSWmOw#scrollTo=zq_a4cZCWUp&printMode=true

```
#Checking if we can use GPU

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

model = model.to(device)
criterion = criterion.to(device)

avail_optimizers = {'Adam':torch.optim.Adam(model.parameters(), lr=learning_rate, betas=(0.9, 0.999), eps=1e-08, weight_decay=0, amsgrad=False),
                    'RMS': torch.optim.RMSprop(model.parameters(), lr=learning_rate, alpha=0.99, eps=1e-08, weight_decay=0, momentum=0, centered=False),
                    'SGD': torch.optim.SGD(model.parameters(), lr=learning_rate, momentum=0, dampening=0, weight_decay=0, nesterov=False)}

optimizer = avail_optimizers[selected_optimizer]

epochs = 20
print(f"selected optimizer and learning rate: {optimizer}")
for n_iter in tqdm(range(epochs)):
    print(f"Epoch {n_iter+1}/{epochs}")

    start_time = time.monotonic()

    train_loss, train_acc = train(model, train_iterator, optimizer, criterion, device)
    test_loss, test_acc = test(model, test_iterator, criterion, device)

    end_time = time.monotonic()

    epoch_mins, epoch_secs = epoch_time(start_time, end_time)

    print(f'Epoch: {n_iter+1:02} | Epoch Time: {epoch_mins}m {epoch_secs}s')
    print(f'\tTrain Loss: {train_loss:.3f} | Train Acc: {train_acc*100:.2f}%')
    print(f'\t Test. Loss: {test_loss:.3f} | Test. Acc: {test_acc*100:.2f}%')

    if tb_rms is not None:
        tb_rms.add_scalars(f'Loss_{learning_rate}', {"Train":train_loss,
                                                    "Test":test_loss}, n_iter)
        tb_rms.add_scalars(f'Accuracy_{learning_rate}', {"Train":train_acc,
                                                         "Test":test_acc}, n_iter)

        tb_rms.add_scalars(f'Train_Accuracy', {f"{learning_rate}":train_acc}, n_iter)
        tb_rms.add_scalars(f'Test_Accuracy', {f"{learning_rate}":test_acc}, n_iter)

        tb_rms.add_scalars(f'Train_loss', {f"{learning_rate}":train_loss}, n_iter)
        tb_rms.add_scalars(f'Test_loss', {f"{learning_rate}":test_loss}, n_iter)
```

```
selected optimizer and learning rate: RMSprop (
Parameter Group 0
  alpha: 0.99
  centered: False
  eps: 1e-08
  lr: 0.01
  momentum: 0
  weight_decay: 0
)
100% 20/20 [06:50<00:00, 20.79s/it]

Epoch 1/20
Training: 99% 465/469 [00:19<00:00, 25.61it/s]

Testing: 96% 76/79 [00:01<00:00, 50.18it/s]
Epoch: 01 | Epoch Time: 0m 21s
      Train Loss: 3.080 | Train Acc: 11.07%
      Test. Loss: 2.302 | Test. Acc: 11.36%
Epoch 2/20
Training: 99% 466/469 [00:17<00:00, 25.70it/s]

Testing: 97% 77/79 [00:01<00:00, 47.52it/s]
Epoch: 02 | Epoch Time: 0m 19s
      Train Loss: 2.302 | Train Acc: 11.12%
      Test. Loss: 2.302 | Test. Acc: 11.36%
Epoch 3/20
Training: 99% 466/469 [00:18<00:00, 26.27it/s]

Testing: 99% 78/79 [00:01<00:00, 47.52it/s]
Epoch: 03 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 11.05%
      Test. Loss: 2.301 | Test. Acc: 11.36%
Epoch 4/20
Training: 100% 468/469 [00:18<00:00, 27.47it/s]

Testing: 96% 76/79 [00:01<00:00, 46.81it/s]
Epoch: 04 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 11.16%
      Test. Loss: 2.301 | Test. Acc: 11.36%
Epoch 5/20
Training: 99% 466/469 [00:18<00:00, 27.45it/s]

Testing: 96% 76/79 [00:01<00:00, 47.51it/s]
Epoch: 05 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 11.16%
      Test. Loss: 2.302 | Test. Acc: 10.24%
Epoch 6/20
Training: 99% 465/469 [00:18<00:00, 24.94it/s]

Testing: 94% 74/79 [00:01<00:00, 46.97it/s]
Epoch: 06 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 11.11%
      Test. Loss: 2.302 | Test. Acc: 10.24%
Epoch 7/20
Training: 100% 468/469 [00:18<00:00, 25.01it/s]

Testing: 100% 79/79 [00:01<00:00, 53.29it/s]
Epoch: 07 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 10.95%
      Test. Loss: 2.301 | Test. Acc: 11.36%
Epoch 8/20
Training: 100% 468/469 [00:18<00:00, 26.10it/s]

Testing: 96% 76/79 [00:01<00:00, 47.74it/s]
Epoch: 08 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 11.12%
      Test. Loss: 2.302 | Test. Acc: 11.36%
Epoch 9/20
Training: 100% 467/469 [00:18<00:00, 24.99it/s]

Testing: 94% 74/79 [00:01<00:00, 46.79it/s]
Epoch: 09 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 11.15%
      Test. Loss: 2.302 | Test. Acc: 11.36%
Epoch 10/20
Training: 100% 468/469 [00:18<00:00, 25.60it/s]

Testing: 94% 74/79 [00:01<00:00, 49.32it/s]
Epoch: 10 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 11.17%
      Test. Loss: 2.302 | Test. Acc: 11.36%
Epoch 11/20
Training: 100% 469/469 [00:19<00:00, 27.21it/s]

Testing: 97% 77/79 [00:01<00:00, 49.34it/s]
Epoch: 11 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 11.08%
      Test. Loss: 2.301 | Test. Acc: 11.36%
Epoch 12/20
Training: 99% 465/469 [00:18<00:00, 25.03it/s]

Testing: 97% 77/79 [00:01<00:00, 44.64it/s]
Epoch: 12 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 11.13%
      Test. Loss: 2.302 | Test. Acc: 10.24%
Epoch 13/20
Training: 99% 465/469 [00:19<00:00, 24.69it/s]

Testing: 95% 75/79 [00:01<00:00, 47.20it/s]
Epoch: 13 | Epoch Time: 0m 21s
      Train Loss: 2.302 | Train Acc: 11.13%
      Test. Loss: 2.302 | Test. Acc: 10.24%
Epoch 14/20
Training: 100% 469/469 [00:18<00:00, 28.17it/s]

Testing: 94% 74/79 [00:01<00:00, 47.48it/s]
Epoch: 14 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 11.10%
      Test. Loss: 2.302 | Test. Acc: 11.36%
Epoch 15/20
Training: 100% 469/469 [00:18<00:00, 27.51it/s]

Testing: 94% 74/79 [00:01<00:00, 45.02it/s]
Epoch: 15 | Epoch Time: 0m 20s
      Train Loss: 2.302 | Train Acc: 11.12%
```

```
Train Loss: 2.302 | Train Acc: 11.10%
Test. Loss: 2.302 | Test. Acc: 11.36%
Epoch 16/20
Training: 99%                                466/469 [00:18<00:00, 27.06it/s]

Testing: 96%                                76/79 [00:01<00:00, 48.20it/s]
Epoch: 16 | Epoch Time: 0m 20s
Train Loss: 2.302 | Train Acc: 11.20%
Test. Loss: 2.302 | Test. Acc: 10.13%
Epoch 17/20
Training: 99%                                465/469 [00:18<00:00, 25.18it/s]

Testing: 96%                                76/79 [00:01<00:00, 46.87it/s]
Epoch: 17 | Epoch Time: 0m 20s
Train Loss: 2.302 | Train Acc: 11.12%
Test. Loss: 2.302 | Test. Acc: 10.24%
Epoch 18/20
Training: 100%                               469/469 [00:18<00:00, 28.70it/s]

Testing: 96%                                76/79 [00:01<00:00, 48.26it/s]
Epoch: 18 | Epoch Time: 0m 20s
Train Loss: 2.302 | Train Acc: 11.14%
Test. Loss: 2.303 | Test. Acc: 10.24%
Epoch 19/20
Training: 100%                               468/469 [00:18<00:00, 26.44it/s]

Testing: 95%                                75/79 [00:01<00:00, 47.05it/s]
Epoch: 19 | Epoch Time: 0m 20s
Train Loss: 2.302 | Train Acc: 11.12%
Test. Loss: 2.302 | Test. Acc: 11.36%
Epoch 20/20
Training: 100%                               469/469 [00:19<00:00, 27.84it/s]

Testing: 97%                                77/79 [00:01<00:00, 47.77it/s]
Epoch: 20 | Epoch Time: 0m 21s
Train Loss: 2.302 | Train Acc: 11.13%
Test. Loss: 2.302 | Test. Acc: 11.36%
selected optimizer and learning rate: RMSprop (
Parameter Group 0
  alpha: 0.99
  centered: False
  eps: 1e-08
  lr: 0.001
  momentum: 0
  weight_decay: 0
)
100%                                         20/20 [06:46<00:00, 20.38s/it]

Epoch 1/20
Training: 100%                               469/469 [00:18<00:00, 26.94it/s]

Testing: 97%                                77/79 [00:01<00:00, 48.90it/s]
Epoch: 01 | Epoch Time: 0m 20s
Train Loss: 0.352 | Train Acc: 88.68%
Test. Loss: 0.083 | Test. Acc: 97.40%
Epoch 2/20
Training: 100%                               468/469 [00:18<00:00, 27.50it/s]

Testing: 97%                                77/79 [00:01<00:00, 48.92it/s]
Epoch: 02 | Epoch Time: 0m 20s
Train Loss: 0.127 | Train Acc: 96.01%
Test. Loss: 0.053 | Test. Acc: 98.24%
Epoch 3/20
Training: 100%                               468/469 [00:18<00:00, 25.81it/s]

Testing: 96%                                76/79 [00:02<00:00, 32.32it/s]
Epoch: 03 | Epoch Time: 0m 20s
Train Loss: 0.096 | Train Acc: 96.89%
Test. Loss: 0.038 | Test. Acc: 98.66%
Epoch 4/20
Training: 100%                               468/469 [00:19<00:00, 26.80it/s]

Testing: 97%                                77/79 [00:01<00:00, 47.45it/s]
Epoch: 04 | Epoch Time: 0m 20s
Train Loss: 0.078 | Train Acc: 97.67%
Test. Loss: 0.041 | Test. Acc: 98.64%
Epoch 5/20
Training: 100%                               469/469 [00:18<00:00, 28.31it/s]

Testing: 95%                                75/79 [00:01<00:00, 49.00it/s]
Epoch: 05 | Epoch Time: 0m 20s
Train Loss: 0.070 | Train Acc: 97.81%
Test. Loss: 0.036 | Test. Acc: 98.82%
Epoch 6/20
Training: 100%                               467/469 [00:18<00:00, 25.59it/s]

Testing: 96%                                76/79 [00:01<00:00, 48.41it/s]
Epoch: 06 | Epoch Time: 0m 20s
Train Loss: 0.060 | Train Acc: 98.14%
Test. Loss: 0.039 | Test. Acc: 98.67%
Epoch 7/20
Training: 99%                                466/469 [00:17<00:00, 26.16it/s]

Testing: 97%                                77/79 [00:01<00:00, 45.61it/s]
Epoch: 07 | Epoch Time: 0m 20s
Train Loss: 0.055 | Train Acc: 98.27%
Test. Loss: 0.036 | Test. Acc: 98.83%
Epoch 8/20
Training: 100%                               467/469 [00:18<00:00, 24.63it/s]

Testing: 97%                                77/79 [00:01<00:00, 46.22it/s]
Epoch: 08 | Epoch Time: 0m 20s
Train Loss: 0.052 | Train Acc: 98.39%
Test. Loss: 0.037 | Test. Acc: 98.69%
Epoch 9/20
Training: 99%                                466/469 [00:18<00:00, 25.61it/s]

Testing: 96%                                76/79 [00:01<00:00, 47.30it/s]
Epoch: 09 | Epoch Time: 0m 20s
Train Loss: 0.048 | Train Acc: 98.46%
Test. Loss: 0.028 | Test. Acc: 99.11%
Epoch 10/20
Training: 100%                               468/469 [00:18<00:00, 25.96it/s]

Testing: 94%                                74/79 [00:01<00:00, 45.38it/s]
```

```
Epoch: 10 | Epoch Time: 0m 20s
    Train Loss: 0.046 | Train Acc: 98.58%
    Test. Loss: 0.029 | Test. Acc: 98.95%
Epoch 11/20
Training: 100%                                468/469 [00:18<00:00, 27.21it/s]

Testing: 100%                                79/79 [00:01<00:00, 52.04it/s]
Epoch: 11 | Epoch Time: 0m 20s
    Train Loss: 0.044 | Train Acc: 98.60%
    Test. Loss: 0.026 | Test. Acc: 99.16%
Epoch 12/20
Training: 100%                                467/469 [00:18<00:00, 26.50it/s]

Testing: 95%                                75/79 [00:01<00:00, 50.09it/s]
Epoch: 12 | Epoch Time: 0m 20s
    Train Loss: 0.042 | Train Acc: 98.65%
    Test. Loss: 0.030 | Test. Acc: 99.10%
Epoch 13/20
Training: 100%                                468/469 [00:18<00:00, 28.07it/s]

Testing: 99%                                78/79 [00:01<00:00, 50.54it/s]
Epoch: 13 | Epoch Time: 0m 20s
    Train Loss: 0.041 | Train Acc: 98.71%
    Test. Loss: 0.033 | Test. Acc: 98.97%
Epoch 14/20
Training: 100%                                467/469 [00:18<00:00, 26.06it/s]

Testing: 97%                                77/79 [00:01<00:00, 49.03it/s]
Epoch: 14 | Epoch Time: 0m 20s
    Train Loss: 0.038 | Train Acc: 98.81%
    Test. Loss: 0.030 | Test. Acc: 99.02%
Epoch 15/20
Training: 100%                                468/469 [00:18<00:00, 25.63it/s]

Testing: 97%                                77/79 [00:01<00:00, 48.13it/s]
Epoch: 15 | Epoch Time: 0m 20s
    Train Loss: 0.037 | Train Acc: 98.82%
    Test. Loss: 0.026 | Test. Acc: 99.20%
Epoch 16/20
Training: 99%                                466/469 [00:18<00:00, 26.44it/s]

Testing: 95%                                75/79 [00:01<00:00, 46.75it/s]
Epoch: 16 | Epoch Time: 0m 20s
    Train Loss: 0.036 | Train Acc: 98.82%
    Test. Loss: 0.040 | Test. Acc: 98.78%
Epoch 17/20
Training: 99%                                466/469 [00:19<00:00, 25.26it/s]

Testing: 92%                                73/79 [00:01<00:00, 46.49it/s]
Epoch: 17 | Epoch Time: 0m 21s
    Train Loss: 0.034 | Train Acc: 98.94%
    Test. Loss: 0.025 | Test. Acc: 99.20%
Epoch 18/20
Training: 100%                                469/469 [00:18<00:00, 27.68it/s]

Testing: 95%                                75/79 [00:01<00:00, 48.40it/s]
Epoch: 18 | Epoch Time: 0m 20s
    Train Loss: 0.036 | Train Acc: 98.84%
    Test. Loss: 0.020 | Test. Acc: 99.25%
Epoch 19/20
Training: 100%                                467/469 [00:18<00:00, 27.63it/s]

Testing: 92%                                73/79 [00:01<00:00, 47.64it/s]
Epoch: 19 | Epoch Time: 0m 20s
    Train Loss: 0.033 | Train Acc: 98.89%
    Test. Loss: 0.029 | Test. Acc: 99.08%
Epoch 20/20
Training: 100%                                467/469 [00:18<00:00, 26.82it/s]

Testing: 95%                                75/79 [00:01<00:00, 46.23it/s]
Epoch: 20 | Epoch Time: 0m 20s
    Train Loss: 0.032 | Train Acc: 99.01%
    Test. Loss: 0.023 | Test. Acc: 99.19%
selected optimizer and learning rate: RMSprop (
Parameter Group 0
    alpha: 0.99
    centered: False
    eps: 1e-08
    lr: 0.0001
    momentum: 0
    weight_decay: 0
)
100%                                20/20 [06:53<00:00, 20.42s/it]

Epoch 1/20
Training: 100%                                469/469 [00:18<00:00, 26.73it/s]

Testing: 99%                                78/79 [00:01<00:00, 47.12it/s]
Epoch: 01 | Epoch Time: 0m 20s
    Train Loss: 1.106 | Train Acc: 66.78%
    Test. Loss: 0.397 | Test. Acc: 89.36%
Epoch 2/20
Training: 100%                                467/469 [00:18<00:00, 27.84it/s]

Testing: 99%                                78/79 [00:01<00:00, 49.52it/s]
Epoch: 02 | Epoch Time: 0m 20s
    Train Loss: 0.556 | Train Acc: 83.21%
    Test. Loss: 0.234 | Test. Acc: 93.77%
Epoch 3/20
Training: 100%                                469/469 [00:18<00:00, 27.82it/s]

Testing: 99%                                78/79 [00:02<00:00, 45.46it/s]
Epoch: 03 | Epoch Time: 0m 20s
    Train Loss: 0.373 | Train Acc: 88.93%
    Test. Loss: 0.178 | Test. Acc: 94.98%
Epoch 4/20
Training: 100%                                468/469 [00:19<00:00, 25.55it/s]

Testing: 97%                                77/79 [00:02<00:00, 45.18it/s]
Epoch: 04 | Epoch Time: 0m 21s
    Train Loss: 0.283 | Train Acc: 91.83%
    Test. Loss: 0.155 | Test. Acc: 95.19%
Epoch 5/20
Training: 100%                                469/469 [00:19<00:00, 25.96it/s]

Testing: 96%                                76/79 [00:01<00:00, 43.01it/s]
```

```
Epoch: 05 | Epoch Time: 0m 21s
      Train Loss: 0.230 | Train Acc: 93.27%
      Test. Loss: 0.133 | Test. Acc: 95.95%
Epoch 6/20
Training: 100%                                468/469 [00:19<00:00, 25.28it/s]

Testing: 95%                                75/79 [00:01<00:00, 47.15it/s]
Epoch: 06 | Epoch Time: 0m 21s
      Train Loss: 0.200 | Train Acc: 94.10%
      Test. Loss: 0.111 | Test. Acc: 96.48%
Epoch 7/20
Training: 99%                                466/469 [00:19<00:00, 25.54it/s]

Testing: 95%                                75/79 [00:01<00:00, 46.90it/s]
Epoch: 07 | Epoch Time: 0m 21s
      Train Loss: 0.180 | Train Acc: 94.71%
      Test. Loss: 0.093 | Test. Acc: 97.17%
Epoch 8/20
Training: 100%                                469/469 [00:18<00:00, 26.31it/s]

Testing: 97%                                77/79 [00:01<00:00, 47.56it/s]
Epoch: 08 | Epoch Time: 0m 20s
      Train Loss: 0.163 | Train Acc: 95.08%
      Test. Loss: 0.087 | Test. Acc: 97.27%
Epoch 9/20
Training: 100%                                468/469 [00:18<00:00, 26.82it/s]

Testing: 95%                                75/79 [00:01<00:00, 46.23it/s]
Epoch: 09 | Epoch Time: 0m 20s
      Train Loss: 0.155 | Train Acc: 95.45%
      Test. Loss: 0.092 | Test. Acc: 97.02%
Epoch 10/20
Training: 99%                                465/469 [00:19<00:00, 25.35it/s]

Testing: 94%                                74/79 [00:01<00:00, 45.70it/s]
Epoch: 10 | Epoch Time: 0m 21s
      Train Loss: 0.142 | Train Acc: 95.75%
      Test. Loss: 0.085 | Test. Acc: 97.24%
Epoch 11/20
Training: 99%                                466/469 [00:18<00:00, 26.67it/s]

Testing: 97%                                77/79 [00:01<00:00, 47.29it/s]
Epoch: 11 | Epoch Time: 0m 20s
```

```
%tensorboard --logdir runs/mnist_LeNet_RMS
```



TensorBoard

SCALARS

TIME SERIES

INACTIVE

- ☐ Show data download links
- ☐ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing

Horizontal Axis

STEP RELATIVE WALL

Runs

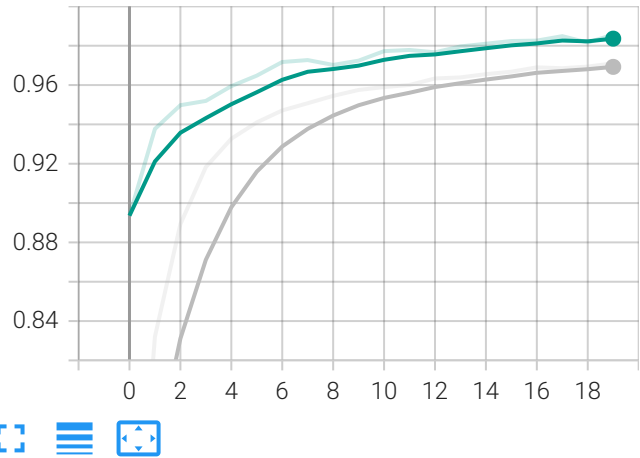
Write a regex to filter runs

- ☐ .
- ☐ Loss_0.01_Test
- ☐ Loss_0.01_Train
- ☐ Accuracy_0.01_Test
- ☐ Accuracy_0.01_Train
- ☐ Train_Accuracy_0.01
- ☐ Test_Accuracy_0.01
- ☐ Train_loss_0.01
- ☐ Test_loss_0.01

TOGGLE ALL RUNS

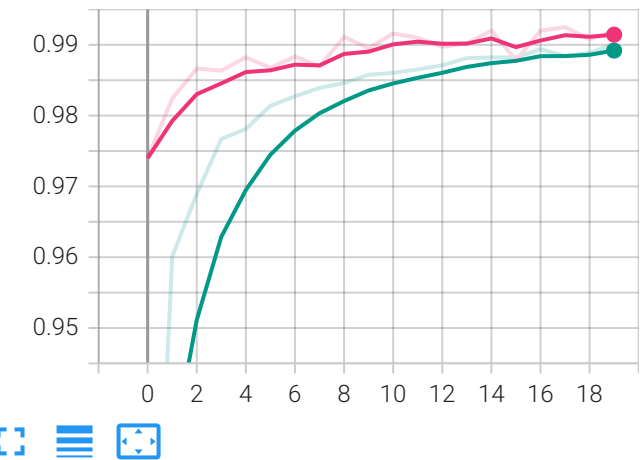
runs/mnist_LeNet_RMS

Accuracy_0.0001
tag: Accuracy_0.0001



Accuracy_0.001

Accuracy_0.001
tag: Accuracy_0.001



Accuracy_0.01

Training: 100% 468/469 [00:18<00:00, 26.86it/s]

▼ CIFAR10

```
train_dataset_cifar10 = datasets.CIFAR10(root='data',
```

As LeNet is accepting one in_channel, for CIFAR10 dataset we have to make some changes so it accepts a 3 channels image.

- we first normalize and transform the data to Tensor and then use the architecture from previous part and make few changes.

```
transform_cifar10=transforms.Compose([transforms.Resize((32,32)),transforms.ToTensor(),transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])
```

```
Test. Loss: 0.056 | Test. Acc: 98.16%
```

```
BATCH_SIZE=128
```

```
train_dataset_cifar10 = datasets.CIFAR10(root='data',
                                          train=True,
                                          transform=transform_cifar10,
                                          download=True)
```

```
test_dataset_cifar10 = datasets.CIFAR10(root='data',
                                          train=False,
                                          transform=transform_cifar10)
```

```
train_loader = data.DataLoader(dataset=train_dataset_cifar10,
                                batch_size=BATCH_SIZE,
                                num_workers=2,
                                shuffle=True)

test_loader = data.DataLoader(dataset=test_dataset_cifar10,
                               batch_size=BATCH_SIZE,
                               num_workers=2,
                               shuffle=False)
```

Downloading <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz> to data/cifar-10-python.tar.gz
170499072/? [00:02<00:00, 88587994.69it/s]

Extracting data/cifar-10-python.tar.gz to data

```
#for CIFAR10
class LeNet_CIFAR10(nn.Module):
    def __init__(self, output_dim):
        super().__init__()

        self.conv1 = nn.Conv2d(in_channels=3, #only the input channel is changed to 3 because it is colorful
                                out_channels=6,
                                kernel_size=5)

        self.conv2 = nn.Conv2d(in_channels=6,
                                out_channels=16,
                                kernel_size=5)

        self.fc_1 = nn.Linear(16 * 5 * 5, 120) #also here the shape is changed
        self.fc_2 = nn.Linear(120, 84)
        self.fc_3 = nn.Linear(84, output_dim)

    def forward(self, x):

        # x = [batch size, 1, 28, 28]

        x = self.conv1(x)

        # x = [batch size, 6, 24, 24]

        x = F.max_pool2d(x, kernel_size=2, stride=2)

        # x = [batch size, 6, 12, 12]

        x = F.relu(x)

        x = self.conv2(x)

        # x = [batch size, 16, 8, 8]

        x = F.max_pool2d(x, kernel_size=2, stride=2)

        # x = [batch size, 16, 4, 4]

        x = F.relu(x)

        x = x.view(x.shape[0], -1)

        # x = [batch size, 16*4*4 = 256]

        h = x

        x = self.fc_1(x)

        # x = [batch size, 120]

        x = F.relu(x)

        x = self.fc_2(x)

        # x = batch size, 84]

        x = F.relu(x)

        x = self.fc_3(x)

        # x = [batch size, output dim]

        return x, h
```

```
logs = 'runs/CIFAR10_LeNet'
tb_cf = SummaryWriter(logs)
```

CIFAR10 dataset with Adam optimizer and few different learning rates

```
selected_optimizer='Adam'
for learning_rate in [ 0.01, 0.001, 0.0001]:

    OUTPUT_DIM = 10

    model_c = LeNet_CIFAR10(OUTPUT_DIM)

    # def count_parameters(model):
    #     return sum(p.numel() for p in model.parameters() if p.requires_grad)

    # print(f'The model has {count_parameters(model):,} trainable parameters')

    optimizer = optim.Adam(model.parameters())

    criterion = nn.CrossEntropyLoss()

    #Checking if we can use GPU

    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

    model_c = model_c.to(device)
    criterion = criterion.to(device)
```

```
avail_optimizers = {'Adam':torch.optim.Adam(model_c.parameters(), lr=learning_rate, betas=(0.9, 0.999), eps=1e-08, weight_decay=0, amsgrad=False),
                    'RMS': torch.optim.RMSprop(model_c.parameters(), lr=learning_rate, alpha=0.99, eps=1e-08, weight_decay=0, momentum=0, centered=False),
                    'SGD': torch.optim.SGD(model_c.parameters(), lr=learning_rate, momentum=0, dampening=0, weight_decay=0, nesterov=False)}

optimizer = avail_optimizers[selected_optimizer]

epochs = 20
print(f"selected optimizer and learning rate: {optimizer}")
for n_iter in tqdm(range(epochs)):
    print(f"Epoch {n_iter+1}/{epochs}")

    start_time = time.monotonic()

    train_loss, train_acc = train(model_c, train_loader, optimizer, criterion, device)
    test_loss, test_acc = test(model_c, test_loader, criterion, device)

    end_time = time.monotonic()

    epoch_mins, epoch_secs = epoch_time(start_time, end_time)

    print(f'Epoch: {n_iter+1:02} | Epoch Time: {epoch_mins}m {epoch_secs}s')
    print(f'\tTrain Loss: {train_loss:.3f} | Train Acc: {train_acc*100:.2f}%')
    print(f'\t Test. Loss: {test_loss:.3f} | Test. Acc: {test_acc*100:.2f}%')

    if tb_cf is not None:
        tb_cf.add_scalars(f'Loss_{learning_rate}', {"Train":train_loss,
                                                    "Test":test_loss}, n_iter)
        tb_cf.add_scalars(f'Accuracy_{learning_rate}', {"Train":train_acc,
                                                         "Test":test_acc}, n_iter)

        tb_cf.add_scalars(f'Train_Accuracy', {f"{learning_rate}":train_acc}, n_iter)
        tb_cf.add_scalars(f'Test_Accuracy', {f"{learning_rate}":test_acc}, n_iter)

        tb_cf.add_scalars(f'Train_loss', {f"{learning_rate}":train_loss}, n_iter)
        tb_cf.add_scalars(f'Test_loss', {f"{learning_rate}":test_loss}, n_iter)
```



```
selected optimizer and learning rate: Adam (
Parameter Group 0
  amsgrad: False
  betas: (0.9, 0.999)
  eps: 1e-08
  lr: 0.01
  maximize: False
  weight_decay: 0
)
100% 20/20 [05:13<00:00, 15.39s/it]

Epoch 1/20
Training: 100% 390/391 [00:12<00:00, 30.49it/s]

Testing: 100% 79/79 [00:03<00:00, 33.27it/s]
Epoch: 01 | Epoch Time: 0m 16s
  Train Loss: 1.707 | Train Acc: 36.49%
  Test. Loss: 1.582 | Test. Acc: 41.95%
Epoch 2/20
Training: 99% 387/391 [00:12<00:00, 29.31it/s]

Testing: 97% 77/79 [00:02<00:00, 32.94it/s]
Epoch: 02 | Epoch Time: 0m 15s
  Train Loss: 1.504 | Train Acc: 45.16%
  Test. Loss: 1.464 | Test. Acc: 46.16%
Epoch 3/20
Training: 100% 390/391 [00:12<00:00, 31.94it/s]

Testing: 97% 77/79 [00:02<00:00, 31.60it/s]
Epoch: 03 | Epoch Time: 0m 15s
  Train Loss: 1.440 | Train Acc: 47.88%
  Test. Loss: 1.421 | Test. Acc: 49.16%
Epoch 4/20
Training: 99% 388/391 [00:12<00:00, 30.57it/s]

Testing: 97% 77/79 [00:02<00:00, 32.64it/s]
Epoch: 04 | Epoch Time: 0m 15s
  Train Loss: 1.395 | Train Acc: 49.84%
  Test. Loss: 1.417 | Test. Acc: 49.31%
Epoch 5/20
Training: 100% 391/391 [00:13<00:00, 32.26it/s]

Testing: 99% 78/79 [00:02<00:00, 36.14it/s]
Epoch: 05 | Epoch Time: 0m 15s
  Train Loss: 1.367 | Train Acc: 51.25%
  Test. Loss: 1.363 | Test. Acc: 51.10%
Epoch 6/20
Training: 99% 389/391 [00:12<00:00, 31.73it/s]

Testing: 99% 78/79 [00:02<00:00, 34.35it/s]
Epoch: 06 | Epoch Time: 0m 15s
  Train Loss: 1.354 | Train Acc: 51.83%
  Test. Loss: 1.409 | Test. Acc: 50.14%
Epoch 7/20
Training: 100% 391/391 [00:13<00:00, 32.24it/s]

Testing: 96% 76/79 [00:02<00:00, 34.45it/s]
Epoch: 07 | Epoch Time: 0m 16s
  Train Loss: 1.346 | Train Acc: 52.00%
  Test. Loss: 1.395 | Test. Acc: 49.78%
Epoch 8/20
Training: 99% 388/391 [00:13<00:00, 29.21it/s]

Testing: 97% 77/79 [00:02<00:00, 32.90it/s]
Epoch: 08 | Epoch Time: 0m 15s
  Train Loss: 1.324 | Train Acc: 53.02%
  Test. Loss: 1.422 | Test. Acc: 50.13%
Epoch 9/20
Training: 99% 388/391 [00:13<00:00, 32.57it/s]

Testing: 96% 76/79 [00:03<00:00, 20.92it/s]
Epoch: 09 | Epoch Time: 0m 16s
  Train Loss: 1.312 | Train Acc: 53.47%
  Test. Loss: 1.371 | Test. Acc: 51.30%
Epoch 10/20
Training: 99% 389/391 [00:13<00:00, 30.81it/s]

Testing: 99% 78/79 [00:02<00:00, 34.40it/s]
Epoch: 10 | Epoch Time: 0m 16s
  Train Loss: 1.297 | Train Acc: 53.86%
  Test. Loss: 1.388 | Test. Acc: 50.90%
Epoch 11/20
Training: 100% 390/391 [00:12<00:00, 31.97it/s]

Testing: 96% 76/79 [00:02<00:00, 35.31it/s]
Epoch: 11 | Epoch Time: 0m 15s
  Train Loss: 1.293 | Train Acc: 54.31%
  Test. Loss: 1.380 | Test. Acc: 51.52%
Epoch 12/20
Training: 99% 388/391 [00:12<00:00, 31.17it/s]

Testing: 97% 77/79 [00:02<00:00, 36.44it/s]
Epoch: 12 | Epoch Time: 0m 15s
  Train Loss: 1.287 | Train Acc: 54.64%
  Test. Loss: 1.383 | Test. Acc: 51.03%
Epoch 13/20
Training: 99% 387/391 [00:12<00:00, 30.55it/s]

Testing: 99% 78/79 [00:02<00:00, 34.75it/s]
Epoch: 13 | Epoch Time: 0m 15s
  Train Loss: 1.272 | Train Acc: 54.93%
  Test. Loss: 1.328 | Test. Acc: 53.39%
Epoch 14/20
Training: 99% 389/391 [00:12<00:00, 32.52it/s]

Testing: 96% 76/79 [00:02<00:00, 34.27it/s]
Epoch: 14 | Epoch Time: 0m 15s
  Train Loss: 1.266 | Train Acc: 55.27%
  Test. Loss: 1.370 | Test. Acc: 51.38%
Epoch 15/20
Training: 99% 387/391 [00:12<00:00, 28.44it/s]

Testing: 94% 74/79 [00:02<00:00, 32.20it/s]
Epoch: 15 | Epoch Time: 0m 15s
  Train Loss: 1.258 | Train Acc: 55.40%
```

```
      train Loss: 1.258 | train Acc: 55.49%
      Test. Loss: 1.311 | Test. Acc: 54.49%
Epoch 16/20
Training: 99%                                388/391 [00:12<00:00, 30.72it/s]

Testing: 97%                                77/79 [00:02<00:00, 34.32it/s]
Epoch: 16 | Epoch Time: 0m 15s
      Train Loss: 1.245 | Train Acc: 56.02%
      Test. Loss: 1.333 | Test. Acc: 53.55%
Epoch 17/20
Training: 99%                                389/391 [00:12<00:00, 29.97it/s]

Testing: 96%                                76/79 [00:02<00:00, 33.75it/s]
Epoch: 17 | Epoch Time: 0m 15s
      Train Loss: 1.250 | Train Acc: 55.73%
      Test. Loss: 1.353 | Test. Acc: 53.13%
Epoch 18/20
Training: 99%                                389/391 [00:12<00:00, 31.73it/s]

Testing: 97%                                77/79 [00:02<00:00, 36.92it/s]
Epoch: 18 | Epoch Time: 0m 15s
      Train Loss: 1.235 | Train Acc: 56.65%
      Test. Loss: 1.310 | Test. Acc: 53.97%
Epoch 19/20
Training: 100%                               390/391 [00:12<00:00, 32.06it/s]

Testing: 97%                                77/79 [00:02<00:00, 33.67it/s]
Epoch: 19 | Epoch Time: 0m 15s
      Train Loss: 1.226 | Train Acc: 56.82%
      Test. Loss: 1.311 | Test. Acc: 54.40%
Epoch 20/20
Training: 99%                                388/391 [00:12<00:00, 28.87it/s]

Testing: 96%                                76/79 [00:02<00:00, 33.88it/s]
Epoch: 20 | Epoch Time: 0m 15s
      Train Loss: 1.220 | Train Acc: 57.16%
      Test. Loss: 1.360 | Test. Acc: 53.92%
selected optimizer and learning rate: Adam (
Parameter Group 0
  amsgrad: False
  betas: (0.9, 0.999)
  eps: 1e-08
  lr: 0.001
  maximize: False
  weight_decay: 0
)
100%                                         20/20 [05:13<00:00, 16.06s/it]

Epoch 1/20
Training: 99%                                388/391 [00:12<00:00, 32.31it/s]

Testing: 97%                                77/79 [00:02<00:00, 35.56it/s]
Epoch: 01 | Epoch Time: 0m 15s
      Train Loss: 1.738 | Train Acc: 36.38%
      Test. Loss: 1.540 | Test. Acc: 43.34%
Epoch 2/20
Training: 99%                                389/391 [00:12<00:00, 31.28it/s]

Testing: 97%                                77/79 [00:02<00:00, 33.57it/s]
Epoch: 02 | Epoch Time: 0m 15s
      Train Loss: 1.457 | Train Acc: 47.55%
      Test. Loss: 1.380 | Test. Acc: 50.81%
Epoch 3/20
Training: 99%                                389/391 [00:12<00:00, 29.90it/s]

Testing: 96%                                76/79 [00:02<00:00, 32.74it/s]
Epoch: 03 | Epoch Time: 0m 15s
      Train Loss: 1.333 | Train Acc: 52.33%
      Test. Loss: 1.343 | Test. Acc: 51.52%
Epoch 4/20
Training: 99%                                388/391 [00:12<00:00, 32.26it/s]

Testing: 97%                                77/79 [00:02<00:00, 34.92it/s]
Epoch: 04 | Epoch Time: 0m 15s
      Train Loss: 1.242 | Train Acc: 55.83%
      Test. Loss: 1.236 | Test. Acc: 56.26%
Epoch 5/20
Training: 99%                                387/391 [00:12<00:00, 29.85it/s]

Testing: 96%                                76/79 [00:02<00:00, 32.66it/s]
Epoch: 05 | Epoch Time: 0m 15s
      Train Loss: 1.170 | Train Acc: 58.72%
      Test. Loss: 1.169 | Test. Acc: 58.79%
Epoch 6/20
Training: 100%                               391/391 [00:13<00:00, 33.16it/s]

Testing: 96%                                76/79 [00:02<00:00, 33.77it/s]
Epoch: 06 | Epoch Time: 0m 15s
      Train Loss: 1.112 | Train Acc: 60.72%
      Test. Loss: 1.140 | Test. Acc: 60.03%
Epoch 7/20
Training: 99%                                389/391 [00:13<00:00, 30.71it/s]

Testing: 97%                                77/79 [00:02<00:00, 33.35it/s]
Epoch: 07 | Epoch Time: 0m 15s
      Train Loss: 1.069 | Train Acc: 62.29%
      Test. Loss: 1.107 | Test. Acc: 61.22%
Epoch 8/20
Training: 100%                               390/391 [00:13<00:00, 31.51it/s]

Testing: 96%                                76/79 [00:02<00:00, 35.43it/s]
Epoch: 08 | Epoch Time: 0m 15s
      Train Loss: 1.026 | Train Acc: 63.75%
      Test. Loss: 1.099 | Test. Acc: 61.57%
Epoch 9/20
Training: 99%                                387/391 [00:12<00:00, 30.91it/s]

Testing: 96%                                76/79 [00:02<00:00, 32.56it/s]
Epoch: 09 | Epoch Time: 0m 15s
      Train Loss: 1.000 | Train Acc: 64.59%
      Test. Loss: 1.079 | Test. Acc: 62.61%
Epoch 10/20
Training: 100%                               390/391 [00:12<00:00, 32.56it/s]

Testing: 97%                                77/79 [00:02<00:00, 33.98it/s]
```

Epoch: 10 | Epoch Time: 0m 15s
Train Loss: 0.964 | Train Acc: 65.99%
Test. Loss: 1.071 | Test. Acc: 62.57%

Epoch 11/20
Training: 99% 389/391 [00:12<00:00, 30.11it/s]

Testing: 94% 74/79 [00:02<00:00, 30.53it/s]

Epoch: 11 | Epoch Time: 0m 15s
Train Loss: 0.934 | Train Acc: 67.16%
Test. Loss: 1.082 | Test. Acc: 62.56%

Epoch 12/20
Training: 99% 387/391 [00:12<00:00, 30.83it/s]

Testing: 97% 77/79 [00:02<00:00, 34.41it/s]

Epoch: 12 | Epoch Time: 0m 15s
Train Loss: 0.915 | Train Acc: 67.65%
Test. Loss: 1.087 | Test. Acc: 62.73%

Epoch 13/20
Training: 99% 389/391 [00:12<00:00, 30.87it/s]

Testing: 99% 78/79 [00:02<00:00, 32.28it/s]

Epoch: 13 | Epoch Time: 0m 15s
Train Loss: 0.885 | Train Acc: 68.68%
Test. Loss: 1.035 | Test. Acc: 64.29%

Epoch 14/20
Training: 99% 387/391 [00:12<00:00, 29.80it/s]

Testing: 94% 74/79 [00:02<00:00, 32.49it/s]

Epoch: 14 | Epoch Time: 0m 15s
Train Loss: 0.860 | Train Acc: 69.84%
Test. Loss: 1.049 | Test. Acc: 63.95%

Epoch 15/20
Training: 99% 387/391 [00:12<00:00, 31.07it/s]

Testing: 97% 77/79 [00:02<00:00, 31.15it/s]

Epoch: 15 | Epoch Time: 0m 15s
Train Loss: 0.842 | Train Acc: 70.24%
Test. Loss: 1.050 | Test. Acc: 63.81%

Epoch 16/20
Training: 100% 391/391 [00:12<00:00, 33.01it/s]

Testing: 97% 77/79 [00:02<00:00, 34.48it/s]

Epoch: 16 | Epoch Time: 0m 15s
Train Loss: 0.820 | Train Acc: 71.27%
Test. Loss: 1.027 | Test. Acc: 64.81%

Epoch 17/20
Training: 100% 390/391 [00:12<00:00, 31.52it/s]

Testing: 96% 76/79 [00:02<00:00, 33.92it/s]

Epoch: 17 | Epoch Time: 0m 15s
Train Loss: 0.794 | Train Acc: 72.00%
Test. Loss: 1.052 | Test. Acc: 64.38%

Epoch 18/20
Training: 99% 389/391 [00:13<00:00, 32.06it/s]

Testing: 97% 77/79 [00:03<00:00, 24.54it/s]

Epoch: 18 | Epoch Time: 0m 16s
Train Loss: 0.783 | Train Acc: 72.29%
Test. Loss: 1.048 | Test. Acc: 64.18%

Epoch 19/20
Training: 99% 389/391 [00:13<00:00, 30.50it/s]

Testing: 97% 77/79 [00:03<00:00, 24.93it/s]

Epoch: 19 | Epoch Time: 0m 16s
Train Loss: 0.765 | Train Acc: 72.84%
Test. Loss: 1.046 | Test. Acc: 64.65%

Epoch 20/20
Training: 99% 388/391 [00:13<00:00, 30.59it/s]

Testing: 96% 76/79 [00:02<00:00, 32.94it/s]

Epoch: 20 | Epoch Time: 0m 15s
Train Loss: 0.745 | Train Acc: 73.80%
Test. Loss: 1.064 | Test. Acc: 64.72%

selected optimizer and learning rate: Adam (

Parameter Group 0

- amsgrad: False
- betas: (0.9, 0.999)
- eps: 1e-08
- lr: 0.0001
- maximize: False
- weight_decay: 0

)

100% 20/20 [05:21<00:00, 16.14s/it]

Epoch 1/20
Training: 99% 389/391 [00:13<00:00, 29.80it/s]

Testing: 97% 77/79 [00:02<00:00, 34.16it/s]

Epoch: 01 | Epoch Time: 0m 15s
Train Loss: 2.044 | Train Acc: 25.06%
Test. Loss: 1.844 | Test. Acc: 31.57%

Epoch 2/20
Training: 100% 390/391 [00:13<00:00, 30.98it/s]

Testing: 96% 76/79 [00:02<00:00, 33.15it/s]

Epoch: 02 | Epoch Time: 0m 15s
Train Loss: 1.779 | Train Acc: 35.12%
Test. Loss: 1.706 | Test. Acc: 37.74%

Epoch 3/20
Training: 100% 390/391 [00:13<00:00, 29.38it/s]

Testing: 96% 76/79 [00:02<00:00, 32.61it/s]

Epoch: 03 | Epoch Time: 0m 15s
Train Loss: 1.666 | Train Acc: 39.61%
Test. Loss: 1.615 | Test. Acc: 41.46%

Epoch 4/20
Training: 99% 388/391 [00:12<00:00, 29.59it/s]

Testing: 96% 76/79 [00:02<00:00, 32.59it/s]

Epoch: 04 | Epoch Time: 0m 15s
Train Loss: 1.596 | Train Acc: 42.36%
Test. Loss: 1.558 | Test. Acc: 43.43%

Epoch 5/20
Training: 99% 388/391 [00:12<00:00, 31.02it/s]

Testing: 97% 77/79 [00:02<00:00, 33.59it/s]

```
Epoch: 05 | Epoch Time: 0m 15s
      Train Loss: 1.543 | Train Acc: 44.44%
      Test. Loss: 1.525 | Test. Acc: 44.51%
Epoch 6/20
Training: 100%                               390/391 [00:12<00:00, 32.26it/s]

Testing: 97%                               77/79 [00:02<00:00, 33.67it/s]
Epoch: 06 | Epoch Time: 0m 15s
      Train Loss: 1.502 | Train Acc: 45.97%
      Test. Loss: 1.495 | Test. Acc: 46.39%
Epoch 7/20
Training: 99%                               388/391 [00:12<00:00, 29.27it/s]

Testing: 97%                               77/79 [00:02<00:00, 35.50it/s]
Epoch: 07 | Epoch Time: 0m 15s
      Train Loss: 1.471 | Train Acc: 47.08%
      Test. Loss: 1.461 | Test. Acc: 47.30%
Epoch 8/20
Training: 99%                               387/391 [00:13<00:00, 29.24it/s]

Testing: 97%                               77/79 [00:02<00:00, 33.47it/s]
Epoch: 08 | Epoch Time: 0m 15s
      Train Loss: 1.446 | Train Acc: 48.02%
      Test. Loss: 1.439 | Test. Acc: 48.52%
Epoch 9/20
Training: 99%                               387/391 [00:13<00:00, 28.51it/s]

Testing: 95%                               75/79 [00:02<00:00, 31.77it/s]
Epoch: 09 | Epoch Time: 0m 16s
      Train Loss: 1.423 | Train Acc: 48.79%
      Test. Loss: 1.419 | Test. Acc: 49.38%
Epoch 10/20
Training: 99%                               389/391 [00:13<00:00, 28.70it/s]

Testing: 97%                               77/79 [00:02<00:00, 31.61it/s]
Epoch: 10 | Epoch Time: 0m 16s
      Train Loss: 1.404 | Train Acc: 49.65%
```

```
%tensorboard --logdir runs/CIFAR10_LeNet
```



```
logs = SummaryWriter('runs/CIFAR10_LeNet_RMS')
tb_crf = SummaryWriter(logs)

      Test. Loss: 1.316 | Test. Acc: 53.21%
```

using different optimizer

```
selected_optimizer='RMS'
for learning_rate in [ 0.01, 0.001, 0.0001]:

    OUTPUT_DIM = 10

    model_c = LeNet_CIFAR10(OUTPUT_DIM)

    # def count_parameters(model):
    #     return sum(p.numel() for p in model.parameters() if p.requires_grad)

    # print(f'The model has {count_parameters(model):,} trainable parameters')

    optimizer = optim.RMSprop(model_c.parameters())

    criterion = nn.CrossEntropyLoss()

    #Checking if we can use GPU

    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

```
model_c = model_c.to(device)
criterion = criterion.to(device)

avail_optimizers = {'Adam':torch.optim.Adam(model_c.parameters(), lr=learning_rate, betas=(0.9, 0.999), eps=1e-08, weight_decay=0, amsgrad=False),
                    'RMS': torch.optim.RMSprop(model_c.parameters(), lr=learning_rate, alpha=0.99, eps=1e-08, weight_decay=0, momentum=0, centered=False),
                    'SGD': torch.optim.SGD(model_c.parameters(), lr=learning_rate, momentum=0, dampening=0, weight_decay=0, nesterov=False)}

optimizer = avail_optimizers[selected_optimizer]

epochs = 20
print(f"selected optimizer and learning rate: {optimizer}")
for n_iter in tqdm(range(epochs)):
    print(f"Epoch {n_iter+1}/{epochs}")

    start_time = time.monotonic()

    train_loss, train_acc = train(model_c, train_loader, optimizer, criterion, device)
    test_loss, test_acc = test(model_c, test_loader, criterion, device)

    end_time = time.monotonic()

    epoch_mins, epoch_secs = epoch_time(start_time, end_time)

    print(f'Epoch: {n_iter+1:02} | Epoch Time: {epoch_mins}m {epoch_secs}s')
    print(f'\tTrain Loss: {train_loss:.3f} | Train Acc: {train_acc*100:.2f}%')
    print(f'\t Test. Loss: {test_loss:.3f} | Test. Acc: {test_acc*100:.2f}%')

    if tb_cf is not None:
        tb_cf.add_scalars(f'Loss_{learning_rate}', {"Train":train_loss,
                                                    "Test":test_loss}, n_iter)
        tb_cf.add_scalars(f'Accuracy_{learning_rate}', {"Train":train_acc,
                                                         "Test":test_acc}, n_iter)

        tb_cf.add_scalars(f'Train_Accuracy', {f"{learning_rate}":train_acc}, n_iter)
        tb_cf.add_scalars(f'Test_Accuracy', {f"{learning_rate}":test_acc}, n_iter)

        tb_cf.add_scalars(f'Train_loss', {f"{learning_rate}":train_loss}, n_iter)
        tb_cf.add_scalars(f'Test_loss', {f"{learning_rate}":test_loss}, n_iter)
```

```
selected optimizer and learning rate: RMSprop (
Parameter Group 0
  alpha: 0.99
  centered: False
  eps: 1e-08
  lr: 0.01
  momentum: 0
  weight_decay: 0
)
100% 20/20 [05:14<00:00, 15.72s/it]

Epoch 1/20
Training: 99% 389/391 [00:12<00:00, 32.00it/s]

Testing: 97% 77/79 [00:02<00:00, 33.65it/s]
Epoch: 01 | Epoch Time: 0m 15s
  Train Loss: 3.660 | Train Acc: 18.63%
  Test. Loss: 1.935 | Test. Acc: 27.36%
Epoch 2/20
Training: 99% 387/391 [00:12<00:00, 29.91it/s]

Testing: 100% 79/79 [00:03<00:00, 24.53it/s]
Epoch: 02 | Epoch Time: 0m 15s
  Train Loss: 1.917 | Train Acc: 28.40%
  Test. Loss: 1.876 | Test. Acc: 29.77%
Epoch 3/20
Training: 99% 389/391 [00:12<00:00, 31.95it/s]

Testing: 97% 77/79 [00:02<00:00, 34.04it/s]
Epoch: 03 | Epoch Time: 0m 15s
  Train Loss: 1.868 | Train Acc: 30.60%
  Test. Loss: 1.838 | Test. Acc: 31.41%
Epoch 4/20
Training: 99% 387/391 [00:12<00:00, 31.72it/s]

Testing: 97% 77/79 [00:02<00:00, 33.64it/s]
Epoch: 04 | Epoch Time: 0m 15s
  Train Loss: 1.779 | Train Acc: 33.74%
  Test. Loss: 1.688 | Test. Acc: 36.50%
Epoch 5/20
Training: 99% 388/391 [00:13<00:00, 29.94it/s]

Testing: 97% 77/79 [00:02<00:00, 33.18it/s]
Epoch: 05 | Epoch Time: 0m 15s
  Train Loss: 1.687 | Train Acc: 37.64%
  Test. Loss: 1.660 | Test. Acc: 37.04%
Epoch 6/20
Training: 99% 387/391 [00:13<00:00, 21.14it/s]

Testing: 96% 76/79 [00:02<00:00, 35.15it/s]
Epoch: 06 | Epoch Time: 0m 16s
  Train Loss: 1.620 | Train Acc: 40.57%
  Test. Loss: 1.588 | Test. Acc: 42.65%
Epoch 7/20
Training: 99% 387/391 [00:12<00:00, 31.17it/s]

Testing: 99% 78/79 [00:02<00:00, 33.33it/s]
Epoch: 07 | Epoch Time: 0m 15s
  Train Loss: 1.552 | Train Acc: 43.74%
  Test. Loss: 1.519 | Test. Acc: 44.80%
Epoch 8/20
Training: 99% 387/391 [00:13<00:00, 28.56it/s]

Testing: 99% 78/79 [00:02<00:00, 32.95it/s]
Epoch: 08 | Epoch Time: 0m 15s
  Train Loss: 1.500 | Train Acc: 45.76%
  Test. Loss: 1.554 | Test. Acc: 45.07%
Epoch 9/20
Training: 99% 387/391 [00:12<00:00, 30.80it/s]

Testing: 97% 77/79 [00:02<00:00, 35.24it/s]
Epoch: 09 | Epoch Time: 0m 15s
  Train Loss: 1.458 | Train Acc: 47.65%
  Test. Loss: 1.669 | Test. Acc: 43.86%
Epoch 10/20
Training: 99% 387/391 [00:12<00:00, 30.30it/s]

Testing: 99% 78/79 [00:02<00:00, 33.24it/s]
Epoch: 10 | Epoch Time: 0m 15s
  Train Loss: 1.422 | Train Acc: 48.82%
  Test. Loss: 1.461 | Test. Acc: 47.72%
Epoch 11/20
Training: 99% 389/391 [00:13<00:00, 29.50it/s]

Testing: 99% 78/79 [00:02<00:00, 35.64it/s]
Epoch: 11 | Epoch Time: 0m 15s
  Train Loss: 1.399 | Train Acc: 49.92%
  Test. Loss: 1.501 | Test. Acc: 46.84%
Epoch 12/20
Training: 99% 388/391 [00:12<00:00, 30.93it/s]

Testing: 96% 76/79 [00:02<00:00, 33.85it/s]
Epoch: 12 | Epoch Time: 0m 15s
  Train Loss: 1.381 | Train Acc: 50.88%
  Test. Loss: 1.552 | Test. Acc: 46.24%
Epoch 13/20
Training: 99% 389/391 [00:12<00:00, 30.34it/s]

Testing: 97% 77/79 [00:02<00:00, 33.28it/s]
Epoch: 13 | Epoch Time: 0m 15s
  Train Loss: 1.355 | Train Acc: 51.79%
  Test. Loss: 1.535 | Test. Acc: 46.45%
Epoch 14/20
Training: 99% 389/391 [00:13<00:00, 30.91it/s]

Testing: 96% 76/79 [00:02<00:00, 33.37it/s]
Epoch: 14 | Epoch Time: 0m 15s
  Train Loss: 1.339 | Train Acc: 52.57%
  Test. Loss: 1.438 | Test. Acc: 49.23%
Epoch 15/20
Training: 99% 387/391 [00:12<00:00, 30.15it/s]

Testing: 96% 76/79 [00:02<00:00, 33.32it/s]
Epoch: 15 | Epoch Time: 0m 15s
  Train Loss: 1.315 | Train Acc: 53.32%
```

```
      train Loss: 1.315 | Train Acc: 53.32%
      Test. Loss: 1.549 | Test. Acc: 46.94%
Epoch 16/20
Training: 99%                                389/391 [00:13<00:00, 30.33it/s]

Testing: 97%                                77/79 [00:02<00:00, 34.12it/s]
Epoch: 16 | Epoch Time: 0m 15s
      Train Loss: 1.307 | Train Acc: 53.93%
      Test. Loss: 1.674 | Test. Acc: 43.06%
Epoch 17/20
Training: 100%                               390/391 [00:13<00:00, 30.12it/s]

Testing: 96%                                76/79 [00:02<00:00, 32.52it/s]
Epoch: 17 | Epoch Time: 0m 15s
      Train Loss: 1.289 | Train Acc: 54.35%
      Test. Loss: 1.526 | Test. Acc: 47.93%
Epoch 18/20
Training: 100%                               390/391 [00:13<00:00, 32.18it/s]

Testing: 96%                                76/79 [00:02<00:00, 33.49it/s]
Epoch: 18 | Epoch Time: 0m 15s
      Train Loss: 1.284 | Train Acc: 54.95%
      Test. Loss: 1.585 | Test. Acc: 44.58%
Epoch 19/20
Training: 100%                               390/391 [00:12<00:00, 31.04it/s]

Testing: 96%                                76/79 [00:02<00:00, 31.87it/s]
Epoch: 19 | Epoch Time: 0m 15s
      Train Loss: 1.269 | Train Acc: 55.24%
      Test. Loss: 1.408 | Test. Acc: 51.03%
Epoch 20/20
Training: 99%                                387/391 [00:12<00:00, 30.42it/s]

Testing: 97%                                77/79 [00:02<00:00, 34.94it/s]
Epoch: 20 | Epoch Time: 0m 15s
      Train Loss: 1.258 | Train Acc: 55.67%
      Test. Loss: 1.477 | Test. Acc: 49.67%
selected optimizer and learning rate: RMSprop (
Parameter Group 0
  alpha: 0.99
  centered: False
  eps: 1e-08
  lr: 0.001
  momentum: 0
  weight_decay: 0
)
100%                                         20/20 [05:17<00:00, 15.89s/it]

Epoch 1/20
Training: 99%                                389/391 [00:13<00:00, 29.47it/s]

Testing: 95%                                75/79 [00:02<00:00, 32.71it/s]
Epoch: 01 | Epoch Time: 0m 15s
      Train Loss: 1.708 | Train Acc: 37.69%
      Test. Loss: 1.623 | Test. Acc: 40.85%
Epoch 2/20
Training: 100%                               391/391 [00:13<00:00, 32.75it/s]

Testing: 99%                                78/79 [00:03<00:00, 22.99it/s]
Epoch: 02 | Epoch Time: 0m 16s
      Train Loss: 1.442 | Train Acc: 48.17%
      Test. Loss: 1.382 | Test. Acc: 50.37%
Epoch 3/20
Training: 99%                                388/391 [00:13<00:00, 29.05it/s]

Testing: 95%                                75/79 [00:02<00:00, 30.67it/s]
Epoch: 03 | Epoch Time: 0m 15s
      Train Loss: 1.336 | Train Acc: 52.14%
      Test. Loss: 1.370 | Test. Acc: 50.55%
Epoch 4/20
Training: 99%                                387/391 [00:12<00:00, 28.11it/s]

Testing: 96%                                76/79 [00:02<00:00, 33.36it/s]
Epoch: 04 | Epoch Time: 0m 15s
      Train Loss: 1.261 | Train Acc: 54.76%
      Test. Loss: 1.313 | Test. Acc: 53.37%
Epoch 5/20
Training: 99%                                387/391 [00:13<00:00, 29.47it/s]

Testing: 95%                                75/79 [00:02<00:00, 32.50it/s]
Epoch: 05 | Epoch Time: 0m 16s
      Train Loss: 1.205 | Train Acc: 56.94%
      Test. Loss: 1.236 | Test. Acc: 55.93%
Epoch 6/20
Training: 99%                                388/391 [00:13<00:00, 30.11it/s]

Testing: 97%                                77/79 [00:02<00:00, 32.32it/s]
Epoch: 06 | Epoch Time: 0m 15s
      Train Loss: 1.152 | Train Acc: 59.01%
      Test. Loss: 1.224 | Test. Acc: 57.12%
Epoch 7/20
Training: 99%                                389/391 [00:13<00:00, 30.12it/s]

Testing: 97%                                77/79 [00:02<00:00, 33.25it/s]
Epoch: 07 | Epoch Time: 0m 15s
      Train Loss: 1.111 | Train Acc: 60.39%
      Test. Loss: 1.237 | Test. Acc: 56.28%
Epoch 8/20
Training: 99%                                388/391 [00:12<00:00, 30.75it/s]

Testing: 95%                                75/79 [00:02<00:00, 33.74it/s]
Epoch: 08 | Epoch Time: 0m 15s
      Train Loss: 1.070 | Train Acc: 61.87%
      Test. Loss: 1.154 | Test. Acc: 58.97%
Epoch 9/20
Training: 100%                               391/391 [00:13<00:00, 32.23it/s]

Testing: 97%                                77/79 [00:02<00:00, 32.51it/s]
Epoch: 09 | Epoch Time: 0m 15s
      Train Loss: 1.041 | Train Acc: 63.05%
      Test. Loss: 1.201 | Test. Acc: 57.45%
Epoch 10/20
Training: 100%                               390/391 [00:12<00:00, 32.50it/s]

Testing: 97%                                77/79 [00:02<00:00, 33.10it/s]
```

```
Epoch: 10 | Epoch Time: 0m 15s
      Train Loss: 1.001 | Train Acc: 64.34%
      Test. Loss: 1.103 | Test. Acc: 60.68%

Epoch 11/20
Training: 99%                                387/391 [00:13<00:00, 28.73it/s]

Testing: 97%                                77/79 [00:02<00:00, 32.32it/s]

Epoch: 11 | Epoch Time: 0m 15s
      Train Loss: 0.969 | Train Acc: 65.67%
      Test. Loss: 1.128 | Test. Acc: 60.82%

Epoch 12/20
Training: 99%                                389/391 [00:12<00:00, 31.50it/s]

Testing: 99%                                78/79 [00:02<00:00, 32.91it/s]

Epoch: 12 | Epoch Time: 0m 15s
      Train Loss: 0.943 | Train Acc: 66.74%
      Test. Loss: 1.103 | Test. Acc: 61.72%
```

▼ BONUS

```
Training: 99%                                387/391 [00:13<00:00, 28.73it/s]
```

Reference: <https://www.youtube.com/watch?v=1ZbLA7ofasY>

```
      Test. Loss: 1.150 | Test. Acc: 60.76%
```

```
logs+=·'runs/mnist_Bonus'
tb_cf+=·SummaryWriter(logs)

      Testin: 96%                                76/79 [00:02<00:00, 34.48it/s]

OUTPUT_DIM = 10

model = LeNet(OUTPUT_DIM)

def activation_hook(inst, inp, out):
    tb_cf.add_histogram(repr(inst), out)

model.conv1.register_forward_hook(activation_hook)
model.conv2.register_forward_hook(activation_hook)
model.cpu()

EPOCHS = 10

# best_valid_loss = float('inf')
# optimizer = optim.Adam(model.parameters())
learning_rate = 0.001
optimizer = optim.Adam(model.parameters(), lr=learning_rate)
criterion = nn.CrossEntropyLoss()

#Checking if we can use GPU

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

model = model.to(device)
criterion = criterion.to(device)

print("Optimizer:", optimizer)
for epoch in trange(EPOCHS, desc="Epochs"):

    start_time = time.monotonic()

    train_loss, train_acc = train(model, train_iterator, optimizer, criterion, device)
    test_loss, test_acc = test(model, test_iterator, criterion, device)

    end_time = time.monotonic()

    epoch_mins, epoch_secs = epoch_time(start_time, end_time)

    print(f'Epoch: {epoch+1:02} | Epoch Time: {epoch_mins}m {epoch_secs}s')
    print(f'\tTrain Loss: {train_loss:.3f} | Train Acc: {train_acc*100:.2f}%')
    print(f'\t Test. Loss: {test_loss:.3f} | Val. Acc: {test_acc*100:.2f}%')

    if tb_cf is not None:
        tb_cf.add_scalars(f'Loss_{learning_rate}', {"Train":train_loss,
                                                    "Test":test_loss}, epoch)
        tb_cf.add_scalars(f'Accuracy_{learning_rate}', {"Train":train_acc,
                                                         "Test":test_acc}, epoch)

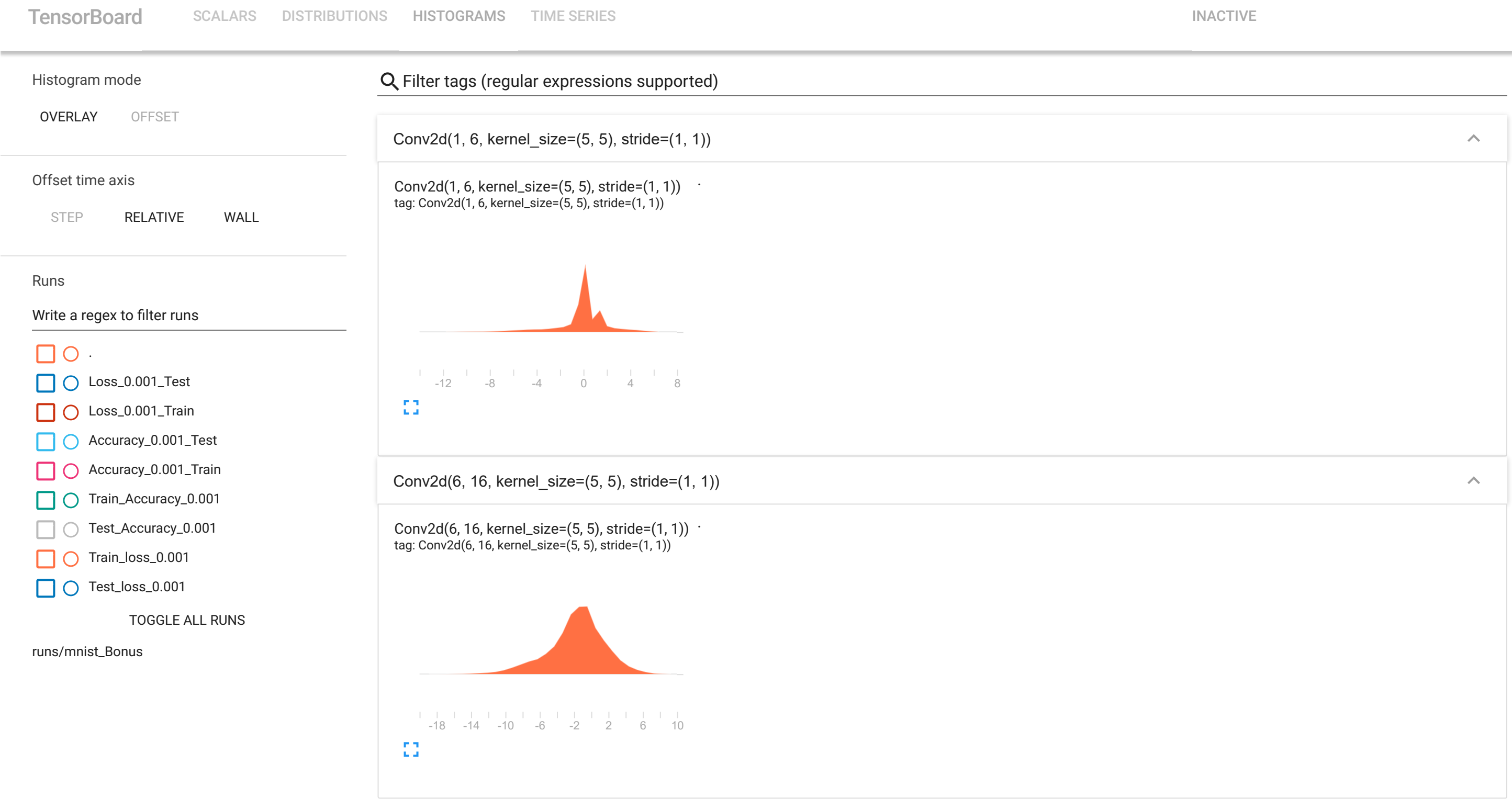
        tb_cf.add_scalars(f'Train_Accuracy', {f"{learning_rate}":train_acc}, epoch)
        tb_cf.add_scalars(f'Test_Accuracy', {f"{learning_rate}":test_acc}, epoch)

        tb_cf.add_scalars(f'Train_loss', {f"{learning_rate}":train_loss}, epoch)
        tb_cf.add_scalars(f'Test_loss', {f"{learning_rate}":test_loss}, epoch)
```



```
Optimizer: Adam (  
Parameter Group 0  
    amsgrad: False  
    betas: (0.9, 0.999)  
    eps: 1e-08  
    lr: 0.001  
    maximize: False  
    weight_decay: 0  
)  
Epochs: 100%  
10/10 [09:05<00:00, 53.91s/it]
```

%tensorboard --logdir runs/mnist_Bonus



```
print(f'\tTrain Loss: {train_loss:.3f} | Train Acc: {train_acc*100:.2f}%')
print(f'\t Test. Loss: {test_loss:.3f} | Val. Acc: {test_acc*100:.2f}%')

# if tb is not None:
#     tb.add_scalars(f'Loss_{learning_rate}', {"Train":train_loss,
#                                             "Test":test_loss}, epoch)
#     tb.add_scalars(f'Accuracy_{learning_rate}', {"Train":train_acc,
#                                                  "Test":test_acc}, epoch)

#     tb.add_scalars(f'Train_Accuracy', {f"{learning_rate}":train_acc}, epoch)
#     tb.add_scalars(f'Test_Accuracy', {f"{learning_rate}":test_acc}, epoch)

#     tb.add_scalars(f'Train_loss', {f"{learning_rate}":train_loss}, epoch)
#     tb.add_scalars(f'Test_loss', {f"{learning_rate}":test_loss}, epoch)
```

```
Optimizer: Adam (
Parameter Group 0
  amsgrad: False
  betas: (0.9, 0.999)
  eps: 1e-08
  lr: 0.001
  maximize: False
  weight_decay: 0
)
```

Epochs: 100%5/5 [04:58<00:00, 59.93s/it]

```
Epoch: 01 | Epoch Time: 1m 0s
  Train Loss: 1.734 | Train Acc: 36.35%
  Test. Loss: 1.524 | Val. Acc: 44.28%
Epoch: 02 | Epoch Time: 0m 58s
  Train Loss: 1.439 | Train Acc: 47.84%
  Test. Loss: 1.385 | Val. Acc: 49.78%
Epoch: 03 | Epoch Time: 0m 59s
  Train Loss: 1.312 | Train Acc: 52.82%
  Test. Loss: 1.281 | Val. Acc: 54.01%
Epoch: 04 | Epoch Time: 0m 59s
  Train Loss: 1.231 | Train Acc: 56.07%
  Test. Loss: 1.256 | Val. Acc: 54.90%
Epoch: 05 | Epoch Time: 1m 0s
  Train Loss: 1.172 | Train Acc: 58.29%
  Test. Loss: 1.209 | Val. Acc: 56.29%
```

```
from tensorboard import notebook
notebook.list()
```

Known TensorBoard instances:

- port 6007: logdir runs/mnist_LeNet (started 0:33:26 ago; pid 7019)
- port 6008: logdir runs/CIFAR10_BONUS (started 0:29:40 ago; pid 7137)
- port 6006: logdir runs/CIFAR10_LeNet_RMS (started 0:54:22 ago; pid 6266)
- port 6009: logdir runs/CIFAR10_LeNet (started 0:18:10 ago; pid 7506)
- port 6010: logdir runs/mnist_Bonus (started 0:05:38 ago; pid 7975)

```
%tensorboard --logdir runs/CIFAR10_BONUS
```

Reusing TensorBoard on port 6008 (pid 7137), started 0:29:46 ago. (Use '!kill 7137' to kill it.)

TensorBoard

SCALARS

DISTRIBUTIONS

HISTOGRAMS

TIME SERIES

INACTIVE

Histogram mode

OVERLAY

OFFSET

Offset time axis

STEP

RELATIVE

WALL

Runs

Write a regex to filter runs

Loss_0.001_Test

Loss_0.001_Train

Accuracy_0.001_Test

Accuracy_0.001_Train

Train_Accuracy_0.001

Test_Accuracy_0.001

Train_loss_0.001

Test_loss_0.001


TOGGLE ALL RUNS

runs/CIFAR10_BONUS

Q Filter tags (regular expressions supported)

Conv2d(3, 6, kernel_size=(5, 5), stride=(1, 1))

Conv2d(3, 6, kernel_size=(5, 5), stride=(1, 1)) · tag: Conv2d(3, 6, kernel_size=(5, 5), stride=(1, 1))



Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))

Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1)) · tag: Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))



```
!wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
from colab_pdf import colab_pdf
colab_pdf('DDA06_LeNet_Nabawi309498.ipynb')
```

```
--2022-06-17 18:12:57-- https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1864 (1.8K) [text/plain]
Saving to: 'colab_pdf.py'

colab_pdf.py      100%[=====>]    1.82K  --.-KB/s    in 0s

2022-06-17 18:12:57 (42.7 MB/s) - 'colab_pdf.py' saved [1864/1864]

Mounted at /content/drive/

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Extracting templates from packages: 100%
[NbConvertApp] Converting notebook /content/drive/MyDrive/Colab Notebooks/DDA06_LeNet_Nabawi309498.ipynb to pdf
[NbConvertApp] ERROR | Notebook JSON is invalid: data must be valid exactly by one of oneOf definition

Failed validating <unset> in notebook:

On instance:
<unset>
[NbConvertApp] Writing 206848 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', './notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', './notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 135615 bytes to /content/drive/My Drive/DDA06_LeNet_Nabawi309498.pdf
'File ready to be Downloaded and Saved to Drive'
```