.ipynb

July 4, 2022

[1]:
```
%config Completer.use_jedi = False
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torchvision import datasets, transforms

import time
```

[2]:
```
class Arguments():
    def __init__(self):
        self.batch_size = 64
        self.test_batch_size = 64
        self.epochs = 20
        self.lr = 0.02
        self.seed = 1
        self.log_interval = 1 # Log info at each batch
        self.precision_fractional = 3

args = Arguments()

_ = torch.manual_seed(args.seed)
```

1 PySft

[3]:
```
import syft as sy  # import the Pysyft library
hook = sy.TorchHook(torch)  # hook PyTorch to add extra functionalities like␣
 ↪Federated and Encrypted Learning

# simulation functions
def connect_to_workers(n_workers):
    return [
        sy.VirtualWorker(hook, id=f"worker{i+1}")
        for i in range(n_workers)
    ]
def connect_to_crypto_provider():
```

```
    return sy.VirtualWorker(hook, id="crypto_provider")

workers = connect_to_workers(n_workers=2)
crypto_provider = connect_to_crypto_provider()
```

2 Access and secret

The workers then split their data in batches and secret share their data between each others. Local worker (like us) never had access to the data.

[4]:
```
# We don't use the whole dataset for efficiency purpose, but feel free to␣
↪increase these numbers
n_train_items = 640#can be changed
n_test_items = 640#can be changed

def get_private_data_loaders(precision_fractional, workers, crypto_provider):

    def one_hot_of(index_tensor):
        """
        Transform to one hot tensor

        Example:
            [0, 3, 9]
            =>
            [[1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
             [0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],
             [0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]]

        """
        onehot_tensor = torch.zeros(*index_tensor.shape, 10) # 10 classes for␣
↪MNIST
        onehot_tensor = onehot_tensor.scatter(1, index_tensor.view(-1, 1), 1)
        return onehot_tensor

    def secret_share(tensor):
        """
        Transform to fixed precision and secret share a tensor
        """
        return (
            tensor
            .fix_precision(precision_fractional=precision_fractional)
            .share(*workers, crypto_provider=crypto_provider,␣
↪requires_grad=True)
        )

    transformation = transforms.Compose([
```

```
        transforms.ToTensor(),
        transforms.Normalize((0.1307,), (0.3081,))
    ])

    train_loader = torch.utils.data.DataLoader(
        datasets.MNIST('../data', train=True, download=True,␣
↪transform=transformation),
        batch_size=args.batch_size
    )

    private_train_loader = [
        (secret_share(data), secret_share(one_hot_of(target)))
        for i, (data, target) in enumerate(train_loader)
        if i < n_train_items / args.batch_size
    ]

    test_loader = torch.utils.data.DataLoader(
        datasets.MNIST('../data', train=False, download=True,␣
↪transform=transformation),
        batch_size=args.test_batch_size
    )

    private_test_loader = [
        (secret_share(data), secret_share(target.float()))
        for i, (data, target) in enumerate(test_loader)
        if i < n_test_items / args.test_batch_size
    ]

    return private_train_loader, private_test_loader


private_train_loader, private_test_loader = get_private_data_loaders(
    precision_fractional=args.precision_fractional,
    workers=workers,
    crypto_provider=crypto_provider
)
```

```
1.1%

Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz to
../data/MNIST/raw/train-images-idx3-ubyte.gz

100.1%

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

5.5%5%

Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz to
../data/MNIST/raw/train-labels-idx1-ubyte.gz
```

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

180.4%

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 to
../data/MNIST/raw/t10k-labels-idx1-ubyte.gz
Extracting ../data/MNIST/raw/t10k-labels-idx1-ubyte.gz to ../data/MNIST/raw
Processing…
Done!
```

3 Model specification

[5]:
```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(28 * 28, 128)
        self.fc2 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, 10)

    def forward(self, x):
        x = x.view(-1, 28 * 28)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

4 Training and testing functions

[6]:
```
def train(args, model, private_train_loader, optimizer, epoch):
    model.train()
    for batch_idx, (data, target) in enumerate(private_train_loader): # <-- now␣
 ↪it is a private dataset
        start_time = time.time()

        optimizer.zero_grad()

        output = model(data)

        # loss = F.nll_loss(output, target)  <-- not possible here
        batch_size = output.shape[0]
        loss = ((output - target)**2).sum().refresh()/batch_size

        loss.backward()
```

```
        optimizer.step()

        if batch_idx % args.log_interval == 0:
            loss = loss.get().float_precision()
            print('Train Epoch: {} [{}/{} ({:.0f}%)]\tLoss: {:.6f}\tTime: {:.
↪3f}s'.format(
                epoch, batch_idx * args.batch_size, len(private_train_loader) *␣
↪args.batch_size,
                100. * batch_idx / len(private_train_loader), loss.item(), time.
↪time() - start_time))
```

[7]:
```
def test(args, model, private_test_loader):
    model.eval()
    test_loss = 0
    correct = 0
    with torch.no_grad():
        for data, target in private_test_loader:
            start_time = time.time()

            output = model(data)
            pred = output.argmax(dim=1)
            correct += pred.eq(target.view_as(pred)).sum()

    correct = correct.get().float_precision()
    print('\nTest set: Accuracy: {}/{} ({:.0f}%)\n'.format(
        correct.item(), len(private_test_loader)* args.test_batch_size,
        100. * correct.item() / (len(private_test_loader) * args.
↪test_batch_size)))
```

5 Trainning

[8]:
```
model = Net()
model = model.fix_precision().share(*workers, crypto_provider=crypto_provider,␣
↪requires_grad=True)

optimizer = optim.SGD(model.parameters(), lr=args.lr)
optimizer = optimizer.fix_precision()

for epoch in range(1, args.epochs + 1):
    train(args, model, private_train_loader, optimizer, epoch)
    test(args, model, private_test_loader)
```

```
/home/fjunyuan/.local/lib/python3.8/site-
packages/syft/frameworks/torch/tensors/interpreters/additive_shared.py:122:
UserWarning: Use dtype instead of field
```

```
  warnings.warn("Use dtype instead of field")
Train Epoch: 1 [0/640 (0%)]     Loss: 1.128000  Time: 7.122s
Train Epoch: 1 [64/640 (10%)]   Loss: 1.013000  Time: 8.410s
Train Epoch: 1 [128/640 (20%)]  Loss: 0.988000  Time: 8.142s
Train Epoch: 1 [192/640 (30%)]  Loss: 0.901000  Time: 7.185s
Train Epoch: 1 [256/640 (40%)]  Loss: 0.887000  Time: 6.988s
Train Epoch: 1 [320/640 (50%)]  Loss: 0.875000  Time: 7.344s
Train Epoch: 1 [384/640 (60%)]  Loss: 0.852000  Time: 7.173s
Train Epoch: 1 [448/640 (70%)]  Loss: 0.849000  Time: 7.142s
Train Epoch: 1 [512/640 (80%)]  Loss: 0.829000  Time: 8.226s
Train Epoch: 1 [576/640 (90%)]  Loss: 0.839000  Time: 8.102s

Test set: Accuracy: 233.0/640 (36%)

Train Epoch: 2 [0/640 (0%)]     Loss: 0.777000  Time: 7.340s
Train Epoch: 2 [64/640 (10%)]   Loss: 0.732000  Time: 7.606s
Train Epoch: 2 [128/640 (20%)]  Loss: 0.792000  Time: 8.453s
Train Epoch: 2 [192/640 (30%)]  Loss: 0.713000  Time: 8.054s
Train Epoch: 2 [256/640 (40%)]  Loss: 0.701000  Time: 7.574s
Train Epoch: 2 [320/640 (50%)]  Loss: 0.705000  Time: 7.435s
Train Epoch: 2 [384/640 (60%)]  Loss: 0.704000  Time: 7.949s
Train Epoch: 2 [448/640 (70%)]  Loss: 0.714000  Time: 7.650s
Train Epoch: 2 [512/640 (80%)]  Loss: 0.709000  Time: 7.618s
Train Epoch: 2 [576/640 (90%)]  Loss: 0.742000  Time: 7.692s

Test set: Accuracy: 359.0/640 (56%)

Train Epoch: 3 [0/640 (0%)]     Loss: 0.668000  Time: 6.621s
Train Epoch: 3 [64/640 (10%)]   Loss: 0.598000  Time: 6.548s
Train Epoch: 3 [128/640 (20%)]  Loss: 0.696000  Time: 6.273s
Train Epoch: 3 [192/640 (30%)]  Loss: 0.601000  Time: 7.431s
Train Epoch: 3 [256/640 (40%)]  Loss: 0.584000  Time: 6.623s
Train Epoch: 3 [320/640 (50%)]  Loss: 0.597000  Time: 6.638s
Train Epoch: 3 [384/640 (60%)]  Loss: 0.605000  Time: 6.403s
Train Epoch: 3 [448/640 (70%)]  Loss: 0.625000  Time: 6.455s
Train Epoch: 3 [512/640 (80%)]  Loss: 0.623000  Time: 6.311s
Train Epoch: 3 [576/640 (90%)]  Loss: 0.670000  Time: 6.066s

Test set: Accuracy: 401.0/640 (63%)

Train Epoch: 4 [0/640 (0%)]     Loss: 0.586000  Time: 6.373s
Train Epoch: 4 [64/640 (10%)]   Loss: 0.501000  Time: 6.751s
Train Epoch: 4 [128/640 (20%)]  Loss: 0.620000  Time: 6.876s
Train Epoch: 4 [192/640 (30%)]  Loss: 0.516000  Time: 6.587s
Train Epoch: 4 [256/640 (40%)]  Loss: 0.507000  Time: 6.195s
Train Epoch: 4 [320/640 (50%)]  Loss: 0.517000  Time: 6.075s
Train Epoch: 4 [384/640 (60%)]  Loss: 0.535000  Time: 6.084s
```

```
Train Epoch: 4 [448/640 (70%)]  Loss: 0.555000  Time: 7.216s
Train Epoch: 4 [512/640 (80%)]  Loss: 0.554000  Time: 6.603s
Train Epoch: 4 [576/640 (90%)]  Loss: 0.612000  Time: 6.714s

Test set: Accuracy: 432.0/640 (68%)

Train Epoch: 5 [0/640 (0%)]     Loss: 0.530000  Time: 6.788s
Train Epoch: 5 [64/640 (10%)]   Loss: 0.441000  Time: 6.616s
Train Epoch: 5 [128/640 (20%)]  Loss: 0.570000  Time: 7.035s
Train Epoch: 5 [192/640 (30%)]  Loss: 0.457000  Time: 7.138s
Train Epoch: 5 [256/640 (40%)]  Loss: 0.450000  Time: 6.405s
Train Epoch: 5 [320/640 (50%)]  Loss: 0.458000  Time: 6.456s
Train Epoch: 5 [384/640 (60%)]  Loss: 0.486000  Time: 6.544s
Train Epoch: 5 [448/640 (70%)]  Loss: 0.507000  Time: 6.563s
Train Epoch: 5 [512/640 (80%)]  Loss: 0.503000  Time: 7.044s
Train Epoch: 5 [576/640 (90%)]  Loss: 0.572000  Time: 6.831s

Test set: Accuracy: 452.0/640 (71%)

Train Epoch: 6 [0/640 (0%)]     Loss: 0.487000  Time: 7.746s
Train Epoch: 6 [64/640 (10%)]   Loss: 0.390000  Time: 7.019s
Train Epoch: 6 [128/640 (20%)]  Loss: 0.523000  Time: 6.706s
Train Epoch: 6 [192/640 (30%)]  Loss: 0.411000  Time: 6.841s
Train Epoch: 6 [256/640 (40%)]  Loss: 0.409000  Time: 6.239s
Train Epoch: 6 [320/640 (50%)]  Loss: 0.411000  Time: 6.206s
Train Epoch: 6 [384/640 (60%)]  Loss: 0.444000  Time: 6.623s
Train Epoch: 6 [448/640 (70%)]  Loss: 0.468000  Time: 6.851s
Train Epoch: 6 [512/640 (80%)]  Loss: 0.462000  Time: 6.674s
Train Epoch: 6 [576/640 (90%)]  Loss: 0.537000  Time: 6.838s

Test set: Accuracy: 458.0/640 (72%)

Train Epoch: 7 [0/640 (0%)]     Loss: 0.449000  Time: 6.204s
Train Epoch: 7 [64/640 (10%)]   Loss: 0.353000  Time: 6.545s
Train Epoch: 7 [128/640 (20%)]  Loss: 0.490000  Time: 6.274s
Train Epoch: 7 [192/640 (30%)]  Loss: 0.380000  Time: 6.300s
Train Epoch: 7 [256/640 (40%)]  Loss: 0.377000  Time: 6.291s
Train Epoch: 7 [320/640 (50%)]  Loss: 0.377000  Time: 6.267s
Train Epoch: 7 [384/640 (60%)]  Loss: 0.410000  Time: 6.311s
Train Epoch: 7 [448/640 (70%)]  Loss: 0.438000  Time: 6.277s
Train Epoch: 7 [512/640 (80%)]  Loss: 0.429000  Time: 6.243s
Train Epoch: 7 [576/640 (90%)]  Loss: 0.509000  Time: 6.463s

Test set: Accuracy: 464.0/640 (72%)

Train Epoch: 8 [0/640 (0%)]     Loss: 0.418000  Time: 6.371s
Train Epoch: 8 [64/640 (10%)]   Loss: 0.329000  Time: 6.285s
Train Epoch: 8 [128/640 (20%)]  Loss: 0.461000  Time: 6.294s
```

```
Train Epoch: 8 [192/640 (30%)]  Loss: 0.349000  Time: 6.667s
Train Epoch: 8 [256/640 (40%)]  Loss: 0.351000  Time: 6.367s
Train Epoch: 8 [320/640 (50%)]  Loss: 0.348000  Time: 6.295s
Train Epoch: 8 [384/640 (60%)]  Loss: 0.382000  Time: 6.419s
Train Epoch: 8 [448/640 (70%)]  Loss: 0.414000  Time: 6.328s
Train Epoch: 8 [512/640 (80%)]  Loss: 0.402000  Time: 6.323s
Train Epoch: 8 [576/640 (90%)]  Loss: 0.485000  Time: 6.399s

Test set: Accuracy: 473.0/640 (74%)

Train Epoch: 9 [0/640 (0%)]     Loss: 0.392000  Time: 7.137s
Train Epoch: 9 [64/640 (10%)]   Loss: 0.303000  Time: 7.304s
Train Epoch: 9 [128/640 (20%)]  Loss: 0.437000  Time: 7.145s
Train Epoch: 9 [192/640 (30%)]  Loss: 0.327000  Time: 8.104s
Train Epoch: 9 [256/640 (40%)]  Loss: 0.328000  Time: 7.361s
Train Epoch: 9 [320/640 (50%)]  Loss: 0.322000  Time: 8.650s
Train Epoch: 9 [384/640 (60%)]  Loss: 0.359000  Time: 7.486s
Train Epoch: 9 [448/640 (70%)]  Loss: 0.390000  Time: 6.788s
Train Epoch: 9 [512/640 (80%)]  Loss: 0.378000  Time: 6.768s
Train Epoch: 9 [576/640 (90%)]  Loss: 0.462000  Time: 6.895s

Test set: Accuracy: 478.0/640 (75%)

Train Epoch: 10 [0/640 (0%)]    Loss: 0.369000  Time: 7.763s
Train Epoch: 10 [64/640 (10%)]  Loss: 0.284000  Time: 7.823s
Train Epoch: 10 [128/640 (20%)] Loss: 0.412000  Time: 8.615s
Train Epoch: 10 [192/640 (30%)] Loss: 0.307000  Time: 6.964s
Train Epoch: 10 [256/640 (40%)] Loss: 0.310000  Time: 6.919s
Train Epoch: 10 [320/640 (50%)] Loss: 0.304000  Time: 7.292s
Train Epoch: 10 [384/640 (60%)] Loss: 0.337000  Time: 7.472s
Train Epoch: 10 [448/640 (70%)] Loss: 0.371000  Time: 6.948s
Train Epoch: 10 [512/640 (80%)] Loss: 0.356000  Time: 7.138s
Train Epoch: 10 [576/640 (90%)] Loss: 0.442000  Time: 8.508s

Test set: Accuracy: 487.0/640 (76%)

Train Epoch: 11 [0/640 (0%)]    Loss: 0.349000  Time: 7.835s
Train Epoch: 11 [64/640 (10%)]  Loss: 0.264000  Time: 7.415s
Train Epoch: 11 [128/640 (20%)] Loss: 0.388000  Time: 9.328s
Train Epoch: 11 [192/640 (30%)] Loss: 0.287000  Time: 9.295s
Train Epoch: 11 [256/640 (40%)] Loss: 0.291000  Time: 7.787s
Train Epoch: 11 [320/640 (50%)] Loss: 0.284000  Time: 8.170s
Train Epoch: 11 [384/640 (60%)] Loss: 0.320000  Time: 8.380s
Train Epoch: 11 [448/640 (70%)] Loss: 0.352000  Time: 7.215s
Train Epoch: 11 [512/640 (80%)] Loss: 0.340000  Time: 7.226s
Train Epoch: 11 [576/640 (90%)] Loss: 0.423000  Time: 7.814s

Test set: Accuracy: 483.0/640 (75%)
```

```
Train Epoch: 12 [0/640 (0%)]    Loss: 0.331000  Time: 6.787s
Train Epoch: 12 [64/640 (10%)]  Loss: 0.250000  Time: 6.820s
Train Epoch: 12 [128/640 (20%)] Loss: 0.367000  Time: 6.936s
Train Epoch: 12 [192/640 (30%)] Loss: 0.273000  Time: 6.909s
Train Epoch: 12 [256/640 (40%)] Loss: 0.276000  Time: 6.907s
Train Epoch: 12 [320/640 (50%)] Loss: 0.266000  Time: 6.877s
Train Epoch: 12 [384/640 (60%)] Loss: 0.304000  Time: 6.877s
Train Epoch: 12 [448/640 (70%)] Loss: 0.336000  Time: 7.158s
Train Epoch: 12 [512/640 (80%)] Loss: 0.322000  Time: 6.925s
Train Epoch: 12 [576/640 (90%)] Loss: 0.405000  Time: 6.819s

Test set: Accuracy: 494.0/640 (77%)

Train Epoch: 13 [0/640 (0%)]    Loss: 0.314000  Time: 6.896s
Train Epoch: 13 [64/640 (10%)]  Loss: 0.237000  Time: 6.916s
Train Epoch: 13 [128/640 (20%)] Loss: 0.349000  Time: 7.229s
Train Epoch: 13 [192/640 (30%)] Loss: 0.258000  Time: 7.992s
Train Epoch: 13 [256/640 (40%)] Loss: 0.261000  Time: 6.925s
Train Epoch: 13 [320/640 (50%)] Loss: 0.254000  Time: 6.963s
Train Epoch: 13 [384/640 (60%)] Loss: 0.287000  Time: 7.002s
Train Epoch: 13 [448/640 (70%)] Loss: 0.322000  Time: 6.905s
Train Epoch: 13 [512/640 (80%)] Loss: 0.305000  Time: 6.927s
Train Epoch: 13 [576/640 (90%)] Loss: 0.388000  Time: 6.949s

Test set: Accuracy: 496.0/640 (78%)

Train Epoch: 14 [0/640 (0%)]    Loss: 0.300000  Time: 6.901s
Train Epoch: 14 [64/640 (10%)]  Loss: 0.220000  Time: 6.950s
Train Epoch: 14 [128/640 (20%)] Loss: 0.331000  Time: 6.993s
Train Epoch: 14 [192/640 (30%)] Loss: 0.247000  Time: 7.179s
Train Epoch: 14 [256/640 (40%)] Loss: 0.247000  Time: 7.014s
Train Epoch: 14 [320/640 (50%)] Loss: 0.239000  Time: 7.028s
Train Epoch: 14 [384/640 (60%)] Loss: 0.272000  Time: 6.929s
Train Epoch: 14 [448/640 (70%)] Loss: 0.309000  Time: 6.988s
Train Epoch: 14 [512/640 (80%)] Loss: 0.291000  Time: 7.034s
Train Epoch: 14 [576/640 (90%)] Loss: 0.377000  Time: 7.001s

Test set: Accuracy: 504.0/640 (79%)

Train Epoch: 15 [0/640 (0%)]    Loss: 0.291000  Time: 7.039s
Train Epoch: 15 [64/640 (10%)]  Loss: 0.208000  Time: 6.992s
Train Epoch: 15 [128/640 (20%)] Loss: 0.314000  Time: 7.036s
Train Epoch: 15 [192/640 (30%)] Loss: 0.236000  Time: 7.072s
Train Epoch: 15 [256/640 (40%)] Loss: 0.235000  Time: 7.077s
Train Epoch: 15 [320/640 (50%)] Loss: 0.226000  Time: 7.032s
Train Epoch: 15 [384/640 (60%)] Loss: 0.258000  Time: 7.343s
Train Epoch: 15 [448/640 (70%)] Loss: 0.293000  Time: 7.019s
```

```
Train Epoch: 15 [512/640 (80%)] Loss: 0.278000  Time: 7.064s
Train Epoch: 15 [576/640 (90%)] Loss: 0.364000  Time: 7.069s

Test set: Accuracy: 508.0/640 (79%)

Train Epoch: 16 [0/640 (0%)]    Loss: 0.278000  Time: 7.092s
Train Epoch: 16 [64/640 (10%)]  Loss: 0.200000  Time: 7.145s
Train Epoch: 16 [128/640 (20%)] Loss: 0.300000  Time: 7.013s
Train Epoch: 16 [192/640 (30%)] Loss: 0.226000  Time: 7.920s
Train Epoch: 16 [256/640 (40%)] Loss: 0.227000  Time: 7.374s
Train Epoch: 16 [320/640 (50%)] Loss: 0.216000  Time: 7.303s
Train Epoch: 16 [384/640 (60%)] Loss: 0.247000  Time: 7.307s
Train Epoch: 16 [448/640 (70%)] Loss: 0.282000  Time: 7.375s
Train Epoch: 16 [512/640 (80%)] Loss: 0.265000  Time: 7.386s
Train Epoch: 16 [576/640 (90%)] Loss: 0.348000  Time: 7.394s

Test set: Accuracy: 511.0/640 (80%)

Train Epoch: 17 [0/640 (0%)]    Loss: 0.262000  Time: 7.460s
Train Epoch: 17 [64/640 (10%)]  Loss: 0.189000  Time: 7.381s
Train Epoch: 17 [128/640 (20%)] Loss: 0.291000  Time: 7.614s
Train Epoch: 17 [192/640 (30%)] Loss: 0.215000  Time: 7.529s
Train Epoch: 17 [256/640 (40%)] Loss: 0.216000  Time: 7.486s
Train Epoch: 17 [320/640 (50%)] Loss: 0.204000  Time: 7.482s
Train Epoch: 17 [384/640 (60%)] Loss: 0.235000  Time: 7.442s
Train Epoch: 17 [448/640 (70%)] Loss: 0.270000  Time: 7.478s
Train Epoch: 17 [512/640 (80%)] Loss: 0.252000  Time: 7.449s
Train Epoch: 17 [576/640 (90%)] Loss: 0.335000  Time: 7.495s

Test set: Accuracy: 511.0/640 (80%)

Train Epoch: 18 [0/640 (0%)]    Loss: 0.253000  Time: 7.538s
Train Epoch: 18 [64/640 (10%)]  Loss: 0.181000  Time: 7.501s
Train Epoch: 18 [128/640 (20%)] Loss: 0.276000  Time: 7.586s
Train Epoch: 18 [192/640 (30%)] Loss: 0.207000  Time: 7.567s
Train Epoch: 18 [256/640 (40%)] Loss: 0.206000  Time: 7.521s
Train Epoch: 18 [320/640 (50%)] Loss: 0.194000  Time: 7.555s
Train Epoch: 18 [384/640 (60%)] Loss: 0.223000  Time: 7.568s
Train Epoch: 18 [448/640 (70%)] Loss: 0.260000  Time: 7.519s
Train Epoch: 18 [512/640 (80%)] Loss: 0.242000  Time: 7.588s
Train Epoch: 18 [576/640 (90%)] Loss: 0.324000  Time: 7.988s

Test set: Accuracy: 516.0/640 (81%)

Train Epoch: 19 [0/640 (0%)]    Loss: 0.243000  Time: 7.536s
Train Epoch: 19 [64/640 (10%)]  Loss: 0.172000  Time: 7.566s
Train Epoch: 19 [128/640 (20%)] Loss: 0.266000  Time: 7.658s
Train Epoch: 19 [192/640 (30%)] Loss: 0.200000  Time: 7.570s
```

```
Train Epoch: 19 [256/640 (40%)] Loss: 0.197000  Time: 8.042s
Train Epoch: 19 [320/640 (50%)] Loss: 0.185000  Time: 7.618s
Train Epoch: 19 [384/640 (60%)] Loss: 0.216000  Time: 7.622s
Train Epoch: 19 [448/640 (70%)] Loss: 0.251000  Time: 7.597s
Train Epoch: 19 [512/640 (80%)] Loss: 0.231000  Time: 7.635s
Train Epoch: 19 [576/640 (90%)] Loss: 0.313000  Time: 7.671s

Test set: Accuracy: 519.0/640 (81%)

Train Epoch: 20 [0/640 (0%)]    Loss: 0.235000  Time: 7.509s
Train Epoch: 20 [64/640 (10%)]  Loss: 0.167000  Time: 7.688s
Train Epoch: 20 [128/640 (20%)] Loss: 0.255000  Time: 7.711s
Train Epoch: 20 [192/640 (30%)] Loss: 0.192000  Time: 7.670s
Train Epoch: 20 [256/640 (40%)] Loss: 0.191000  Time: 7.662s
Train Epoch: 20 [320/640 (50%)] Loss: 0.178000  Time: 7.614s
Train Epoch: 20 [384/640 (60%)] Loss: 0.209000  Time: 7.640s
Train Epoch: 20 [448/640 (70%)] Loss: 0.241000  Time: 7.775s
Train Epoch: 20 [512/640 (80%)] Loss: 0.223000  Time: 7.693s
Train Epoch: 20 [576/640 (90%)] Loss: 0.303000  Time: 8.037s

Test set: Accuracy: 515.0/640 (80%)
```