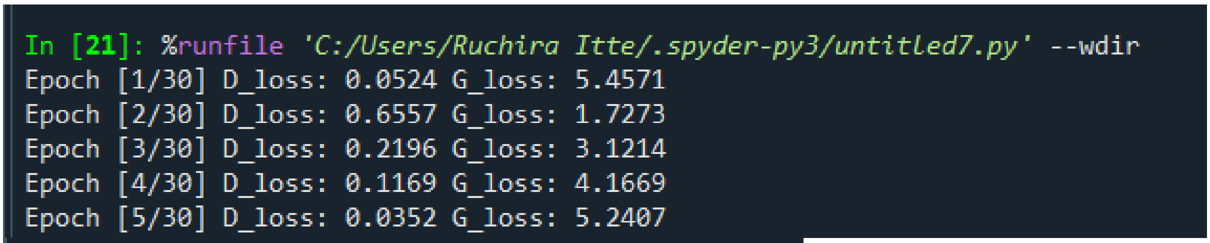
Practical-5

Aim: **Train a generative adversarial network (GAN) for generating realistic images.**

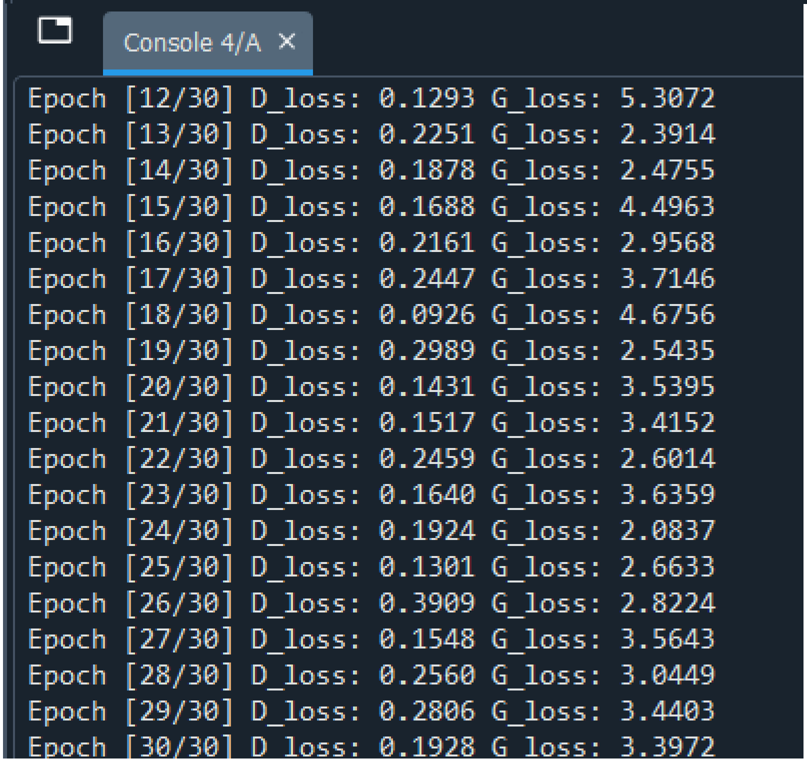
Code:

```
import torch
import torch.nn as nn
import torch.optim as optim
import torchvision
from torchvision import datasets, transforms
import matplotlib.pyplot as plt

# Device setup
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# Data (MNIST: 28x28 grayscale)
transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.5,), (0.5,))
])
train_dataset = datasets.MNIST(root="./data", train=True, download=True, transform=transform)
dataloader = torch.utils.data.DataLoader(train_dataset, batch_size=64, shuffle=True)

# Hyperparameters
latent_dim = 64
lr = 0.0002
epochs = 30

# Generator
class Generator(nn.Module):
    def __init__(self, latent_dim):
        super(Generator, self).__init__()
        self.model = nn.Sequential(
            nn.Linear(latent_dim, 128),
            nn.ReLU(True),
            nn.Linear(128, 256),
            nn.ReLU(True),
            nn.Linear(256, 512),
            nn.ReLU(True),
            nn.Linear(512, 28*28),
            nn.Tanh()
        )

    def forward(self, z):
        img = self.model(z)
        return img.view(-1, 1, 28, 28)

# Discriminator
class Discriminator(nn.Module):
    def __init__(self):
        super(Discriminator, self).__init__()
        self.model = nn.Sequential(
            nn.Linear(28*28, 512),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(512, 256),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(256, 1),
            nn.Sigmoid()
        )
    def forward(self, img):
        return self.model(img.view(-1, 28*28))

# Init models
generator = Generator(latent_dim).to(device)
discriminator = Discriminator().to(device)

# Loss & Optimizers
criterion = nn.BCELoss()
optimizer_G = optim.Adam(generator.parameters(), lr=lr)
optimizer_D = optim.Adam(discriminator.parameters(), lr=lr)

# Training
for epoch in range(epochs):
    for i, (imgs, _) in enumerate(dataloader):
        real = imgs.to(device)
        batch_size = real.size(0)

        valid = torch.ones(batch_size, 1, device=device)
        fake = torch.zeros(batch_size, 1, device=device)

        # Train Discriminator
        z = torch.randn(batch_size, latent_dim, device=device)
        gen_imgs = generator(z)
        real_loss = criterion(discriminator(real), valid)
        fake_loss = criterion(discriminator(gen_imgs.detach()), fake)
        d_loss = (real_loss + fake_loss) / 2

        optimizer_D.zero_grad()
        d_loss.backward()
        optimizer_D.step()

        # Train Generator
        z = torch.randn(batch_size, latent_dim, device=device)
        gen_imgs = generator(z)
        g_loss = criterion(discriminator(gen_imgs), valid)

        optimizer_G.zero_grad()
        g_loss.backward()
        optimizer_G.step()

    print(f"Epoch [{epoch+1}/{epochs}] D_loss: {d_loss.item():.4f} G_loss: {g_loss.item():.4f}")

# Generate sample images
z = torch.randn(16, latent_dim, device=device)
gen_imgs = generator(z).cpu().detach()

grid = torchvision.utils.make_grid(gen_imgs, nrow=4, normalize=True)
plt.imshow(grid.permute(1, 2, 0))
plt.axis("off")
plt.show()
```

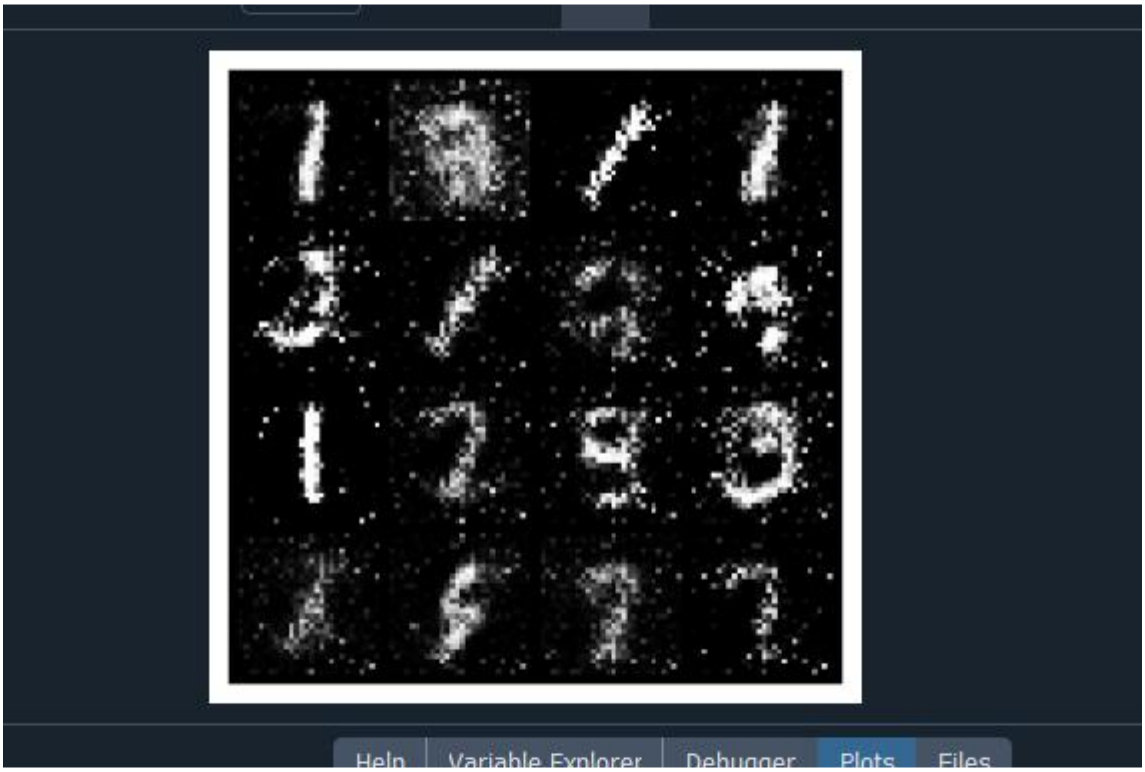
Output:



```
In [21]: %runfile 'C:/Users/Ruchira Itte/.spyder-py3/untitled7.py' --wdir
Epoch [1/30] D_loss: 0.0524 G_loss: 5.4571
Epoch [2/30] D_loss: 0.6557 G_loss: 1.7273
Epoch [3/30] D_loss: 0.2196 G_loss: 3.1214
Epoch [4/30] D_loss: 0.1169 G_loss: 4.1669
Epoch [5/30] D_loss: 0.0352 G_loss: 5.2407
```



```
Epoch [12/30] D_loss: 0.1293 G_loss: 5.3072
Epoch [13/30] D_loss: 0.2251 G_loss: 2.3914
Epoch [14/30] D_loss: 0.1878 G_loss: 2.4755
Epoch [15/30] D_loss: 0.1688 G_loss: 4.4963
Epoch [16/30] D_loss: 0.2161 G_loss: 2.9568
Epoch [17/30] D_loss: 0.2447 G_loss: 3.7146
Epoch [18/30] D_loss: 0.0926 G_loss: 4.6756
Epoch [19/30] D_loss: 0.2989 G_loss: 2.5435
Epoch [20/30] D_loss: 0.1431 G_loss: 3.5395
Epoch [21/30] D_loss: 0.1517 G_loss: 3.4152
Epoch [22/30] D_loss: 0.2459 G_loss: 2.6014
Epoch [23/30] D_loss: 0.1640 G_loss: 3.6359
Epoch [24/30] D_loss: 0.1924 G_loss: 2.0837
Epoch [25/30] D_loss: 0.1301 G_loss: 2.6633
Epoch [26/30] D_loss: 0.3909 G_loss: 2.8224
Epoch [27/30] D_loss: 0.1548 G_loss: 3.5643
Epoch [28/30] D_loss: 0.2560 G_loss: 3.0449
Epoch [29/30] D_loss: 0.2806 G_loss: 3.4403
Epoch [30/30] D_loss: 0.1928 G_loss: 3.3972
```


Help
Variable Explorer
Debugger
Plots
Files