Close

Amardeep Singh

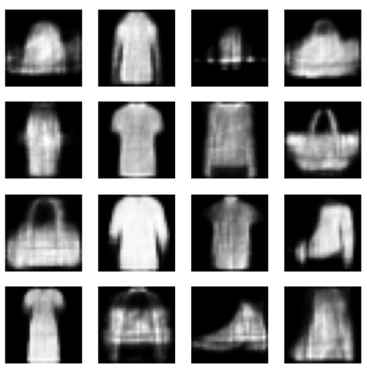
Amardeep Singh

```
# ================= IMPORTS =================
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms
from torch.utils.data import DataLoader
import matplotlib.pyplot as plt

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

# ================= DATASET (FASHION-MNIST) =================
batch_size = 128

transform = transforms.Compose([
    transforms.ToTensor()  # values in [0,1]
])

train_dataset = datasets.FashionMNIST(
    root="./data", train=True, transform=transform, download=True
)
test_dataset = datasets.FashionMNIST(
    root="./data", train=False, transform=transform, download=True
)

train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
test_loader  = DataLoader(test_dataset,  batch_size=batch_size, shuffle=False)

# ================= VAE MODEL =================
class VAE(nn.Module):
    def __init__(self, latent_dim=20):
        super().__init__()
        # Encoder
        self.fc1 = nn.Linear(28*28, 400)
        self.fc_mu = nn.Linear(400, latent_dim)
        self.fc_logvar = nn.Linear(400, latent_dim)
        # Decoder
        self.fc2 = nn.Linear(latent_dim, 400)
        self.fc3 = nn.Linear(400, 28*28)

    def encode(self, x):
        h = torch.relu(self.fc1(x))
        return self.fc_mu(h), self.fc_logvar(h)

    def reparameterize(self, mu, logvar):
        std = torch.exp(0.5 * logvar)
        eps = torch.randn_like(std)
        return mu + eps * std

    def decode(self, z):
        h = torch.relu(self.fc2(z))
        return torch.sigmoid(self.fc3(h))

    def forward(self, x):
        mu, logvar = self.encode(x)
        z = self.reparameterize(mu, logvar)
        recon = self.decode(z)
        return recon, mu, logvar

# ================= LOSS FUNCTION =================
def vae_loss(recon_x, x, mu, logvar):
    # Reconstruction (BCE) + KL Divergence
    recon_loss = nn.functional.binary_cross_entropy(recon_x, x, reduction='sum')
    kl_loss = -0.5 * torch.sum(1 + logvar - mu.pow(2) - logvar.exp())
    return recon_loss + kl_loss, recon_loss, kl_loss

# ================= INITIALIZE =================
latent_dim = 20
model = VAE(latent_dim).to(device)
optimizer = optim.Adam(model.parameters(), lr=1e-3)

# ================= TRAINING =================
epochs = 30
train_losses, val_losses = [], []

for epoch in range(1, epochs + 1):
    model.train()
    train_loss = 0.0

    for x, _ in train_loader:
        x = x.view(-1, 28*28).to(device)
        recon, mu, logvar = model(x)
        loss, _, _ = vae_loss(recon, x, mu, logvar)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        train_loss += loss.item()

    train_loss /= len(train_loader.dataset)
    train_losses.append(train_loss)

    # Validation
    model.eval()
    val_loss = 0.0
    with torch.no_grad():
        for x, _ in test_loader:
            x = x.view(-1, 28*28).to(device)
            recon, mu, logvar = model(x)
            loss, _, _ = vae_loss(recon, x, mu, logvar)
            val_loss += loss.item()

    val_loss /= len(test_loader.dataset)
    val_losses.append(val_loss)

    print(f"Epoch {epoch}/{epochs} | Train Loss: {train_loss:.2f} | Val Loss: {val_loss:.2f}")

# ================= RECONSTRUCTION =================
model.eval()
with torch.no_grad():
    x, _ = next(iter(test_loader))
    x = x.view(-1, 28*28).to(device)
    recon, _, _ = model(x)

x = x.view(-1, 1, 28, 28).cpu()
recon = recon.view(-1, 1, 28, 28).cpu()

plt.figure(figsize=(8,4))
for i in range(10):
    plt.subplot(2,10,i+1)
    plt.imshow(x[i][0], cmap='gray')
    plt.axis('off')
    plt.subplot(2,10,i+11)
    plt.imshow(recon[i][0], cmap='gray')
    plt.axis('off')
plt.suptitle("Fashion-MNIST | Top: Original | Bottom: Reconstructed")
plt.show()

# ================= GENERATE NEW SAMPLES =================
with torch.no_grad():
    z = torch.randn(16, latent_dim).to(device)
    samples = model.decode(z).view(-1, 1, 28, 28).cpu()

plt.figure(figsize=(4,4))
for i in range(16):
    plt.subplot(4,4,i+1)
    plt.imshow(samples[i][0], cmap='gray')
    plt.axis('off')
plt.suptitle("Generated Samples (Fashion-MNIST)")
plt.show()

# ================= LOSS CURVES =================
plt.figure(figsize=(6,4))
plt.plot(train_losses, label="Train Loss")
plt.plot(val_losses, label="Validation Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.title("VAE Loss Curves (Fashion-MNIST, 30 Epochs)")
plt.legend()
plt.grid(True)
plt.show()
```

```
100%|██████████| 26.4M/26.4M [00:02<00:00, 11.1MB/s]
100%|██████████| 29.5k/29.5k [00:00<00:00, 190kB/s]
100%|██████████| 4.42M/4.42M [00:01<00:00, 3.58MB/s]
100%|██████████| 5.15k/5.15k [00:00<00:00, 13.9MB/s]
Epoch 1/30 | Train Loss: 284.58 | Val Loss: 262.04
Epoch 2/30 | Train Loss: 256.01 | Val Loss: 254.22
Epoch 3/30 | Train Loss: 250.36 | Val Loss: 250.54
Epoch 4/30 | Train Loss: 247.64 | Val Loss: 248.12
Epoch 5/30 | Train Loss: 245.90 | Val Loss: 246.90
Epoch 6/30 | Train Loss: 244.91 | Val Loss: 246.66
Epoch 7/30 | Train Loss: 244.15 | Val Loss: 245.38
Epoch 8/30 | Train Loss: 243.54 | Val Loss: 244.96
Epoch 9/30 | Train Loss: 243.00 | Val Loss: 245.11
Epoch 10/30 | Train Loss: 242.57 | Val Loss: 244.14
Epoch 11/30 | Train Loss: 242.19 | Val Loss: 243.71
Epoch 12/30 | Train Loss: 241.86 | Val Loss: 243.54
Epoch 13/30 | Train Loss: 241.61 | Val Loss: 243.32
Epoch 14/30 | Train Loss: 241.33 | Val Loss: 243.12
Epoch 15/30 | Train Loss: 241.09 | Val Loss: 243.06
Epoch 16/30 | Train Loss: 240.96 | Val Loss: 242.63
Epoch 17/30 | Train Loss: 240.77 | Val Loss: 242.71
Epoch 18/30 | Train Loss: 240.65 | Val Loss: 242.42
Epoch 19/30 | Train Loss: 240.45 | Val Loss: 242.24
Epoch 20/30 | Train Loss: 240.31 | Val Loss: 242.23
Epoch 21/30 | Train Loss: 240.24 | Val Loss: 242.12
Epoch 22/30 | Train Loss: 240.12 | Val Loss: 241.99
Epoch 23/30 | Train Loss: 240.02 | Val Loss: 241.76
Epoch 24/30 | Train Loss: 239.89 | Val Loss: 242.02
Epoch 25/30 | Train Loss: 239.81 | Val Loss: 241.96
Epoch 26/30 | Train Loss: 239.70 | Val Loss: 242.54
Epoch 27/30 | Train Loss: 239.65 | Val Loss: 241.59
Epoch 28/30 | Train Loss: 239.59 | Val Loss: 241.52
Epoch 29/30 | Train Loss: 239.50 | Val Loss: 241.40
Epoch 30/30 | Train Loss: 239.41 | Val Loss: 241.36
```

Fashion-MNIST | Top: Original | Bottom: Reconstructed

Generated Samples (Fashion-MNIST)



VAE Loss Curves (Fashion-MNIST, 30 Epochs)