RA2311047010040

Index

மரம் வளர்ப்போம்
மழை வளம் காப்போம்

Superior® Note Books

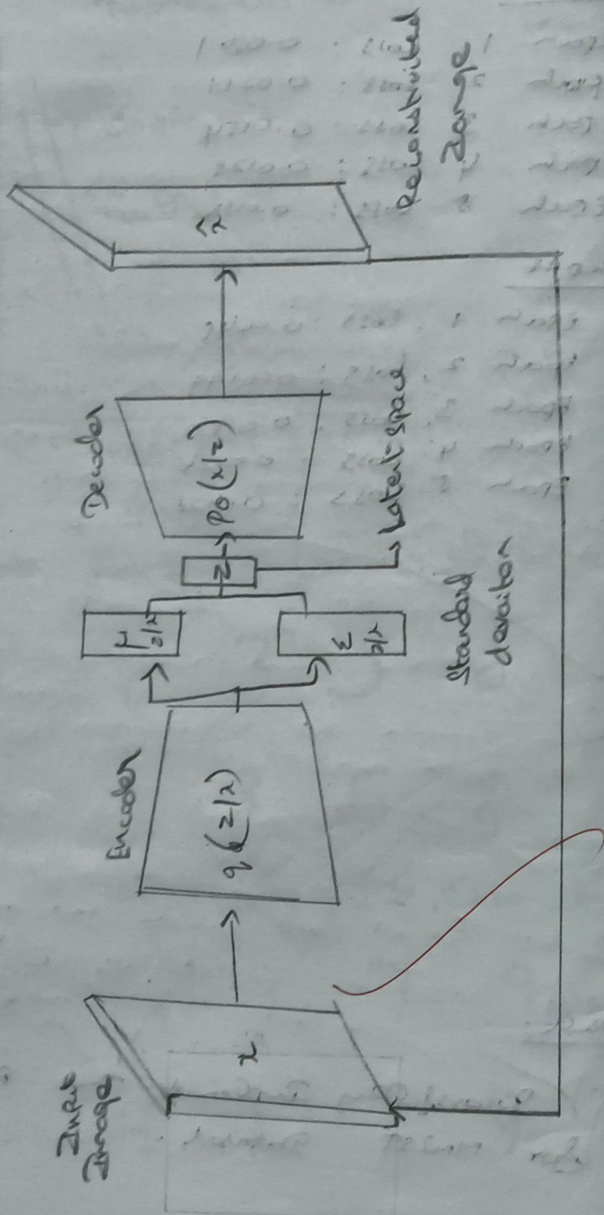
Name : Vishwa S

Subject : DL

Std. : .... Sec. : .... Roll No. ....

School : ....

No.	Date	Title	Marks	Signature
1	24/7/25	Exploring the DL platforms.		
2	31/7/25	Implement a classifier using open source dataset		
3	7/8/25	Study the classifier with respect to statistical parameters.		18/8
4	14/8/25	Build a simple feed forward neural network to Recognize Handwritten characters		22/8
5	22/8/25	Study of Activation functions & Their role.		9/9/25
6	9/9/25	Implement Gradient Descent & Backpropagation in Deep neural Network		16/9
7	16/9/25	Build a CNN model to classify cat & dog image		23/9/25
8 Ex 9 Z	30/9/25	Build a RNN Network.		30/9
9 Ex 8	9/10/25	Experiment using LSTM		9/10
10	9/10/25	AutoEncoder		
11	9/10/25	VARIATIONAL AUTOENCODER		
12	17/10/25	Deep convolutional GAN		



Architecture of VAE

9/10/25

Lab 11: Experiments using Variational Autoencoder (VAE)

Aim:

To implement a variational Autoencoder to learn probabilistic latent representations of MNIST image and generate new images from the learned latent space.

Objectives

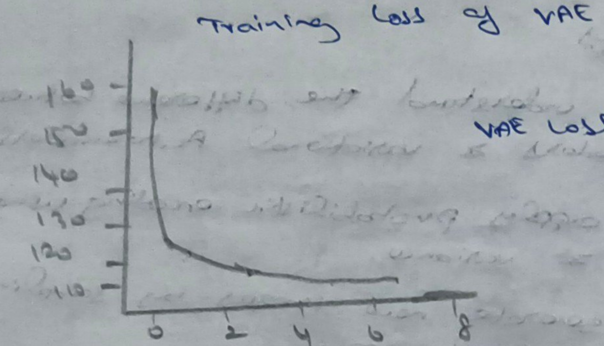
1) To understand the difference between Autoencoders & variational Autoencoders.
2) To apply probabilistic encoding using mean & variance.
3) To generate new images by sampling from the latent space.
4) To visualize the learned latent distribution & reconstructed images

Observation

- The VAE extends the Autoencoder by introducing probabilistic latent variables
- It learns a distribution rather than a fixed encoding, allowing sampling & generation of new data.
- The model is trained using a reconstruction loss and a KL divergence loss to balance accuracy & latent regularization
- The generated samples resemble hand written digits, showing that the model learned the data distribution.

Output.

Epoch 1: Loss : 164.378
Epoch 2 : Loss : 121.2820
Epoch 3 : Loss : 114.5211
Epoch 4 : Loss : 111.6013
Epoch 5 : Loss : 109.8342
Epoch 6 : Loss : 108.6767
Epoch 7 : Loss : 107.8209
Epoch 8 : Loss : 107.1637
Epoch 9 : Loss : 106.6916
Epoch 10 : Loss : 106.2222



Top : original Images | Bottom : VAE Reconstruction

0 2 6 4 0 9 8 5

0 2 6 4 0 9 8 5

Generated Digits from Random Latent Space

2 2 8 3 4 8 3 8

Pseudocode.

BEGIN

St-1 : Import libraries

St-2 : Load & normalize MNIST dataset

St-3 : Define encoder

- Input : 784-dimensional image
- Dense hidden layers
- Output : mean (μ) and log-variance (σ^2)

St-4 : Sample latent vector $z = \mu + \sigma * \varepsilon$

St-5 : Define decoder

- Input : z
- Dense layers to reconstruct 784-dimensional output

St-6 : Define loss function

- Reconstruction loss + KL divergence

St-7 : Train VAE using training data

St-8 : Generate new images by sampling from latent space

St-9 : Visualize reconstructed and generated images

END.

Result:

Successfully Implemented the VAE for MNIST dataset.

dltlab11.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Share V

Connect T4

[ ]
```
!pip install torch torchvision matplotlib
```

```
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.8.0+cu126)
Requirement already satisfied: torchvision in /usr/local/lib/python3.12/dist-packages (0.23.0+cu126)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch) (3.20.0)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from torch) (3.5)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch) (3.1.6)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch) (9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch) (11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch) (10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch) (11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch) (12.5.4.2)
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in /usr/local/lib/python3.12/dist-packages (from torch) (2.27.3)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.85)
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch) (1.11.1.6)
Requirement already satisfied: triton==3.4.0 in /usr/local/lib/python3.12/dist-packages (from torch) (3.4.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from torchvision) (2.0.2)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.12/dist-packages (from torchvision) (11.3.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (25.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.2.5)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->torch) (3.0.3)
```

[ ]
```python
# vae_mnist.py
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms
from torch.utils.data import DataLoader
```

CO dltlab11.ipynb

File Edit View Insert Runtime Tools Help

Share V

Commands + Code + Text ▷ Run all Connect

```
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")

# ---- Load MNIST dataset
tf = transforms.Compose([transforms.ToTensor()])
train_data = datasets.MNIST(root='data', train=True, transform=tf, download=True)
train_loader = DataLoader(train_data, batch_size=128, shuffle=True)

# ---- Variational Autoencoder
class VAE(nn.Module):
    def __init__(self):
        super(VAE, self).__init__()
        self.fc1 = nn.Linear(784, 400)
        self.fc_mu = nn.Linear(400, 20)      # Mean of latent vector
        self.fc_logvar = nn.Linear(400, 20) # Log variance of latent vector
        self.fc2 = nn.Linear(20, 400)
        self.fc3 = nn.Linear(400, 784)
        self.relu = nn.ReLU()
        self.sigmoid = nn.Sigmoid()

    def encode(self, x):
        h1 = self.relu(self.fc1(x))
        mu = self.fc_mu(h1)
        logvar = self.fc_logvar(h1)
        return mu, logvar

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

    def decode(self, z):
        h2 = self.relu(self.fc2(z))
        return self.sigmoid(self.fc3(h2))

    def forward(self, x):
        x = x.view(-1, 784)
        mu, logvar = self.encode(x)
```

CO dltlab11.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Connect T4

Share V

[ ]

```
# ---- Loss Function (Reconstruction + KL Divergence)
def vae_loss(recon_x, x, mu, logvar):
    BCE = nn.functional.binary_cross_entropy(recon_x, x.view(-1, 784), reduction='sum')
    # KL Divergence: how latent distribution differs from standard normal
    KLD = -0.5 * torch.sum(1 + logvar - mu.pow(2) - logvar.exp())
    return BCE + KLD

# ---- Model, Optimizer
model = VAE().to(device)
optimizer = optim.Adam(model.parameters(), lr=1e-3)

# ---- Training
epochs = 10
train_losses = []

for epoch in range(1, epochs + 1):
    model.train()
    total_loss = 0
    for imgs, _ in train_loader:
        imgs = imgs.to(device)
        optimizer.zero_grad()
        recon, mu, logvar = model(imgs)
        loss = vae_loss(recon, imgs, mu, logvar)
        loss.backward()
        total_loss += loss.item()
        optimizer.step()

    avg_loss = total_loss / len(train_loader.dataset)
    train_losses.append(avg_loss)
    print(f"Epoch [{epoch}/{epochs}] Loss: {avg_loss:.4f}")

# ---- Plot Training Loss
plt.plot(train_losses, label="VAE Loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.title("Training Loss of Variational Autoencoder")
plt.legend()
plt.show()

# ---- Reconstruction Visualization
model.eval()
imgs, _ = next(iter(train_loader))
imgs = imgs[:8].to(device)
with torch.no_grad():
    recons, _, _ = model(imgs)
```

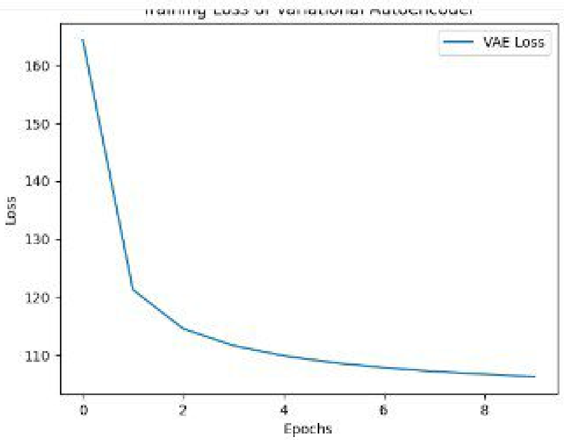
Variables Terminal

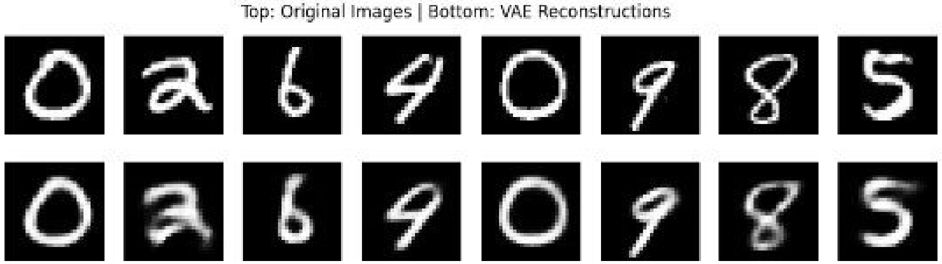
dltlab11.ipynb

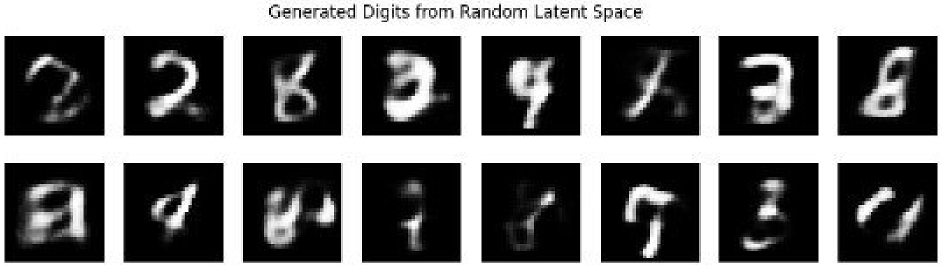
File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```
plt.show()
```







Variables Terminal