ASSIGNMENT
REPORT
CS F437: Generative Artificial Intelligence
ASSIGNMENT-2

By

Student ID	**Name**
2021AAPS3025H	ANSH KANOTRA
2021A7PS2951H	Jatin Jindal
2020A8PS2199H	Manav Sahni

Assignment Set Number: 2

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

PART A - IMAGE GENERATION

Variational Autoencoders

Definition

Variational autoencoders (VAEs) are neural networks that learn to encode input data into a compact representation, called a latent space. Unlike traditional autoencoders, VAEs introduce probabilistic elements, enabling them to generate diverse outputs. They find applications in data compression, generation, and anomaly detection in machine learning.

Following are steps to train it:

1. **Data Preparation: -** Load, normalize, and reshape MNIST images.
2. **Encoder Network: -** Design neural network with probabilistic layers for input-to-latent mapping.
3. **Decoder Network: -** Create a mirrored network for latent-to-output mapping.
4. **Loss Function: -** Define loss balancing reconstruction error and latent space divergence.
5. **Training: -** Train VAE on normalized dataset, optimize parameters to minimize defined loss.
6. **Sampling from Latent Space: -** Generate random samples from learned latent space.
7. **Decoding Test Examples: -** Encode MNIST test examples with trained encoders.
8. **Reconstruction: -** Decode latent representations to reconstruct images.
9. **Evaluation: -** Assess reconstruction quality on the test set.

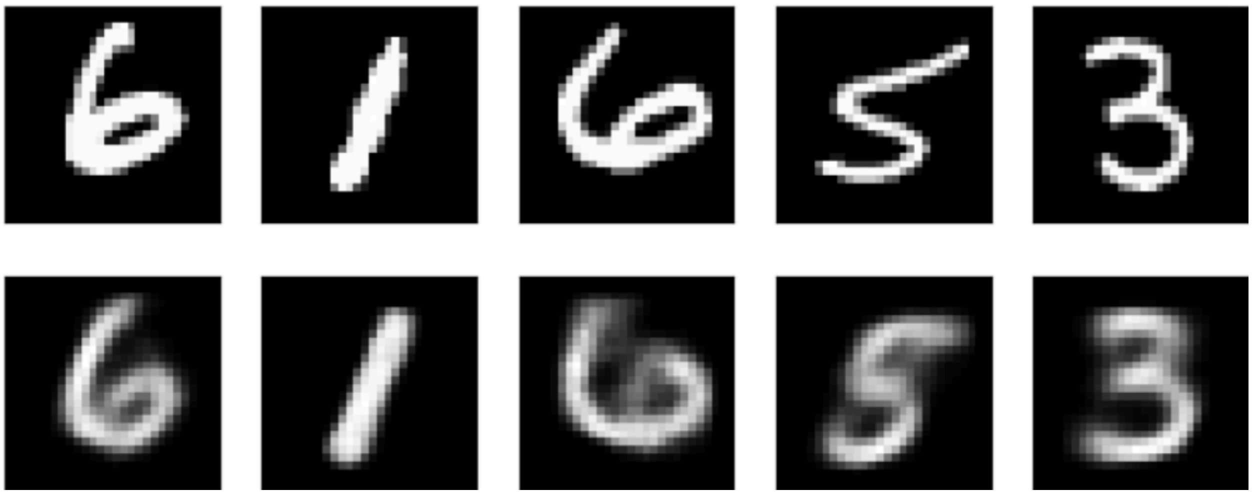
Mean Squared errors

Following are the errors obtained after training same model (with principal component varying) in 10 epochs:

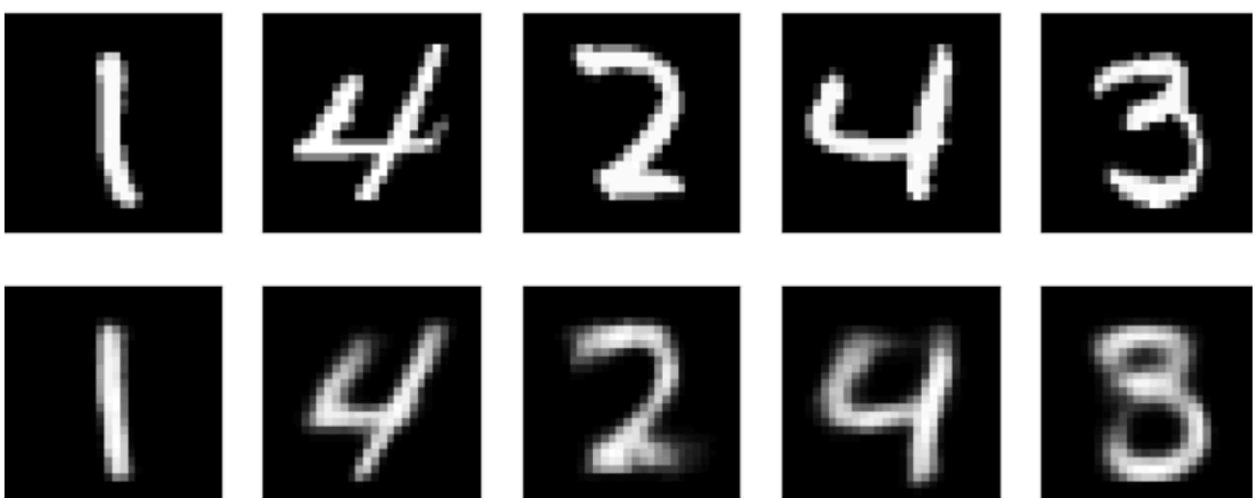
1. Performing Probabilistic PCA reconstruction with 2 principal components... Train MSE: 0.0397
2. Performing Probabilistic PCA reconstruction with 4 principal components... Train MSE: 0.0272
3. Performing Probabilistic PCA reconstruction with 8 principal components... Train MSE: 0.0139
4. Performing Probabilistic PCA reconstruction with 16 principal components... Train MSE: 0.0167
5. Performing Probabilistic PCA reconstruction with 32 principal components... Train MSE: 0.0086
6. Performing Probabilistic PCA reconstruction with 64 principal components... Train MSE: 0.0089

Image Reconstruction examples

Original vs Reconstructed (latent_space = 2)



Original vs Reconstructed (latent_space = 4)



Original vs Reconstructed (latent_space = 8)

Original vs Reconstructed (latent_space = 16)

Original vs Reconstructed (latent_space = 32)

Original vs Reconstructed (latent_space = 64)

GENERATIVE ADVERSERIAL NETWORK

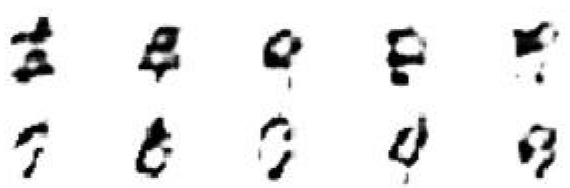
GANs function on the fundamenta concept of the rivalry between the discriminator and generator. The discriminator is continuously working to improve its ability to discern between genuine and fake data, while the generator is continuously striving to create more realistic data to trick the discriminator. Both networks get better over time as a result of this rivalry, and eventually the generator produces extremely realistic data that is identical to genuine data.

1. We first loaded our dataset
2. Created a model for the generator and the discriminator
3. Combined both the models to form a GAN
4. Generated images through the Generator and used the discriminator to classify them
5. Performed training using our dataset to improve the generator.

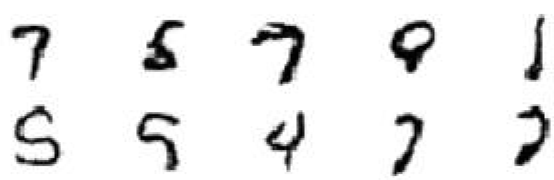
OUTPUTS

Latent Space Vector-

2

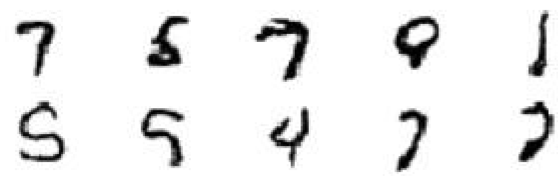


16



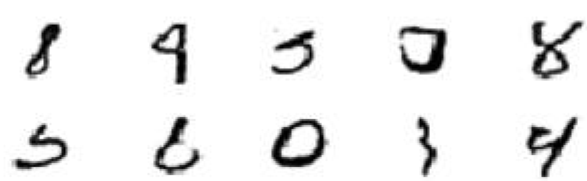
4

32



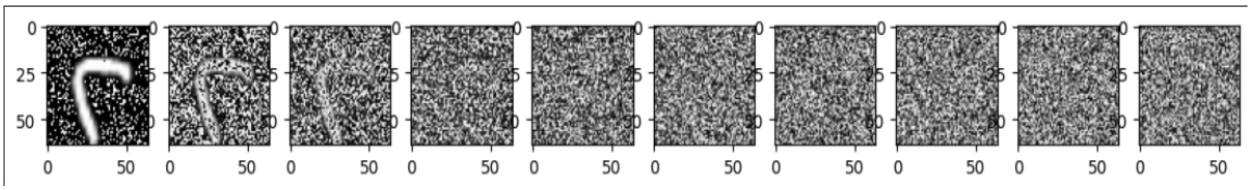
8

64

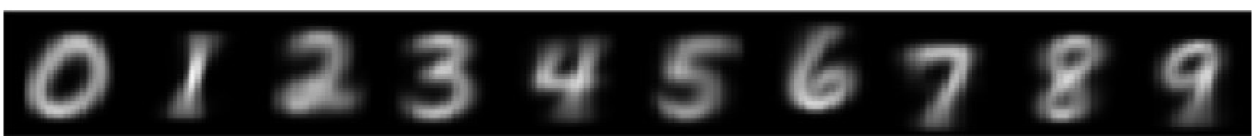


DIFFUSION MODEL

Diffusion models are probabilistic generative models that learn a probability distribution over data by iteratively applying a diffusion process. This process gradually spreads the data distribution until it reaches a stationary distribution. By modeling the diffusion process, these models can generate high-quality samples and perform tasks like image denoising, inpainting, and super-resolution. Diffusion models offer a powerful framework for probabilistic generative modeling and find applications in various domains such as image synthesis, natural language processing, and signal processing in machine learning.



Output



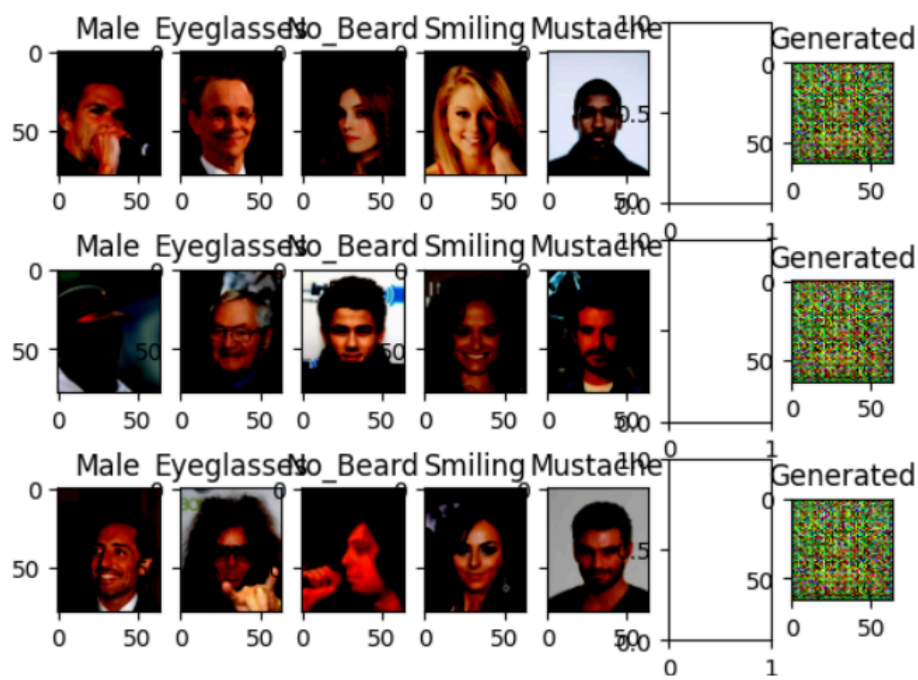
DCGAN - DEEP CONVOLUTIONAL GAN

Alec Radford, Luke Metz, and Soumith Chintala proposed this particular GAN architecture in 2015. Convolutional neural networks (CNNs) are used by DCGANs for both the discriminator and generator networks. DCGANs have been extensively employed to produce artificial pictures of superior quality. They have been used in many different fields, including picture-to-image translation, art creation, realistic face generation, and scene generation. Numerous generative model developments in the past have been based on their architecture and training protocols.

TRAINING

Epoch [1/5]: Gen Loss 2.0022, Enc Loss 0.1449, Disc Loss 0.1956
Epoch [2/5]: Gen Loss 2.0274, Enc Loss 0.1387, Disc Loss 0.2185
Epoch [3/5]: Gen Loss 2.0550, Enc Loss 0.1444, Disc Loss 0.2019
Epoch [4/5]: Gen Loss 2.0622, Enc Loss 0.1331, Disc Loss 0.1941
Epoch [5/5]: Gen Loss 2.0562, Enc Loss 0.1413, Disc Loss 0.1917

OUTPUTS



DATASET

PART B - IMAGE TO IMAGE TRANSLATION