ASSIGNMENT-2

Task-2

NAME- Akash Das (MDS202206)
WEBMAIL- akashd@cmi.ac.in

1. Introduction:

We implemented Generative Adversarial Network (GAN) applied to the MNIST digit dataset within a Python environment. The objective here was to synthesize digit images that are statistically similar to the authentic MNIST dataset.

2. Methodology:

2.1. Data Preprocessing

The dataset comprised 28x28 pixel grayscale images of 10 distinct classes, processed via:

- Rescaling to 32x32 pixels, suitable for the GAN's architecture.
- Normalizing pixel values to [-1, 1], compatible with the generator's tanh activation.

2.2. Model Architecture

i. Discriminator: The Discriminator is given pairs of (Image, Label) to classify as real or fake.

ii. Generator: Here specifically, we give the generator a random codeword from the latent space, as well as a label indicating which class of image we want, and the generator tries to create an image of this class, i.e. it was tasked with creating new images from a noise vector.

2.3. Training Process

The training loop involved:

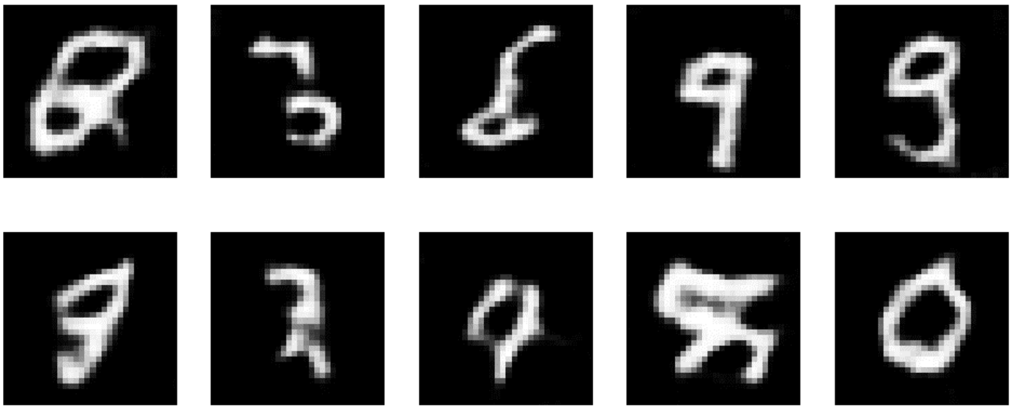
• Alternating between training the Discriminator with real and fake images, and updating the Generator based on the Discriminator's feedback.

• Applying Adam optimizer with specific learning rates and hyperparameters.

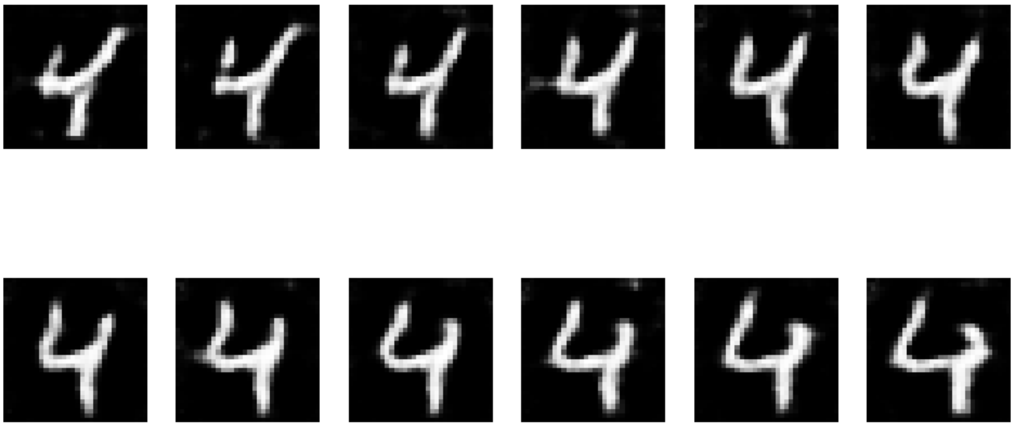
• Utilizing appropriate loss functions for GAN training (Binary Cross-Entropy).

3. Output:

i. Examples of 10 images generated from random codewords by the model –



ii. 1 examples of the following: Pick two random codewords, a and b, and pick 10 uniformly spaced points between a and b on the line from a to b. Show the images generated from these 12 points



4. **Conclusion:**

Thus the implemented GAN successfully generated images mimicking the MNIST digit dataset, underscoring the effectiveness of the chosen architecture and training strategy.