

Advanced Machine Learning

Assignment 2

Roudranil Das	Saikat Bera	Shreyan Chakraborty
MDS202227	MDS202228	MDS202237
roudranil@cmi.ac.in	saikatb@cmi.ac.in	shreyanc@cmi.ac.in

Contents

1 Problem 1: Training a VAE for the FashionMnist dataset	1
1.1 Architecture	2
1.2 Training	2
1.3 Results	2
2 Problem 2: Training a GAN for MNIST dataset	3
2.1 Architecture	3
2.2 Training	3
2.3 Results	3
3 Problem 3: Training a Conditional GAN for the MNIST dataset	5
3.1 Architecture and training	5
3.2 Results	6

About the datasets

In the given datasets we have images consisting english digits and images consisting of dressing items. The images are 28×28 in resolution.

Processing the dataset

The images were loaded from the corresponding csv file, where individual images are stored as rows, the columns being the pixel values. The images were then loaded as tensors and then normalised to be either between 0 and 1 or between -1 and 1 as per model architecture.

1 Problem 1: Training a VAE for the FashionMnist dataset

Variational Autoencoders (VAEs) are a class of generative models that are adept at learning complex data distributions. They work by encoding data into a lower-dimensional latent space and then decoding this representation back to the original data space. VAEs have been successfully applied to tasks such as image generation, denoising, and representation learning.

1.1 Architecture

Our VAE model is defined by an input dimension of 784, corresponding to the flattened 28×28 FashionMNIST images. The model consists of a hidden layer with 64 neurons and a latent space with a dimensionality of 16. The encoder network maps input images to a latent space representation, which is then used to parameterize a normal distribution. The decoder network samples from this distribution to reconstruct the images.

1.2 Training

We trained our model using the Adam optimizer with a learning rate of $1e-3$, commonly referred to as the “Karpathy constant”. The model was trained for 3 epochs, with a batch size of 32. The loss function used was the mean squared error (MSE) for the reconstruction loss and the KL divergence for the latent space regularization.

1.3 Results

The VAE was evaluated based on its ability to reconstruct images and generate new images. The training process resulted in a steady decrease in loss, indicating that the model was learning to reconstruct the input images effectively. Generated images after training showed recognizable fashion items, demonstrating that the VAE could capture the distribution of the FashionMNIST dataset.

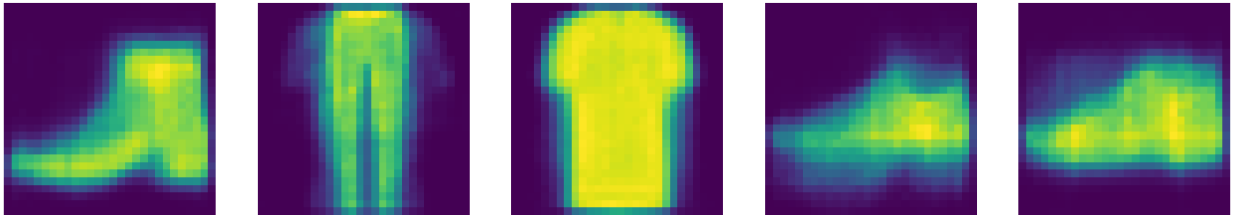


Figure 1: 5 images generated by the model from randomly sampled z in the latent space

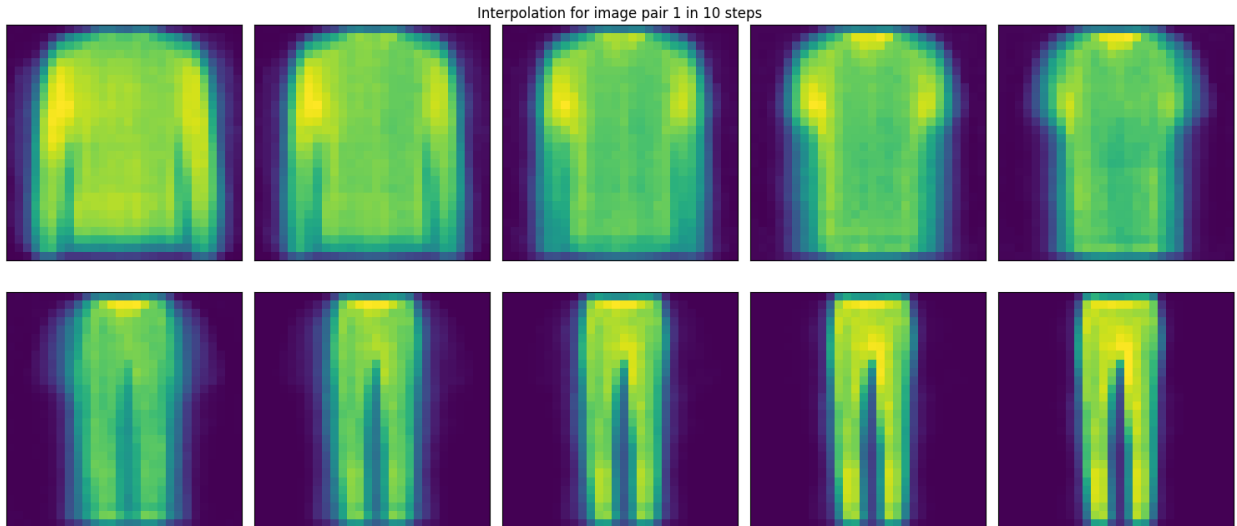


Figure 2: An example of interpolation between 2 images

2 Problem 2: Training a GAN for MNIST dataset

A Generative Adversarial Network (GAN) is a powerful and innovative deep learning framework designed to generate data, typically images or other complex structures, that closely resemble real data. GANs consist of two neural networks, a **Generator** and a **Discriminator**, engaged in a continuous adversarial game. This competition allows a GAN to generate data bearing resemblance to real data.

2.1 Architecture

The GAN consists of a **Generator** and a **Discriminator**. The **Generator** generates images from a latent space of dimension 200, where vectors are sampled with the elements of the vectors following a $\mathcal{U}(-1, 1)$ distribution. The **Generator** utilises transposed convolution operations to upscale the generated images to be of size 28×28 .

The **Discriminator** reads images (all pixel values between -1 and 1), and then outputs probability of the image being real or fake.

The architecture was closely followed with the one described in the original DCGAN paper.

2.2 Training

We trained our model with the Adam optimiser with a learning rate of $2e - 4$ and a beta of 0.5, 0.999. It was trained first for 10 epochs and then for 50 epochs.

2.3 Results

During training we kept 16 randomly generated noise vectors fixed throughout the training process, which we used to evaluate the progress of the model training besides

monitoring the loss. However it seemed that the images generated by the model kind of stabilised after a certain number of epochs.

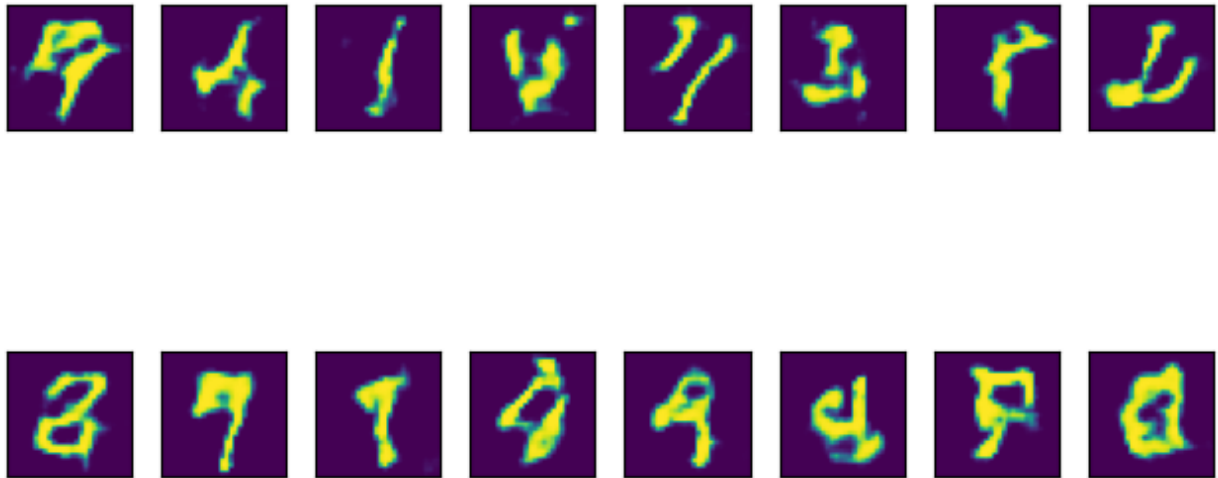


Figure 3: Results of the model on those 16 vectors on training for 16 epochs

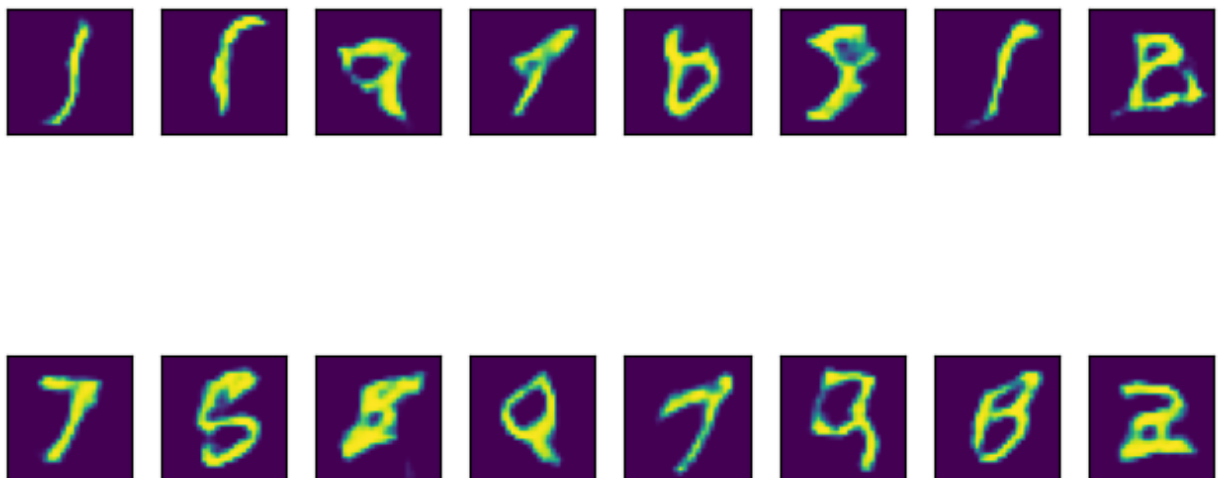


Figure 4: The images had stabilised after 22 epochs

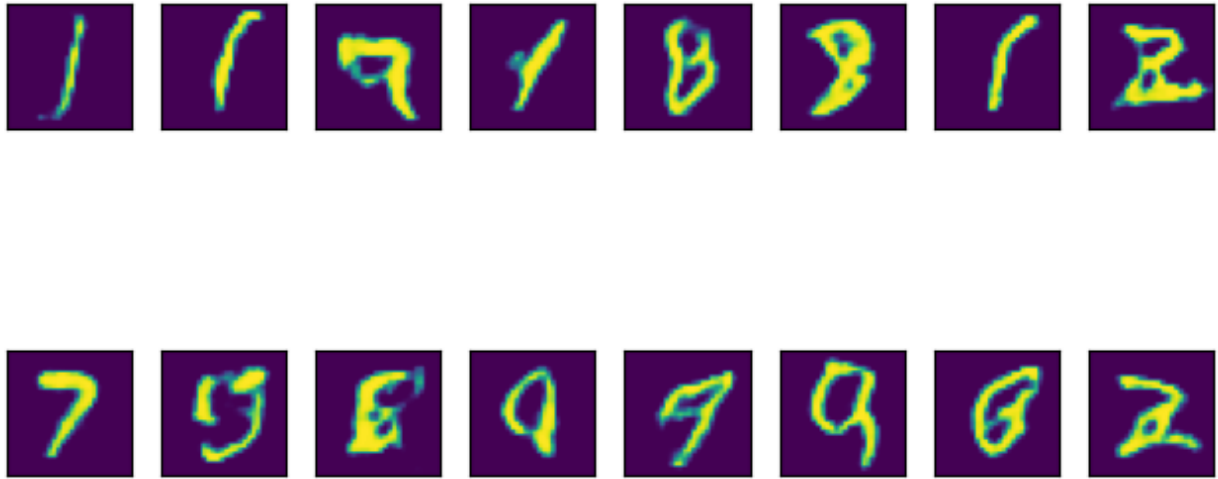


Figure 5: Images have stabilised to this after 34 epochs

3 Problem 3: Training a Conditional GAN for the MNIST dataset

GAN's are already powerful neural networks for the generation of synthetic data. Conditional GAN's take this a step forward by conditioning the generation process on additional information, such as class labels.

3.1 Architecture and training

In this model, the **Generator** and the **Discriminator** are built with linear layers as a normal GAN. The difference is in the training procedure. This time the **Discriminator** is trained on both the real data and the real labels, and for n times before the **Generator** is trained once. The **Generator** generates images from the random noise and the real label is also used. Then the **Discriminator** is trained to differentiate between the two images.

3.2 Results

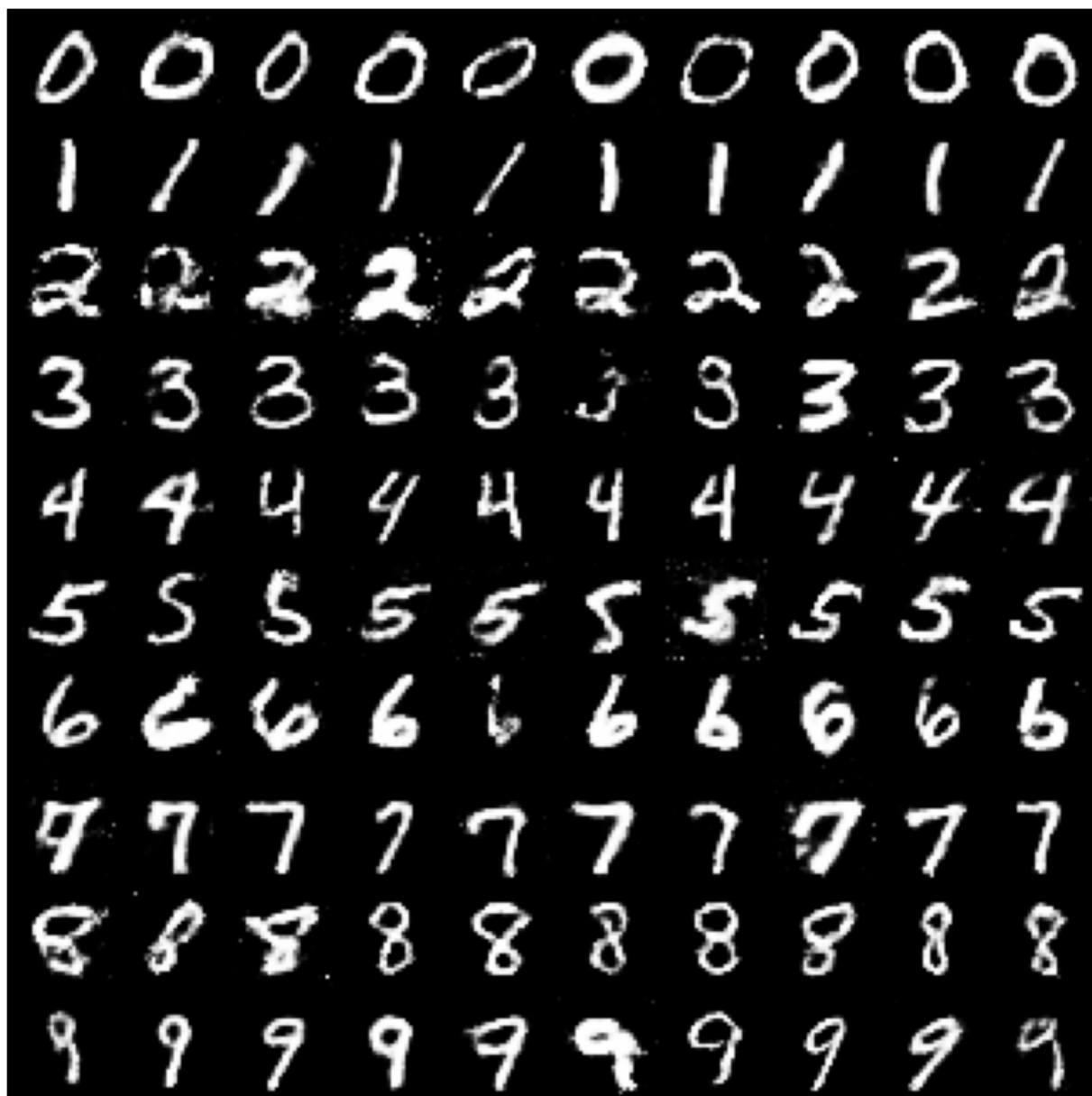


Figure 6: Images generated by the model, each row is for one provided label