CS 663 Course Project

Krishna Wadhwani - 160010031

November, 2018

Introduction

Generative adversarial networks (GANs) are a class of neural networks that are used in unsupervised machine learning. They help to solve such tasks as image generation from descriptions, getting high resolution images from low resolution ones, predicting which drug could treat a certain disease, retrieving images that contain a given pattern, etc.

To explain in brief, GANs are a type of deep neural network architecture made from two neural networks (Generator and Discriminator), putting one network against the remaining network (adversarial). The framework corresponds to a minimax two-player game where the Generator captures the data distribution and the Discriminator estimates the probability that a sample came from the training data rather than the Generator. In a nutshell, after training on a finite unlabeled dataset, a GANs can generate new data from the same kind that might not neccesarily be in the original training set.

In this project, I have implemented the paper "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets" (link), which introduces a variant of GAN called InfoGAN. InfoGAN uses the concept of mutual information from Information Theory to learn a disentangled representation of the dataset in a completely unsupervised manner.

Overview of the Problem Statement

GANs have produced remarkale results in image generation, improving on architectures such as autoencoders and variational autoencoders. In the DCGAN paper, it was shown that a lot of interesting things can be done on GANS when doing arithmetic on the noise vector that is fed as input to the generator. In this paper, the input noise vectors of men with glasses are manipulated to give vectors that result in women with sunglasses once fed into the generator. This shows that there are structures in the noise vectors that have meaningful and consistent effects on the generator output.

However, there is no systematic way to learn these structures. These noise vectors are the only parameter that can be modified to have a desired effect in the images generated and since we basically feed noise into the images, there is no intutition on how to modify it to get a desired effect. The reason for this problem is that the representation of the underlying structures in the images in 'entangled'. That is where infoGANs come into the picture to provide a disentangled representation that allows us to manipulate the structures in a generated images.

InfoGANs

The way InfoGAN works works is by splitting the Generator input into two parts: the traditional noise vector and a new "latent code" vector. These latent codes are then made meaningful by maximizing the

Mutual Information between the code and the generator output. It is expected that these latent codes shall capture the significant features like shape or color or thickness of objects, while the traditional noise vector shall represent some irrelevant details of the image.

In order to strengthen the relationship between latent code c and the image, the authors proposed that the Mutual Information between them should be as large as possible. This was achieved by modifying the loss function to incorporate this mutual information term. As it is difficult to mututal information explicitly, the authors derived a variational lower bound on the mutual information.

Results and Analysis

The algorithm was implemented on MNIST dataset and the results match the ones reported in the paper.

Result after 5 epochs:

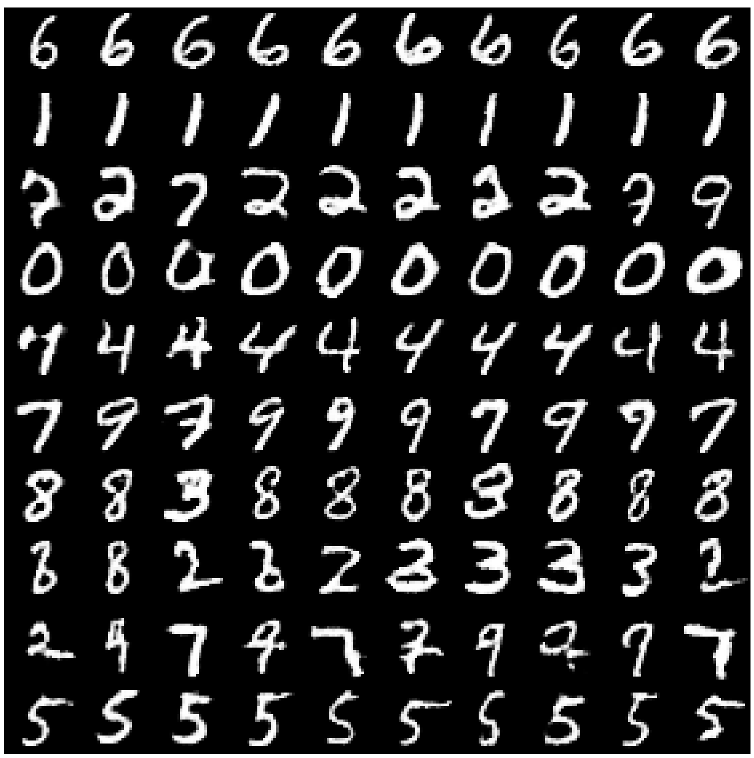

Figure 1: Variation of discrete code on InfoGAN - We are able to almost get a distinct digit corresponding to each of the dimension of discrete code. This is just after 5 epochs so we can see that certain digits such as 3, 4 and 7 do not look natural.

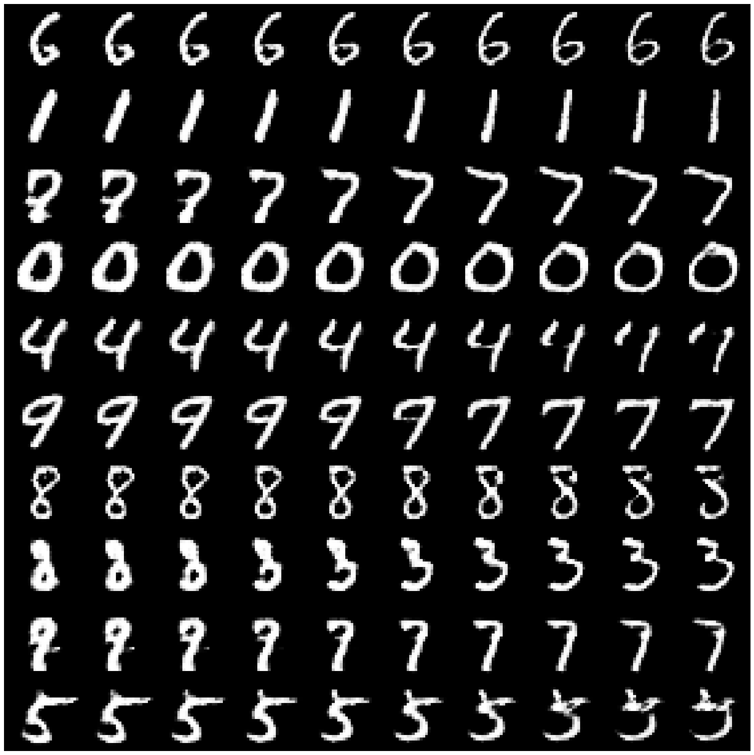

Figure 2: Variation of 1st continuous dimension on InfoGAN - There is a distinct thickness pattern observed here. The thickness of the digits decrease as we move from left to right. This pattern is distinctly evident for all the digits except the digits 8 and 4 while digits 9 and 4 get deformed at reduced thickness.

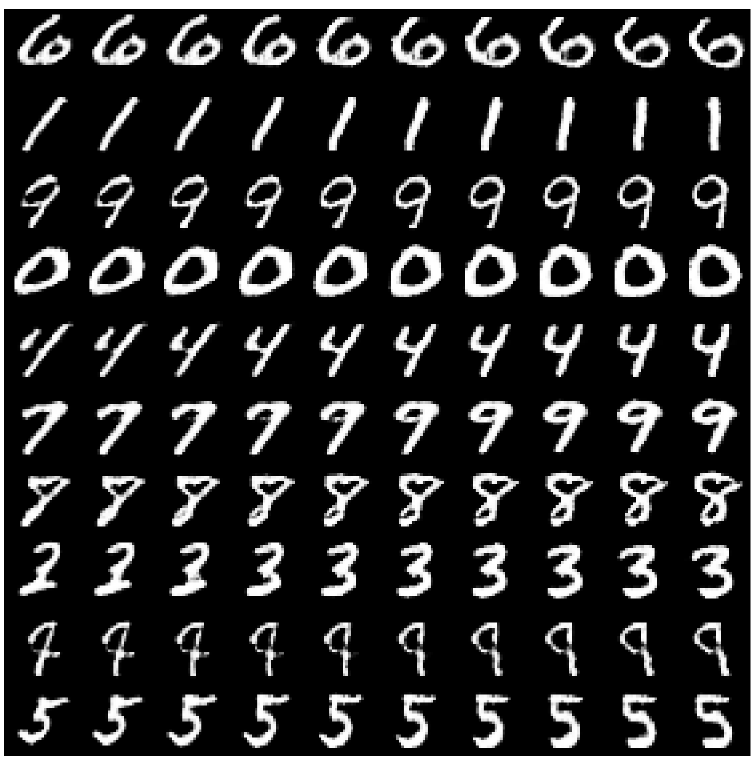

Figure 3: Variation of 2nd continuous dimension on InfoGAN - A rotation pattern is starting to emerge. The pattern looks good for digits 1, 6 and 9.

Result after 200 epochs (PTO)

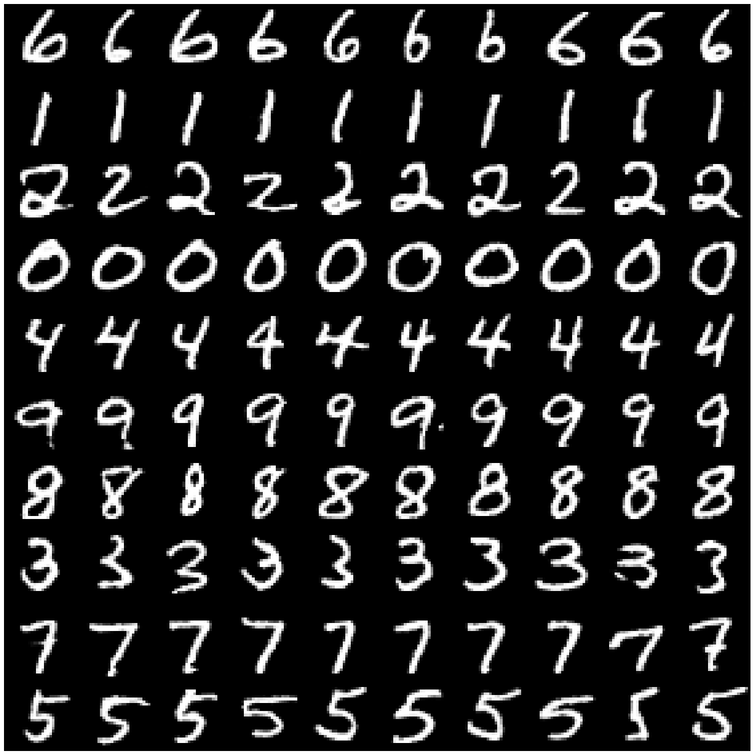

Figure 4: Variation of discrete code on InfoGAN - Distinct digit corresponding to each dimension of discrete code is very much evident and the digits look very natural and vary slightly from each other. Digits look much better as compared to the results after 5 epochs.

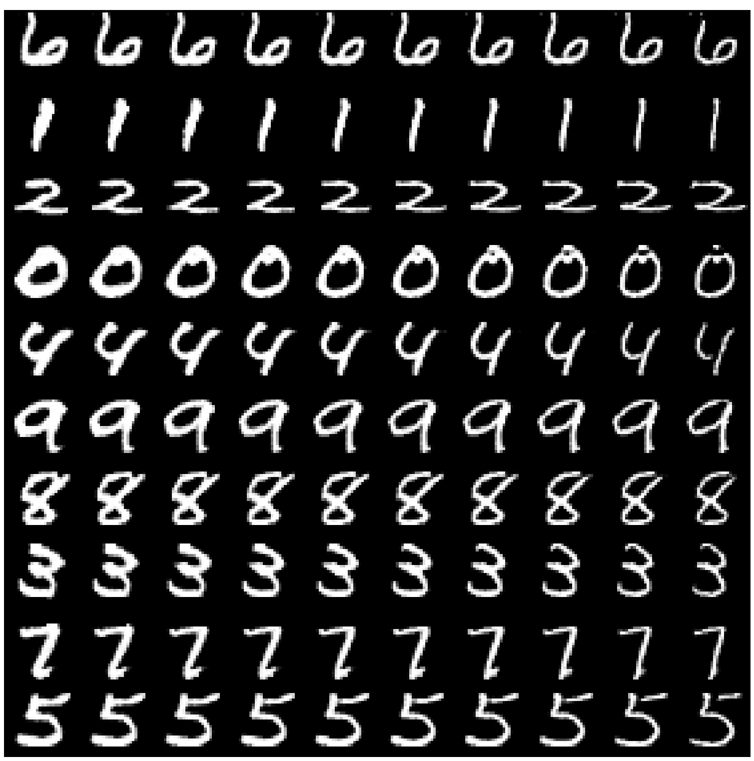

Figure 5: Variation of 1st continuous dimension on InfoGAN - The thickness pattern is evident for all the digits. Thickness reduces from right to left in such a manner that the digits are not stretched but the thickness is slightly changed and hence, the resultant images are very natural looking.

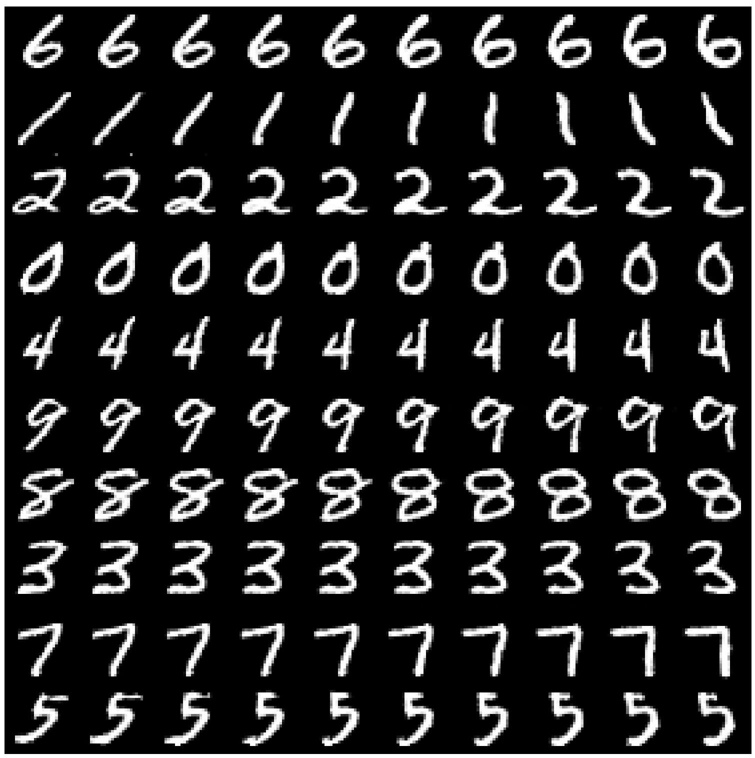

Figure 6: Variation of 2nd continuous dimension on InfoGAN - The rotation pattern is much clearly visible and is wonderfully demonstrated for digits such as 1, 4, 7 and 9 that have an almost vertical edge.