An Exploration into Generative Models

Dipankar Ghosh
dghosh6@uic.edu

Nigel Flower
nflowe3@uic.edu

Kushagradhi Bhowmik
kbhowm2@uic.edu

Sneha Shet
sshet6@uic.edu

Simran Jumani
sjuman2@uic.edu

Abstract

In order to explore generative models for this project, we worked on Variational AutoEncoders (VAE) and Generative Adversarial Network (GAN) architectures to explore how data generative systems work. We implemented these models on three datasets- MNIST, Labelled Faces in the Wild, and CelebA. Our goal in implementing VAEs and GANs both was to compare the quality of images generated.

1 The problem

Our main task in this project was to apply GANs, in this case DCGANs[1] specifically, to generate data along three different datasets and then compare the generated results to another popular generative model - the VAE. We chose to use DCGANs because they use convolutional layers, which allow for more efficient processing of image data. We worked with three different datasets, that are traditionally used when applying GANs. First, we used the MNIST dataset as a preliminary model to ensure that our model was performing correctly and that our code is along the correct trajectory. Once we have trained a GAN on the MNIST digit dataset, we continued with the two datasets relating to faces: the Labeled Faces in the Wild dataset and the CelebA dataset.

2 Dataset

We have generated models for three datasets:

1. The MNIST database of handwritten digits[2] [link]

   It is a collection of images of handwritten digits and is segregated into a training set of 60,000 examples and a test set of 10,000 examples.

2. Labeled Faces in the Wild[3] [link]

   The data set contains more than 13,000 images of faces. We resized the images to size 40x40 to speed up the models.

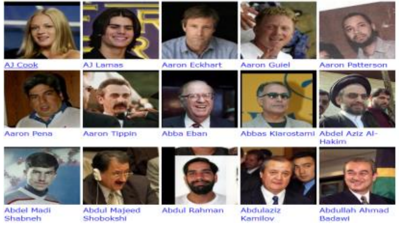


Figure 1: Sample from the 'Label Faces in the Wild' dataset

3. Large-scale CelebFaces Attributes (CelebA)[4] [link]

   This database has over 200,000 celebrity images, each with 40 attribute annotations. We considered a subset of this dataset for both the GANs and VAE. For the VAE model we focused only on the face and cropped out the background.

Figure 2: Sample from the CelebA database

3 Approach

Variational Autoencoder (VAE)[5]

The inability of the traditional autoencoder to generate new images is addressed by using a variational autoencoder. It uses an architecture like the traditional autoencoder. However, it does not learn to morph the data in and out of a compressed representation of itself. Instead, they learn the parameters of the probability distribution that the data came from, in terms of underlying, unobserved latent variables. It is essentially an inference model and a generative model daisy-chained together. We can consider the encoder takes an example *x,* and produces a latent representation *z*, denoted as $q_\theta (z \mid x)$. The input to the decoder is a hidden representation *z*, and it produces a recreated example *x* and is denoted by $p_\Phi (x \mid z)$.

VAE is trained using a loss function with two components:

1. Reconstruction loss - This is the cross-entropy describing the errors between the decoded samples from the latent distribution and the original inputs.
2. The Kullback-Liebler divergence between the latent distribution and the prior (this acts as a regularization term).

The loss over a single datapoint is: $l_i (\theta, \Phi) = - E_{z\sim q\theta (z|xi)} [\log p_\Phi (x_i|z)] + KL[q\theta (z|xi) \| p(z)]$

Generative Adversarial Networks (GAN)[6]

Devised by Ian Goodfellow in 2014, GAN has two differentiable functions (as neural networks) are locked in a game (adversarial process). Generative model G captures the data distribution, while discriminative model D estimates the probability that a sample came from the training data rather than G. Much like the VAE, the generator network generates data from a latent vector. These two networks are adversarial in the sense that they are trying to minimize their costs at the expense of the other network. Both G & D use binary cross entropy as the loss function.

Labelled Faces in the wild:

VAE Architecture:

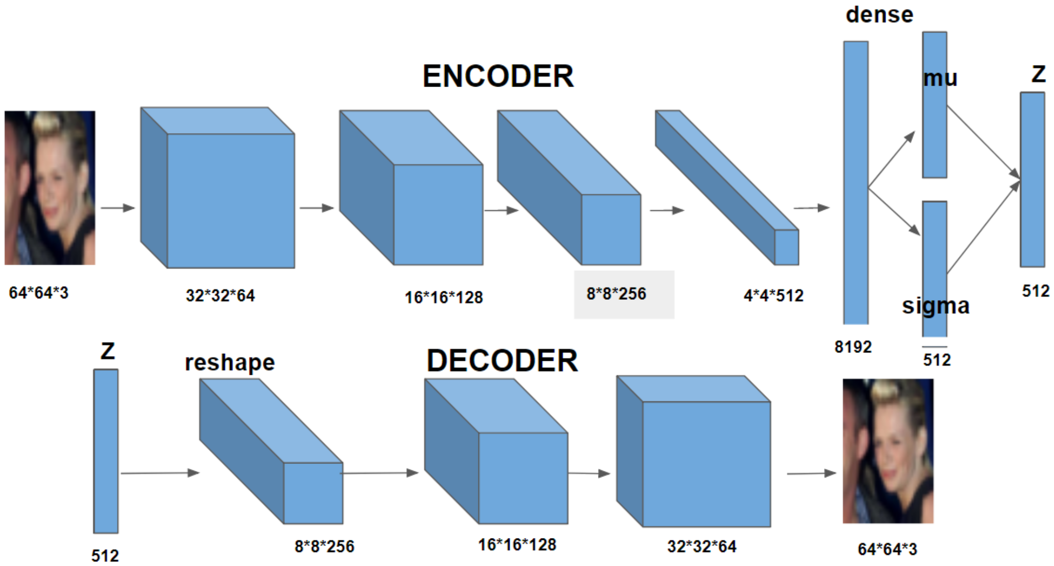


Figure 3: VAE model architecture for LFW dataset

GAN architecture:
The goal here was to train the generator to generate images based on the dataset provided to it, and to be able to fool the discriminator in believing that these were real images. We implemented this model using Tensorflow.

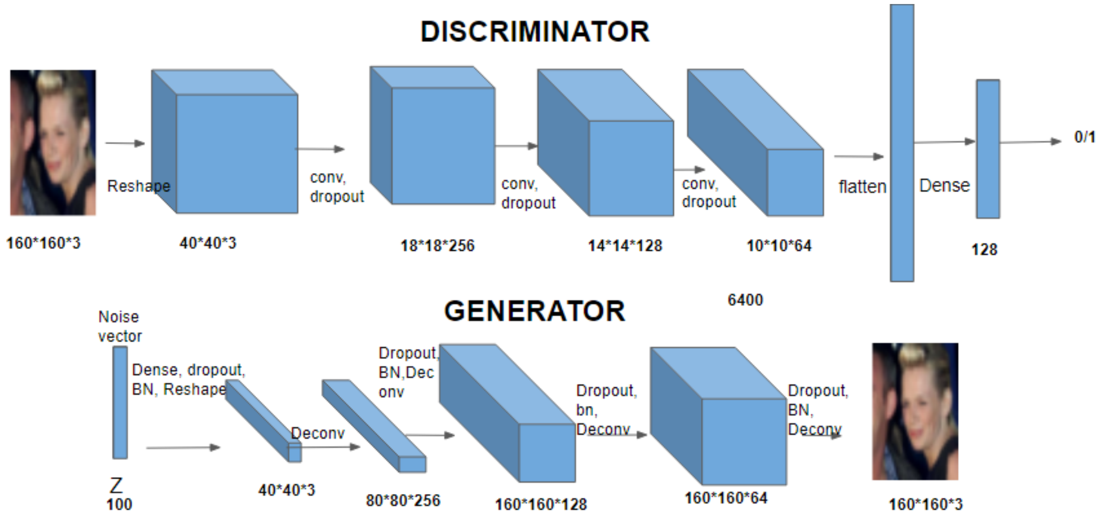


Figure 4: GAN model architecture for LFW dataset

CelebA:

VAE Architecture:

For this model, we chose a latent representation space of dimension 64.

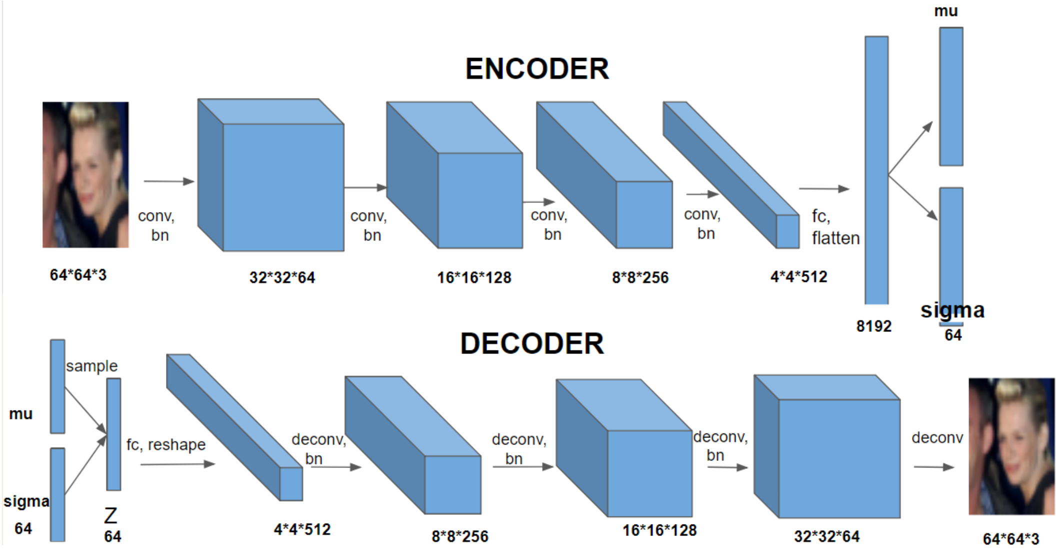


Figure 5: VAE model architecture for CelebA dataset

GAN Architecture:

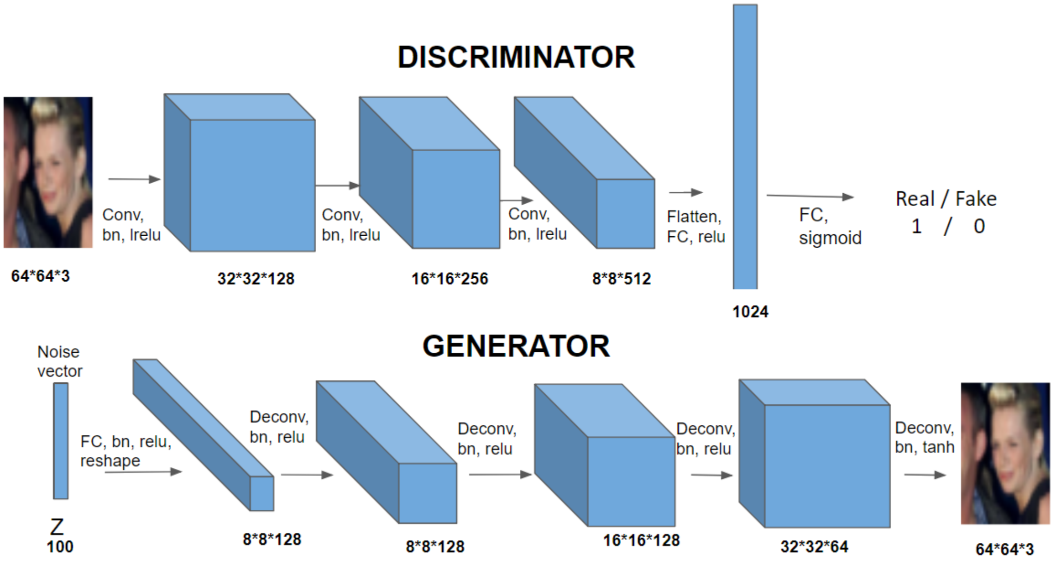


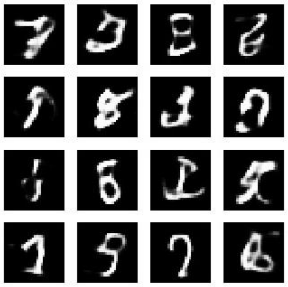
Figure 6: GAN model architecture for CelebA dataset

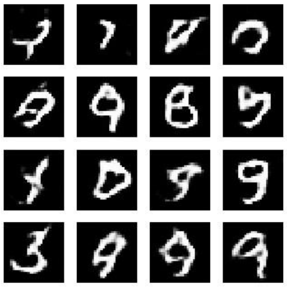
4 Results & Evaluation

In this section, we compare the output from the various datasets and models. The evaluation is qualitative as it is difficult to quantitatively evaluate the various output generated in a meaningful manner.

4.1 Dataset: MNIST

Following are the results produced after training on the MNIST dataset (left VAE, right GAN):





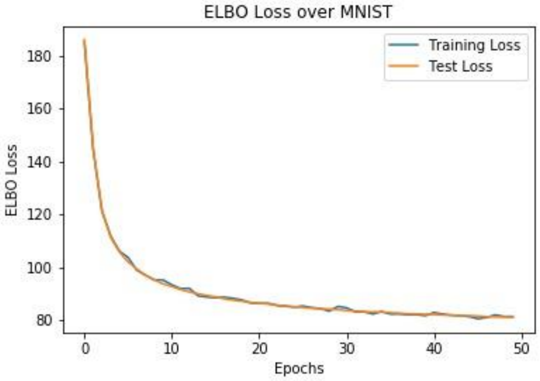


Figure 7: ELBO losses over MNIST for VAE

4.2 Dataset: Labelled Faces in the Wild

4.2.1 Model: VAE

The output generated from the VAE is as below:

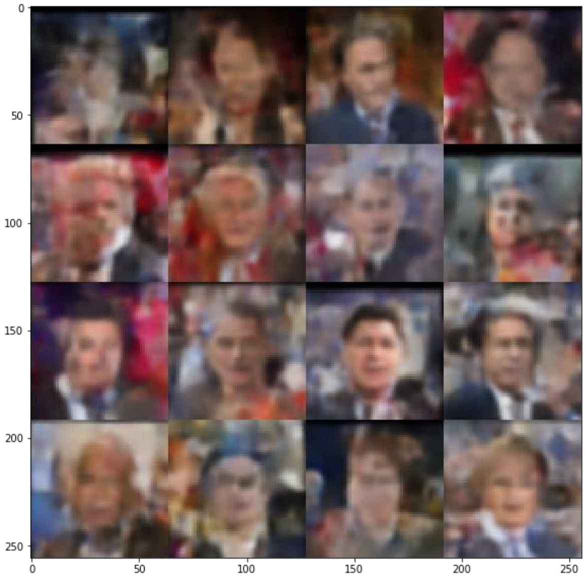


Figure 8: Images generated by VAE on LFW dataset

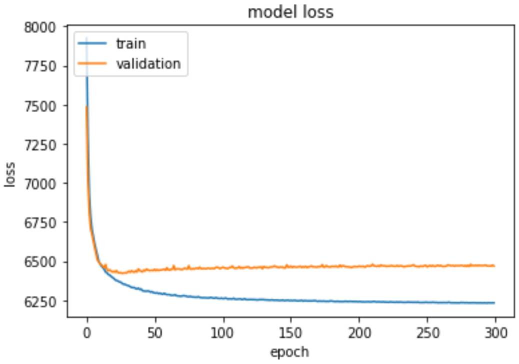


Figure 9: Loss function – VAE on LFW dataset

4.2.2 Model: GAN

The output generated from the GAN by the 40th epoch looks as follows:

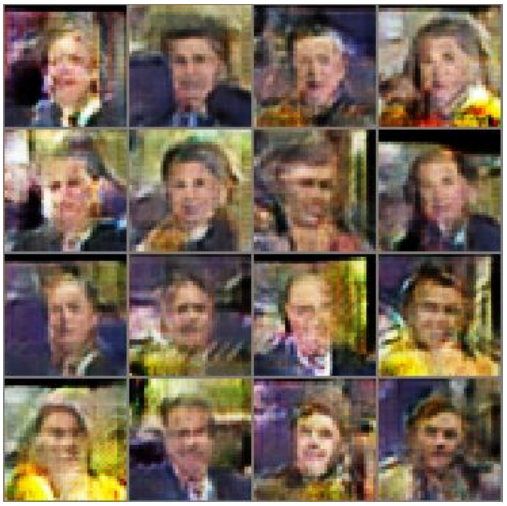


Figure 10: Images generated after 40th epoch

While the images start to resemble the faces of humans, a deeper network trained for a couple of days is required to generate sharper images. As shown below by the loss function, the discriminator does reach a loss of ~0.6 which shows that we significantly succeeded in fooling the discriminator with the generated images.

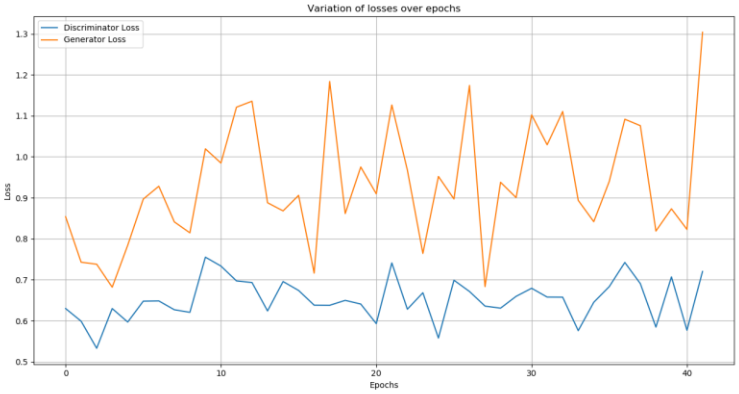


Figure 11: Loss function- discriminator vs generator

4.3 Dataset: CelebA

4.3.1 Model: VAE

For this model, we cropped the part of the images surrounding the center to focus on the faces. While this helps generate the facial features better with less noise, it is still blurry. And as seen with the LFW dataset, the images generated by the VAE model without focusing on the facial features are blurrier than the GANs.

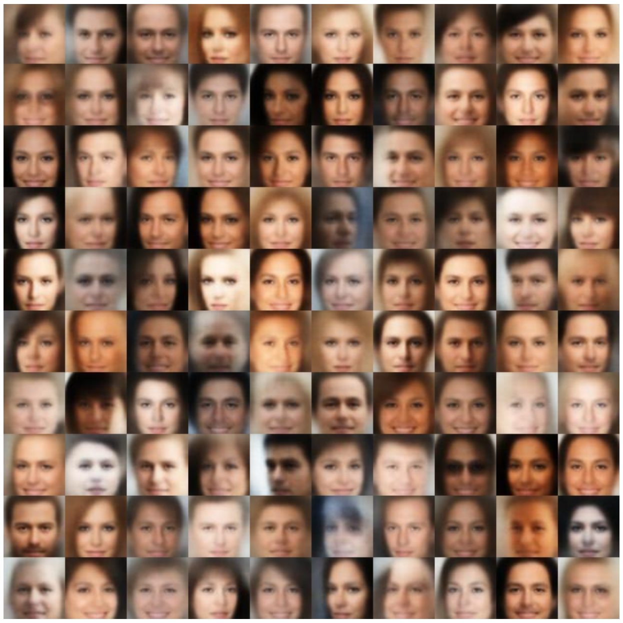


Figure 12: Images generated by VAE on CelebA after 20th epoch

Presented below are the progression of the loss function and the regularization term over the epochs. From the first, we can observe that the reconstruction error progressively reduces, as the reconstructed images becoming more similar to the original over epochs. From the second plot we can see that the encoded latent space increasingly models the target unit Normal distribution.

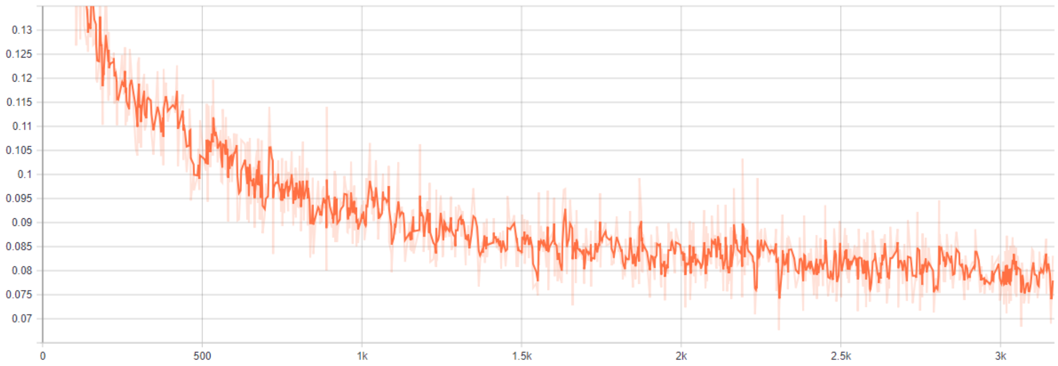


Figure 13: Mean Squared Error Loss for VAE on CelebA

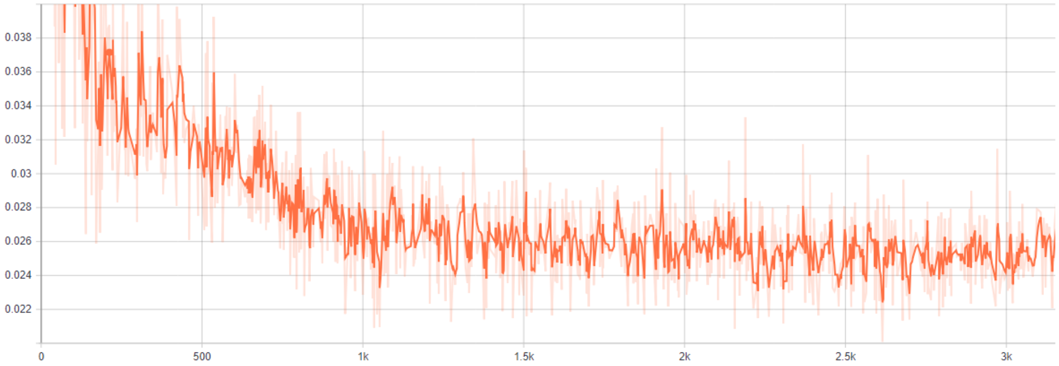


Figure 14: KL Divergence plot for VAE on CelebA

4.3.2 Model: GAN

The output generated from the GAN by the 40th epoch looks as follows:

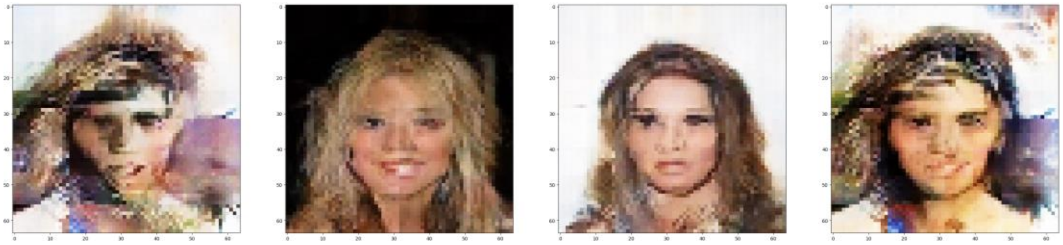


Figure 15: Images generated by GAN on CelebA after 40th epoch

We down-scaled the input images to 64*64 for ease of computation. However, we can use the original resolution and a deeper network to generate sharper images.

In the binary cross entropy loss function plot below with respect to the output of D, we observe that G tries to minimize it and D tries to maximize it. A unique solution exists, with G recovering the training data distribution and D equal to 1/2 everywhere. However, in our plot we reach an equilibrium state with a small gap between the G & D converged losses.

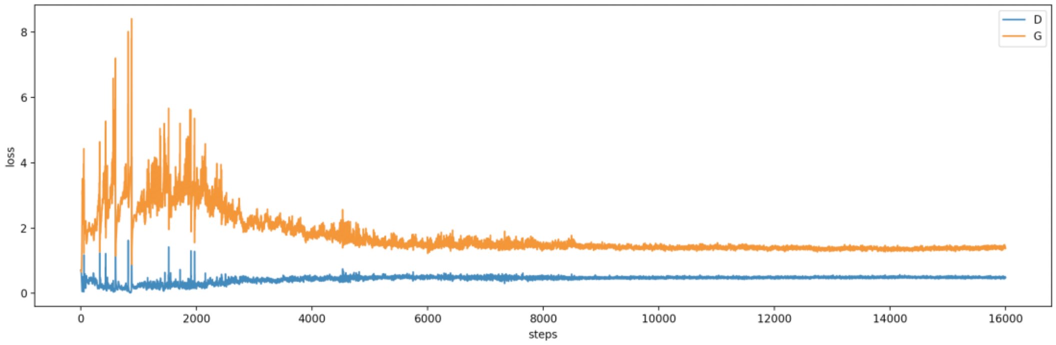


Figure 16: Loss function- discriminator vs generator

Overall the images produced by the GAN seem to have greater finer detail in the faces, notwithstanding the artefacts in the images. The results from the VAE are comparatively blurrier, which is expected due to the loss of information in the lower dimensional latent representation.

5 Summary and Future Work

We learnt how generative networks work. In specific, we saw how VAE & GAN can be used efficiently to generate new data points in a probability distribution from a latent representation. Such models that can generate new data from latent representation can be widely used to represent and manipulate high-dimensional probability distribution, train with less data and provide predictions on missing data, generating video frames, text-to-image applications, and many more.

Our future work will be focused on:

1. Be more systematic in comparing our models and explore different hyperparameter settings for each model
2. Try more complex architecture for GAN in order to get better output
3. Try higher resolution input images for GAN

References

[1] Alec Radford, Luke Metz, Soumith Chintala.. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv:1511.06434v2 [cs.LG], 2016

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. 1998

[3] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49. 2007

[4] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. ICCV 2015

[5] Diederik P Kingma, Max Welling. Auto-Encoding Variational Bayes. arXiv:1312.6114 [stat.ML] 2014

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative Adversarial Networks. arXiv:1406.2661 [stat.ML] 2014