Preliminary Project Report

Abstract

GANs are generative models where we use adversarial process and simultaneously train two models, a generative model and a discriminative model. The generative model G captures the data distribution, and the discriminative model D estimates the probability that a sample came from the training data rather than G. It uses implicit density to generate new samples, unlike traditional methods such as AutoEncoders which minimize explicit density of training samples and produces unrealistic blurry images.
In this project, we aim to use GANs for face aging also known as age synthesis, where we render a face image with natural aging and rejuvenating effects on the individual face. We further discuss various versions of GANs such as Conditional GAN, where we feed the input with a conditioning variable to condition on to both generator and discriminator, Age-cGANs where we focus on identity preserving face aging.

1. Introduction

The applications of face aging include, finding lost children and cross-age face recognition. Traditional face aging are roughly of two types: prototyping and modeling. Prototyping approach of face aging are simple and fast - they estimate average faces within predefined age groups. Since they are based on general rules, personalised information is discarded, producing unrealistic images. On the other hand, modelling approaches using parametric models to simulate aging process. To train a model, we require various age sequences of the same person which is expensive.

To solve this problem, we will prefer generative models which generates new samples by estimating the probability density of given samples. Generative models are of two kinds - explicit and implicit density models. Explicit density models try to maximize the likelihood estimation of training data to generate new samples but the images produced are of low quality. Implicit density models, such as GANs, use game theoretic approach to generate realistic samples.

There are a different types of GANs, for the purpose of face aging we are going to discuss about Conditional GANs where we use target age group as conditional input.

2. Literature Review

2.1. Generative Models

Deep generative models (DGM) are neural networks with many hidden layers trained to approximate complicated, high-dimensional probability distributions using a large number of samples. When trained successfully, we can use the DGMs to estimate the likelihood of each observation and create new samples from the underlying distribution.

As of today, the three most popular generative models are Fully Visible Belief Networks (FVBN), Variational AutoEncoder and GAN.

2.2. Fully Visible Belief Networks (FVBN)

It is an explicit density model. It uses chain rule to decompose the likelihood of an image X into product of 1D distribution.

$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, ..., x_{i-1}) \tag{1}$$

Then, we maximize the likelihood of the training data.

2.2.1 PixelRNN

PixelRNN generates image pixels starting from a corner. It depends on previous pixels which are modelled using an RNN (LSTM). Since we are sequentially generating pixels using RNN, it is very slow.

2.2.2 PixelCNN

PixelCNN also generates image pixels starting from a corner. Its dependency on previous pixels is now modelled using a CNN over context region. Since we can use parallel convolutions, training is faster than PixelRNN. But, generation is still slow as we are proceeding sequentially.

2.3. Variational Autoencoders

In variational autoencoders (VAE), we use interactable density function and optimize the lower bound of the likelihood of the training data. We use an encoder network to

model $q_\phi(z|x)$ where, z is a latent vector and x is an input image. Further, we model $p_\theta(z|x)$ using a decoder network. Following is the objective function for the model,

$$\mathcal{L}(x^{(i)}, \theta, \phi) = \mathbf{E}\left[\log p_\theta(x^{(i)}|z)\right] - D_{KL}\left(q_\phi\left(z|x^{(i)}\right)||p_\theta(z)\right) \quad (2)$$

The variational lower bound for the likelihood,

$$\log p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)}, \theta, \phi) \quad (3)$$

We train our model by maximizing the lower bound

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)}, \theta, \phi) \quad (4)$$

Pros:

- Principled approach to generative models.
- Allows inference of $q(z|x)$ which can be a useful feature representation for other tasks.

Cons:

- It maximizes the lower bound of likelihood. But, the evaluation is not as good as PixelRNN/PixelCNN.
- The generated samples are blurry and of low quality.

2.4. Generative Adversarial Network

Generative Adversarial models uses a game-theoretic approach, where we train two networks - generator network and discriminator network.
The generator network tries to fool the discriminator by generating real looking images. The discriminator network tries to distinguish between real and fake images.
The generator network models a distribution of real looking images from a random noise. And, the discriminator network outputs the likelihood in $(0, 1)$ of real image. We train both the networks jointly using a minimax objective function.

$$\min_{\theta_g}\max_{\theta_d}[\mathbb{E}_{x\sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))] \quad (5)$$

where, $D_{\theta_d}(x)$ is the discriminator output for the real data x and $D_{\theta_d}(G_{\theta_g}(z))$ is the discriminator output for the generated fake data $G(z)$.

for number of training iterations **do**
  for k steps **do**
    • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)})))\right]$$

  end for
  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Update the generator by ascending its stochastic gradient (improved objective):

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

Figure 1. Training pipeline for model

2.5. Conditional GAN

Generative Adversarial Networks can be extended to a conditional probabilistic model if, a conditional variable y is fed into both generator and discriminator as an input layer. In the generator, the prior input noise $p_z(z)$ and y are combined in joint hidden representation. In the discriminator, x and y are presented as input to a discriminative function.

$$\min_{\theta_g}\max_{\theta_d}[\mathbb{E}_{x\sim p_{data}} \log D_{\theta_d}(x|y) + \mathbb{E}_{z\sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z|y)))] \quad (6)$$

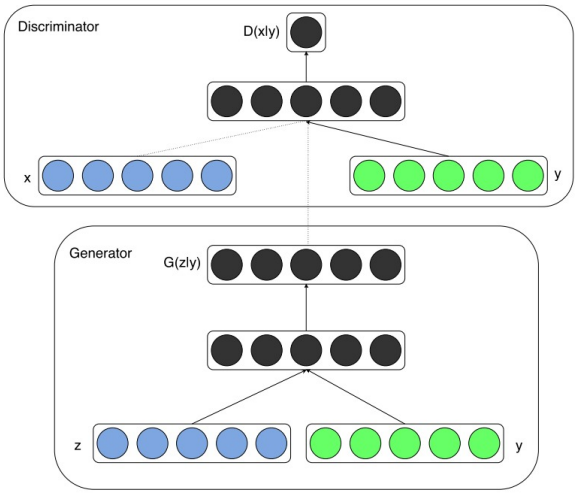

Figure 2. Conditional adversarial network

2.6. Age-cGAN

In Age-cGAN, we use a novel latent vector optimization approach which allows Age-cGAN to reconstruct an input face image preserving the original person's identity. The

Age-cGAN has multiple stages of training. It has 4 networks which get trained in 3 stages.

1. Conditional GAN training: Training of generator and discriminator networks
2. Initial latent vector approximation: Training of encoder network
3. Latent vector optimization: Training of facial recognition network

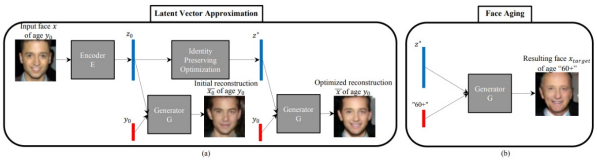

Figure 3. Face aging method using Age-cGAN

2.6.1 Latent Vector Optimization

The ultimate goal of the face aging task is to change the age of the person without changing the identity. Use of initial latent approximations results in loss of identity of the original person in about 50% of the cases. Therefore, initial latent approximations must be improved.
Pixel-wise latent vector optimization can be used for the above purpose. But, it has 2 clear downsides. It increases the blurriness of reconstruction and it focuses on unnecessary details of input face images, which have nothing to do with the person's identity. Instead, identity preserving latent vector approach is used where the difference between the identities in the original and the reconstructed images is expressed as the Euclidean distance between the corresponding embeddings. The objective is to minimize this distance.

$$z^*_{IP} = \arg\min_z \|FR(x) - FR(\bar{x})\|_{L_2} \tag{7}$$

This face aging method can be used for synthetic augmentation of face dataset and for improving the robustness of face recognition solution in cross-age scenarios.

2.7. FaceNet

The face recognition network used in previous method for latent vector optimization is nothing, but an internal implementation of FaceNet CNN. In FaceNet, we train a unified system for face verification, recognition and clustering. We map an input image to a lower dimensional Euclidean space where, the squared L2 distance between the mapping directly corresponds to image similarity. The above mentioned tasks can now be achieved the following way: face verification by assigning a threshold distance, recognition by simple k-NN classification and clustering by k-Means or agglomerative clustering.
The architecture is inspired from the models: ZeilerFergus consisting of multiple interleaved layers, bottleneck layer consisting of $1 \times 1 \times d$ convolutions and Inception model of Szegedy consisting of parallel convolution and pooling layers. Triplet loss is the novel loss function employed in the model.

2.7.1 Triplet Loss

Definitions: Anchor image (x_i^a) - Current image for which the loss function is calculated.
Positive image (x_i^p) - Image belonging to the same class as anchor image.
Negative image (x_i^n) - Image not belonging to the same class as anchor image.
Objective of the Triplet loss function is to maximize the distance between the anchor and positive images while minimizing the one between anchor and negative images. Mathematically,

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha \leq \|f(x_i^a) - f(x_i^n)\|_2^2 \tag{8}$$

where α is the margin between positive and negative pairs. The loss function that is to be minimized is

$$\sum_i^N [\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha - \|f(x_i^a) - f(x_i^n)\|_2^2]_+ \tag{9}$$

2.7.2 Triplet Selection

The selection of triplets is such that we choose those which violate the triplet constraint for faster convergence. Hard positives and hard negatives are selected. Hard positives are positive images that are farthest from the anchor image in the Euclidean space ($argmax_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$). Similarly, hard negatives are negative images that are closest to the anchor image ($argmin_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$). Outliers and poorly labelled images make selection difficult. Thus, the dataset is divided into mini-batches to overcome the above problem. Further, in a mini-batch all the anchor-positive pairs are used while still selecting the hard negative points.

2.7.3 Training

The model was trained using Stochastic Gradient Descent and optimization technique as AdaGrad. The performance of the model was found to be similar irrespective of the architectures used. But, Inception based model reduced the model size and the number of FLOPS.

2.8. Bidirectional DCGAN

In the application on face aging, Deep Convolutional Generative Adversarial Networks works with the objective to maintain personalization and acquire it in the aging face. The framework of the network is as follows:

- **Encoder(E)** : Convolutional encoder is a fully connected convolutional network. An image is x is fed as an input to it which is mapped to a low-dimensional feature vector z.

$$z = E(x) \tag{10}$$

- **Connector(C)** : It merges the feature vector z, age vector a and gender vector g which becomes the conditional input to the generator.

- **Generator(G)** : Generator acts as a decoder. It reconstructs the face image corresponding to the conditional output fed to it, by extracting the feature information. $G(C(z, y))$ is the generated aged face image. Gender condition is added to account for the fact that different genders having different aging characteristics.

- **Discriminator(D)** : It discriminates between the real and the synthesized aged face image and outputs the probability of the generated being a real one.

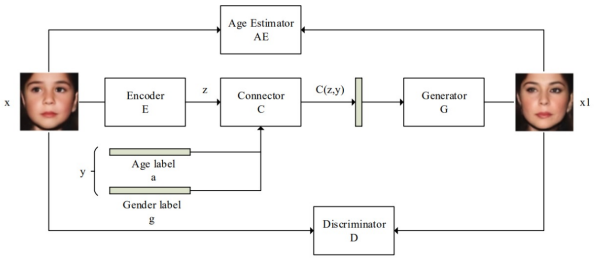

Figure 4. Framework of DCGAN

To generate clearer images, the model employs three loss functions:

- **Feature Space Loss** : It represents the difference between the feature maps of the real and generated face image. Instead of the usual practice of employing a trained network like AlexNet, to reduce the network complexity the output of the middle layer of the Discriminator D is calculated.

$$L_f = E_{x,y}[\|D(G(E(x), y)) - D(x, y)\|^2] \tag{11}$$

- **Pixel Space Loss** : It represents the pixel wise difference between the real and generated images.

$$L_p = E_{x,y}[\|G(E(x), y) - x\|^2] \tag{12}$$

- **Age Loss** : It represents the difference in the age estimates between the real and synthetic face images. This helps generator to synthesize a reasonable aged face.

$$L_a = E_{x,y}[\left\|l^{'} - l^{"}\right\|^2] \tag{13}$$

The objective function obtained by combination of the above loss terms is,

$$\min_{E,G}(\lambda_g L_g + \lambda_p L_p + \lambda_a L_a + \lambda_f L_f) \tag{14}$$

where $\lambda_g, \lambda_p, \lambda_a, \lambda_f$ are weight factors of generator, pixel, age, and feature space losses respectively.

2.9. Progressive Face Aging

Traditional methods of modeling face age progression can be roughly divided into two categories: physical model and prototype-based methods. Physical models mechanically stimulate the changes in facial appearance over time using varying parameters. However, these models don't generalize well and are computationally expensive. Prototyping models, on the other hand, compute the average face of an age group as protoype. The final output is then calculated by adding the difference between given image and prototype. These models are computationally efficient but it cannot map personalized features well.

GAN based models such as cGAN use different age groups as conditioning variables while training the generator. A single model is trained to learn the aging effects between different age groups. Therefore, these models cannot perform following three functions - image quality, identity preservation and aging accuracy.

In PFA-GANs, multiple sub-networks are trained to learn the aging effects between two adjacent age groups. Since, aging is a gradual process, these models perform better than traditional cGANs. In these models, novel encoding scheme is introduced by adding binary gates to control aging flow.
Therefore, the progressive aging framework from the source age group s to target one t can be formulated as follows:

$$X_t = G_{t-1} \circ G_{t-2} \circ .....G_s(X_s) \tag{15}$$

Residual skip connections are also introduced in these subnetworks to prevent memorizing the exact copy of input faces. Target age group image is then found by controlling the sequence of binary gates. The adjacent age group output can be re-written as:

$$X_{i+1} = G_i(X_i) = X_i + \lambda_i G_i(X_i) \tag{16}$$

Here, $\lambda \in \{0, 1\}$ is the binary gate controlling whether the sub-interval G_i is involved in the path to target age group.

Finally, a given face X_s of age group s, the aging from s to group t can be written as

$$X_t = G(X_s, \lambda_{s:t}) \tag{17}$$

Along with these sub-networks, PFA-GAN also contains

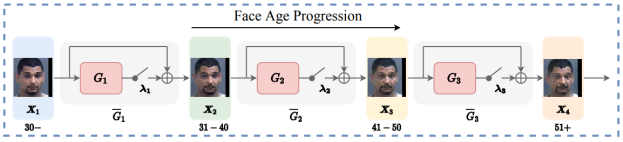

Figure 5. sub-networks in PFA

pre-age estimation network. These networks help to characterize the face-age distribution for an improved aging accuracy. These networks use both age-classification term and age regression term, to check whether the generated face belongs to the target age group or not.

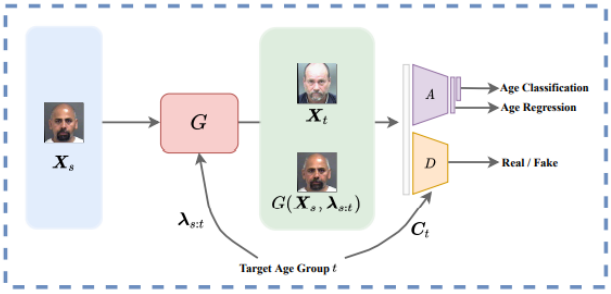

Figure 6. PFA-GAN Framework

2.9.1 Loss function

The overall loss function of PFA-GAN contains the following three components:

- **Adversarial Loss:** To produce high quality images.
- **Age Estimation Loss:** To improve aging accuracy.
- **Identity Consistency Loss:** To preserve identity.

1. **Adversarial Loss:** Unlike normal-GANs, least squared loss function is used as an adversarial loss. It improves the quality of generated images and stabilize the training processes. Given a face X_s from age group s, the output of G from s to a age group t is $G(X_s, \lambda_{s:t})$. In this context, the adversarial loss is given as:

$$L_{adv} = \frac{1}{2} E_{X_s} \left[D([G(X_s, \lambda_{s:t}); C_t]) - 1 \right]^2 \tag{18}$$

2. **Age Estimation Loss:** Age estimation network was included in PFA-network to regularize the face-age distribution by minimizing age-estimation loss which includes both regression and classification loss. The age-estimation loss between estimated age $\hat{y}$ and target age y for the generator age G is defined as:

$$L_{age} = E_{X_s} \left[\|y - \hat{y}\|^2 + l(A(X)W, c_t) \right] \tag{19}$$

3. **Identity Consistency Loss:** To preserve the identity related information of the face, the identity consistency loss has three components. These components are pixel-wise loss, structure similarity loss and feature level loss. The three losses are defined as:

$$L_{pix} = E_{X_s} \left[G(X_s, \lambda_{s:t} - X_s \right] \tag{20}$$

$$L_{ssim} = E_{X_s} \left[1 - SSIM(G(X_s, \lambda_{s:t})), X_s \right] \tag{21}$$

$$L_{fea} = E_{X_s} \left\| \phi(G(X_s, \lambda_{s:t})) - \phi(X_S) \right\|_F^2 \tag{22}$$

3. Preliminary Results

3.1. Dataset

We have used "Cats faces 64*64" dataset available on Kaggle.

3.2. Architecture

The architecture which produced the best results as follows:

Discriminator				
Layer	Filter	Stride	Activation	Output Shape
Conv	4 * 4	2	Leaky ReLU	16 x 32 x 32
Conv	4 * 4	2	Leaky ReLU	32 x 16 x 16
Conv	4 * 4	2	Leaky ReLU	64 x 8 x 8
Conv	4 * 4	2	Leaky ReLU	128 x 4 x 4
Conv	4 * 4	1	-	1 * 1 * 1

Table 1. Architecture of the discriminator network

Generator				
Layer	Filter	Activation	Padding	Output Shape
Deconv	4 * 4	ReLU	0	128 * 4 * 4
Deconv	4 * 4	ReLU	1	64 * 8 * 8
Deconv	4 * 4	ReLU	1	32 * 16 * 16
Deconv	4 * 4	ReLU	1	16 * 32 * 32
Deconv	4 * 4	Tanh	1	3 * 64 * 64

Table 2. Architecture of the generator network

3.3. Experiments

3.3.1 Trial 1

We used the learning rate as 0.002, the number of epochs as 60, the parameters β of Adam optimizer are $\{0.5, 0.999\}$. We started with an architecture as follows:

Discriminator				
Layer	Filter	Stride	Activation	Output Shape
Conv	4 * 4	2	Leaky ReLU	6 * 32 * 32
Conv	4 * 4	2	Leaky ReLU	10 * 16 * 16
Conv	4 * 4	2	Leaky ReLU	20 * 8 * 8
Conv	4 * 4	2	Leaky ReLU	40 * 4 * 4
Conv	4 * 4	1	-	1 * 1 * 1

Table 3. Architecture of the discriminator network

Generator				
Layer	Filter	Activation	Padding	Output Shape
Deconv	4 * 4	ReLU	0	40 * 4 * 4
Deconv	4 * 4	ReLU	1	20 * 8 * 8
Deconv	4 * 4	ReLU	1	10 * 16 * 16
Deconv	4 * 4	ReLU	1	6 * 32 * 32
Deconv	4 * 4	Tanh	1	3 * 64 * 64

Table 4. Architecture of the generator network

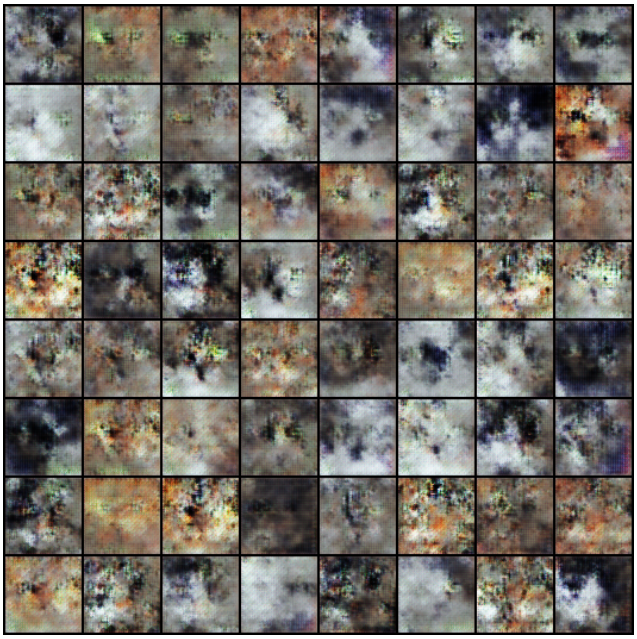

Figure 7. Generated cat images

3.3.2 Trial 2

We used the learning rate as 0.004, the number of epochs as 100, the parameters β of Adam optimizer are $\{0.65, 0.999\}$. We started with an architecture as follows:

Discriminator				
Layer	Filter	Stride	Activation	Output Shape
Conv	4 * 4	2	Leaky ReLU	6 * 32 * 32
Conv	4 * 4	2	Leaky ReLU	10 * 16 * 16
Conv	4 * 4	2	Leaky ReLU	15 * 8 * 8
Conv	4 * 4	2	Leaky ReLU	25 * 4 * 4
Conv	4 * 4	1	-	1 * 1 * 1

Table 5. Architecture of the discriminator network

Generator				
Layer	Filter	Activation	Padding	Output Shape
Deconv	4 * 4	ReLU	0	25 * 4 * 4
Deconv	4 * 4	ReLU	1	15 * 8 * 8
Deconv	4 * 4	ReLU	1	10 * 16 * 16
Deconv	4 * 4	ReLU	1	6 * 32 * 32
Deconv	4 * 4	Tanh	1	3 * 64 * 64

Table 6. Architecture of the generator network

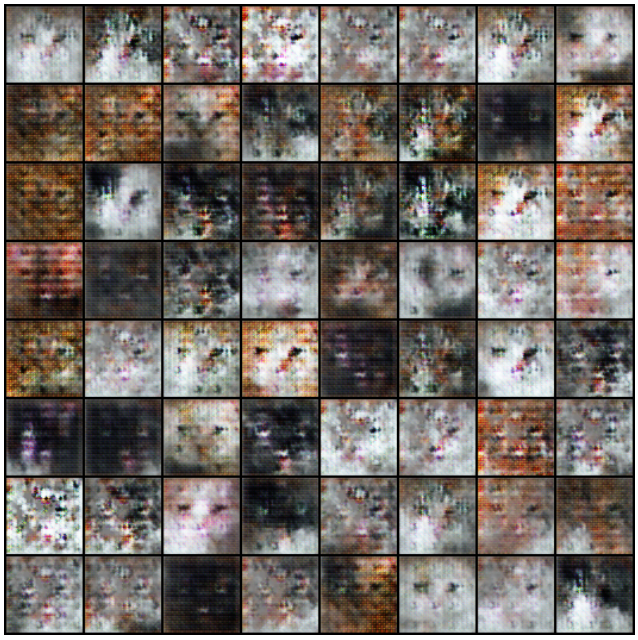

Figure 8. Generated cat images

3.4. Results and Future work

Building on this idea, we will implement age synthesis using Age-cGAN.

Figure 9. Generated cat images

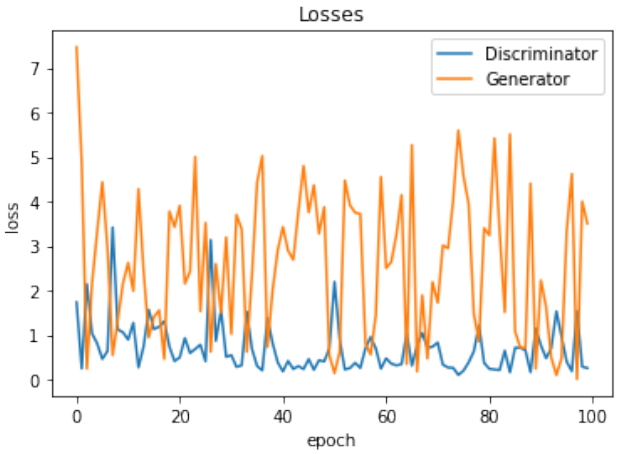

Figure 10. Plot of losses v/s epochs

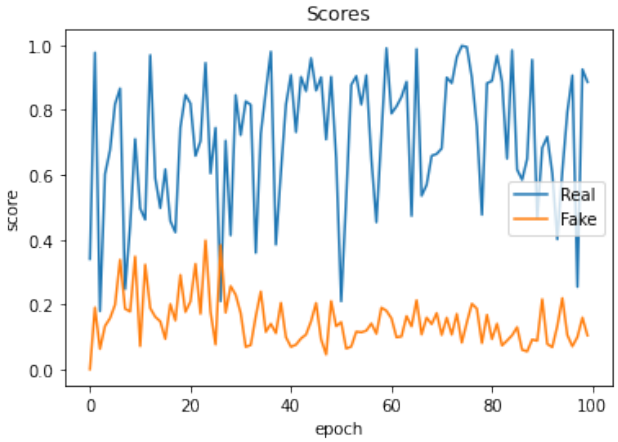

Figure 11. Plot of scores v/s epochs

References

[1] Aäron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu (2016) "Pixel Recurrent Neural Networks"

[2] Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, Koray Kavukcuoglu (2016) "Conditional Image Generation with PixelCNN Decoders"

[3] Diederik P. Kingma, Max Welling (2014) "Auto-Encoding Variational Bayes"

[4] Ian Goodfellow et al. (2014) "Generative Adversarial Nets", NIPS 2014

[5] Mehdi Mirza, Simon Osindero (2014) "Conditional Generative Adversarial Nets"

[6] Grigory Antipov, Moez Baccouche, Jean-Luc Dugelay (2017) "Face Aging with Conditional Generative Adversarial Networks"

[7] Xinhua Liu, Yao Zou, Chengjuan Xie, Hailan Kuang, Xiaolin Ma (2019) "Bidirectional Face Aging Synthesis Based on Improved Deep Convolutional Generative Adversarial Networks"

[8] Zhizhong Huang, Shouzhen Chen, Junping Zhang, Hongming Shan (2020) "PFA-GAN: Progressive Face Aging with Generative Adversarial Network"

[9] Florian Schroff, Dmitry Kalenichenko, James Philbin (2015) "FaceNet: A Unified Embedding for Face Recognition and Clustering"

[10] Matthew D. Zeiler, Rob Fergus (2013) "Visualizing and Understanding Convolutional Networks"

[11] https://towardsdatascience.com/getting-started-with-gans-using-pytorch-78e7c22a14a5