Project Report : Generative Models Using Probabilistic Principal Component Analysis

Group 9 - Generare
Aashima Yuthika(1401071), Deep Patel(1401010), Deval Shah(1401060), Kirtan Modi(1401122), Yash Kotadia(1401114)
School of Engineering and Applied Science, Ahmedabad University
April 23, 2017
Subject: ML & AOBD

***Abstract*—The main aim of this project is to be able to build Generative models using the concept of Probabilistic Principal Component Analysis or PPCA. PPCA essentially tries to fill in the data voids using various methods and assumptions. These data voids are usually known as the latent (unobservable) variables. We are using PPCA along with the General Adversarial Network as the main Generative Model, which uses Neural Networks, and has two components - Generator and Discriminator, which help in creating images that are as close to real world images as possible.**

I. Introduction

Generative models are the models that generate the real like data artificially. Although they are in their beginning stage of the research they are proposed to be applied in many fields like music generation, artificial data generation, user behavior prediction etc. Here we are working with GAN class of generative models. Also for this project we will be making use of the concept of PPCA to extract the principle components of the images and also make them noise free before using them as training set. PPCA is a method to find principal components of the data probabilistically by formulating the problem within a maximum-likelihood framework, based on a specific form of Gaussian latent variable model. This along with the General Adversarial Network is used in this project to generate real like images. When GAN architecture is combined with the aprior information of data in the form of PCA the training process is boosted as compared to GANs.

II. Model implemented by us

A. Deep Convolutional Generative Adversarial Network

At the beginning of the project, we implemented the DCGAN as it is to gain the insight into the word of generative models.Below is the figure which describes the model well:

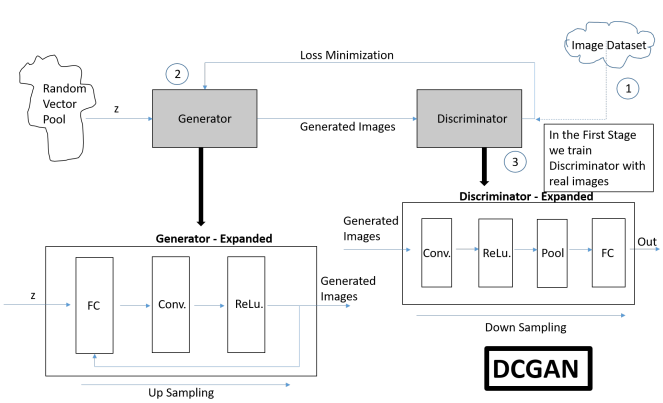


Fig. 1: DCGAN Model

DCGAN, in essence is a Convolutional Neural Network based GANs Model. It has two main blocks: Discriminator (D), Generator (G) The Discriminator learns on the real world dataset and then tries to judge how appropriate the resemblance of the generated images is with the real world images. The Generator's objective is to generate real world like images which can fool the Discriminator. So we can even think of it as a game wherein the Generator tries to generate images that can fool the Discriminator and the Discriminator tries to perfectly differentiate between real-world and fake images. The main objective here is to train the Generator till a point where it starts fooling the Discriminator.The Generator takes feedback from the Discriminator for every image that it generates and tries to update the output image of the underlying neural network through loss minimization. The Discriminator first learns the pattern of the given dataset using CNNs and then helps the generator to generalize it over the data space that it was given.

1) Algorithm

- i Initialize the weights of components randomly.
- ii Train the Discriminator with the training images, so that it collects the main features of the images.
- iii Give input of random vector (z) to the generator and then the generator generates an image by up-sampling z in CNN.
- iv Discriminator classifies the generated image as fake or real and sends its feedback to the generator. The generator then updates its image space of the network through loss minimization to generate realistic images that fools the discriminator.

Repeat steps (ii) through (iii) until convergence (this may also be the number of iterations). The results for this algorithm are discussed in the next section.

B. Our Model

After studying the DCGANs we have designed our own model which uses PCAs of the data and GAN arhitecture. Following figure describes our model in detail:

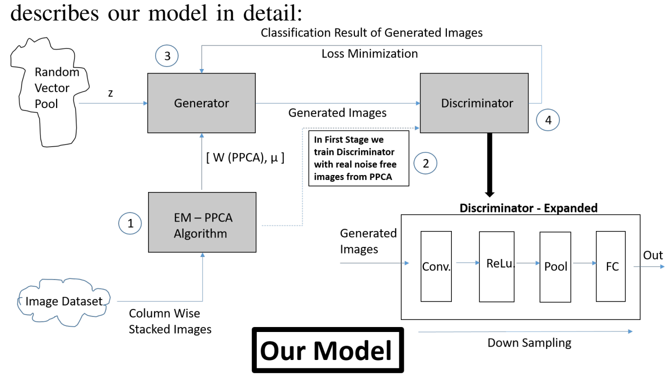


Fig. 2: Our Model

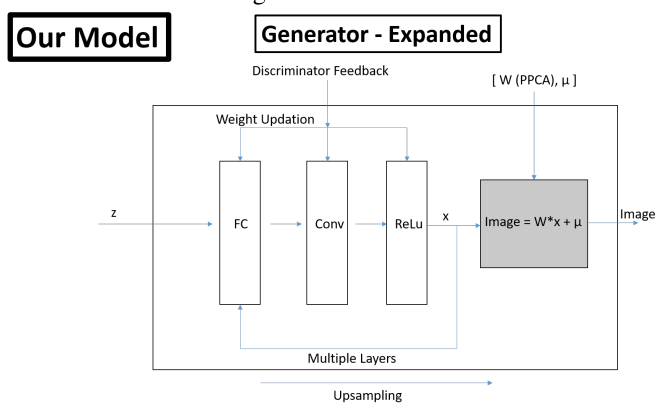


Fig. 3: Generator Block Expanded

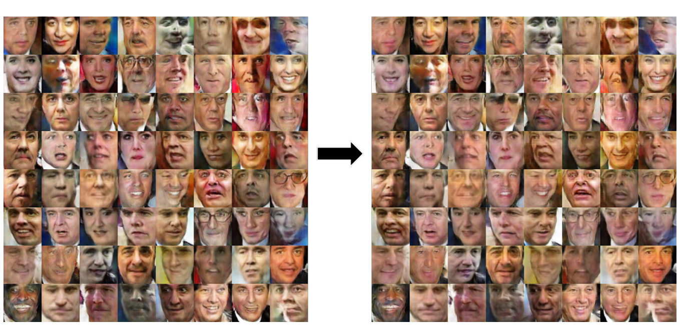
Our model is a hybrid of GAN and PPCA, the architecture, however, is same as that of GAN's. It has a Discriminator as well as a Generator and works the same as DCGANs except that in our model the Generator uses PPCAs as input along with random noise to generate the images. Extraction of PCAs of the input training set by using the PPCA-EM algorithm reduces the noise in the input set. Then we use these PPCAs to generate images and feed it to the Discriminator. The Generator has a well defined latent space in the form of pcas for it to work on, to generate new images.

C. Algorithm

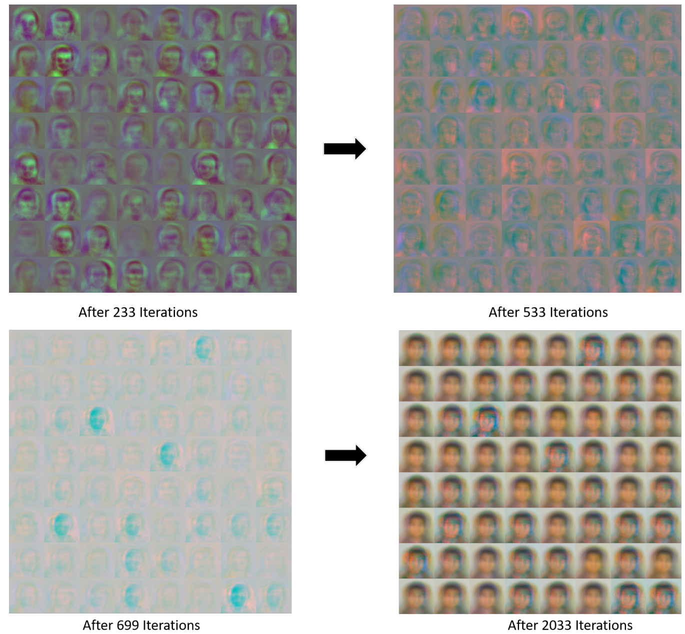
i Initialize model with the random parameters.
ii Apply PPCA-EM algorithm on the image dataset to reduce the noise, if any, and also to extract the PCAs of the data for use in the Generator.
iii Train the Discriminator with the real world images reconstructed from the PCAs obtained above.
iv Feed W(PCAs), data mean (μ), and random vector z in the Generator. The Generator then up-samples the z to x using CNN and then the image is generated using:- $Image = W \times x + \mu$
v The generated image is then fed into the Discriminator which classifies it as Fake or Real and gives feedback to the Generator. The Generator updates the image space accordingly to minimize the generator loss and generate more real looking images.

III. Results

A. DCGAN



B. Our Model



IV. Conclusion

Here are some advantages and disadvantages for the model that we have implemented to generate images.

A. Advantages

- This approach leads us to generate real looking images at a good pace as compared to DCGAN, because we are providing the PCAs of the data as the apriori information to the Generator, and thus the Generator has a good subspace to start with and will consequently converge faster as compared to the General Adversarial Network.
- PCAs can help generalize the structure of the data and thus chances of overfitting a dataset are diminished.
- Noise in the dataset is reduced by the EM-PPCA algorithm, thereby improving the quality of the training dataset and not training the Discriminator using noisy data

B. Disadvantages

- Apriori computation of PCAs is computationally expensive and requires a lot of time.
- During training our model we required 1 hour to compute the PPCAs of 256 color images of 64x64 dimensions.
- The images generated are somewhat blurry (but we can identify that faces are generated)as compared to that generated by DCGANs.

V. Future Work

We have used a fully connected layer in the Generator and are still getting appreciable results, so we are planning to add convolutional layers to the Generator so that it can learn the features of the images as well. We are also working on variational auto encoder technique (Generative Model) along with implementing our current working model on different datasets.

References

[1] Michael E. Tipping; Christopher M. Bishop, *Probabilistic Principal Component Analysis*, Journal of the Royal Statistical Society. Series B (Statistical Methodology) Vol. 61, No.3 (1999), 611-622.
[2] https://bamos.github.io/2016/08/09/deep-completion/
[3] https://github.com/bamos/dcgan-completion.tensorflow