RegGAN Summary

Ahmad Shayaan
as5948

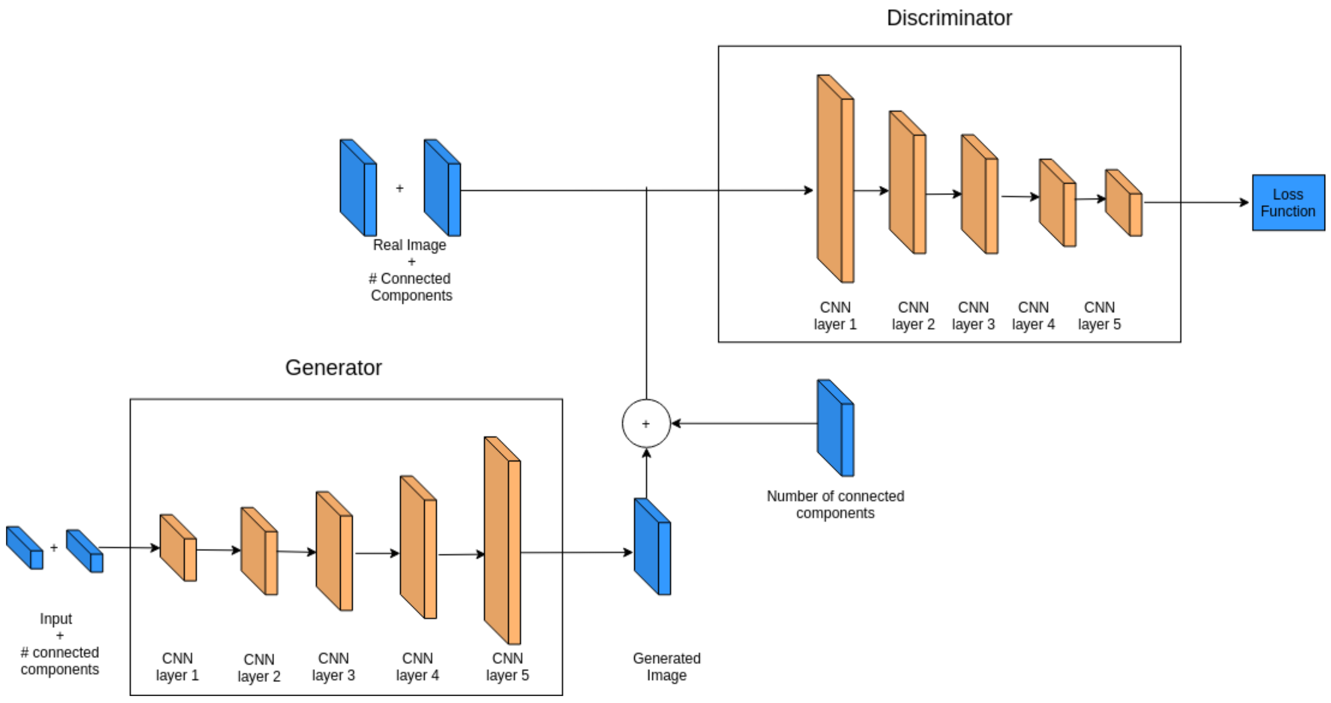
{ahmad.shayaan@columbia.edu}@columbia.edu
Columbia University
January 24, 2021

Conditional Deep Convolutional Generative Adversarial Networks

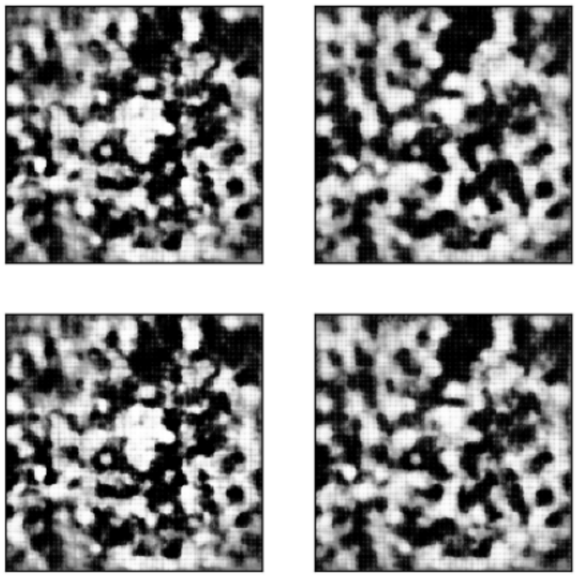
We tired to use a Conditional Deep Convolutional Generative Adversarial Networks (CDC-GAN) to generate the images. CDCGAN are an extension of the vanilla DCGAN in which both the discriminator and the generator are conditioned on some extra information. CDC-GAN have been successfully use to generate medical data link

The conditioning of the model on the extra gives the model an initial starting point to generate data. It also gives the user more control on what type of data we want the mode to generate. We condition both the generator and the discriminator on the number of connected components we want the generated image to have. We hoped that conditioning the model on the number of connected components would provide the model with necessary information so as to generate images with the desired structure. However empirically we found that this is not the case, the model fails to sufficiently leverage the extra information provided.

The high level architecture of the CDCGAN is shown below.



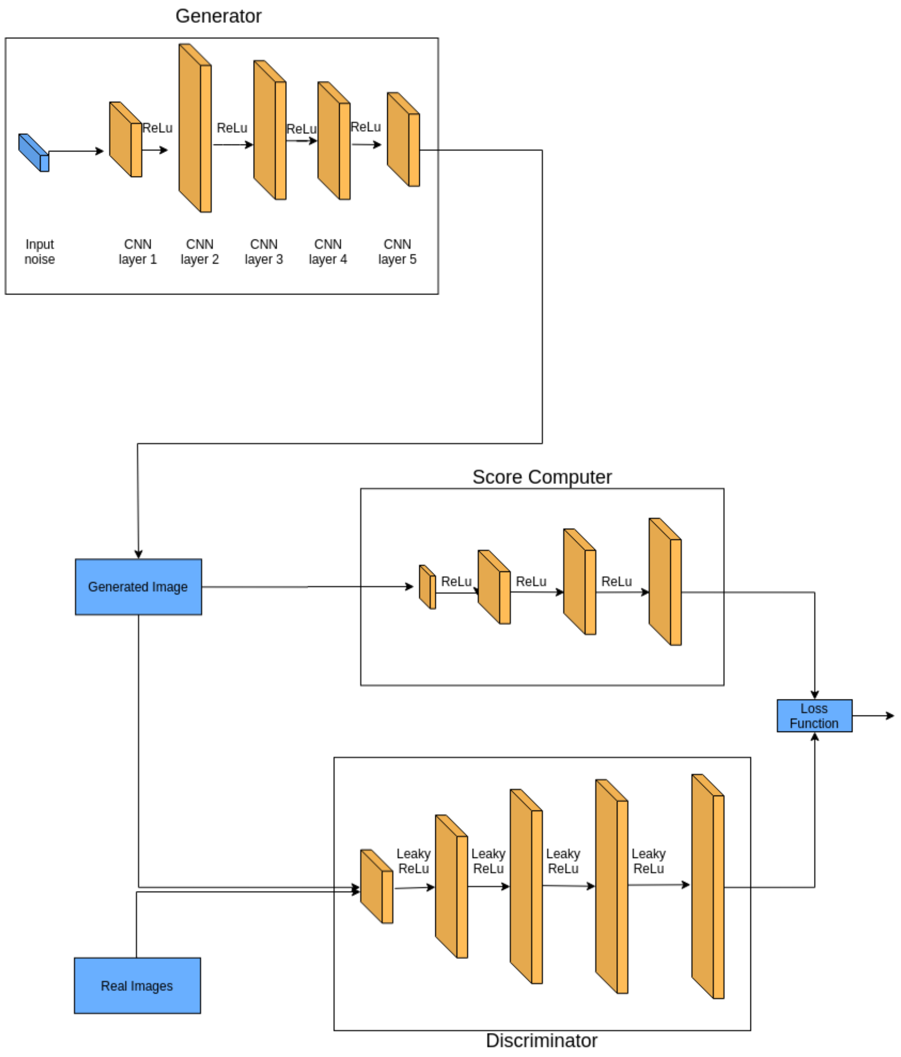
However the CDCGAN was not able to sufficiently capture the structures of the images and generated images as shown below.



It can be seen form the image above that the model is not able to use the extra information given by us.

RegGAN

The RegGAN architecture consists of two discriminator and a single generator. The second discriminator is used to simulate the score function, which was designed by us. The first discriminator is used to differentiate between the images generated by the network and the ones from the dataset. The architecture can be seen below.



Pre-Training of Discriminator

We pre-train the discriminator to learn the score function. This is done so that the second discriminator has a good starting point for the actual training of the network. The discriminator is a CNN with four layers.
During pre-training we feed in the images from the data set to the network. The outputs from the network are then compared to the actual scores given by the score function. The results from the training of the network are shown below. *Pre-training helps the model to get to a good starting point.* Once the discriminator has converged close enough to the score function we freeze the weights of the model. We do so because we want to use the second discriminator, as a pseudo for the score function. For other applications, where the penalty function should evolve with the data the weights of the discriminator can evolve with the training of the generator.

Training RegGan

The Generator and discriminator are also CNN both with 5 layers. We feed in noise to the generator and get outputs as images. These images are then fed into both the discriminators to compute the score and to compare it to the images of the actual data set.

We have two ways in which we back-propagate. In the first one we freeze the weights of the second discriminator and the gradient is only propagated through the the generator and the first discriminator. In the second method we pass the gradient through the second discriminator as well.

A sample of the images generated buy the network can be seen below.