Generative Adversarial Networks : Understanding and Scoring

Lucas Swiniarski
lucas.swiniarski@nyu.edu

Emily Denton
denton@cs.nyu.edu

Rob Fergus
fergus@cs.nyu.edu

Abstract

This project aims at visualizing how a generative adversarial network learns and automatically assess how well it fits the true distribution of the data. With recent advances, it is possible to learn a generator conditioned on a label. We can evaluate the performance of a generator based on how well a classifier learned with generated images generalized on unseen data.

1 Visualizing the learning of a GAN

In order to visualize the learning process of a GAN, we can work on a very simple setup of Gaussian mixtures with two dimensions. I have chosen to learn a Gaussian mixture with one Gaussian with mean $0.5, 0.5$, and one Gaussian with mean $-0.5, -0.5$. The standard deviation is set to 0.1. This experiment allows us to visualize probability density functions as well as the values taken by the discriminator on our input space. Figure 1 displays the real density, 2 shows a well-approximated density by a GAN. We can see that the GAN struggles to get the shape of the pdf right, even though there is no mode dropping. Figure 3 shows a GAN with the Jensen-Shannon divergence minimization. We see the struggles that comes with having a non-Lipschitz discriminator, there is a substantial part of the input space with 0 gradient for the generator, thus there is no way for the generator to learn one part of the distribution. Finally we can see how Wasserstain GAN handles such situation : The discriminator has gradient everywhere, allowing the generator to learn smoothly.

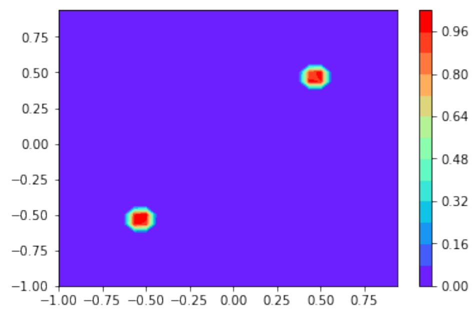

Figure 1: Left : Real distribution pdf.

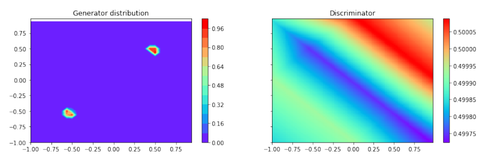

Figure 2: Left : Generated Distribution. Right : Discriminator values.

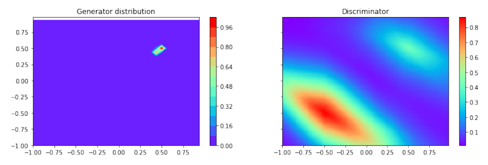

Figure 3: Left - Center Left : DC-GAN generated pdf and discriminator.

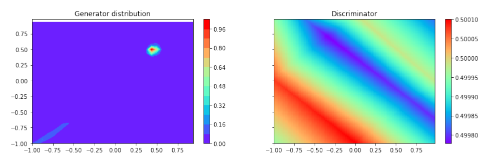

Figure 4: Center Right - Right : W-GAN learning process.

2 Automatic scoring of a learned distribution

There are various ways to assess how well a GAN has learned the distribution. Perhaps the most common way is using the inception score. This score is essentially generating samples and looking at the confidence score an inception network trained on real data gives to a given class. This is helpful to assess the quality of a generated sample, but we can still drop modes of our distribution without impacting the inception score. Another metric consists of optimizing the latent space to generate a given image, then looking at the likelihood of the found latent vector and the reconstruction loss. This is an exhaustive and computationally expensive task as well as depending on optimization errors. I propose a method learning an Auxiliary classifier GAN with a loss and other hyper-parameters of our choice. We also define a fixed architecture and learning procedure for a classifier on the input space. For instance, our input space can be MNIST digits, the labels being the digits values. With a learned generator, we can train the classifier on generated conditional samples. The classifier's accuracy on a test set will give us how well our generator approximated the true data distribution. If the generator is able to learn the true distribution we should get no error on the test set. In reality we always get higher generalization error with a classifier trained on generated samples than a classifier trained on the train set. We could estimate how many unique examples were generated by looking at the generalization error, and interpolating with the network trained on a train-set of various size.

training set	98.68%
DC-GAN	67.7 %
W-GAN	97.55%

The downsides of this method are :

- MNIST is a fairly easy dataset to train a classifier on. As such, very quickly the accuracy difference between models are close.
- Besides MNIST today, there is no image datasets learnable by a GAN that is good enough to be used to train a classifier with descent accuracy on.
- Assessing the effect of learning a conditional GAN on the quality of generated samples is no trivial task.