MODE-SEPARATION GENERATIVE ADVERSARIAL NETWORK

Onkar Salvi
New York University

Raj Mehta
New York University

ABSTRACT

In this paper, we have proposed a variation of Generative Adversarial Network which aims to solve the mode collapsing problem in Generative Adversarial Networks, by using multiple distributions to replicate the distribution of training data. The idea we used is extremely simple, yet very effective. We have used multiple generators that learn different modes in the distribution of the given data. Thus, our model has multiple generators and the mixture of the distribution of all the generators is approximately equal to the distribution of the training data. The distinguishing factor about our network is that we have used a pre-trained classifier, along with the Discriminator, which forces each generator to learn different distributions. Our framework is generalizable as it can be easily combined with other existing variants of GANs to produce diverse samples. We conducted experiments on synthetic data and real world data and compared our results to the traditional GAN framework.

INTRODUCTION

Since the introduction of Generative Adversarial Networks (Ian Goodfellow, n.d.), they have been widely used for generative learning applications such as image generation, text generation, image-to-image translation and image inpainting, to name a few. The framework consists of a discriminator and a generator which play a two-player minimax game, where the generator tries to fool the discriminator by producing samples which resemble those in the training data, while the discriminator tries not to be fooled by training itself to distinguish between the samples from the training data and samples from the generator. What this does is, over a few epochs, the generator learns to produce more realistic images by changing its weights as per the loss of the discriminator.

Training a GAN, however, is a very challenging task. Most of the real-world data distributions are multimodal i.e. the probability distribution which describes the data has multiple peaks, where different subgroups of samples are concentrated. For example, suppose we have a data of weekly calorie consumption of people who work out regularly and people who do not workout at all. This distribution of data will be bimodal. There will be two peaks near the average of the calorie consumption of people belonging to each group. Now suppose we want to train a GAN which generates the possible values of calorie consumption using this data. We would expect the generator to produce calorie consumption values of both types of people with equal probability. However, there is a commonly encountered issue, where the generator will only output samples from a single mode. To understand why, consider the following situation; The generator learns that it can fool the discriminator into thinking that it is outputting realistic values by producing values of people who work out. The discriminator counter by learning that all values of people who do not work out are real and essentially guesses whether values of people who work out are real or fake since they are indistinguishable. The generator exploits the discriminator by switching modes by producing values close to the people who do not workout instead, abandoning the values from the mode of people who work out. The discriminator now assumes that all values of people who do not workout are fake and the values of people who work out are real. This keeps on repeating, with the generator never being able to cover both the modes. This issue in the training of GAN is called mode collapsing.

Our proposed Network, which we would like to term MS-GAN (Mode-Separation Generative Adversarial Network) will not only be able to overcome the mode collapse problem by producing samples from all the modes but will also be able to separately produce samples from individual modes.

RELATED WORK

The most recent work in dealing with mode collapse issues is the Multi-Agent Diverse Generative Adversarial Networks (Arnab Ghosh, 03/09/2018), that proposes a frameworks where they use multiple generators and one discriminator, wherein the discriminator has to identify if the data is for the true or fake distribution and simultaneously also identify the generator that generates the fake sample.

In generative adversarial networks(GAN's), we try to reduce the distance between the two distributions i.e. The original distribution P_r and the distribution produced by the generator P_g. There is a very interesting approach directed in the Wasserstein GAN's paper(Martin Arjovsky & Bottou, n.d.) wherein, they show that certain sequence of distributions do not tend to converge with the Kullback-Liebler, Jenson Shannon or reverse Kullback-Liebler divergence, but do converge with the earth mover's or wassertein distance. In their paper they have proposed an algorithm that solves the mode collapse problem using the Earth Movers (EM) distance.

Another interesting approach to control mode collapse was introduced in the paper VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning(Akash Srivastava, n.d.). Here the authors used a technique in which they introduce an additional reconstructor into the network that reverses the action of the generator by mapping from data to noise. The intuition behind their approach is that, if the reconstructor learns both to map all of the true data to the noise distribution and is an approximate inverse of the generator network, this will encourage the generator network to map from the noise distribution to the entirety of the true data distribution, thus resolving mode collapse.

A few approaches have also been done using autoencoders(Rosca, n.d.), GAN's finds a sub-basis of the latent space, and then take random samples from this sub-basis. We train a GAN to generate a batch of vectors and enforce that they are orthogonal using their dot product, and then take random linear combinations of these vectors. The discriminator then decides whether these linear combinations are convincing latent space encodings. Those that fool the discriminator get decoded into realistic samples.

In this paper, we propose a new framework to solve this mode collapse problem using multiple generators and enforcing each generator to learn a different mode.

PROPOSED MODE-SEPARATION GAN

We now propose our model in which we effectively tackle mode collapse problem. Our main idea is to use mixture of multiple distributions to replicate the distribution of the training data, where each generated distribution covers a single mode.

To implement this, we use multiple generators with a single discriminator and a pre-trained classifier. Intuitively, we can think of our model as the generators analogous to counterfeiters trying to forge a currency distribution without detection and the discriminator as analogous to police, trying to detect this fake currency and the classifier forcing each individual generator in the group to learn a different currency valuation. Thus, a minimax game is established between the generators and the discriminators, with the classifier aiding each generator to learn a different mode of the distribution of the training data.

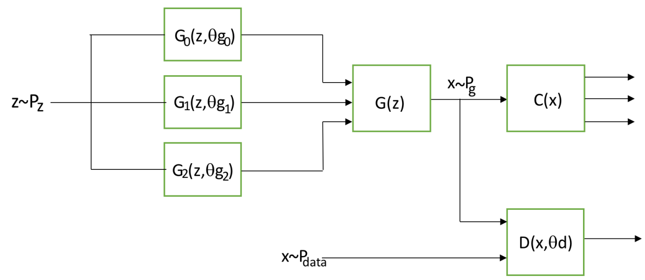

Figure 1. MSGAN Network Architecture.

More formally, Consider we have N generators $G_{1:N}$, A discriminator D and a Classifier C. Each Generator G_n maps z to $G_{n(z)}$, with distribution P_{G_n} and then the equally sampled mixture of N generators together forms the distribution P_G. The Sample G(z) is used as output of the generators. The Discriminator D tries to classify between these samples and the samples from the true distribution Pdata. The pre-trained classifier C performs multiclass classification to classify samples labelled by the indices of their corresponding generators. This forces each generator to create samples form a different distribution.

In other words, $G_{1:N}$, D and C play the following minimax game with value function $V(D,C,G_{1:N})$

.

$$\min_{G_{1:N}} \max_{D} V(D, C, G_{1:N}) = \mathbb{E}_{x\sim P_{\text{data}}}[logD(x)]$$
$$+\, \mathbb{E}_{z\sim P_Z}\, log[1 - D(G(z))] + \textstyle\sum_{i=1}^{N} \mathbb{E}_{z\sim P_Z}[logC_{\text{n}}(G(z))]$$

Algorithm 1 Minibatch stochastic gradient descent training of mode separation generative adversarial nets. Here, we choose the number k = 1, which is the number of time we train the discriminator in an epoch.

for number of iterations **do**

Sample minibatch of m examples $x^{(1)}, \ldots, x^{(m)}$ from data distribution $P_{data}(x)$.

Update the Classifier C by descending along its gradients.

for number of iterations **do**

for k Steps **do**

Sample minibatch of m examples $x^{(1)}, \ldots, x^{(m)}$ from data distribution $P_{data}(x)$.

Sample minibatch of m noise samples $z^{(1)}, \ldots, z^{(m)}$ from the noise prior $P_g(z)$ and distribute the m noise samples to the N generators

Update the Discriminator by ascending its Stocastic Gradient

$$\nabla_{\theta d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log(1 - D(G(z^{(i)}))]$$

Sample minibatch of m noise samples $z^{(1)}, \ldots, z^{(m)}$ from the mixture generator samples G(z) with equal ratio from all the generators

Update the n Discriminator by ascending along its gradients

$$\nabla_{\theta gn} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)}))$$

Update each generator individually by descending along its gradients

$$\nabla_{\theta gn} \frac{1}{K} \sum_{i=1}^{K} \log(C(G_n(z^{(i)}))$$

RESULTS

We conducted experiments on the both real world and synthetic data. For synthetic data, we created a mixture of three gaussian signals with mean at 10, 40 and 70 and each with a standard deviation of 1. We first trained a traditional GAN using this data. The mode collapse problem was observed when training with the traditional GAN. Figure 1 compares the distribution of the data sampled by the generator of the GAN to the distribution of the real data. Next, we trained this with MSGAN, The MSGAN was able to generate data from all the modes, thus solving the mode collapse problem. Figure 2 compares the distribution of the data sampled from the mixture of the distribution of all the generators with the distribution of the real data.

For real world data, we used the MNIST dataset. Due to computational limitations we considered only three numbers from the MNIST dataset and we trained our network using three generators to recreate these three numbers and a discriminator and a pre-trained classifier respectively. After training the network, each generator was able to produce a different digit from the mnist dataset. Figure 4, Figure 5 and Figure 6 show the digits produced by generator 1, generator 2 and generator 3 respectively. The mixture of the distribution of all the three generators will be equal to the distribution of the training data.

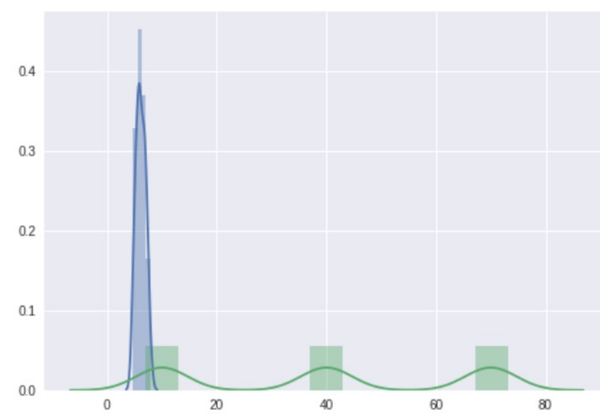

Figure 2. Mode collapse observed when trained by a normal GAN. Training data is in green and data produced by the generator is in blue.

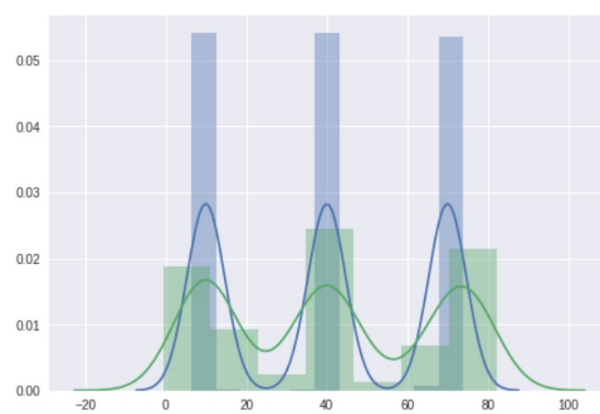

Figure 3. All modes are learnt by the MSGAN model. Training data is in blue and data produced by the mixture of all the generators is in green.

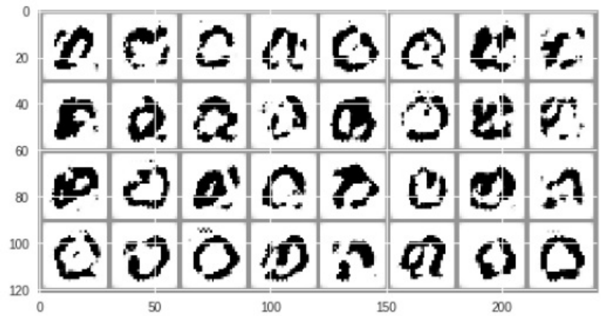

Figure 4. zeros produced by generator 1.

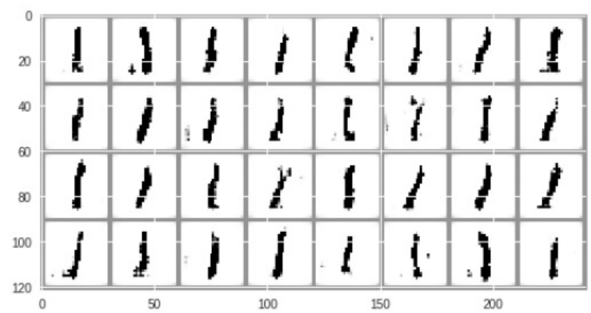

Figure 5. ones produced by generator 2.

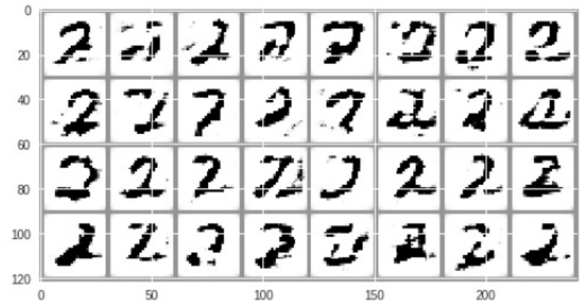

Figure 6. twos produced by generator 3.

References

Akash Srivastava, C. R. M. G., Lazar Valkov. (n.d.). "veegan: Reducing mode collapse in gans using implicit variational learning".

Arnab Ghosh, V. K. (03/09/2018). Multi-agent diverse generative adversarial networks.

Ian Goodfellow, M. M. Y. B., Jean Pouget-Abadie. (n.d.). "generative adversarial nets".

Martin Arjovsky, S. C., & Bottou, L. (n.d.). "wasserstein gan".

Rosca, M. (n.d.). "solving mode collapse with auto-encoder gans".