DISCRIMINATOR REJECTION SAMPLING

ICLR 2019

Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, Augustus Odena

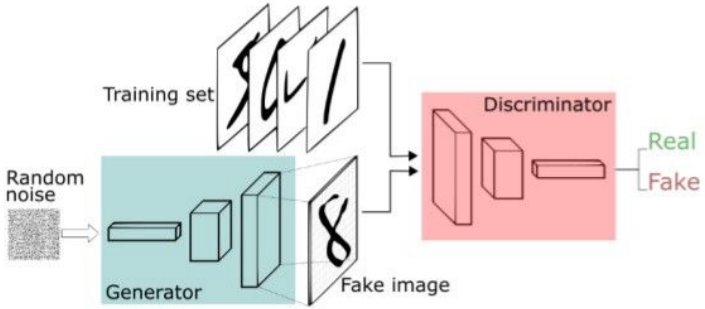
Abstract

- A **rejection sampling scheme** using the discriminator of a GAN to approximately correct errors in the GAN generator distribution
- under quite strict assumptions, this will allow us to recover the data distribution exactly
- **practical algorithm**—called Discriminator Rejection Sampling (DRS)— for real life case where the strict assumption don't hold.

Background - GAN

Generative Adversarial Networks (GANs)

GAN training procedure pits two neural networks against each other, a generator and a discriminator.



$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

log prob of D predicting that real-world data is genuine

log prob of D predicting that G's generated data is not genuine

$$\begin{aligned} L_D &= -\mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] - \mathbb{E}_{z\sim p_z}[1 - \log D(G(z))] \\ L_G &= -\mathbb{E}_{z\sim p_z}[\log D(G(z))] \end{aligned}$$

Background - INCEPTION SCORE

- Relevant for labeled datasets

- Motivation :
  Assume you have a classifier trained on real data P(y|x)
  we want:
  - P(y) to be uniform (high diversity of generated images) - high entropy
  - P(y|G(z)) to be confident - low entropy

$$\mathrm{IS} = \exp\left(\mathbb{E}_{p_G}[KL(p(y\,|\,x)\,||\,p_\theta(y)]\right)$$

Background - FRECHET´ INCEPTION DISTANCE (FID)

- Look at features using pertained classifier
- Measure distance using Fréchet distance (2-Wasserstein) - optimal transport

$$X_g \sim \mathcal{N}(\mu_g, \Sigma_g)$$
$$X_r \sim \mathcal{N}(\mu_r, \Sigma_r)$$

$$\text{FID} = ||\mu_r - \mu_g||^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

- Lower FID is better, corresponding to more similar real and generated samples

Background - SELF-ATTENTION GAN

- SAGAN differs from a vanilla GAN in the following ways:
  - Uses large residual networks
  - spectral normalization (Miyato et al., 2018) in the generator and the discriminator
  - much lower learning rate for the generator
  - SAGAN makes use of self-attention layers
  - whole model is trained using a special hinge version of the adversarial loss

$$
\begin{aligned}
L_D = & -\mathbb{E}_{(x,y)\sim p_{data}}[\min(0, -1 + D(x,y))] \\
& -\mathbb{E}_{z\sim p_z, y\sim p_{data}}[\min(0, -1 - D(G(z),y))], \\
L_G = & -\mathbb{E}_{z\sim p_z, y\sim p_{data}} D(G(z),y),
\end{aligned}
$$

Background - Rejection Sampling

- Sampling from $P_d(x)$ is too hard
- Samples are instead drawn from a proposal distribution $p_g(x)$, which is easier to sample from and which is chosen such that there exists a finite value M such that $M * P_g(x) > P_d(x)$ for $\forall x \in \text{domain}(P_d(x))$.

Algorithm:

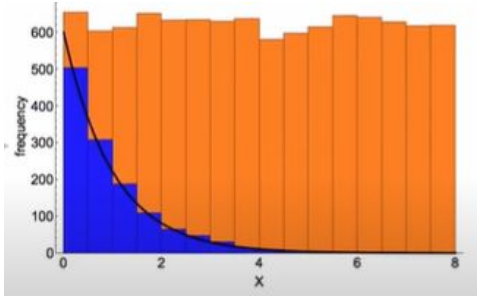
1. Sample U~Unif (0, 1)
2. Sample x from P_g
3. If $U <= P_d(x) / M * P_g(x)$ then Accept x else reject

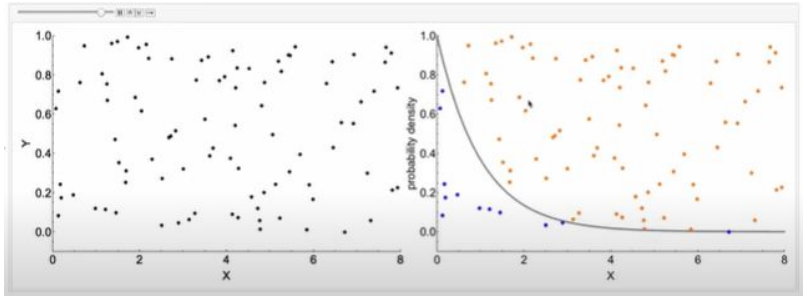
.

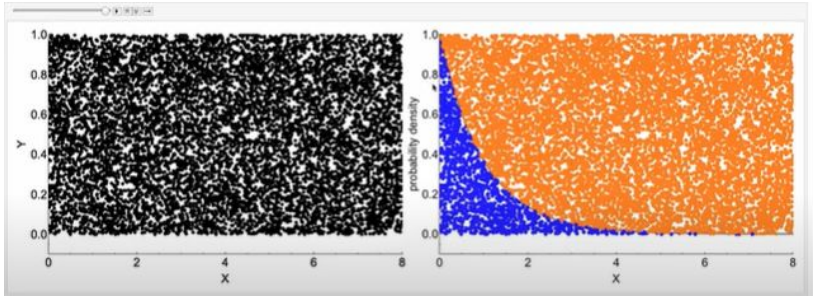
Background - Rejection Sampling Example

$P_d \sim$ Exp (1) $\Rightarrow$ $P_d(x) = e^{-x}$

$P_g \sim$ Uniform [0, 8]



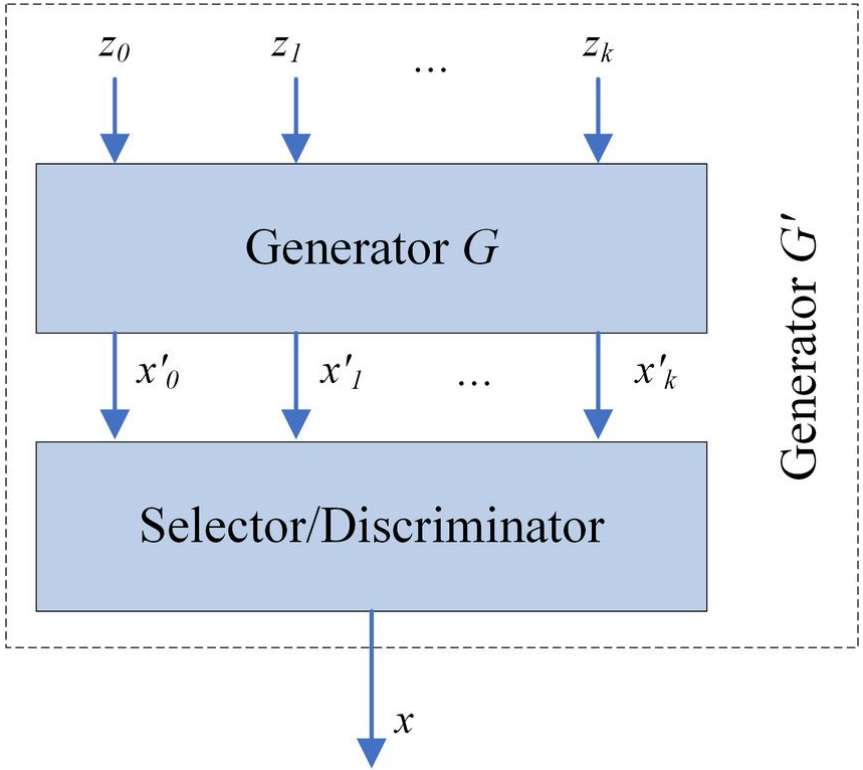




Introduction

Generative Adversarial Networks (GANs)

Instead of trying to improve the training procedure, we (temporarily) accept its flaws and attempt to improve the quality of trained generators by **post-processing** their samples using information from the trained discriminator



REJECTION SAMPLING FOR GANS

THE IDEALIZED VERSION

- P_g and P_d have the same support.
  That is, for all $x \in X$, $P_g(x) != 0$ if and only if $P_d(x) != 0$
- Can compute somehow $P_d(x)/P_g(x)$.
  Then, if $M = MAX_x P_d(x)/P_g(x)$ then $M*P_g(x) > P_d(x)$ for all x.
  In this case, we can exactly sample from P_d.

But how do we compute $P_d(x) / M P_g(x)$?

REJECTION SAMPLING FOR GANS

THE IDEALIZED VERSION

Optimize D in the density function space instead of the parameters space.

Also, Suppose D defined by as Sigmoid activated on a function of x and train by cross-entropy loss.

$$D(x) = \sigma(x) = \frac{1}{1 + e^{-\widetilde{D}(x)}}$$

Then, Given fixed G, can be shown that D that's minimizing D's loss is::

$$D^*(x) = \frac{p_d(x)}{p_d(x) + p_g(x)}$$

REJECTION SAMPLING FOR GANS

THE IDEALIZED VERSION

$$
\begin{aligned}
D^*(x) = \frac{1}{1 + e^{-\widetilde{D}^*(x)}} &= \frac{p_d(x)}{p_d(x) + p_g(x)} \\
1 + e^{-\widetilde{D}^*(x)} &= \frac{p_d(x) + p_g(x)}{p_d(x)} \\
p_d(x) + p_d(x)e^{-\widetilde{D}^*(x)} &= p_d(x) + p_g(x) \\
p_d(x)e^{-\widetilde{D}^*(x)} &= p_g(x) \\
\frac{p_d(x)}{p_g(x)} &= e^{\widetilde{D}^*(x)}
\end{aligned}
$$

REJECTION SAMPLING FOR GANS

THE IDEALIZED VERSION

Now, suppose one last thing:

We can tractably compute $M = \text{MAX}_x P_d(x)/P_g(x)$.
Then we would find that

$$M = p_d(x^*)/p_g(x^*) = e^{\widetilde{D}^*(x^*)}$$

for some (not necessarily unique) x*

Given all these assumptions:

Define $\widetilde{D}^*_M := \widetilde{D}^*(x^*)$ Then $p_d(x)/Mp_g(x)$ can be written as $e^{\widetilde{D}^*(x)-\widetilde{D}^*_M} \in [0,1]$

REJECTION SAMPLING FOR GANS

THE IDEALIZED VERSION

Algorithm:

1. Sample ψ ~ Uniform[0,1]
2. Sample x from P_g
3. if $\psi < e^{\widetilde{D}^*(x) - \widetilde{D}^*_M}$ then Accept x else reject

REJECTION SAMPLING FOR GANS

THE PRACTICAL SCHEME - Difficulties

1. Can’t actually perform optimization over density functions. i.e can’t compute D* . Thus, our acceptance probability won’t necessarily be proportional to $P_d(x)/P_g(x)$.

2. Especially On large datasets, **different support** and the intersection can be **low volume**, i,e $P_d(x)/P_g(x)$ would just evaluate to 0 in most places..

3. Can't draw infinite sample from P_d to calc D^*.

4. In general it won’t be tractable to compute M.

5. Rejection sampling is known to have too low an acceptance probability when the target distribution is high dimensional

REJECTION SAMPLING FOR GANS

THE PRACTICAL SCHEME

On the difficulty of actually computing D*

- Given 2,3 perhaps we shouldn’t make the effort to compute D* and those should not be worry about 1.
- Train regularized D with SGD will not achieve D* but at least won't cause overfitting to the finite number of samples from P_d used in training.

REJECTION SAMPLING FOR GANS

THE PRACTICAL SCHEME

On the difficulty of actually computing M

- Nontrivial to compute M, especially when D* can't be computed.
- In practice:
  - Step 1: Estimation phase - sample and estimate D^*_M
  - Step 2: Sampling phase - run Rejection Sampling given the known maximum
    - if new maximum found, update it for the next samples

- Changing the maximum on the fly, changes the acceptance prob' for the same sample, is it a problem?

REJECTION SAMPLING FOR GANS

THE PRACTICAL SCHEME

Dealing with acceptance probabilities that are too low

- Item 5 suggests that we may end up with acceptance probabilities that are too low to be useful when performing this technique on realistic data-sets.

- If D^*_M is very large, the acceptance probability exp($D^*(x) - D^*_M$) will be close to zero, and almost all samples will be rejected, which is undesirable.

- Instead compute F(X) $$\frac{1}{1+e^{-F(x)}} = e^{\widetilde{D}^*(x) - \widetilde{D}^*_M}$$

REJECTION SAMPLING FOR GANS

THE PRACTICAL SCHEME

Dealing with acceptance probabilities that are too low

$$
\begin{aligned}
F(x) &= \widetilde{D}^*(x) - \log(e^{\widetilde{D}^*_M} - e^{\widetilde{D}^*(x)}) \\
&= \widetilde{D}^*(x) - \log(\frac{e^{\widetilde{D}^*_M}}{e^{\widetilde{D}^*_M}} e^{\widetilde{D}^*_M} - \frac{e^{\widetilde{D}^*_M}}{e^{\widetilde{D}^*_M}} e^{\widetilde{D}^*(x)}) \\
&= \widetilde{D}^*(x) - \widetilde{D}^*_M - \log(1 - e^{\widetilde{D}^*(x) - \widetilde{D}^*_M})
\end{aligned}
$$

In practice: $\hat{F}(x) = \widetilde{D}^*(x) - \widetilde{D}^*_M - \log(1 - e^{\widetilde{D}^*(x) - \widetilde{D}^*_M - \epsilon}) - \gamma$

γ is a hyperparameter modulating overall acceptance probability. For very positive, all samples will be rejected. For very negative , all samples will be accepted.

REJECTION SAMPLING FOR GANS

THE PRACTICAL SCHEME

Algorithm

Data: generator **G** and discriminator **D**
Result: Filtered samples from **G**
$D^* \leftarrow$ `KeepTraining(`D`);`
$\bar{M} \leftarrow$ `BurnIn(`$\boldsymbol{G}$, D^*`);`
$samples \leftarrow \emptyset$;
while $|samples| < N$ **do**
  $x \leftarrow$ `GetSample(`$\boldsymbol{G}$`);`
  $ratio \leftarrow e^{\tilde{D}^*(x)}$;
  $\bar{M} \leftarrow$ `Maximum(`$\bar{M}$, $ratio$`);`
  $p \leftarrow \sigma(\hat{F}(x, \bar{M}, \epsilon, \gamma))$;
  $\psi \leftarrow$ `RandomUniform(0,1);`
  if $\psi \leq p$ **then**
    `Append(`X, $samples$`);`
  end
end

Experiments

MIXTURE OF 25 GAUSSIANS

mixture of twenty-five 2D isotropic Gaussian dist' (each with standard deviation of 0.05) arranged in a grid

- Generator and Discriminator are NN with 4 FC with ReLu activations

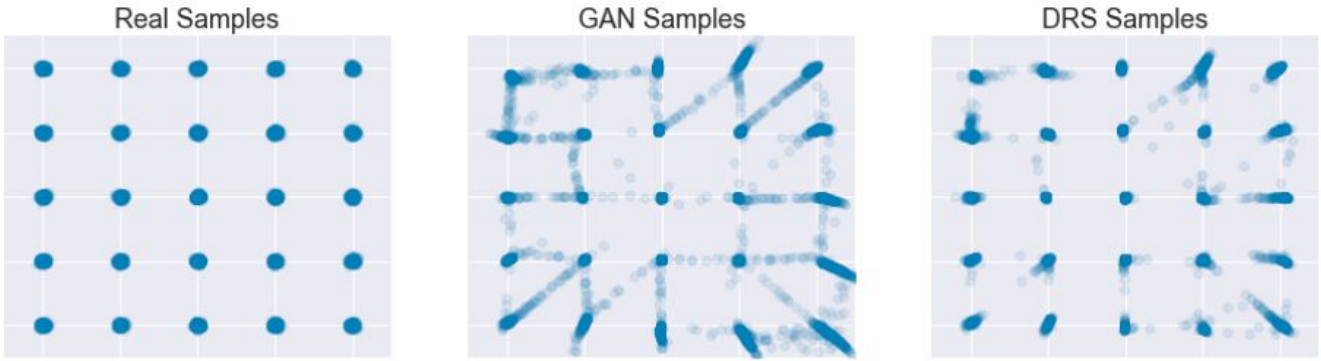


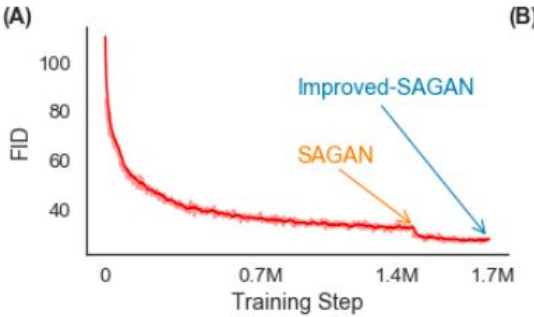
Table 1: Results with and without DRS on 10,000 generated samples from a model of a 2D grid of Gaussian components.

	# of recovered modes	% "high quality"	std of "high quality" samples
Without DRS	24.8 ± 0.4	70 ± 9	0.11 ± 0.01
With DRS	24.8 ± 0.4	$\mathbf{90 \pm 2}$	$\mathbf{0.10 \pm 0.01}$

Experiments

Improved SAGAN

-



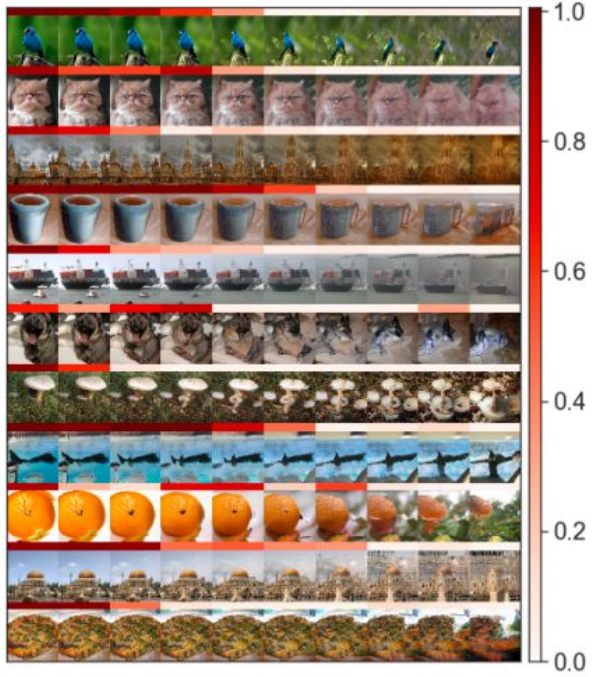


Table 2: Results with and without DRS on 50K ImageNet samples. Low FID and high IS are better.

	SAGAN		Improved-SAGAN	
	IS	FID	IS	FID
Without DRS	52.34 ± 0.45	18.21 ± 0.14	62.36 ± 0.35	14.79 ± 0.06
With DRS	61.44 ± 0.09	17.14 ± 0.09	$\mathbf{76.08 \pm 0.30}$	$\mathbf{13.57 \pm 0.13}$

MH-GAN (Metropolis-Hastings GAN)

Link