Wasserstein GAN

Changzhi Sun

East China Normal University

changzhisun@stu.ecnu.edu.cn

April 27, 2017

Outline

Introduction

Generative Model

Real distribution P_r, approximately distribution P_θ

1. Directly learn the prob density function P_θ
   - P_θ is some differentiable function such that $P_\theta(x) \geq 0$ and $\int P_\theta(x)dx = 1$
   - Optimize P_θ through MLE
2. Learn a function that transforms an existing distribution Z into P_θ
   - g_θ is some differentiable function, Z is a common distribution (usually uniform or Gaussian), and $P_\theta = g_\theta(Z)$

$$\max_{\theta \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} \log P_\theta(x^{(i)})$$

- In the limit, this is equivalent to minimizing the KL-divergence $KL(P_r||P_\theta)$

Why Is This True

$$KL(P||Q) = \int_x P(x) \log \frac{P(x)}{Q(x)} \, dx$$

In the limit

$$\begin{aligned}
\lim_{m\to\infty} \max_{\theta\in\mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} \log P_\theta(x^{(i)}) &= \max_{\theta\in\mathbb{R}^d} \int_x P_r(x) \log P_\theta(x) \, dx \\
&= \min_{\theta\in\mathbb{R}^d} - \int_x P_r(x) \log P_\theta(x) \, dx \\
&= \min_{\theta\in\mathbb{R}^d} \int_x P_r(x) \log P_r(x) \, dx - \int_x P_r(x) \log P_\theta(x) \, dx \\
&= \min_{\theta\in\mathbb{R}^d} KL(P_r \| P_\theta)
\end{aligned}$$

Why Is This True

- If $Q(x) = 0$ at an x where $P(x) > 0$, the KL divergence goes to $+\infty$
- This is bad for the MLE if P_θ has low dimensional support, because it'll be very unlikely that all of P_r lies within that support
- If even a single data point lies outside P_θ's support, the KL divergence will explode

Solution: add random noise to P_θ when training the MLE

- This ensures the distribution is defined everywhere
- Even if we learn a good density P_θ, it may be computationally expensive to sample from P_θ

Introduction

- Learning a g_θ (a generator) to transform a known distribution Z
- Given a trained g_θ, simply sample random noise $z \sim Z$, and evaluate $g_\theta(z)$
- we dont explicitly know what P_θ, but in practice this isn't that important

To train g_θ, we need a measure of distance between distributions

- Different metrics (different definitions of distance) induce different sets of convergent sequences
- Distance d is weaker than distance d', if every sequence that converges under d' converges under d
- Given a distance d, we can treat $d(P_r, P_\theta)$as a loss function
- Mapping $\theta \mapsto P_\theta$ is continuous (which will be true if g_θ is a neural net)

Different Distances

- The Total Variation (TV) distance is

$$\delta(P_r, P_g) = \sup_A |P_r(A) - P_g(A)|$$

- The Kullback-Leibler (KL) divergence is

$$KL(P_r \| P_g) = \int_x \log\left(\frac{P_r(x)}{P_g(x)}\right) P_r(x)\, dx$$

  This isn't symmetric. The reverse KL divergence is defined as $KL(P_g \| P_r)$.

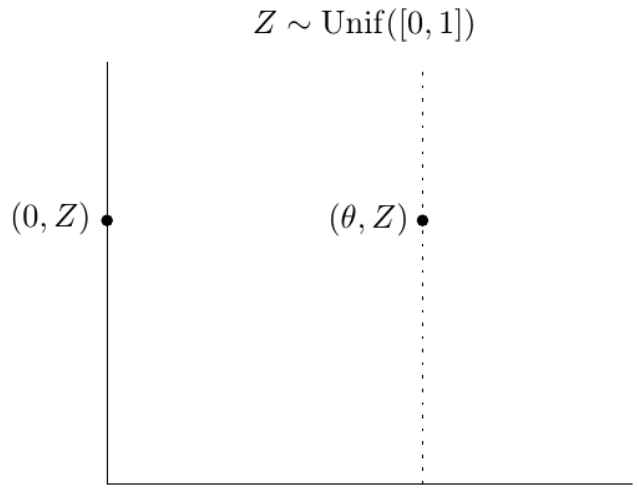
- The Jenson-Shannon (JS) divergence: Let M be the mixture distribution $M = P_r/2 + P_g/2$ Then

$$JS(P_r, P_g) = \frac{1}{2} KL(P_r \| P_m) + \frac{1}{2} KL(P_g \| P_m)$$

- Finally, the Earth Mover (EM) or Wasserstein distance: Let $\Pi(P_r, P_g)$ be the set of all joint distributions γ whose marginal distributions are P_r and P_g. Then.

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[\, \|x - y\| \,\right]$$

Different Distances



- $\mathrm{KL}(\mathbb{P}_\theta \| \mathbb{P}_0) = \begin{cases} \infty & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases}$.
- $\mathrm{JS}(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases}$.
- $\delta(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} 1 & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases}$.
- $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$.

Different Distances

- There exist sequences of distributions that don't converge under the JS, KL, reverse KL, or TV divergence, but which do converge under the EM distance
- For the JS, KL, reverse KL, and TV divergence, there are cases where the gradient is always 0
- This is a contrived example because the supports are disjoint

Different Distances

Theorem 1. *Let $\mathbb{P}_r$ be a fixed distribution over $\mathcal{X}$. Let Z be a random variable (e.g Gaussian) over another space $\mathcal{Z}$. Let $g : \mathcal{Z} \times \mathbb{R}^d \to \mathcal{X}$ be a function, that will be denoted $g_\theta(z)$ with z the first coordinate and θ the second. Let $\mathbb{P}_\theta$ denote the distribution of $g_\theta(Z)$. Then,*

1. *If g is continuous in θ, so is $W(\mathbb{P}_r, \mathbb{P}_\theta)$.*
2. *If g is locally Lipschitz and satisfies regularity assumption 1, then $W(\mathbb{P}_r, \mathbb{P}_\theta)$ is continuous everywhere, and differentiable almost everywhere.*
3. *Statements 1-2 are false for the Jensen-Shannon divergence $JS(\mathbb{P}_r, \mathbb{P}_\theta)$ and all the KLs.*

Different Distances

Theorem 2. *Let $\mathbb{P}$ be a distribution on a compact space $\mathcal{X}$ and $(\mathbb{P}_n)_{n\in\mathbb{N}}$ be a sequence of distributions on $\mathcal{X}$. Then, considering all limits as $n \to \infty$,*

1. *The following statements are equivalent*
   - *$\delta(\mathbb{P}_n, \mathbb{P}) \to 0$ with δ the total variation distance.*
   - *$JS(\mathbb{P}_n, \mathbb{P}) \to 0$ with JS the Jensen-Shannon divergence.*
2. *The following statements are equivalent*
   - *$W(\mathbb{P}_n, \mathbb{P}) \to 0$.*
   - *$\mathbb{P}_n \xrightarrow{\mathcal{D}} \mathbb{P}$ where $\xrightarrow{\mathcal{D}}$ represents convergence in distribution for random variables.*
3. *$KL(\mathbb{P}_n \| \mathbb{P}) \to 0$ or $KL(\mathbb{P} \| \mathbb{P}_n) \to 0$ imply the statements in (1).*
4. *The statements in (1) imply the statements in (2).*

Wasserstein Distances

Unfortunately, computing the Wasserstein distance exactly is intractable. Let's repeat the definition.

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma} \left[\|x - y\| \right]$$

The paper now shows how we can compute an approximation of this.

A result from Kantorovich-Rubinstein duality shows W is equivalent to

$$W(P_r, P_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_\theta}[f(x)]$$

where the supremum is taken over all 1-Lipschitz functions.

What Does Lipschitz Mean

Let d_X and d_Y be distance functions on spaces X and Y. A function $f : X \to Y$ is K-Lipschitz if for all $x_1, x_2 \in X$,

$$d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2)$$

Intuitively, the slope of a K-Lipschitz function never exceeds K, for a more general definition of slope.

The supremum over K-Lipschitz functions $\{f : \|f\|_L \leq K\}$ is still intractable, but now it's easier to approximate. Suppose we have a parametrized function family $\{f_w\}_{w \in \mathcal{W}}$, where w are the weights and $\mathcal{W}$ is the set of all possible weights. Further suppose these functions are all K-Lipschitz for some K. Then we have

$$\begin{aligned} \max_{w \in \mathcal{W}} \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{x \sim P_\theta}[f_w(x)] &\leq \sup_{\|f\|_L \leq K} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_\theta}[f(x)] \\ &= K \cdot W(P_r, P_\theta) \end{aligned}$$

What Does Lipschitz Mean

1. we don't even need to know what K is!
2. K will get absorbed into the hyperparam tuning(learning rate α)

- Train $P_\theta = g_\theta(Z)$ to match P_r
- Given a fixed g_θ, compute the optimal f_w for the Wasserstein distance

$$\begin{aligned}\nabla_\theta W(P_r, P_\theta) &= \nabla_\theta(\mathbb{E}_{x\sim P_r}[f_w(x)] - \mathbb{E}_{z\sim Z}[f_w(g_\theta(z))]) \\ &= -\mathbb{E}_{z\sim Z}[\nabla_\theta f_w(g_\theta(z))]\end{aligned}$$

Wasserstein GAN

Theorem 3. *Let $\mathbb{P}_r$ be any distribution. Let $\mathbb{P}_\theta$ be the distribution of $g_\theta(Z)$ with Z a random variable with density p and g_θ a function satisfying assumption 1. Then, there is a solution $f : \mathcal{X} \to \mathbb{R}$ to the problem*

$$\max_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)]$$

and we have

$$\nabla_\theta W(\mathbb{P}_r, \mathbb{P}_\theta) = -\mathbb{E}_{z \sim p(z)}[\nabla_\theta f(g_\theta(z))]$$

when both terms are well-defined.

Wasserstein GAN

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

Compare & Contrast: Standard GANs

GANs

1. the discriminator max

$$\frac{1}{m}\sum_{i=1}^{m}\log D(x^{(i)}) + \frac{1}{m}\sum_{i=1}^{m}\log(1 - D(g_\theta(z^{(i)})))$$

   where we constraint $D(x)$ to always be a probability $p \in (0, 1)$
2. in the limit, discriminator objective is JS divergence
3. in practice we we never train D to convergence

WGANs

1. nothing requires f_w to output a probability
2. it is the Wasserstein distance instead
3. should train f_w to convergence before each generator update

Empirical Results

1. a meaningful loss metric that correlates with the generators convergence and sample quality
2. improved stability of the optimization process

Empirical Results

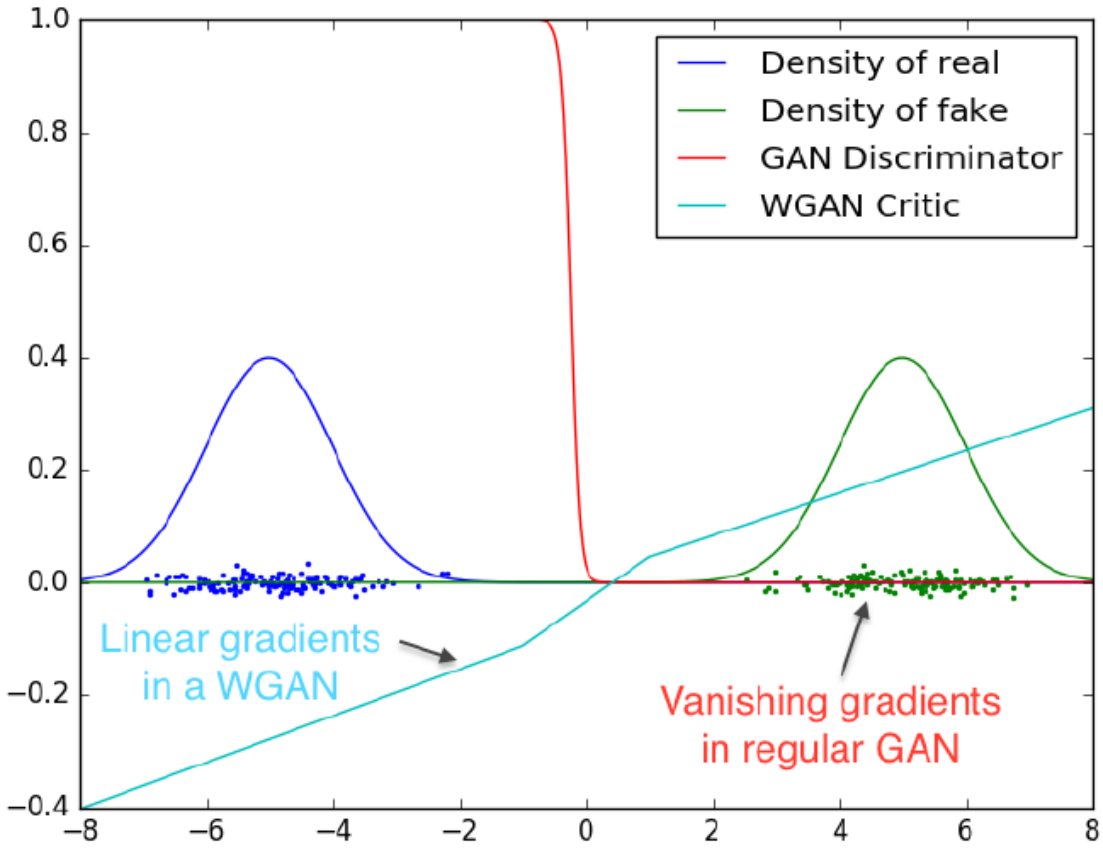
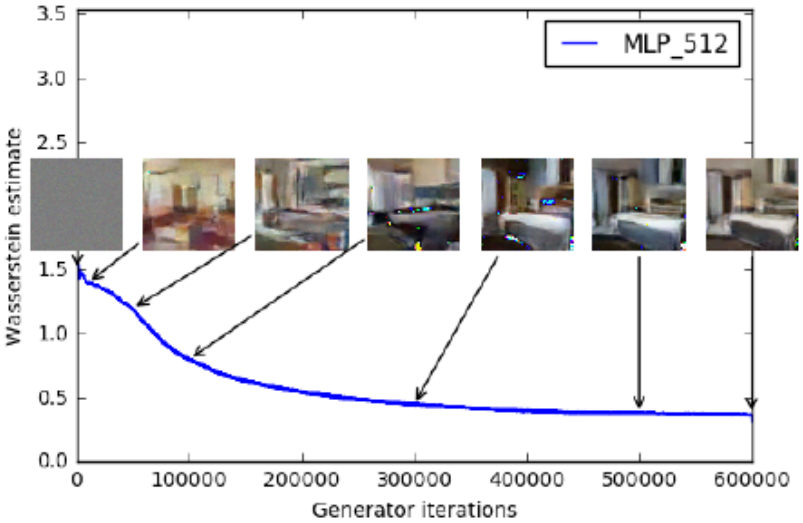
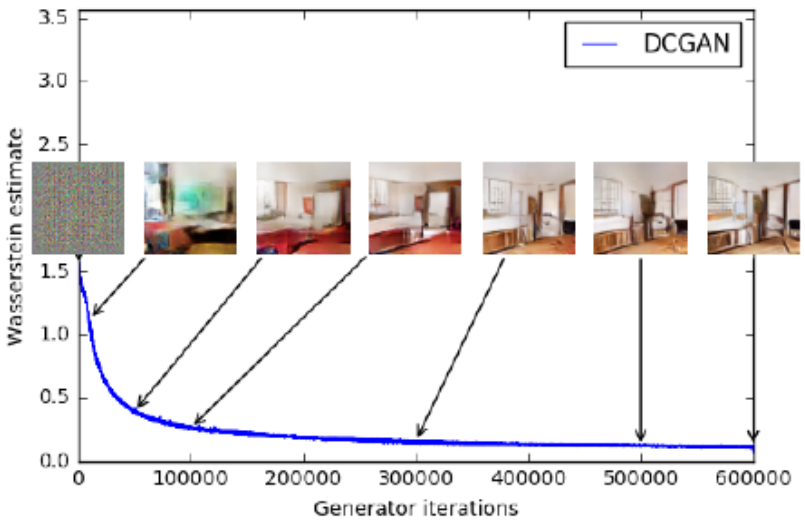
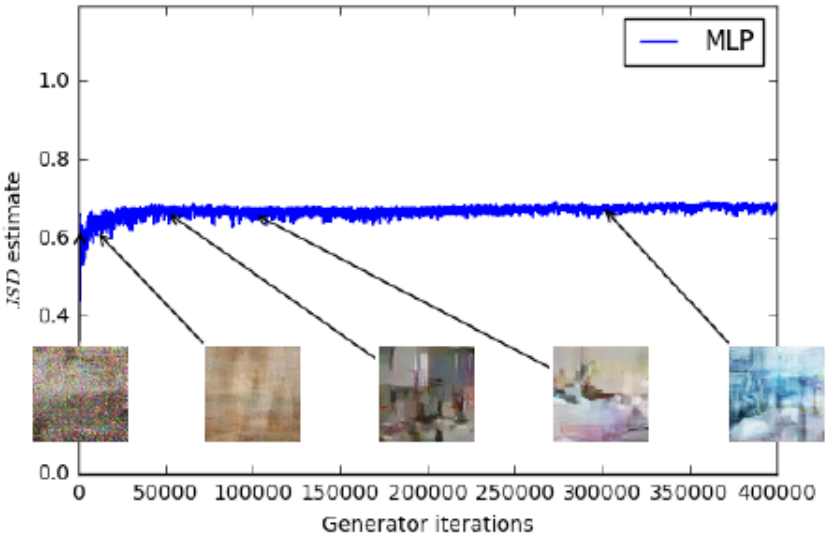


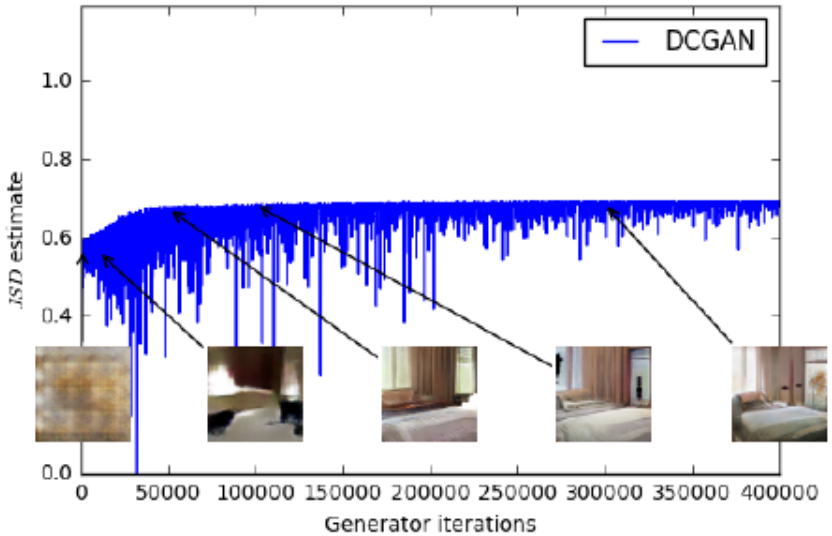
Figure 2: Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the traditional GAN discriminator saturates and results in vanishing gradients. Our WGAN critic provides very clean gradients on all parts of the space.

Empirical Results

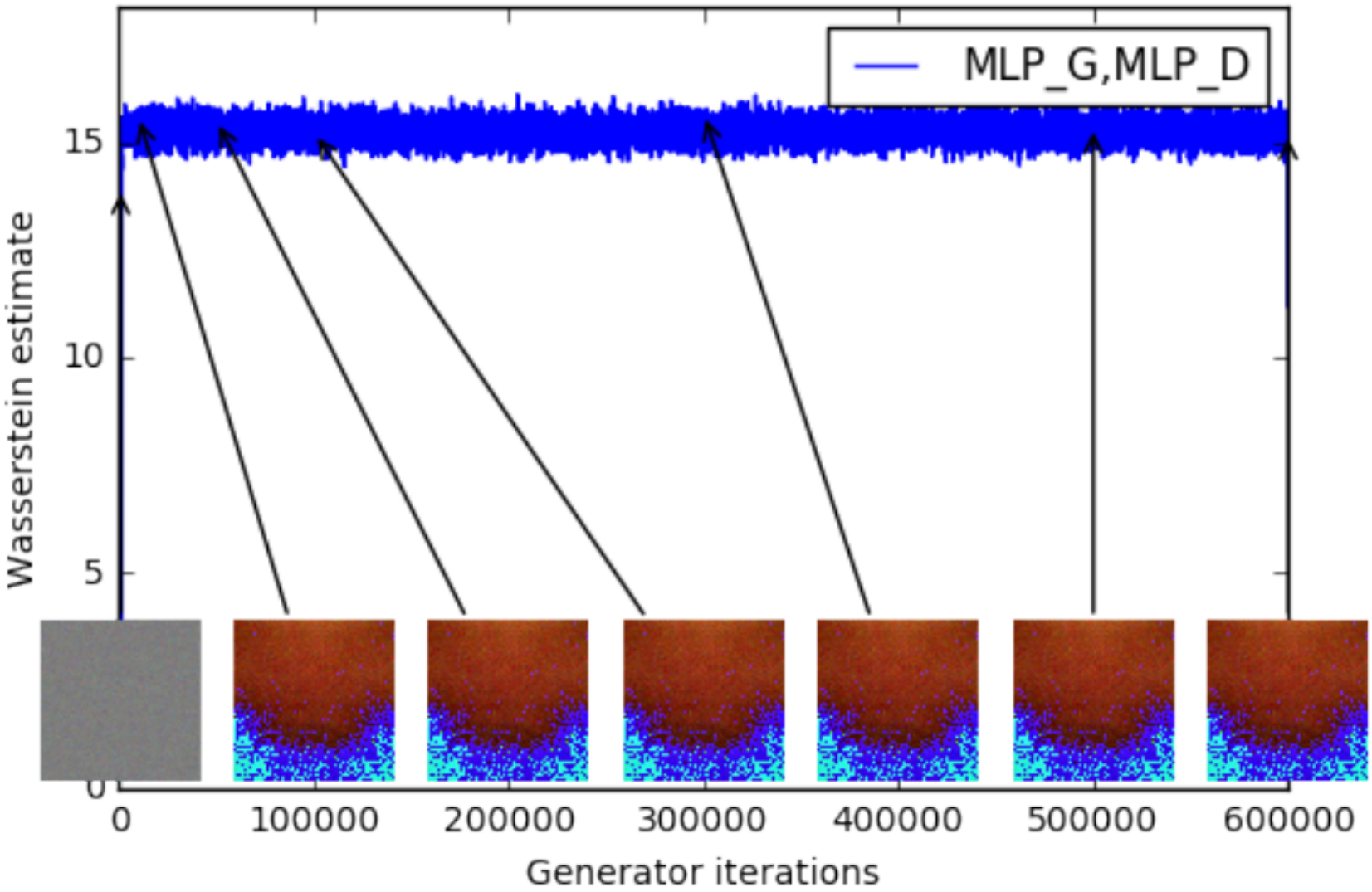








Empirical Results



Empirical Results

Figure 5: Algorithms trained with a DCGAN generator. Left: WGAN algorithm. Right: standard GAN formulation. Both algorithms produce high quality samples.

Figure 6: Algorithms trained with a generator without batch normalization and constant number of filters at every layer (as opposed to duplicating them every time as in [18]). Aside from taking out batch normalization, the number of parameters is therefore reduced by a bit more than an order of magnitude. Left: WGAN algorithm. Right: standard GAN formulation. As we can see the standard GAN failed to learn while the WGAN still was able to produce samples.

Empirical Results

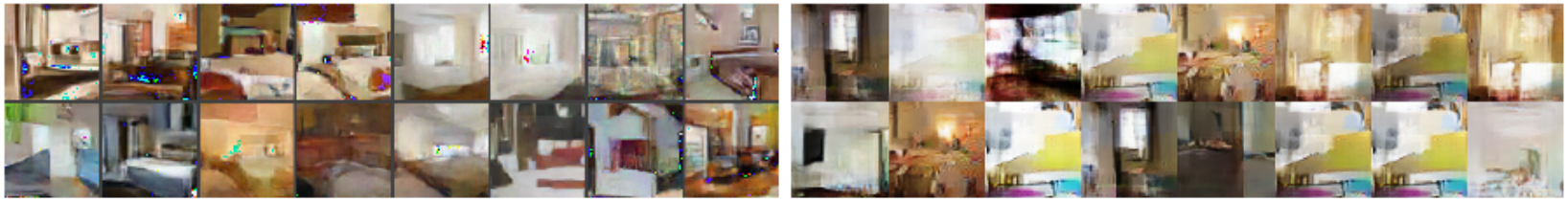


Figure 7: Algorithms trained with an MLP generator with 4 layers and 512 units with ReLU nonlinearities. The number of parameters is similar to that of a DCGAN, but it lacks a strong inductive bias for image generation. Left: WGAN algorithm. Right: standard GAN formulation. The WGAN method still was able to produce samples, lower quality than the DCGAN, and of higher quality than the MLP of the standard GAN. Note the significant degree of mode collapse in the GAN MLP.

Thanks Q&A