On Different Distances Between Distributions and Generative Adversarial Networks

Martin Arjovsky

Unsupervised learning

- We have samples $\{x^{(i)}\}_{i=1}^{m}$ from an unknown distribution $\mathbb{P}_r$

Unsupervised learning

- We have samples $\{x^{(i)}\}_{i=1}^{m}$ from an unknown distribution $\mathbb{P}_r$

- We want to approximate it by $\mathbb{P}_{\boldsymbol{\theta}}$ a parametric distribution that's close to $\mathbb{P}_r$ in some sense.

Unsupervised learning

- We have samples $\{x^{(i)}\}_{i=1}^{m}$ from an unknown distribution $\mathbb{P}_r$

- We want to approximate it by $\mathbb{P}_{\boldsymbol{\theta}}$ a parametric distribution that's close to $\mathbb{P}_r$ in some sense.
- Close how?

Maximum Likelihood

- Maximum likelihood:

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

Maximum Likelihood

- Maximum likelihood:

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

- Assumptions: continuous with full support.

Maximum Likelihood

- Maximum likelihood:

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

- Assumptions: continuous with full support.
- Problems: restricted capacity distributes mass.
  Modeling low dimensional distributions is impossible.

Kullback-Leibler Divergence

- Closeness measured by KL divergence (equivalent to ML):

$$\min_{\theta \in \mathbb{R}^d} KL(\mathbb{P}_r \| \mathbb{P}_\theta) = \int_{\mathcal{X}} P_r(x) \log \frac{P_r(x)}{P_\theta(x)} \, \mathrm{d}x$$

Kullback-Leibler Divergence

- Closeness measured by KL divergence (equivalent to ML):

$$\min_{\theta \in \mathbb{R}^d} KL(\mathbb{P}_r \| \mathbb{P}_\theta) = \int_{\mathcal{X}} P_r(x) \log \frac{P_r(x)}{P_\theta(x)} \, \mathrm{d}x$$

- When $P_r(x) > 0, P_\theta(x) \to 0$ integrand goes to infinity: high cost for mode dropping.

Kullback-Leibler Divergence

- Closeness measured by KL divergence (equivalent to ML):

$$\min_{\theta \in \mathbb{R}^d} KL(\mathbb{P}_r \| \mathbb{P}_\theta) = \int_{\mathcal{X}} P_r(x) \log \frac{P_r(x)}{P_\theta(x)} \, \mathrm{d}x$$

- When $P_r(x) > 0, P_\theta(x) \to 0$ integrand goes to infinity: high cost for mode dropping.
- When $P_\theta(x) > 0, P_r(x) \to 0$ integrand goes to 0: low cost for fake looking samples.

Generative Adversarial Networks

- Let $\mathbb{P}_\theta$ be the law of $g_\theta(Z)$ for some simple (e.g. Gaussian) r.v Z, passed through a complex function.

Generative Adversarial Networks

- Let $\mathbb{P}_\theta$ be the law of $g_\theta(Z)$ for some simple (e.g. Gaussian) r.v Z, passed through a complex function.

- Discriminator maximizes and generator minimizes

$$L(\phi, \theta) = \mathbb{E}_{x \sim \mathbb{P}_r}[\log D_\phi(x)] + \mathbb{E}_{z \sim p_Z}[\log(1 - D_\phi(g_\theta(z)))]$$

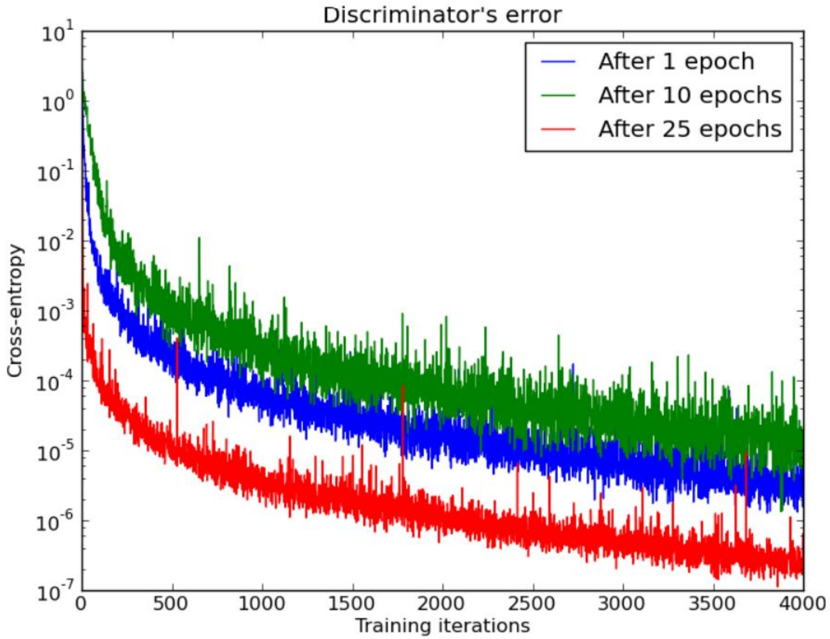
Generative Adversarial Networks

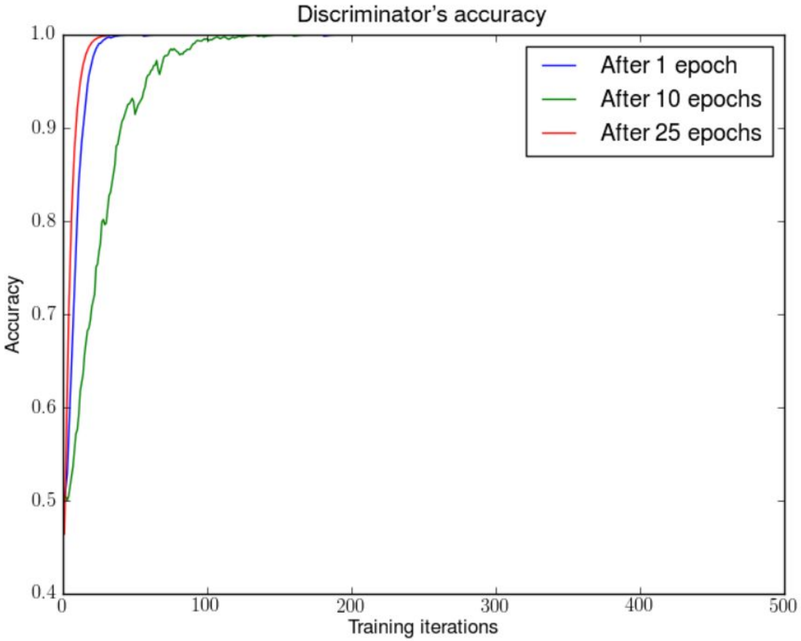
- Under optimal discriminator, minimizes

$$\min_{\theta \in \mathbb{R}^d} JSD(\mathbb{P}_r \| \mathbb{P}_\theta) = KL(\mathbb{P}_r \| \mathbb{P}_m) + KL(\mathbb{P}_\theta \| \mathbb{P}_m)$$

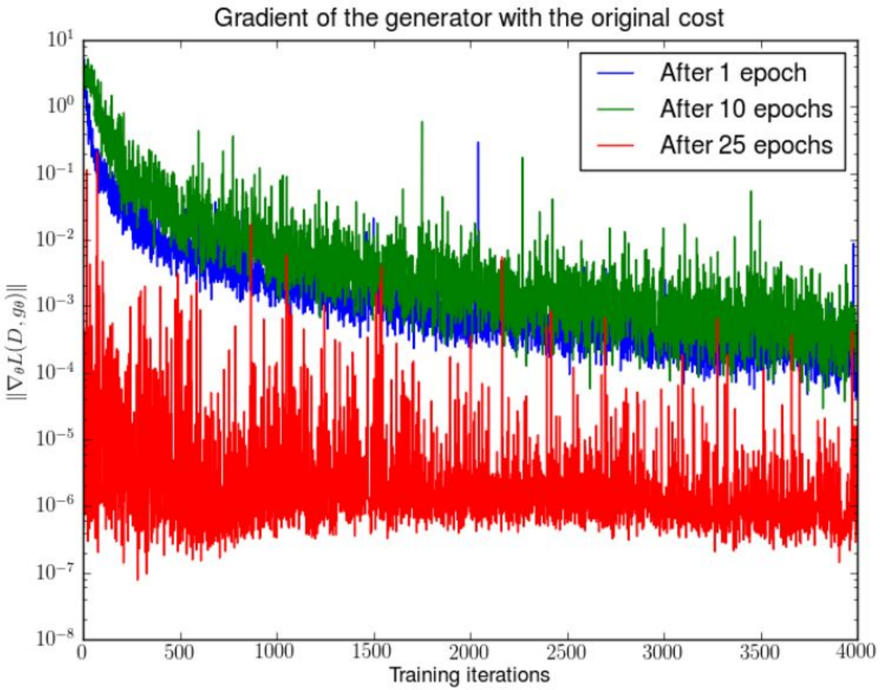
- Problems: vanishing gradients very quickly when D’s accuracy is high.

Discriminator is pretty good...





Vanishing gradients, original cost



Alternate update

- Alternate update that has less vanishing gradients

$$\Delta\theta \propto \mathbb{E}_{z\sim p_Z}[\nabla_\theta \log(D_\phi(g_\theta(z)))]$$

Alternate update

- Alternate update that has less vanishing gradients

$$\Delta\theta \propto \mathbb{E}_{z\sim p_Z}[\nabla_\theta \log(D_\phi(g_\theta(z)))]$$

- Under optimality optimizes

$$KL(\mathbb{P}_\theta \| \mathbb{P}_r) - 2JSD(\mathbb{P}_r \| \mathbb{P}_\theta)$$

Alternate update

- Alternate update that has less vanishing gradients

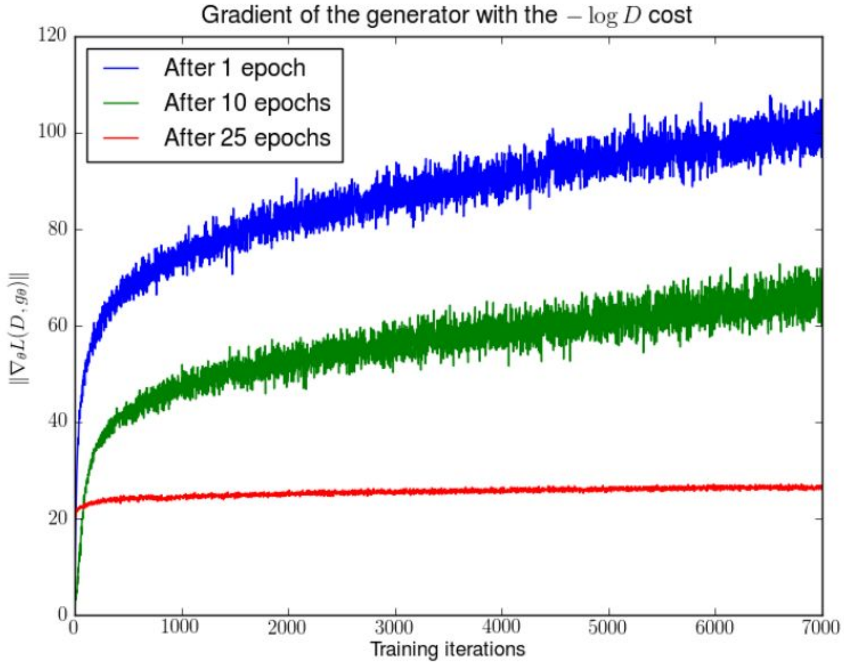
$$\Delta\theta \propto \mathbb{E}_{z\sim p_Z}[\nabla_\theta \log(D_\phi(g_\theta(z)))]$$

- Under optimality optimizes

$$KL(\mathbb{P}_\theta \| \mathbb{P}_r) - 2JSD(\mathbb{P}_r \| \mathbb{P}_\theta)$$

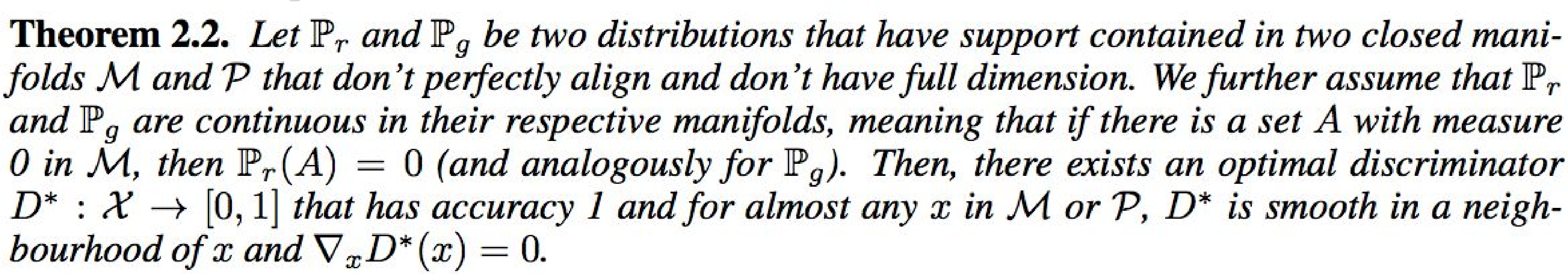
- Problems: JSD with the wrong sign, reverse KL has high mode dropping. Still unstable when D is good.

High variance updates



Problems of GANs (and divergences)

- When $\mathbb{P}_r$ and $\mathbb{P}_\theta$ lie on low dimensional manifolds, there's always a perfect discriminator, that provides no usable gradients.



Theorem 2.2. *Let $\mathbb{P}_r$ and $\mathbb{P}_g$ be two distributions that have support contained in two closed manifolds $\mathcal{M}$ and $\mathcal{P}$ that don't perfectly align and don't have full dimension. We further assume that $\mathbb{P}_r$ and $\mathbb{P}_g$ are continuous in their respective manifolds, meaning that if there is a set A with measure 0 in $\mathcal{M}$, then $\mathbb{P}_r(A) = 0$ (and analogously for $\mathbb{P}_g$). Then, there exists an optimal discriminator $D^* : \mathcal{X} \to [0, 1]$ that has accuracy 1 and for almost any x in $\mathcal{M}$ or $\mathcal{P}$, D^* is smooth in a neighbourhood of x and $\nabla_x D^*(x) = 0$.*

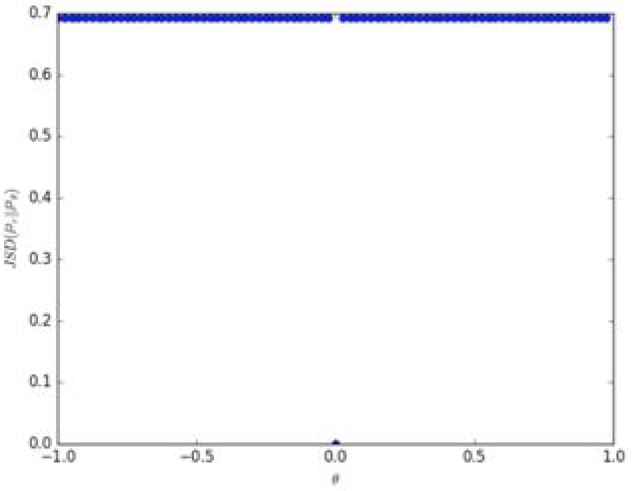
Problems of GANs (and divergences)

- When $\mathbb{P}_r$ and $\mathbb{P}_\theta$ lie on low dimensional manifolds, there's always a perfect discriminator, that provides no usable gradients.
- Under the same assumptions

$$JSD(\mathbb{P}_r||\mathbb{P}_\theta) = \log 2$$
$$KL(\mathbb{P}_r||\mathbb{P}_\theta) = +\infty$$
$$KL(\mathbb{P}_\theta||\mathbb{P}_r) = +\infty$$

Problems of JSD, KLs et al.

- $JSD(\mathbb{P}_r||\mathbb{P}_\theta)$ Doesn’t need to be a continuous function of θ .
- Learning parallel lines.



Distances between distributions

- The topology JSD induces on probability measures is too big, therefore few mappings to this space are continuous.

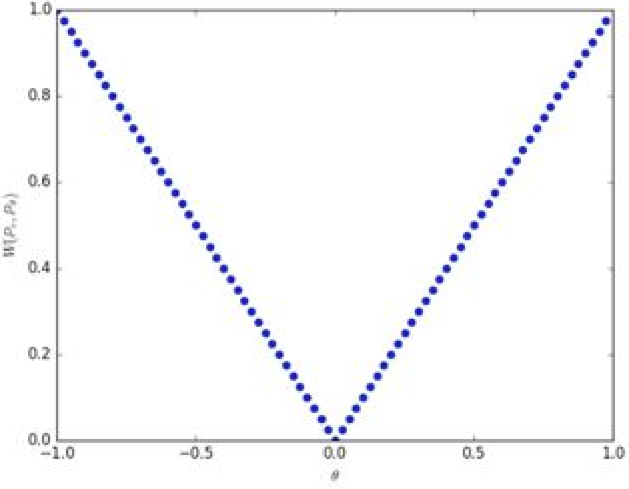
Distances between distributions

- The topology JSD induces on probability measures is too big, therefore few mappings to this space are continuous.
- We can use the weak* topology, given by Wasserstein

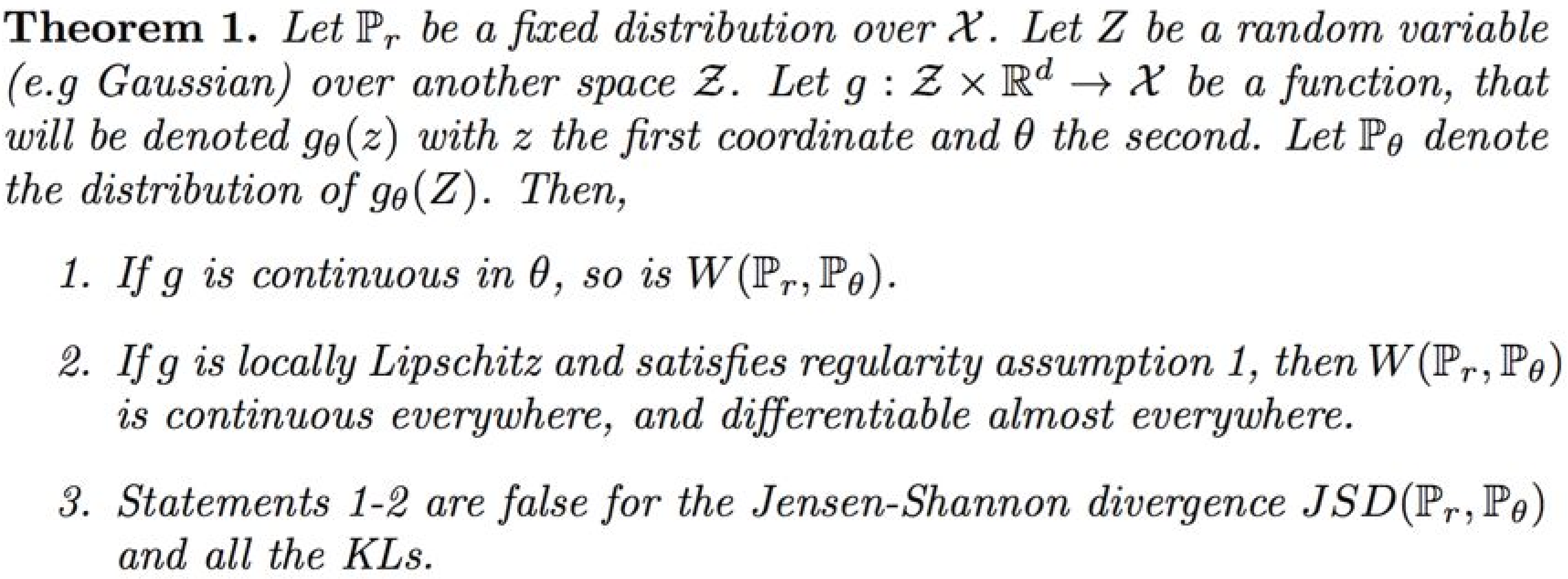
$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \int_{\mathcal{X} \times \mathcal{X}} \|x - y\| \, \mathrm{d}\gamma(x, y)$$

Wasserstein distance

- Wasserstein loss $W(\mathbb{P}_r, \mathbb{P}_\theta)$ is continuous (lines ex):



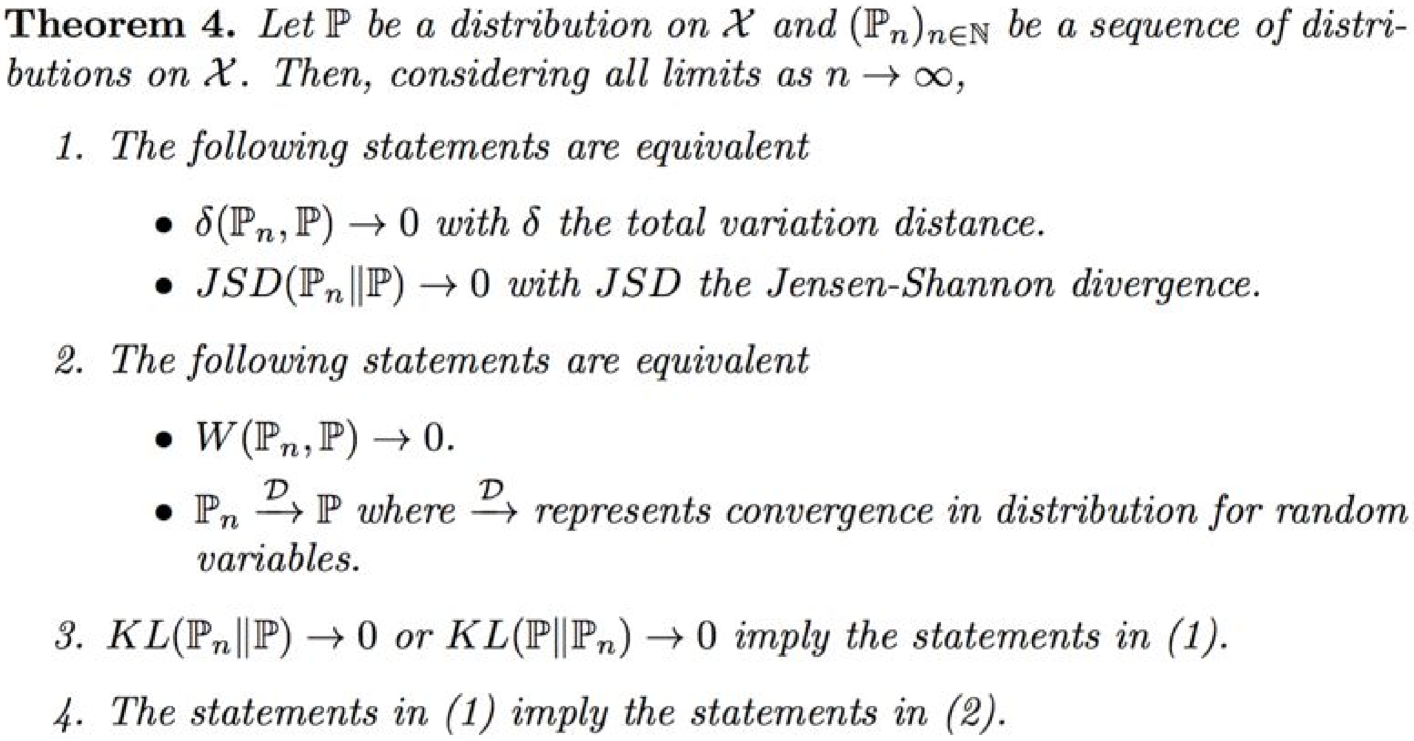
Regularity of Wasserstein



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 $JSD(\mathbb{P}_r, \mathbb{P}_\theta)$ and all the KLs.*

Hierarchy of distances



Theorem 4. *Let $\mathbb{P}$ be a distribution on $\mathcal{X}$ 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.*
   - $JSD(\mathbb{P}_n \| \mathbb{P}) \to 0$ *with JSD 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).*

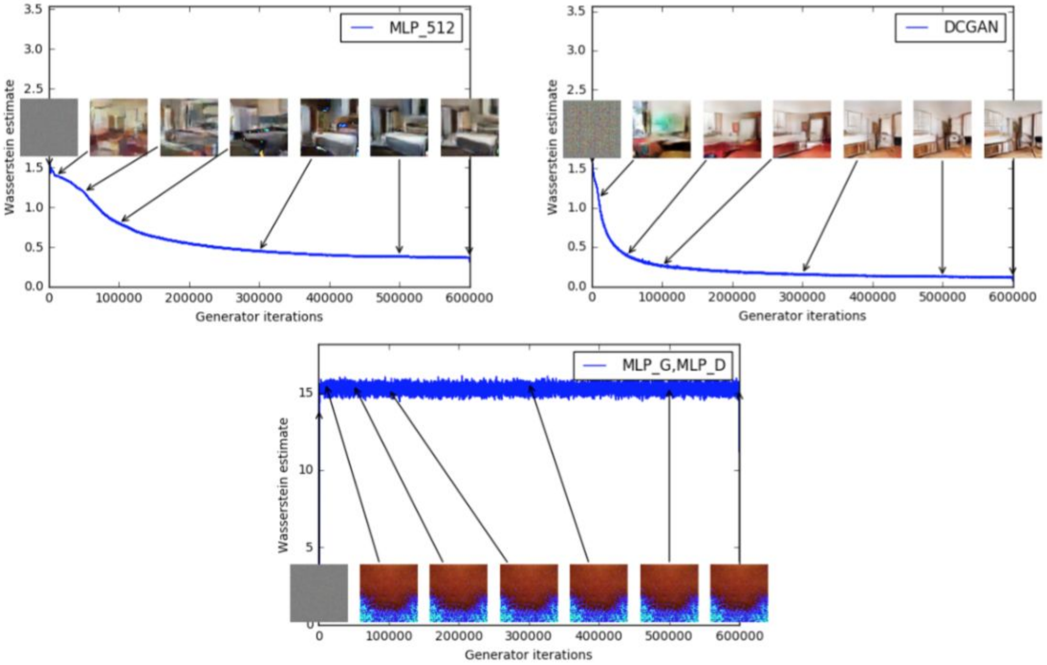
Idea: optimize Wasserstein

- Wasserstein has a dual problem

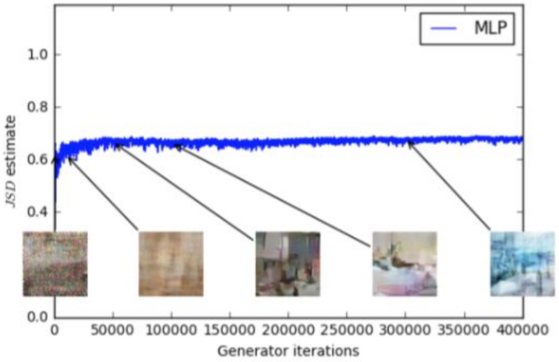
$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \max_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)]$$

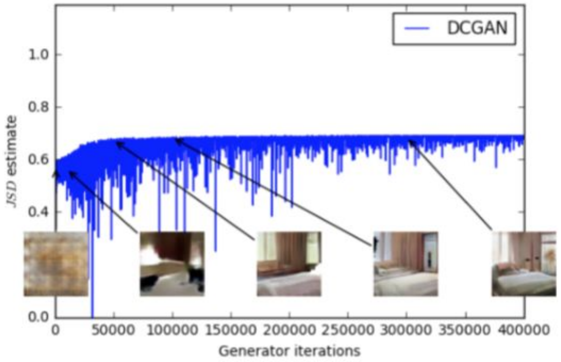
- Idea: train one net f to maximize the dual, then do gradient descent on \theta.

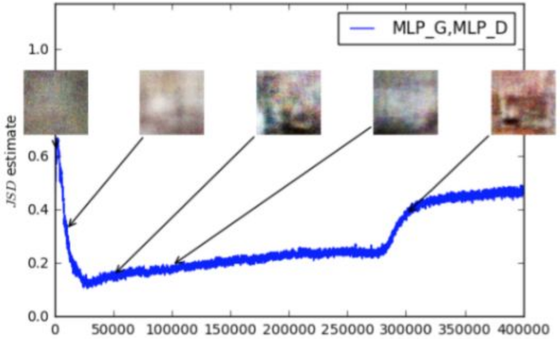
WGAN loss correlates with sample quality!



Correlation for a normal GAN is terrible







Improved model stability

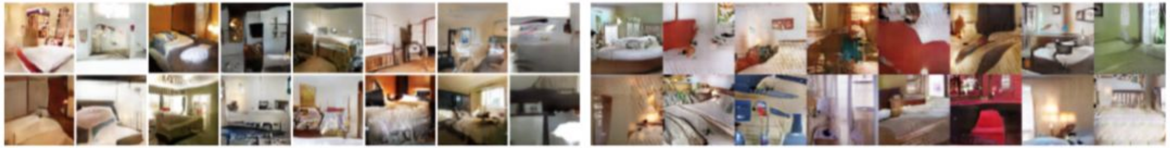


Figure 5: Algorithms trained with a DCGAN generator. Left: WGAN algorithm. Right: standard GAN formulation. Both algorithms produce high quality samples.

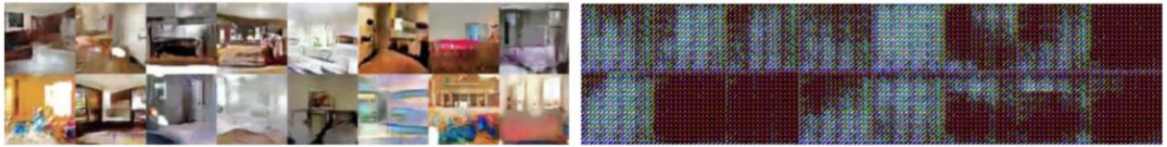


Figure 6: Algorithms trained with a generator without batch normalization and constant number of filters at every layer (as opposed to duplicating them every time as in [17]). Aside from taking out batch normalization, the number of parameters is therefore reduced by a bit more than an order of magnitude. 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.

Improved model stability (cont.)

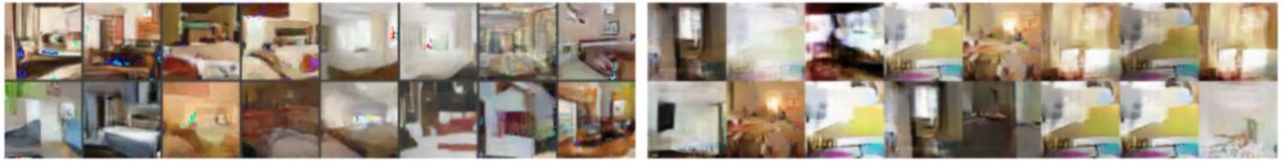


Figure 7: Algorithms trained with an MLP generator with 4 layers and 512 units with ReLU nonlinearities. The number of parameters is similar to that of a DCGAN, but it lacks a strong inductive bias for image generation. Left: WGAN algorithm. Right: standard GAN formulation. The WGAN method still was 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.

Further work needed

- Weight clipping is a terrible way to enforce Lipschitz constraints!

Further work needed

- Weight clipping is a terrible way to enforce Lipschitz constraints!
- There are many Wasserstein distances aside from EM:

Further work needed

- Weight clipping is a terrible way to enforce Lipschitz constraints!
- There are many Wasserstein distances aside from EM:
  - What properties do they share? Make them different?

Further work needed

- Weight clipping is a terrible way to enforce Lipschitz constraints!
- There are many Wasserstein distances aside from EM:
  - What properties do they share? Make them different?
  - How much mode dropping / sample quality focused are they?

Further work needed

- Weight clipping is a terrible way to enforce Lipschitz constraints!
- There are many Wasserstein distances aside from EM:
  - What properties do they share? Make them different?
  - How much mode dropping / sample quality focused are they?
  - How do we optimize them? They all have duals, but they are much more complicated. (E.g. for W2 replace lipschitz by convex and convex conj).

Further work needed

- Weight clipping is a terrible way to enforce Lipschitz constraints!
- There are many Wasserstein distances aside from EM:
  - What properties do they share? Make them different?
  - How much mode dropping / sample quality focused are they?
  - How do we optimize them? They all have duals, but they are much more complicated. (E.g. for W2 replace lipschitz by convex and convex conj).
- Wasserstein requires a metric in X.

Further work needed

- Weight clipping is a terrible way to enforce Lipschitz constraints!
- There are many Wasserstein distances aside from EM:
  - What properties do they share? Make them different?
  - How much mode dropping / sample quality focused are they?
  - How do we optimize them? They all have duals, but they are much more complicated. (E.g. for W2 replace lipschitz by convex and convex conj).
- Wasserstein requires a metric in X.
  - Which one is wgan using? (Some combination of features / samples L2?)

Further work needed

- Weight clipping is a terrible way to enforce Lipschitz constraints!
- There are many Wasserstein distances aside from EM:
  - What properties do they share? Make them different?
  - How much mode dropping / sample quality focused are they?
  - How do we optimize them? They all have duals, but they are much more complicated. (E.g. for W2 replace lipschitz by convex and convex conj).
- Wasserstein requires a metric in X.
  - Which one is wgan using? (Some combination of features / samples L2?)
  - Can we optimize W for a given metric? (And construct geodesics!)

Further work needed

- Weight clipping is a terrible way to enforce Lipschitz constraints!
- There are many Wasserstein distances aside from EM:
  - What properties do they share? Make them different?
  - How much mode dropping / sample quality focused are they?
  - How do we optimize them? They all have duals, but they are much more complicated. (E.g. for W2 replace lipschitz by convex and convex conj).
- Wasserstein requires a metric in X.
  - Which one is wgan using? (Some combination of features / samples L2?)
  - Can we optimize W for a given metric? (And construct geodesics!)
  - Can we learn the metric simultaneously? (And learn geodesics!)

That's all Folks!