

Deep Generative Models

Lecture 10

Roman Isachenko



Ozon Masters

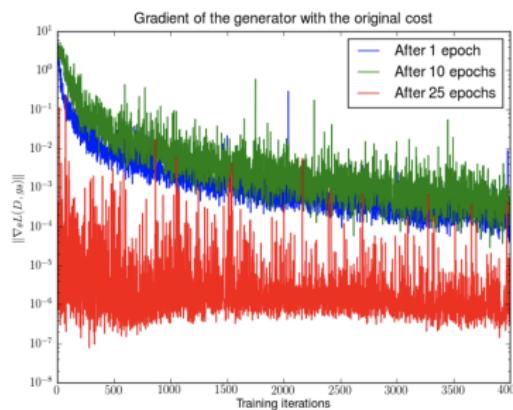
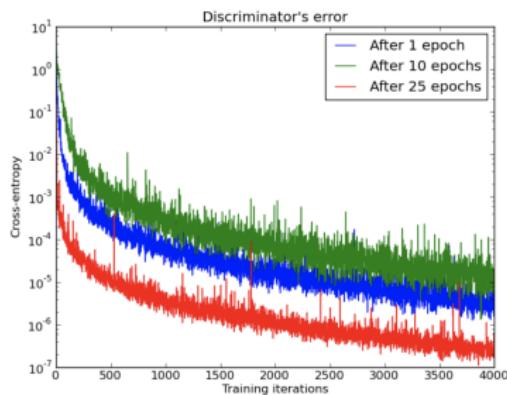
Spring, 2021

Vanishing gradients

Objective

$$\min_G \max_D V(G, D) = \min_G \max_D [\mathbb{E}_{\pi(x)} \log D(x) + \mathbb{E}_{p(z)} \log(1 - D(G(z)))]$$

Early in learning, G is poor, D can reject samples with high confidence. In this case, $\log(1 - D(G(z)))$ saturates.



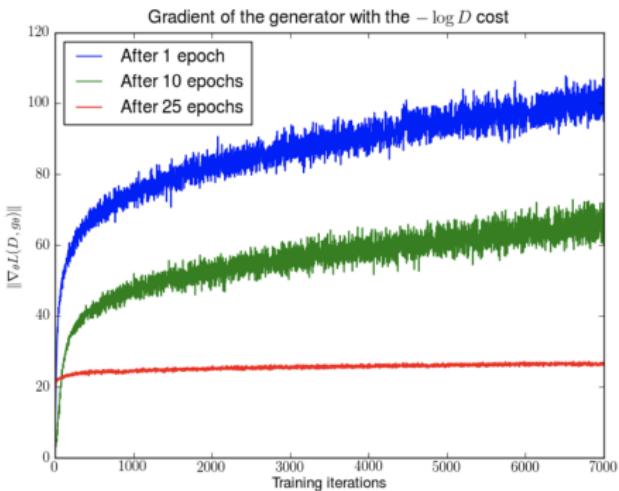
Vanishing gradients

Objective

$$\min_G \max_D V(G, D) = \min_G \max_D [\mathbb{E}_{\pi(x)} \log D(x) + \mathbb{E}_{p(z)} \log(1 - D(G(z)))]$$

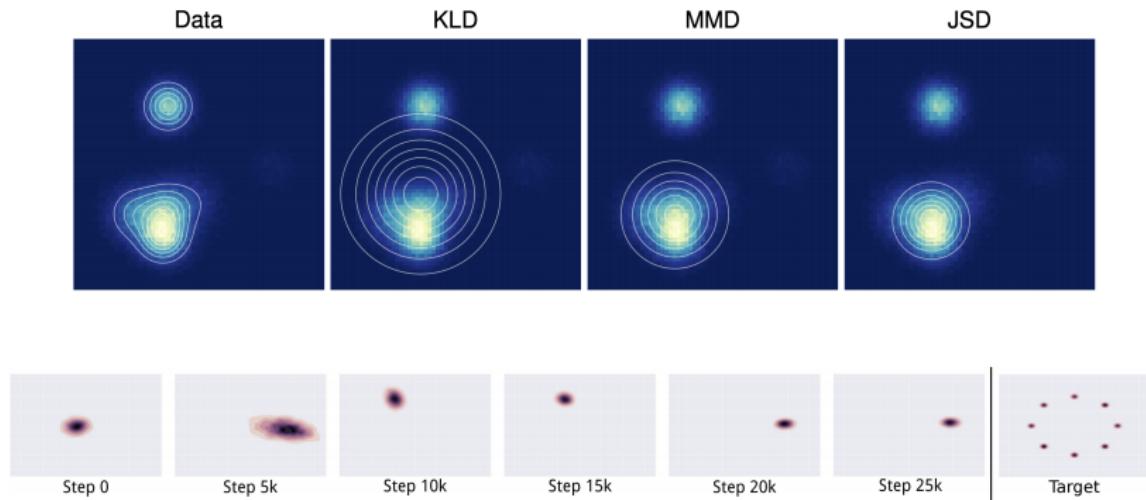
Non-saturating GAN

- ▶ Maximize $\log D(G(z))$ instead of $\log(1 - D(G(z)))$.
- ▶ Gradients are getting much stronger, but the training is unstable (with increasing mean and variance).



Mode collapse

The phenomena where the generator of a GAN collapses to one or few distribution modes.



Alternate architectures, adding regularization terms, injecting small noise perturbations and other millions bags and tricks are used to avoid the mode collapse.

Goodfellow I. J. et al. Generative Adversarial Networks, 2014

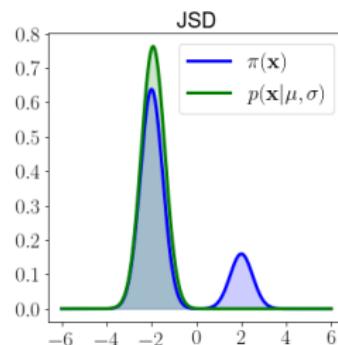
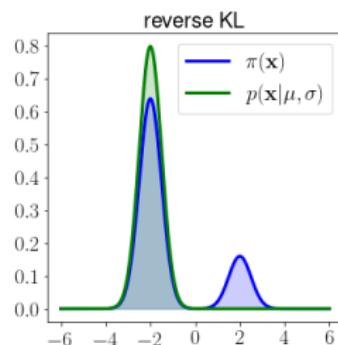
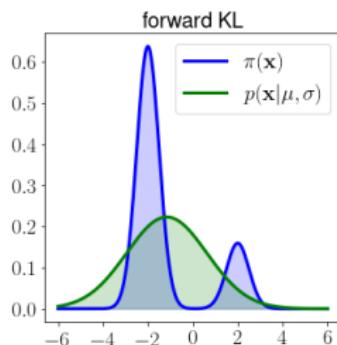
Metz L. et al. Unrolled Generative Adversarial Networks, 2016

Jensen-Shannon vs Kullback-Leibler

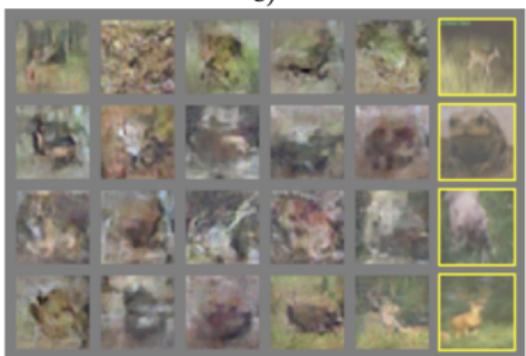
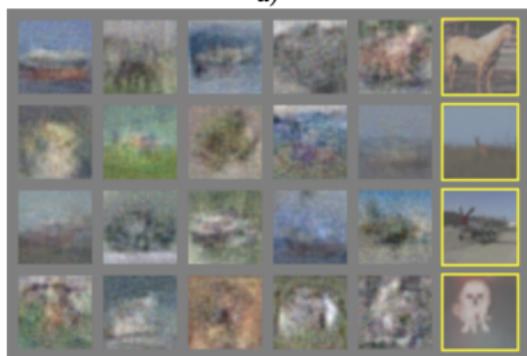
Mode covering vs mode seeking

$$KL(\pi||p) = \int \pi(\mathbf{x}) \log \frac{\pi(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x}, \quad KL(p||\pi) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\pi(\mathbf{x})} d\mathbf{x}$$

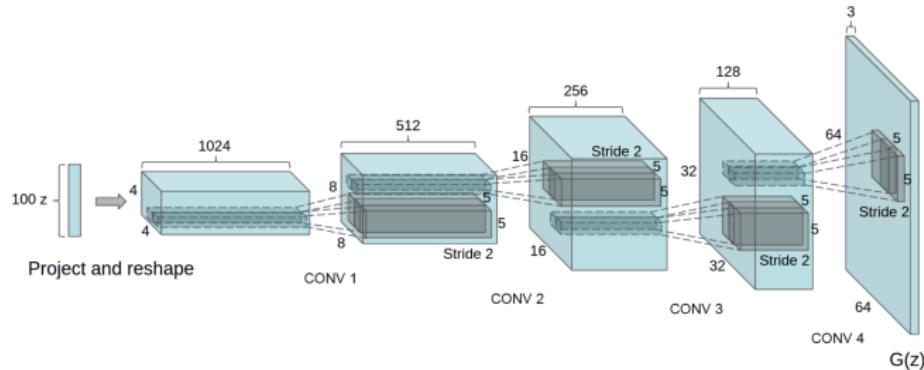
$$JSD(\pi||p) = \frac{1}{2} \left[KL \left(\pi(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x})}{2} \right) + KL \left(p(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x})}{2} \right) \right]$$



Vanilla GAN results



Deep Convolutional GAN

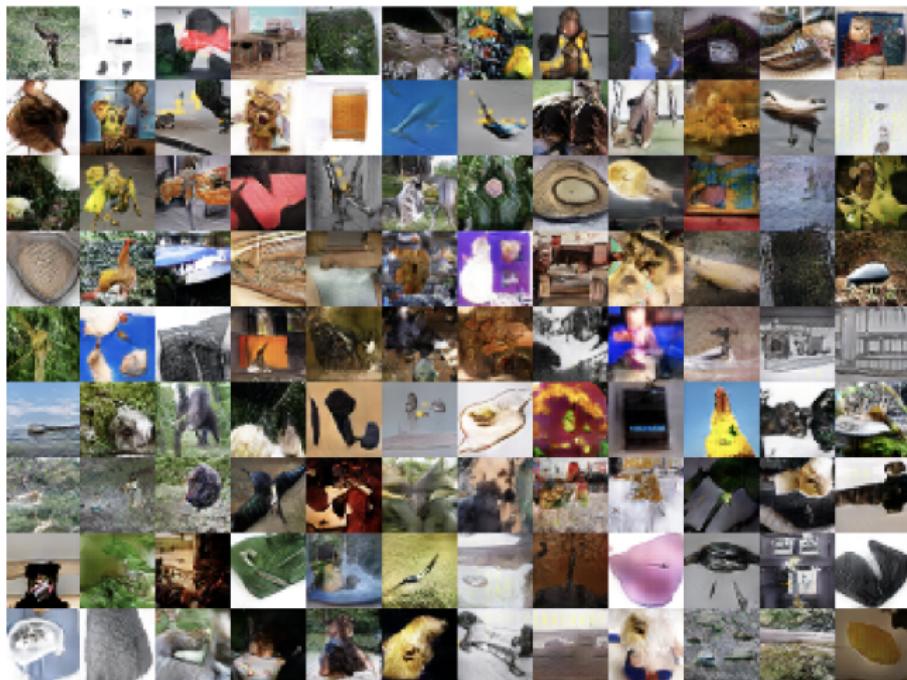


- ▶ Mean-pooling instead of max-pooling.
- ▶ Transposed convolutions in the generator for upsampling.
- ▶ Downsample with strided convolutions and average pooling.
- ▶ ReLU for generator, Leaky-ReLU (0.2) for discriminator.
- ▶ Output nonlinearity: tanh for Generator, sigmoid for discriminator.
- ▶ Batch Normalization used to prevent mode collapse (not applied at the output of G and input of D).
- ▶ Adam: small LR = 2e-4; small momentum: 0.5, batch-size: 128.

Radford A., Metz L., Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015

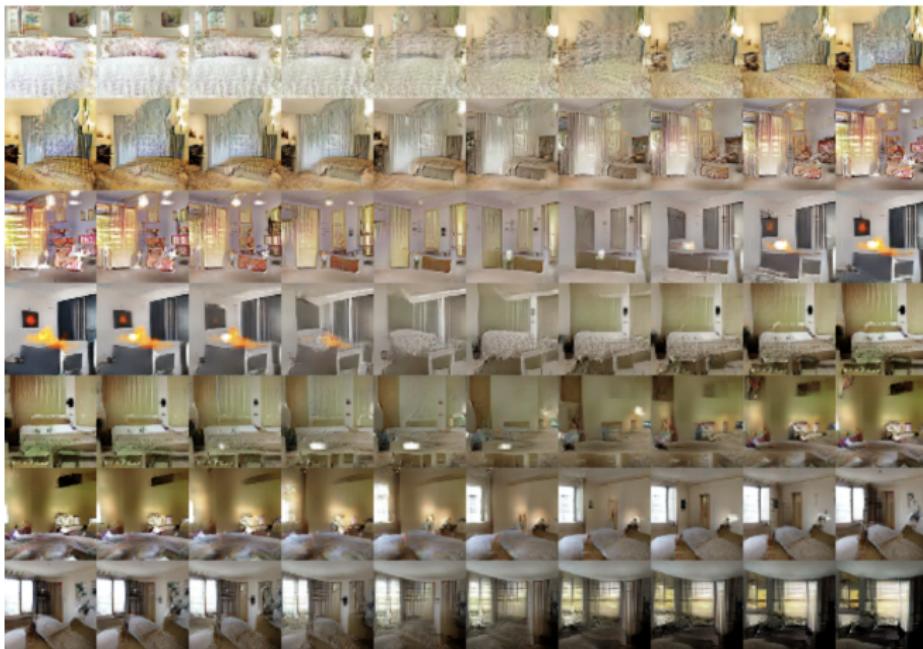
Deep Convolutional GAN

ImageNet samples



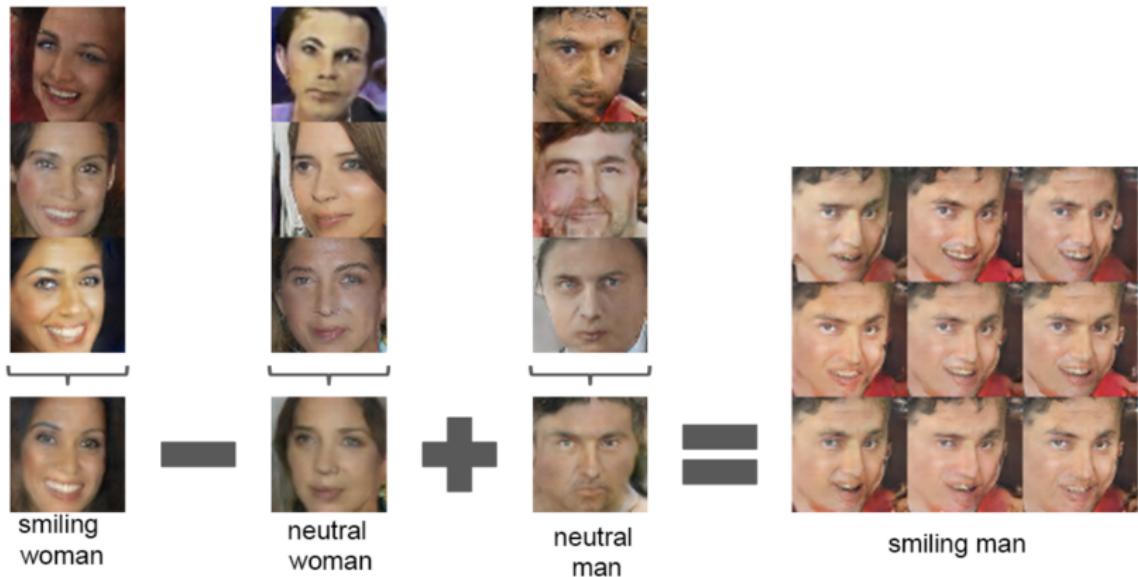
Deep Convolutional GAN

Smooth interpolations



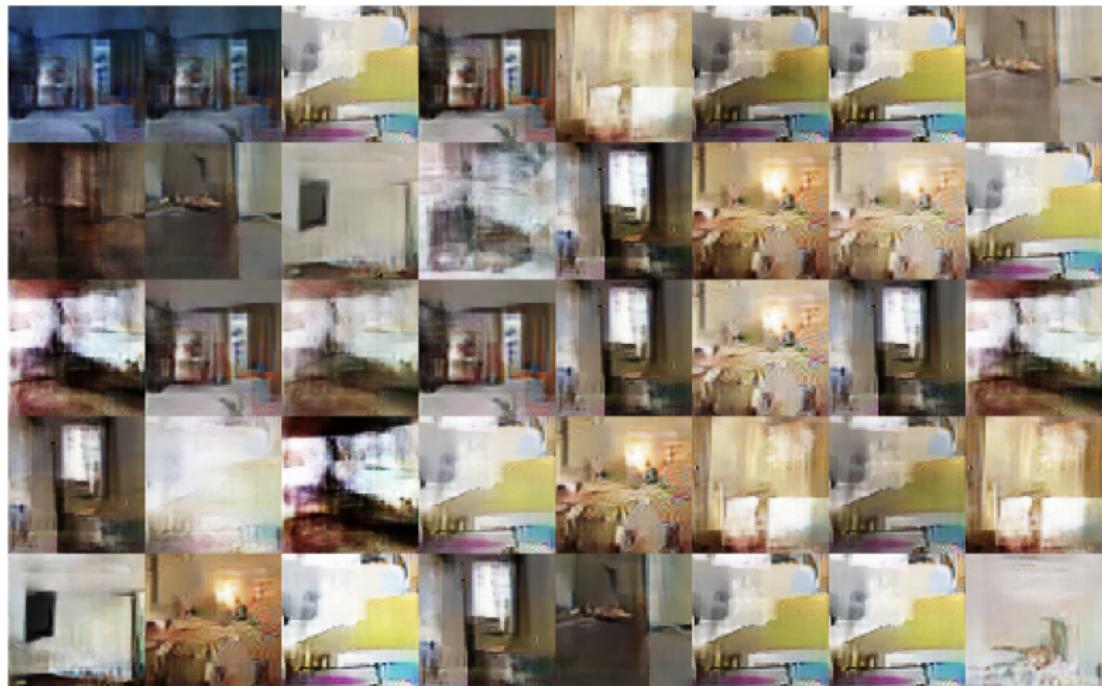
Deep Convolutional GAN

Vector arithmetic



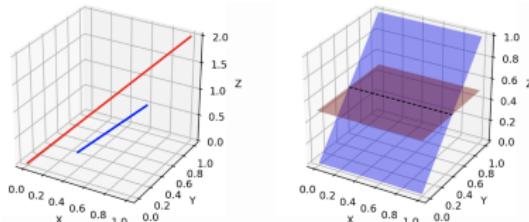
Deep Convolutional GAN

Mode collapse



Informal theoretical results

- ▶ Since z usually has lower dimensionality compared to x , manifold $G(z)$ has a measure 0 in x space. Hence, support of $p(x|\theta)$ lies on low-dimensional manifold.
- ▶ Distribution of real images $\pi(x)$ is also concentrated on a low dimensional manifold.



- ▶ If $\pi(x)$ and $p(x|\theta)$ have disjoint supports, then there is a smooth optimal discriminator. We are not able to learn anything by backproping through it.
- ▶ For such low-dimensional disjoint manifolds

$$KL(\pi||p) = KL(p||\pi) = \infty, \quad JSD(\pi||p) = \log 2$$

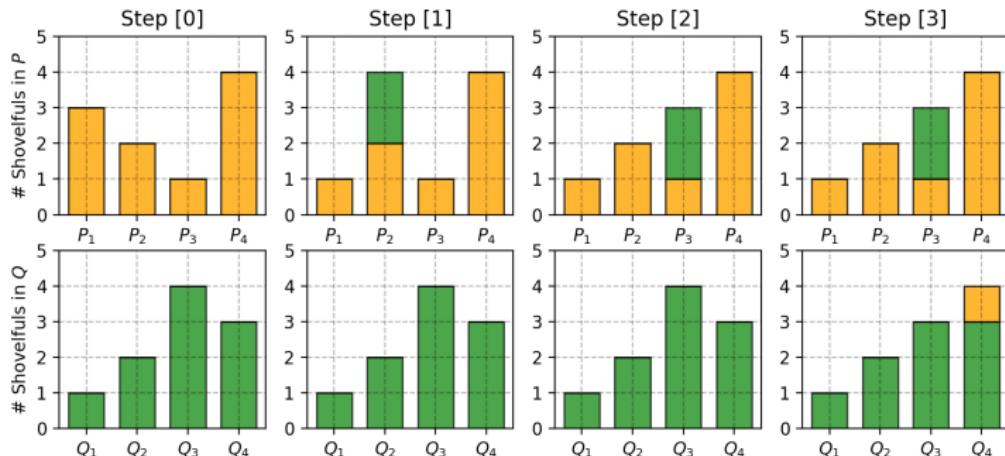
- ▶ Adding continuous noise to the inputs of the discriminator smoothes the distributions of the probability mass.

Weng L. From GAN to WGAN, 2019

Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017

Wasserstein distance (discrete)

Also called Earth Mover's distance. The minimum cost of moving and transforming a pile of dirt in the shape of one probability distribution to the shape of the other distribution.



$$W(P, Q) = 2(\text{step 1}) + 2(\text{step 2}) + 1(\text{step 3}) = 5$$

Wasserstein distance

$$W(\pi, p) = \inf_{\gamma \in \prod(\pi, p)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \|\mathbf{x} - \mathbf{y}\| = \inf_{\gamma \in \prod(\pi, p)} \int \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

- ▶ $\prod(\pi, p)$ – the set of all joint distributions $\gamma(\mathbf{x}, \mathbf{y})$ with marginals π and p ($\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} = p(\mathbf{y})$, $\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y} = \pi(\mathbf{x})$)
- ▶ $\gamma(\mathbf{x}, \mathbf{y})$ – transportation plan (the amount of "dirt" that should be transported from point \mathbf{x} to point \mathbf{y}).
- ▶ $\gamma(\mathbf{x}, \mathbf{y})$ – the amount, $\|\mathbf{x} - \mathbf{y}\|$ – the distance.

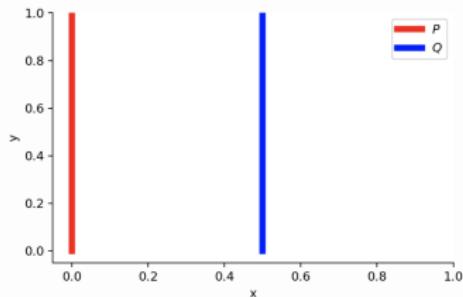
For better understanding of transportation plan function γ , try to write down the plan for previous discrete case.

Wasserstein distance vs KL vs JSD

Consider 2d distributions

$$\pi(x, y) = (0, U[0, 1])$$

$$p(x, y|\theta) = (\theta, U[0, 1])$$



- $\theta = 0$. Distributions are the same

$$KL(\pi||p) = KL(p||\pi) = JSD(p||\pi) = W(\pi, p) = 0$$

- $\theta \neq 0$

$$KL(\pi||p) = \int_{U[0,1]} 1 \log \frac{1}{0} dy = \infty = KL(p||\pi)$$

$$JSD(\pi||p) = \frac{1}{2} \left(\int_{U[0,1]} 1 \log \frac{1}{1/2} dy + \int_{U[0,1]} 1 \log \frac{1}{1/2} dy \right) = \log 2$$

$$W(\pi, p) = |\theta|$$

Wasserstein distance vs KL vs JSD

Theorem 1

Let $G(\mathbf{z}, \theta)$ be any feedforward neural network, and $p(\mathbf{z})$ a prior over \mathbf{z} such that $\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \|\mathbf{z}\| < \infty$. Then therefore $W(\pi, p)$ is continuous everywhere and differentiable almost everywhere.

Theorem 2

Let π be a distribution on a compact space \mathcal{X} and $\{p_t\}_{t=1}^{\infty}$ be a sequence of distributions on \mathcal{X} .

$$KL(\pi || p_t) \rightarrow 0 \text{ (or } KL(p_t || \pi) \rightarrow 0) \quad (1)$$

$$JSD(\pi || p_t) \rightarrow 0 \quad (2)$$

$$W(\pi || p_t) \rightarrow 0 \quad (3)$$

Then, considering limits as $t \rightarrow \infty$, (1) implies (2), (2) implies (3).

Wasserstein distance

$$W(\pi||p) = \inf_{\gamma \in \Pi(\pi, p)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \|\mathbf{x} - \mathbf{y}\| = \inf_{\gamma \in \Pi(\pi, p)} \int \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

- ▶ $\Pi(\pi, p)$ – the set of all joint distributions $\gamma(\mathbf{x}, \mathbf{y})$ with marginals π and p ($\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} = p(\mathbf{y})$, $\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y} = \pi(\mathbf{x})$)
- ▶ $\gamma(\mathbf{x}, \mathbf{y})$ – transportation plan (the amount of "dirt" that should be transported from point \mathbf{x} to point \mathbf{y}).
- ▶ $\gamma(\mathbf{x}, \mathbf{y})$ – the amount, $\|\mathbf{x} - \mathbf{y}\|$ – the distance.

For better understanding of transportation plan function γ , try to write down the plan for previous discrete case.

Wasserstein distance vs KL vs JSD

Theorem 1

Let $G(\mathbf{z}, \theta)$ be any feedforward neural network, and $p(\mathbf{z})$ a prior over \mathbf{z} such that $\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \|\mathbf{z}\| < \infty$. Then therefore $W(\pi||p)$ is continuous everywhere and differentiable almost everywhere.

Theorem 2

Let π be a distribution on a compact space \mathcal{X} and $\{p_t\}_{t=1}^{\infty}$ be a sequence of distributions on \mathcal{X} .

$$KL(\pi||p_t) \rightarrow 0 \text{ (or } KL(p_t||\pi) \rightarrow 0) \quad (4)$$

$$JSD(\pi||p_t) \rightarrow 0 \quad (5)$$

$$W(\pi||p_t) \rightarrow 0 \quad (6)$$

Then, considering limits as $t \rightarrow \infty$, (1) implies (2), (2) implies (3).

Wasserstein GAN

Wasserstein distance

$$W(\pi||p) = \inf_{\gamma \in \prod(\pi, p)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \|\mathbf{x} - \mathbf{y}\| = \inf_{\gamma \in \prod(\pi, p)} \int \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

The infimum across all possible joint distributions in $\prod(\pi, p)$ is intractable.

Kantorovich-Rubinstein duality

$$W(\pi||p) = \frac{1}{K} \max_{\|f\|_L \leq K} [\mathbb{E}_{\mathbf{x} \sim \pi} f(\mathbf{x}) - \mathbb{E}_{\mathbf{x} \sim p} f(\mathbf{x})],$$

where $\|f\|_L \leq K$ are K -Lipschitz continuous functions
 $(f : \mathcal{X} \rightarrow \mathbb{R})$

$$|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq K \|\mathbf{x}_1 - \mathbf{x}_2\|, \quad \text{for all } \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}.$$

Wasserstein GAN

Kantorovich-Rubinstein duality

$$W(\pi || p) = \frac{1}{K} \max_{\|f\|_L \leq K} [\mathbb{E}_{\mathbf{x} \sim \pi} f(\mathbf{x}) - \mathbb{E}_{\mathbf{x} \sim p} f(\mathbf{x})],$$

- ▶ Now we have to ensure that f is K -Lipschitz continuous.
- ▶ Let $f(\mathbf{x}, \phi)$ be parametrized by parameters ϕ .
- ▶ If parameters ϕ lie in a compact set Φ then $f(\mathbf{x}, \phi)$ will be K -Lipschitz continuous function.
- ▶ Let the parameters be clamped to a fixed box $\Phi \in [-0.01, 0.01]^d$ after each gradient update.

$$\max_{\phi \in \Phi} [\mathbb{E}_{\mathbf{x} \sim \pi} f(\mathbf{x}, \phi) - \mathbb{E}_{\mathbf{x} \sim p} f(\mathbf{x}, \phi)] \leq \max_{\|f\|_L \leq K} [\mathbb{E}_{\mathbf{x} \sim \pi} f(\mathbf{x}) - \mathbb{E}_{\mathbf{x} \sim p} f(\mathbf{x})] = K \cdot W(\pi || p)$$

Wasserstein GAN

Vanilla GAN objective

$$\min_G \max_D \mathbb{E}_{\pi(x)} \log D(x) + \mathbb{E}_{p(z)} \log(1 - D(G(z)))$$

WGAN objective

$$\min_G W(\pi || p) = \min_G \max_{\phi \in \Phi} [\mathbb{E}_{\pi(x)} f(x, \phi) - \mathbb{E}_{p(z)} f(G(z), \phi)].$$

- ▶ Discriminator D is similar to the function f , but not the same (it is not a classifier anymore). In the WGAN model, function f is usually called *critic*.
- ▶ "Weight clipping is a clearly terrible way to enforce a Lipschitz constraint". If the clipping parameter is large, it is hard to train the critic till optimality. If the clipping parameter is too small, it could lead to vanishing gradients.

Wasserstein GAN

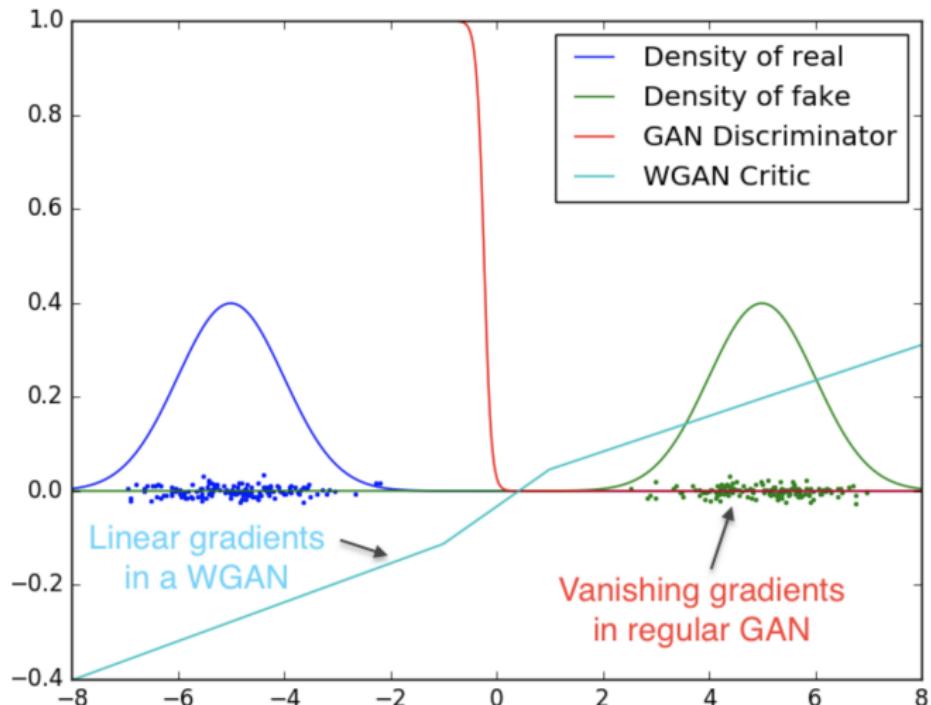
Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

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.

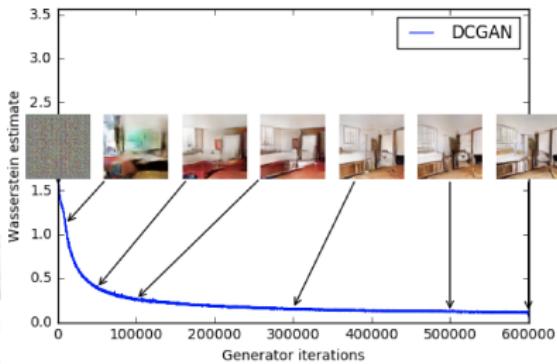
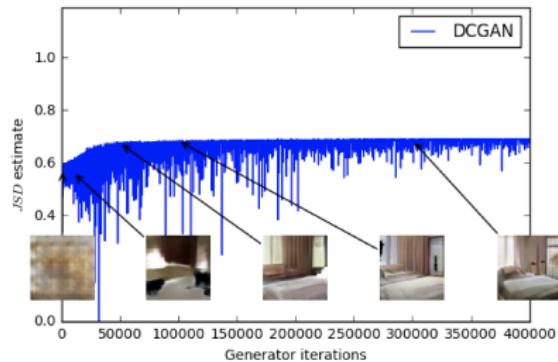
Require: : w_0 , initial critic parameters. θ_0 , initial generator's parameters.

```
1: while  $\theta$  has not converged do
2:   for  $t = 0, \dots, n_{\text{critic}}$  do
3:     Sample  $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$  a batch from the real data.
4:     Sample  $\{z^{(i)}\}_{i=1}^m \sim p(z)$  a batch of prior samples.
5:      $g_w \leftarrow \nabla_w [\frac{1}{m} \sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))]$ 
6:      $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$ 
7:      $w \leftarrow \text{clip}(w, -c, c)$ 
8:   end for
9:   Sample  $\{z^{(i)}\}_{i=1}^m \sim p(z)$  a batch of prior samples.
10:   $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))$ 
11:   $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$ 
12: end while
```

Wasserstein GAN



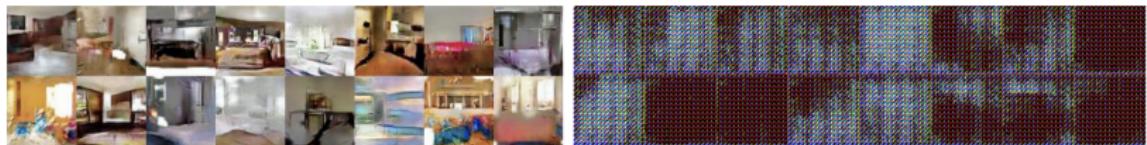
Wasserstein GAN



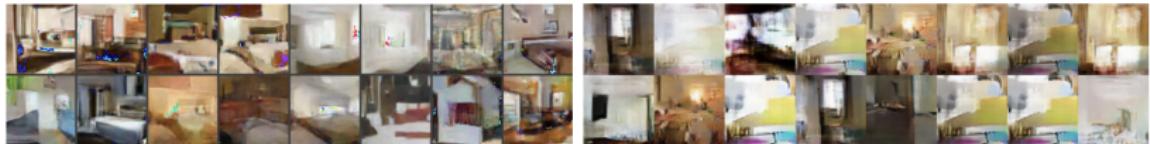
- ▶ JSD correlates poorly with the sample quality. Stays constant nearly maximum value $\log 2 \approx 0.69$.
- ▶ W is highly correlated with the sample quality.

Wasserstein GAN

WGAN converged without batch norm and constant number of filters



"In no experiment did we see evidence of mode collapse for the WGAN algorithm."



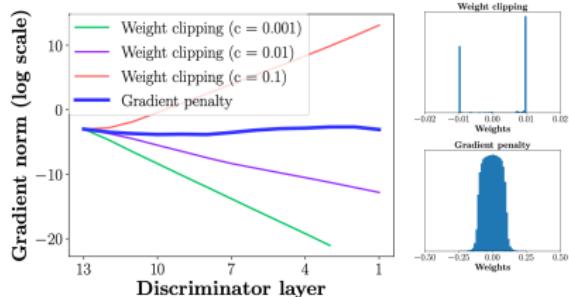
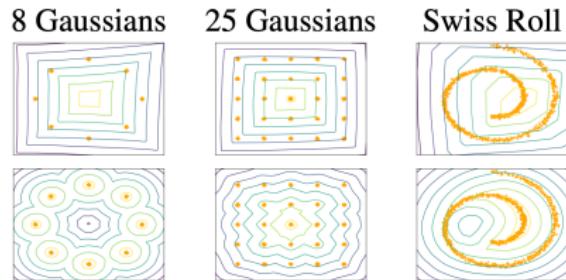
Wasserstein GAN with Gradient Penalty

The generator distribution is fixed and equal to the real distribution + Gaussian noise.

Problems with weight clipping:

- ▶ The critic ignores higher moments of the data distribution.
- ▶ The gradients either grow or decay exponentially.

Gradient penalty makes the gradients more stable.



Wasserstein GAN with Gradient Penalty

Theorem

Let $\pi(\mathbf{x})$ and $p(\mathbf{x})$ be two distribution in \mathcal{X} , a compact metric space. Then, there is 1-Lipschitz function f^* which is the optimal solution of

$$\max_{\|f\|_L \leq 1} [\mathbb{E}_{\mathbf{x} \sim \pi} f(\mathbf{x}) - \mathbb{E}_{\mathbf{x} \sim p} f(\mathbf{x})].$$

Let γ be the optimal transportation plan between $\pi(\mathbf{x})$ and $p(\mathbf{x})$. Then, if f^* is differentiable $\gamma(x = y) = 0$ and $\mathbf{x}_t = t\mathbf{x} + (1 - t)\mathbf{y}$ with $t \in [0, 1]$ it holds that

$$\mathbb{P}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \left[\nabla f^*(\mathbf{x}_t) = \frac{\mathbf{y} - \mathbf{x}_t}{\|\mathbf{y} - \mathbf{x}_t\|} \right] = 1.$$

Corollary

f^* has gradient norm 1 almost everywhere under $\pi(\mathbf{x})$ and $p(\mathbf{x})$.

Wasserstein GAN with Gradient Penalty

A differentiable function is 1-Lipschitz if and only if it has gradients with norm at most 1 everywhere.

Gradient penalty

$$W(\pi || p) = \underbrace{\mathbb{E}_{\mathbf{x} \sim \pi} f(\mathbf{x}) - \mathbb{E}_{\mathbf{x} \sim p} f(\mathbf{x})}_{\text{original critic loss}} + \lambda \underbrace{\mathbb{E}_{\hat{\mathbf{x}}} \left[(\|\nabla_{\hat{\mathbf{x}}} f(\hat{\mathbf{x}})\|_2 - 1)^2 \right]}_{\text{gradient penalty}},$$

- ▶ Samples $\hat{\mathbf{x}}$ are uniformly sampled along straight lines between pairs of points from the data distribution $\pi(\mathbf{x})$ and the generator distribution $p(\mathbf{x}|\theta)$.
- ▶ Enforcing the unit gradient norm constraint everywhere is intractable, it turns out to be sufficient to enforce it only along these straight lines.

Wasserstein GAN with Gradient Penalty

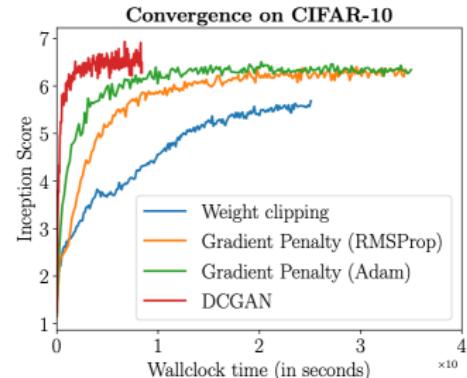
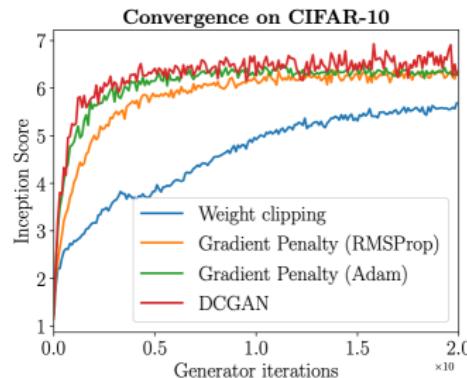
Algorithm 1 WGAN with gradient penalty. We use default values of $\lambda = 10$, $n_{\text{critic}} = 5$, $\alpha = 0.0001$, $\beta_1 = 0$, $\beta_2 = 0.9$.

Require: The gradient penalty coefficient λ , the number of critic iterations per generator iteration n_{critic} , the batch size m , Adam hyperparameters α, β_1, β_2 .

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

```
1: while  $\theta$  has not converged do
2:   for  $t = 1, \dots, n_{\text{critic}}$  do
3:     for  $i = 1, \dots, m$  do
4:       Sample real data  $\mathbf{x} \sim \mathbb{P}_r$ , latent variable  $\mathbf{z} \sim p(\mathbf{z})$ , a random number  $\epsilon \sim U[0, 1]$ .
5:        $\tilde{\mathbf{x}} \leftarrow G_\theta(\mathbf{z})$ 
6:        $\hat{\mathbf{x}} \leftarrow \epsilon \mathbf{x} + (1 - \epsilon) \tilde{\mathbf{x}}$ 
7:        $L^{(i)} \leftarrow D_w(\tilde{\mathbf{x}}) - D_w(\mathbf{x}) + \lambda(\|\nabla_{\hat{\mathbf{x}}} D_w(\hat{\mathbf{x}})\|_2 - 1)^2$ 
8:     end for
9:      $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^m L^{(i)}, w, \alpha, \beta_1, \beta_2)$ 
10:   end for
11:   Sample a batch of latent variables  $\{\mathbf{z}^{(i)}\}_{i=1}^m \sim p(\mathbf{z})$ .
12:    $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^m -D_w(G_\theta(\mathbf{z})), \theta, \alpha, \beta_1, \beta_2)$ 
13: end while
```

Wasserstein GAN with Gradient Penalty



Nonlinearity (G)	[ReLU, LeakyReLU, $\frac{\text{softplus}(2x+2)}{2} - 1$, tanh]
Nonlinearity (D)	[ReLU, LeakyReLU, $\frac{\text{softplus}(2x+2)}{2} - 1$, tanh]
Depth (G)	[4, 8, 12, 20]
Depth (D)	[4, 8, 12, 20]
Batch norm (G)	[True, False]
Batch norm (D ; layer norm for WGAN-GP)	[True, False]
Base filter count (G)	[32, 64, 128]
Base filter count (D)	[32, 64, 128]

Min. score	Only GAN	Only WGAN-GP	Both succeeded	Both failed
1.0	0	8	192	0
3.0	1	88	110	1
5.0	0	147	42	11
7.0	1	104	5	90
9.0	0	0	0	200

Wasserstein GAN with Gradient Penalty

DCGAN	LSGAN	WGAN (clipping)	WGAN-GP (ours)	
Baseline (G : DCGAN, D : DCGAN)				
G : No BN and a constant number of filters, D : DCGAN				
G : 4-layer 512-dim ReLU MLP, D : DCGAN				
No normalization in either G or D				
Gated multiplicative nonlinearities everywhere in G and D				
tanh nonlinearities everywhere in G and D				
101-layer ResNet G and D				

Spectral Normalization GAN

How else could we enforce Lipschitzness?

Fact 1

$$\|\mathbf{g}\|_L = \sup_{\mathbf{x}} \sigma(\nabla \mathbf{g}(\mathbf{x}))$$

Here $\sigma(\mathbf{A})$ – spectral norm of matrix \mathbf{A} .

$$\sigma(\mathbf{A}) = \max_{\mathbf{h} \neq 0} \frac{\|\mathbf{Ah}\|_2}{\|\mathbf{h}\|_2} = \max_{\|\mathbf{h}\|_2 \leq 1} \|\mathbf{Ah}\|_2 = \lambda_{\max}(\mathbf{A}),$$

where $\lambda_{\max}(\mathbf{A})$ is the largest singular value of \mathbf{A} .

Fact 2

$$\|\mathbf{g}_1 \circ \mathbf{g}_2\|_L \leq \|\mathbf{g}_1\|_L \cdot \|\mathbf{g}_2\|_L$$

Spectral Normalization GAN

Let consider the critic $f(\mathbf{x}, \phi)$ of the following form:

$$f(\mathbf{x}, \phi) = \mathbf{W}_{K+1} a_K (\mathbf{W}_K a_{K-1} (\dots a_1 (\mathbf{W}_1 \mathbf{x}) \dots)).$$

This feedforward network is a superposition of simple functions.

- ▶ a_k is a pointwise nonlinearities. We assume that $\|a_k\|_L = 1$ (it holds for ReLU).
- ▶ $\mathbf{g}(\mathbf{h}) = \mathbf{W}\mathbf{h}$ is a linear transformation.

$$\|\mathbf{g}\|_L = \sup_{\mathbf{x}} \sigma(\nabla \mathbf{g}(\mathbf{x})) = \sigma(\mathbf{W}).$$

Critic spectral norm

$$\|f\|_L \leq \prod_{k=1}^{K+1} \sigma(\mathbf{W}_k).$$

If we replace the weights in the critic by $\mathbf{W}_k^{SN} = \mathbf{W}_k / \sigma(\mathbf{W}_k)$, we will get $\|f\|_L \leq 1$.

Spectral Normalization GAN

If we apply singular value decomposition to compute the $\sigma(\mathbf{W})$ at each round of the algorithm, the algorithm becomes computationally heavy.

Power iteration

- ▶ \mathbf{u} – random vector.
- ▶ repeat

$$\mathbf{v} = \frac{\mathbf{W}^T \mathbf{u}}{\|\mathbf{W}^T \mathbf{u}\|}, \quad \mathbf{u} = \frac{\mathbf{Wv}}{\|\mathbf{Wv}\|}$$

- ▶ approximate the spectral norm

$$\sigma(\mathbf{W}) \approx \mathbf{u}^T \mathbf{Wv}$$

Spectral Normalization GAN

Algorithm 1 SGD with spectral normalization

- Initialize $\tilde{\mathbf{u}}_l \in \mathcal{R}^{d_l}$ for $l = 1, \dots, L$ with a random vector (sampled from isotropic distribution).
- For each update and each layer l :
 1. Apply power iteration method to a unnormalized weight W^l :

$$\tilde{\mathbf{v}}_l \leftarrow (W^l)^T \tilde{\mathbf{u}}_l / \| (W^l)^T \tilde{\mathbf{u}}_l \|_2 \quad (20)$$

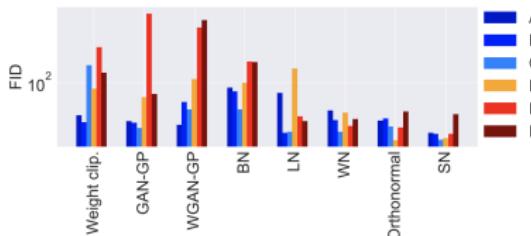
$$\tilde{\mathbf{u}}_l \leftarrow W^l \tilde{\mathbf{v}}_l / \| W^l \tilde{\mathbf{v}}_l \|_2 \quad (21)$$

2. Calculate \bar{W}_{SN} with the spectral norm:

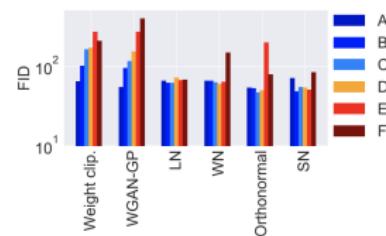
$$\bar{W}_{\text{SN}}^l(W^l) = W^l / \sigma(W^l), \text{ where } \sigma(W^l) = \tilde{\mathbf{u}}_l^T W^l \tilde{\mathbf{v}}_l \quad (22)$$

3. Update W^l with SGD on mini-batch dataset \mathcal{D}_M with a learning rate α :

$$W^l \leftarrow W^l - \alpha \nabla_{W^l} \ell(\bar{W}_{\text{SN}}^l(W^l), \mathcal{D}_M) \quad (23)$$



(a) CIFAR-10



(b) STL-10

Divergences

What do we have?

- ▶ Forward KL divergence in maximum likelihood estimation
- ▶ Reverse KL in variational inference
- ▶ JS divergence in vanilla gan
- ▶ Wasserstein distance in WGAN

Divergence minimization

$$\min_p D(\pi || p)$$

What is a divergence?

Let \mathcal{S} be the set of all possible probability distributions. Then $D : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$ is a divergence if

- ▶ $D(\pi || p) \geq 0$ for all $\pi, p \in \mathcal{S}$;
- ▶ $D(\pi || p) = 0$ if and only if $\pi \equiv p$.

f-divergence family

f-divergence

$$D_f(\pi || p) = \mathbb{E}_p f\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\right) = \int p(\mathbf{x}) f\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\right) d\mathbf{x}.$$

Here $f : \mathbb{R}_+ \rightarrow \mathbb{R}$ is a convex, lower-semicontinuous function satisfying $f(1) = 0$.

Fenchel conjugate

$$f^*(t) = \sup_{u \in \text{dom}_f} (ut - f(u)), \quad f^{**} = f, \quad f(u) = \sup_{t \in \text{dom}_{f^*}} (ut - f^*(t))$$

Name	$D_f(P Q)$	Generator $f(u)$	$T^*(x)$
Kullback-Leibler	$\int p(x) \log \frac{p(x)}{q(x)} dx$	$u \log u$	$1 + \log \frac{p(x)}{q(x)}$
Reverse KL	$\int q(x) \log \frac{q(x)}{p(x)} dx$	$-\log u$	$-\frac{q(x)}{p(x)}$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)} dx$	$(u-1)^2$	$2\left(\frac{p(x)}{q(x)} - 1\right)$
Squared Hellinger	$\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx$	$(\sqrt{u}-1)^2$	$(\sqrt{\frac{p(x)}{q(x)}} - 1) \cdot \sqrt{\frac{q(x)}{p(x)}}$
Jensen-Shannon	$\frac{1}{2} \int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} dx$	$-(u+1) \log \frac{1+u}{2} + u \log u$	$\log \frac{2p(x)}{p(x)+q(x)}$
GAN	$\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} dx - \log(4)$	$u \log u - (u+1) \log(u+1)$	$\log \frac{p(x)}{p(x)+q(x)}$

f-divergence family

Variational divergence estimation

$$\begin{aligned} D_f(\pi || p) &= \int p(\mathbf{x}) f\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\right) d\mathbf{x} \\ &= \int p(\mathbf{x}) \sup_{t \in \text{dom}_{f^*}} \left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})} t - f^*(t) \right) d\mathbf{x} \\ &= \int \sup_{t \in \text{dom}_{f^*}} (\pi(\mathbf{x}) t - p(\mathbf{x}) f^*(t)) d\mathbf{x} \\ &\geq \sup_{T \in \mathcal{T}} \int (\pi(\mathbf{x}) T(\mathbf{x}) - p(\mathbf{x}) f^*(T(\mathbf{x}))) d\mathbf{x} \\ &= \sup_{T \in \mathcal{T}} [\mathbb{E}_\pi T(\mathbf{x}) - \mathbb{E}_p f^*(T(\mathbf{x}))] \end{aligned}$$

Here $\mathcal{T} : \mathcal{X} \rightarrow \mathbb{R}$ is an arbitrary class of functions.

The lower bound is tight for $T^*(\mathbf{x}) = f'\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\right)$.

f-divergence family

Variational divergence estimation

$$D_f(\pi || p) \geq \sup_{T \in \mathcal{T}} [\mathbb{E}_\pi T(\mathbf{x}) - \mathbb{E}_p f^*(T(\mathbf{x}))]$$



(a) GAN

(b) KL

(c) Squared Hellinger

Summary

- ▶ Mode collapse and vanishing gradients are the two main problems of vanilla GAN.
- ▶ Lots of tips and tricks has to be used to make the GAN training is stable and scalable.
- ▶ Wasserstein distance is more appropriate objective function for distribution matching problem.