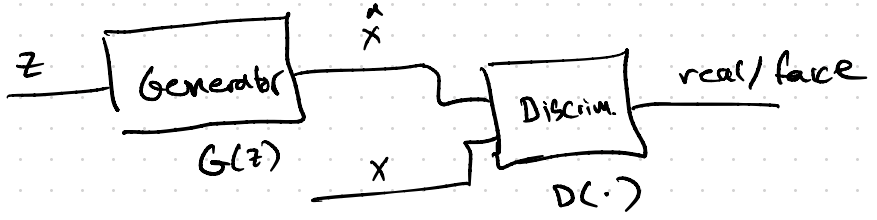
GAN



Original mini-max loss GANs (Goodfellow et al. 2014)

$$\min_G \max_D \; \mathbb{E}_{x \sim p_r}[\log D(x)] + \mathbb{E}_{z \sim p_f}[\log(1 - D(G(z)))]$$

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

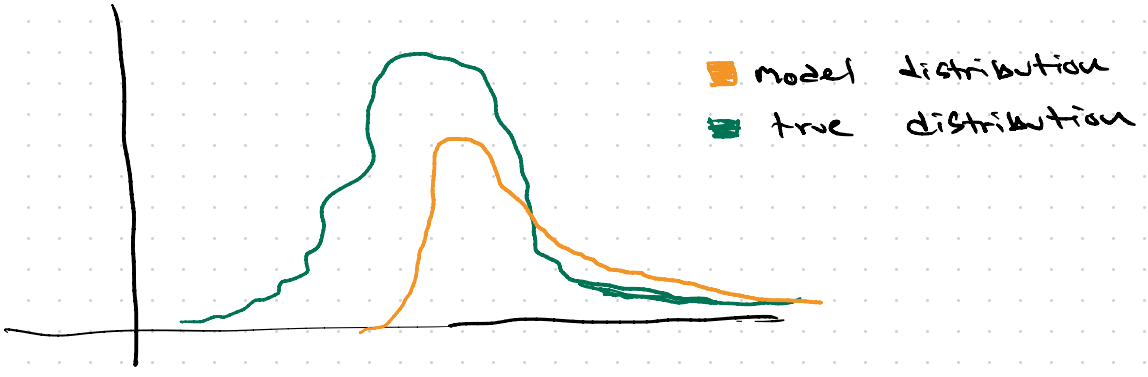
  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.



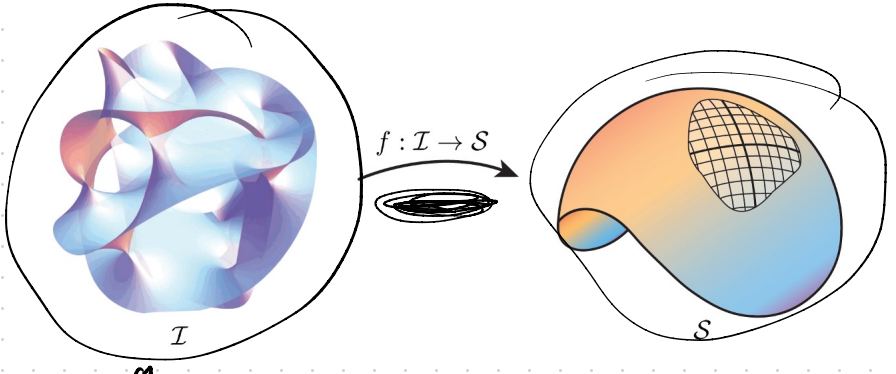
- Classic approach: compute statistics on real data to try and estimate parameters of the true distribution

  ⇒ Problem: parameters can be a very loose description of the underlying distribution.

  ⇒ hard to estimate the full distribution well.

- GANs short-circuit this whole paradigm... !

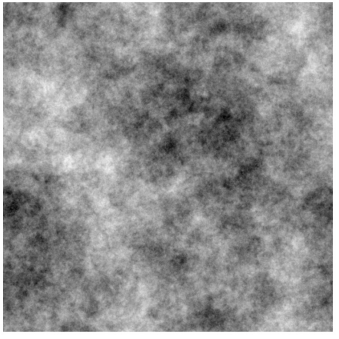
instead, they define an auxiliary task (fooling the discriminator).



Cartoon of natural image manifold

1/*f* noise

(what the world looks like if all you care about are pairwise correlations)



An example of how GANs are changing the approach to image compression

hific.GitHub.io

Simple statistics don't capture the distribution

Mode collapse Problem

- Generator only cares about fooling the discriminator
  ⇒ if the discriminator is bad only on specific "modes" the generator will focus only on these modes

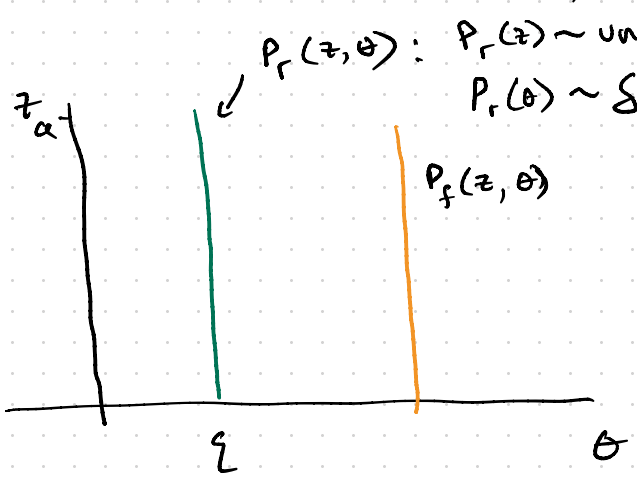
- Vanishing gradient. The discriminator should be making some errors initially in order for the generator to have a gradient to train on

Problems with many distance metrics

Arjovsky - 2017

↑

ex. from



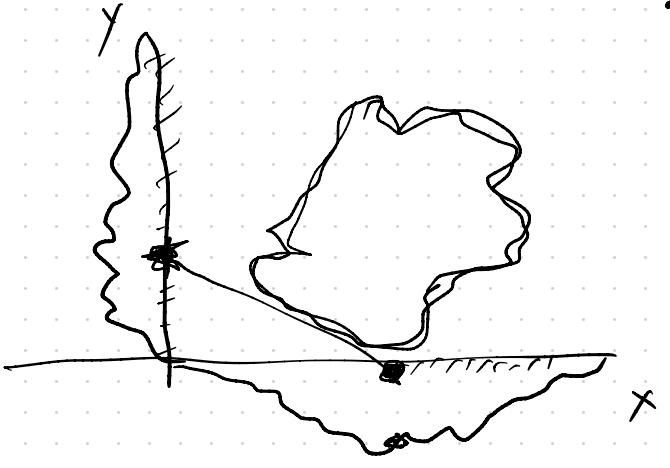
- K-L divergence: $KL(P_r, P_f) = \begin{cases} +\infty & q \neq \theta \\ 0 & \text{otherwise} \end{cases}$
- Total variation distance

- Earthmover's distance (AKA Wassertien metric)

$W(P_r, P_f) = |q - \theta|$ ← This is more informative! (And friendly to gradient descent)

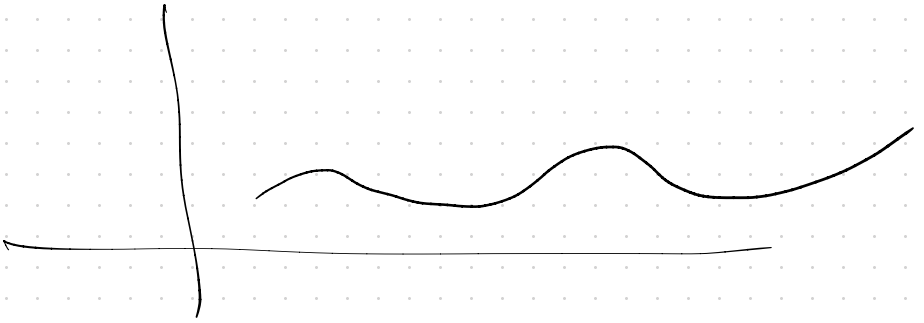
$$W(P_r, P_f) := \inf_{\gamma \in \Pi(P_r, P_f)} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x - y\|\right]$$

↖ Joint distributions that have marginals P_r, P_f



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

maximization over all 1-lipschitz functions



The point: It's really hard to compute this in high dimensions. We have to make an approximation.

$$\min_G \max_D \; \mathbb{E}_{x \sim p_r}[C(x)] - \mathbb{E}_{z \sim p_f}[C(z)]$$

- "Wasserstein Gans work because they fail"

arxiv.org/pdf/2103.01678.pdf

Gist: The wasserstein GANs use an approximation to the wasserstein metric that is really bad. Their empirically-good performance can't be explained by using the wasserstein metric, it's something else.

- likely the regularization of the discriminator.

Algorithm 1: WGAN-GP

Input: N_G - number of generator updates, N_D - number of discriminator updates per one generator update, λ - gradient penalty regularisation parameter

for N_G *iterations* **do**

  for N_D *iterations* **do**

    Sample a batch p_n^* from p^*

    Sample a batch p_n^θ from p^θ

    Ascent α wrt.

    $$\mathcal{L}_D(\alpha) := \mathcal{V}(D_\alpha, p_n^*, p_n^\theta) - \lambda \mathcal{R}(D_\alpha, p_n^*, p_n^\theta)$$

  end

  Sample a batch p_n^* from p^*

  Sample a batch p_n^θ from p^θ

  Descent θ wrt. $\mathcal{L}_G(\theta) := \mathcal{V}(D_\alpha, p_n^*, p_n^\theta)$

end

$$\mathcal{V}(D_\alpha, p_n^*, p_n^\theta) := \mathbb{E}_{x\sim p_n^*}[D_\alpha(x)] - \mathbb{E}_{x\sim p_n^\theta}[D_\alpha(x)]$$

↓ critic ↓ critic

$$\mathcal{R}(D_\alpha, p_n^*, p_n^\theta) := \mathbb{E}_{x\sim\tau}[(\|\nabla_x D_\alpha(x)\| - 1)^2]$$

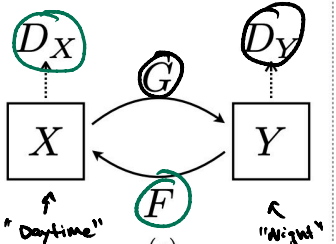
↑

approximately enforce 1-Lipschitz

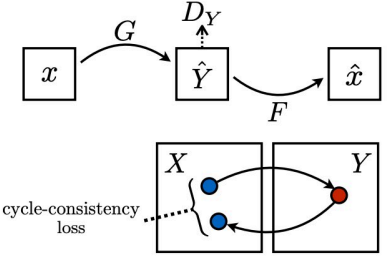
↓

"smoothness"

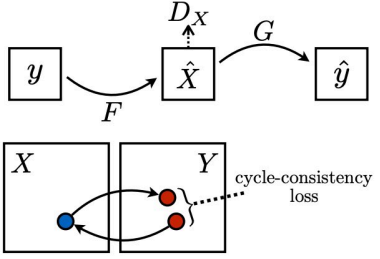
Cycle GANs


D_X
D_Y
G
X
Y
F
"Daytime"
"Night"

(a)


x
G
D_Y
$\hat{Y}$
F
$\hat{x}$
cycle-consistency loss
X
Y

(b)


y
F
D_X
$\hat{X}$
G
$\hat{y}$
X
Y
cycle-consistency loss

(c)

original GAN loss

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))]$$

$\hat{x}$

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$$

$\hat{y}$

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F),$$

$$G^*, F^* = \arg \min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y)$$