Deep Generative Models

Lecture 5

Roman Isachenko

Moscow Institute of Physics and Technology
Yandex School of Data Analysis

2025, Autumn

Recap of Previous Lecture

Assumptions

- Let $c \sim \text{Categorical}(\boldsymbol{\pi})$, where
$$\boldsymbol{\pi} = (\pi_1, \dots, \pi_K), \quad \pi_k = P(c = k), \quad \sum_{k=1}^{K} \pi_k = 1.$$
- Suppose the VAE includes a discrete latent variable c with prior $p(c) = \text{Uniform}\{1, \dots, K\}$.

ELBO

$$\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(c|\mathbf{x})} \log p_{\boldsymbol{\theta}}(\mathbf{x}|c) - \mathrm{KL}(q_{\boldsymbol{\phi}}(c|\mathbf{x}) \| p(c)) \to \max_{\boldsymbol{\phi},\boldsymbol{\theta}}.$$
$$\mathrm{KL}(q_{\boldsymbol{\phi}}(c|\mathbf{x}) \| p(c)) = -\mathrm{H}(q_{\boldsymbol{\phi}}(c|\mathbf{x})) + \log K.$$

- Our encoder must output the discrete distribution $q_{\boldsymbol{\phi}}(c|\mathbf{x})$.
- We'll require an analogue of the reparameterization trick for discrete $q_{\boldsymbol{\phi}}(c|\mathbf{x})$.
- Our decoder $p_{\boldsymbol{\theta}}(\mathbf{x}|c)$ has input the discrete variable c.

Oord A., Vinyals O., Kavukcuoglu K. Neural Discrete Representation Learning, 2017

Recap of Previous Lecture

Assumptions

- Let $c \sim \mathrm{Categorical}(\boldsymbol{\pi})$, where
$$\boldsymbol{\pi} = (\pi_1, \dots, \pi_K), \quad \pi_k = P(c = k), \quad \sum_{k=1}^{K} \pi_k = 1.$$
- Suppose the VAE employs a discrete latent code c, with prior $p(c) = \text{Uniform}\{1, \dots, K\}$.

ELBO

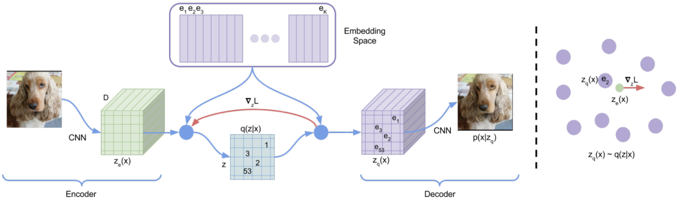
$$\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{q_{\phi}(c|\mathbf{x})} \log p_{\boldsymbol{\theta}}(\mathbf{x}|c) - \mathrm{KL}(q_{\phi}(c|\mathbf{x}) \,\|\, p(c)) \to \max_{\boldsymbol{\phi},\boldsymbol{\theta}}.$$
$$\mathrm{KL}(q_{\phi}(c|\mathbf{x}) \,\|\, p(c)) = -\mathrm{H}(q_{\phi}(c|\mathbf{x})) + \log K.$$

Vector Quantization

Define the codebook $\{\mathbf{e}_k\}_{k=1}^{K}$, where $\mathbf{e}_k \in \mathbb{R}^L$ and K is the size of the dictionary.
$$\mathbf{z}_q = \mathbf{q}(\mathbf{z}) = \mathbf{e}_{k^*}, \quad \text{where } k^* = \arg\min_k \|\mathbf{z} - \mathbf{e}_k\|.$$

Oord A., Vinyals O., Kavukcuoglu K. Neural Discrete Representation Learning, 2017

Recap of Previous Lecture



Deterministic Variational Posterior

$$q_\phi(c_{ij} = k^*|\mathbf{x}) = \begin{cases} 1, & \text{if } \; k^* = \arg\min_k \|[\mathbf{z}_e]_{ij} - \mathbf{e}_k\|; \\ 0, & \text{otherwise.} \end{cases}$$

ELBO

$$\mathcal{L}_{\phi,\theta}(\mathbf{x}) = \mathbb{E}_{q_\phi(c|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{e}_c) - \log K = \log p_\theta(\mathbf{x}|\mathbf{z}_q) - \log K.$$

Straight-Through Gradient Estimation

$$\frac{\partial \log p(\mathbf{x}|\mathbf{z}_q, \boldsymbol{\theta})}{\partial \phi} = \frac{\partial \log p_\theta(\mathbf{x}|\mathbf{z}_q)}{\partial \mathbf{z}_q} \cdot {\color{red}\frac{\partial \mathbf{z}_q}{\partial \phi}} \approx \frac{\partial \log p_\theta(\mathbf{x}|\mathbf{z}_q)}{\partial \mathbf{z}_q} \cdot \frac{\partial \mathbf{z}_e}{\partial \phi}$$

Oord A., Vinyals O., Kavukcuoglu K. Neural Discrete Representation Learning, 2017

Recap of Previous Lecture

Theorem

$$\frac{1}{n}\sum_{i=1}^{n} \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i) \,\|\, p(\mathbf{z})) = \mathrm{KL}(q_{\text{agg},\phi}(\mathbf{z}) \,\|\, p(\mathbf{z})) + \mathbb{I}_q[\mathbf{x}, \mathbf{z}].$$

ELBO Surgery

$$\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{\phi,\theta}(\mathbf{x}_i) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)} \log p_\theta(\mathbf{x}_i|\mathbf{z})}_{\text{Reconstruction Loss}} - \underbrace{\mathbb{I}_q[\mathbf{x}, \mathbf{z}]}_{\text{MI}} - \underbrace{\mathrm{KL}(q_{\text{agg},\phi}(\mathbf{z}) \,\|\, p(\mathbf{z}))}_{\text{Marginal KL}}$$

Optimal Prior

$$\mathrm{KL}(q_{\text{agg},\phi}(\mathbf{z}) \,\|\, p(\mathbf{z})) = 0 \;\Leftrightarrow\; p(\mathbf{z}) = q_{\text{agg}}(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} q_\phi(\mathbf{z}|\mathbf{x}_i).$$

Thus, the optimal prior distribution $p(\mathbf{z})$ is the aggregated variational posterior $q_{\text{agg},\phi}(\mathbf{z})$.

Hoffman M. D., Johnson M. J. ELBO surgery: yet another way to carve up the variational evidence lower bound, 2016

Recap of Previous Lecture

- Standard Gaussian $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I}) \Rightarrow$ over-regularization.
- $p(\mathbf{z}) = q_{\text{agg},\phi}(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} q_\phi(\mathbf{z}|\mathbf{x}_i) \Rightarrow$ overfitting and extremely high computational cost.

Revisiting ELBO

$$\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x}_i) = \text{RL} - \text{MI} - \text{KL}(q_{\text{agg},\phi}(\mathbf{z}) \,\|\, p_{\boldsymbol{\lambda}}(\mathbf{z}))$$

This is the forward KL divergence with respect to $p_{\boldsymbol{\lambda}}(\mathbf{z})$.

ELBO with Learnable VAE Prior

$$\mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) + \log p_{\boldsymbol{\lambda}}(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right]$$
$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) + \underbrace{\Big(\log p(f_{\boldsymbol{\lambda}}(\mathbf{z})) + \log|\det(\mathbf{J}_\mathbf{f})|\Big)}_{\text{flow-based prior}} - \log q_\phi(\mathbf{z}|\mathbf{x})\Big]$$

$$\mathbf{z} = \mathbf{f}_{\boldsymbol{\lambda}}^{-1}(\mathbf{z}^*) = \mathbf{g}_{\boldsymbol{\lambda}}(\mathbf{z}^*), \quad \mathbf{z}^* \sim p(\mathbf{z}^*) = \mathcal{N}(0, \mathbf{I})$$

Chen X. et al. Variational Lossy Autoencoder, 2016

Outline

1. Likelihood-Free Learning
2. Generative Adversarial Networks (GAN)
3. Wasserstein Distance
4. Wasserstein GAN

Outline

1. Likelihood-Free Learning
2. Generative Adversarial Networks (GAN)
3. Wasserstein Distance
4. Wasserstein GAN

Outline

1. Likelihood-Free Learning
2. Generative Adversarial Networks (GAN)
3. Wasserstein Distance
4. Wasserstein GAN

Likelihood-Based Models

Poor Likelihood
High-Quality Samples

$$p_1(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{N}(\mathbf{x}|\mathbf{x}_i, \epsilon\mathbf{I})$$

If ϵ is very small, this model produces excellent, sharp samples but achieves poor likelihoods on test data.

Theis L., Oord A., Bethge M. A Note on the Evaluation of Generative Models, 2015

Likelihood-Based Models

Poor Likelihood
High-Quality Samples

$$p_1(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{N}(\mathbf{x}|\mathbf{x}_i, \epsilon\mathbf{I})$$

If ϵ is very small, this model produces excellent, sharp samples but achieves poor likelihoods on test data.

High Likelihood
Poor Samples

$$p_2(\mathbf{x}) = 0.01p(\mathbf{x}) + 0.99p_{\text{noise}}(\mathbf{x})$$

$$\log\left[0.01p(\mathbf{x}) + 0.99p_{\text{noise}}(\mathbf{x})\right] \geq$$
$$\geq \log\left[0.01p(\mathbf{x})\right] = \log p(\mathbf{x}) - \log 100$$

This model contains mostly noisy, irrelevant samples; for high dimensions, $\log p(\mathbf{x})$ scales linearly with m.

Theis L., Oord A., Bethge M. A Note on the Evaluation of Generative Models, 2015

Likelihood-Based Models

Poor Likelihood
High-Quality Samples

$$p_1(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{N}(\mathbf{x}|\mathbf{x}_i, \epsilon\mathbf{I})$$

If ϵ is very small, this model produces excellent, sharp samples but achieves poor likelihoods on test data.

High Likelihood
Poor Samples

$$p_2(\mathbf{x}) = 0.01p(\mathbf{x}) + 0.99p_{\text{noise}}(\mathbf{x})$$

$$\log\left[0.01p(\mathbf{x}) + 0.99p_{\text{noise}}(\mathbf{x})\right] \geq$$
$$\geq \log\left[0.01p(\mathbf{x})\right] = \log p(\mathbf{x}) - \log 100$$

This model contains mostly noisy, irrelevant samples; for high dimensions, $\log p(\mathbf{x})$ scales linearly with m.

- Likelihood isn't always a suitable metric for evaluating generative models.
- Sometimes, the likelihood function can't even be computed exactly.

Theis L., Oord A., Bethge M. A Note on the Evaluation of Generative Models, 2015

Likelihood-Free Learning

Motivation

We're interested in approximating the true data distribution $p_{\text{data}}(\mathbf{x})$. Instead of searching over all distributions, let's learn a model $p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$.

Likelihood-Free Learning

Motivation

We're interested in approximating the true data distribution $p_{\text{data}}(\mathbf{x})$. Instead of searching over all distributions, let's learn a model $p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$.

Suppose we have two sets of samples:

- $\{\mathbf{x}_i\}_{i=1}^{n_1} \sim p_{\text{data}}(\mathbf{x})$ — real data;
- $\{\mathbf{x}_i\}_{i=1}^{n_2} \sim p_{\boldsymbol{\theta}}(\mathbf{x})$ — generated (fake) data.

Likelihood-Free Learning

Motivation

We're interested in approximating the true data distribution $p_{\text{data}}(\mathbf{x})$. Instead of searching over all distributions, let's learn a model $p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$.

Suppose we have two sets of samples:

- $\{\mathbf{x}_i\}_{i=1}^{n_1} \sim p_{\text{data}}(\mathbf{x})$ — real data;
- $\{\mathbf{x}_i\}_{i=1}^{n_2} \sim p_{\boldsymbol{\theta}}(\mathbf{x})$ — generated (fake) data.

Define a discriminative model (classifier):

$$p(y = 1|\mathbf{x}) = P(\mathbf{x} \sim p_{\text{data}}(\mathbf{x})); \quad p(y = 0|\mathbf{x}) = P(\mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x}))$$

Likelihood-Free Learning

Motivation

We're interested in approximating the true data distribution $p_{\text{data}}(\mathbf{x})$. Instead of searching over all distributions, let's learn a model $p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$.

Suppose we have two sets of samples:

- $\{\mathbf{x}_i\}_{i=1}^{n_1} \sim p_{\text{data}}(\mathbf{x})$ — real data;
- $\{\mathbf{x}_i\}_{i=1}^{n_2} \sim p_{\boldsymbol{\theta}}(\mathbf{x})$ — generated (fake) data.

Define a discriminative model (classifier):

$$p(y = 1|\mathbf{x}) = P(\mathbf{x} \sim p_{\text{data}}(\mathbf{x})); \quad p(y = 0|\mathbf{x}) = P(\mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x}))$$

Assumption

The generative model $p_{\boldsymbol{\theta}}(\mathbf{x})$ matches $p_{\text{data}}(\mathbf{x})$ if a discriminative model $p(y|\mathbf{x})$ can't distinguish between them — that is, if $p(y = 1|\mathbf{x}) = 0.5$ for every $\mathbf{x}$.

Generative Adversarial Networks (GAN)

- The more expressive the discriminator, the closer we get to the optimal $p_{\boldsymbol{\theta}}(\mathbf{x})$.
- Standard classifiers are trained by minimizing cross-entropy loss $-\mathbb{E}_{\hat{p}(\mathbf{x},y)} \log p(y|\mathbf{x})$ with $\hat{p}(\mathbf{x}, y) = \frac{1}{2}[y = 1]p_{\text{data}}(\mathbf{x}) + \frac{1}{2}[y = 0]p_{\boldsymbol{\theta}}(\mathbf{x})$.

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

Generative Adversarial Networks (GAN)

- The more expressive the discriminator, the closer we get to the optimal $p_{\boldsymbol{\theta}}(\mathbf{x})$.
- Standard classifiers are trained by minimizing cross-entropy loss $-\mathbb{E}_{\hat{p}(\mathbf{x},y)} \log p(y|\mathbf{x})$ with $\hat{p}(\mathbf{x}, y) = \frac{1}{2}[y = 1]p_{\text{data}}(\mathbf{x}) + \frac{1}{2}[y = 0]p_{\boldsymbol{\theta}}(\mathbf{x})$.

Cross-Entropy for Discriminator

$$\min_{p(y|\mathbf{x})} \left[-\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log p(y = 1|\mathbf{x}) - \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log p(y = 0|\mathbf{x})\right]$$

$$\max_{p(y|\mathbf{x})} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log p(y = 1|\mathbf{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log p(y = 0|\mathbf{x})\right]$$

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

Generative Adversarial Networks (GAN)

- The more expressive the discriminator, the closer we get to the optimal $p_{\boldsymbol{\theta}}(\mathbf{x})$.
- Standard classifiers are trained by minimizing cross-entropy loss $-\mathbb{E}_{\hat{p}(\mathbf{x},y)} \log p(y|\mathbf{x})$ with $\hat{p}(\mathbf{x}, y) = \frac{1}{2}[y = 1]p_{\text{data}}(\mathbf{x}) + \frac{1}{2}[y = 0]p_{\boldsymbol{\theta}}(\mathbf{x})$.

Cross-Entropy for Discriminator

$$\min_{p(y|\mathbf{x})} \left[-\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log p(y = 1|\mathbf{x}) - \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log p(y = 0|\mathbf{x})\right]$$

$$\max_{p(y|\mathbf{x})} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log p(y = 1|\mathbf{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log p(y = 0|\mathbf{x})\right]$$

Generative Model

Suppose $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, where $p(\mathbf{z})$ is a base distribution, and $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \delta(\mathbf{x} - \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z}))$ is deterministic.

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

Generative Adversarial Networks (GAN)

Cross-Entropy for Discriminative Model

$$\max_{p(y|\mathbf{x})} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log p(y = 1|\mathbf{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log p(y = 0|\mathbf{x})\right]$$

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

Generative Adversarial Networks (GAN)

Cross-Entropy for Discriminative Model

$$\max_{p(y|\mathbf{x})} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log p(y=1|\mathbf{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log p(y=0|\mathbf{x})\right]$$

- **Discriminator:** A classifier $p_\phi(y=1|\mathbf{x}) = D_\phi(\mathbf{x}) \in [0,1]$, distinguishing real and generated samples. The discriminator aims to **maximize** cross-entropy.
- **Generator:** The generative model $\mathbf{x} = \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$, $\mathbf{z} \sim p(\mathbf{z})$, seeks to fool the discriminator. The generator aims to **minimize** cross-entropy.

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

Generative Adversarial Networks (GAN)

Cross-Entropy for Discriminative Model

$$\max_{p(y|\mathbf{x})} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log p(y = 1|\mathbf{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log p(y = 0|\mathbf{x})\right]$$

- **Discriminator:** A classifier $p_{\boldsymbol{\phi}}(y = 1|\mathbf{x}) = D_{\boldsymbol{\phi}}(\mathbf{x}) \in [0, 1]$, distinguishing real and generated samples. The discriminator aims to **maximize** cross-entropy.
- **Generator:** The generative model $\mathbf{x} = \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$, $\mathbf{z} \sim p(\mathbf{z})$, seeks to fool the discriminator. The generator aims to **minimize** cross-entropy.

GAN Objective

$$\min_{G} \max_{D} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log(1 - D(\mathbf{x}))\right]$$

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

Generative Adversarial Networks (GAN)

Cross-Entropy for Discriminative Model

$$\max_{p(y|\mathbf{x})} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log p(y=1|\mathbf{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log p(y=0|\mathbf{x}) \right]$$

- **Discriminator:** A classifier $p_{\phi}(y=1|\mathbf{x}) = D_{\phi}(\mathbf{x}) \in [0,1]$, distinguishing real and generated samples. The discriminator aims to **maximize** cross-entropy.
- **Generator:** The generative model $\mathbf{x} = \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$, $\mathbf{z} \sim p(\mathbf{z})$, seeks to fool the discriminator. The generator aims to **minimize** cross-entropy.

GAN Objective

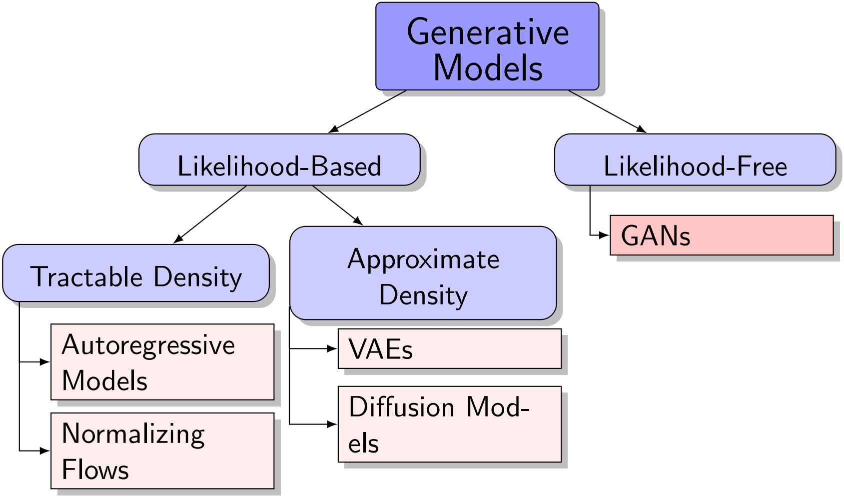
$$\min_{G} \max_{D} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log(1 - D(\mathbf{x})) \right]$$

$$\min_{G} \max_{D} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(\mathbf{G}(\mathbf{z}))) \right]$$

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

Outline

Generative Models Zoo



GAN Optimality

Theorem

The minimax game

$$\min_{G} \max_{D} \Big[\underbrace{\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(\mathbf{G}(\mathbf{z})))}_{V(G,D)} \Big]$$

achieves its global optimum when $p_{\text{data}}(\mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{x})$, and $D^*(\mathbf{x}) = 0.5$.

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

GAN Optimality

Theorem

The minimax game

$$\min_G \max_D \Big[\underbrace{\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(\mathbf{G}(\mathbf{z})))}_{V(G,D)} \Big]$$

achieves its global optimum when $p_{\text{data}}(\mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{x})$, and $D^*(\mathbf{x}) = 0.5$.

Proof (Fixed G)

$$V(G, D) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log(1 - D(\mathbf{x}))$$

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

GAN Optimality

Theorem

The minimax game

$$\min_G \max_D \Big[\underbrace{\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(\mathbf{G}(\mathbf{z})))}_{V(G,D)}\Big]$$

achieves its global optimum when $p_{\text{data}}(\mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{x})$, and $D^*(\mathbf{x}) = 0.5$.

Proof (Fixed G)

$$\begin{aligned} V(G, D) &= \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log(1 - D(\mathbf{x})) \\ &= \int \underbrace{[p_{\text{data}}(\mathbf{x}) \log D(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x}) \log(1 - D(\mathbf{x}))]}_{y(D)} d\mathbf{x} \end{aligned}$$

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

GAN Optimality

Theorem

The minimax game

$$\min_{G} \max_{D} \Big[\underbrace{\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(\mathbf{G}(\mathbf{z})))}_{V(G,D)} \Big]$$

achieves its global optimum when $p_{\text{data}}(\mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{x})$, and $D^*(\mathbf{x}) = 0.5$.

Proof (Fixed G)

$$\begin{aligned} V(G, D) &= \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log(1 - D(\mathbf{x})) \\ &= \int \underbrace{[p_{\text{data}}(\mathbf{x}) \log D(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x}) \log(1 - D(\mathbf{x}))]}_{y(D)} d\mathbf{x} \end{aligned}$$

$$\frac{dy(D)}{dD} = \frac{p_{\text{data}}(\mathbf{x})}{D(\mathbf{x})} - \frac{p_{\boldsymbol{\theta}}(\mathbf{x})}{1 - D(\mathbf{x})} = 0 \quad \Rightarrow \quad D^*(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})}$$

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

GAN Optimality

Proof Continued (Fixed $D = D^*$)

$$V(G, D^*) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log \left(\frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})} \right) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log \left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})} \right)$$

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

GAN Optimality

Proof Continued (Fixed $D = D^*$)

$$V(G, D^*) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log \left(\frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})} \right) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log \left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})} \right)$$

$$= \mathrm{KL} \left(p_{\text{data}}(\mathbf{x}) \parallel \frac{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})}{2} \right) + \mathrm{KL} \left(p_{\boldsymbol{\theta}}(\mathbf{x}) \parallel \frac{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})}{2} \right) - 2 \log 2$$

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

GAN Optimality

Proof Continued (Fixed $D = D^*$)

$$V(G, D^*) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log \left(\frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})} \right) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log \left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})} \right)$$

$$= \mathrm{KL} \left(p_{\text{data}}(\mathbf{x}) \,\|\, \frac{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})}{2} \right) + \mathrm{KL} \left(p_{\boldsymbol{\theta}}(\mathbf{x}) \,\|\, \frac{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})}{2} \right) - 2 \log 2$$

$$= 2\, \mathrm{JSD}(p_{\text{data}}(\mathbf{x}) \,\|\, p_{\boldsymbol{\theta}}(\mathbf{x})) - 2 \log 2.$$

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

GAN Optimality

Proof Continued (Fixed $D = D^*$)

$$V(G, D^*) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log \left(\frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})} \right) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log \left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})} \right)$$

$$= \mathrm{KL} \left(p_{\text{data}}(\mathbf{x}) \parallel \frac{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})}{2} \right) + \mathrm{KL} \left(p_{\boldsymbol{\theta}}(\mathbf{x}) \parallel \frac{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})}{2} \right) - 2 \log 2$$

$$= 2\, \mathrm{JSD}(p_{\text{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})) - 2 \log 2.$$

Jensen-Shannon Divergence (Symmetric KL Divergence)

$$\mathrm{JSD}(p_{\text{data}}(\mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{x})) = \frac{1}{2} \left[\mathrm{KL} \left(p_{\text{data}}(\mathbf{x}) \| \star \right) + \mathrm{KL} \left(p_{\boldsymbol{\theta}}(\mathbf{x}) \| \star \right) \right]$$

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

GAN Optimality

Proof Continued (Fixed $D = D^*$)

$$V(G, D^*) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log \left(\frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})} \right) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \log \left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})} \right)$$
$$= \mathrm{KL}\left(p_{\text{data}}(\mathbf{x}) \,\|\, \frac{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})}{2} \right) + \mathrm{KL}\left(p_{\boldsymbol{\theta}}(\mathbf{x}) \,\|\, \frac{p_{\text{data}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})}{2} \right) - 2\log 2$$
$$= 2\,\mathrm{JSD}(p_{\text{data}}(\mathbf{x}) \,\|\, p_{\boldsymbol{\theta}}(\mathbf{x})) - 2\log 2.$$

Jensen-Shannon Divergence (Symmetric KL Divergence)

$$\mathrm{JSD}(p_{\text{data}}(\mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{x})) = \frac{1}{2}\left[\mathrm{KL}\left(p_{\text{data}}(\mathbf{x}) \| \star\right) + \mathrm{KL}\left(p_{\boldsymbol{\theta}}(\mathbf{x}) \| \star\right)\right]$$

This can be regarded as a proper distance metric!

$$V(G^*, D^*) = -2\log 2, \quad p_{\text{data}}(\mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{x}), \quad D^*(\mathbf{x}) = 0.5.$$

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

GAN Optimality

Theorem

The following minimax game

$$\min_G \max_D \Big[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(\mathbf{G}(\mathbf{z}))) \Big]$$

achieves its global optimum precisely when $p_{\text{data}}(\mathbf{x}) = p_\theta(\mathbf{x})$, and $D^*(\mathbf{x}) = 0.5$.

Expectations

If the generator can express **any** function and the discriminator is **optimal** at every step, the generator **will converge** to the target distribution.

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

GAN Optimality

Theorem

The following minimax game

$$\min_G \max_D \Big[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(\mathbf{G}(\mathbf{z}))) \Big]$$

achieves its global optimum precisely when $p_{\text{data}}(\mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{x})$, and $D^*(\mathbf{x}) = 0.5$.

Expectations

If the generator can express **any** function and the discriminator is **optimal** at every step, the generator **will converge** to the target distribution.

Reality

- Generator updates are performed in parameter space, and the discriminator is often imperfectly optimized.
- Generator and discriminator losses typically oscillate during GAN training.

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

GAN Training

Assume both generator and discriminator are parametric models: $D_{\boldsymbol{\phi}}(\mathbf{x})$ and $\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$.

Objective

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D_{\boldsymbol{\phi}}(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D_{\boldsymbol{\phi}}(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})))\right]$$

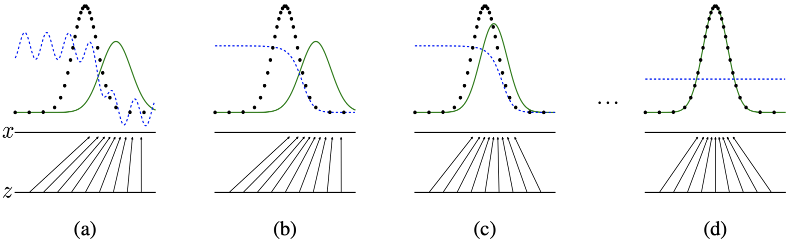
Goodfellow I. J. et al. Generative Adversarial Networks, 2014

GAN Training

Assume both generator and discriminator are parametric models: $D_{\boldsymbol{\phi}}(\mathbf{x})$ and $\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$.

Objective

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D_{\boldsymbol{\phi}}(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D_{\boldsymbol{\phi}}(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})))\right]$$



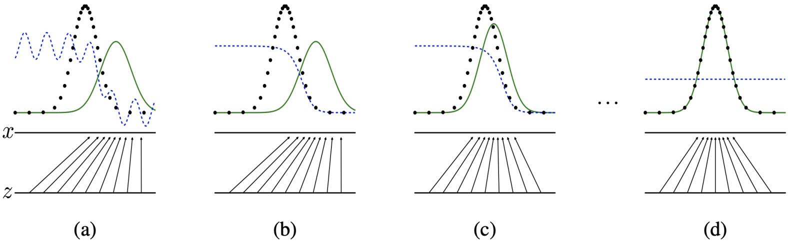
Goodfellow I. J. et al. Generative Adversarial Networks, 2014

GAN Training

Assume both generator and discriminator are parametric models: $D_{\phi}(\mathbf{x})$ and $\mathbf{G}_{\theta}(\mathbf{z})$.

Objective

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D_{\phi}(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D_{\phi}(\mathbf{G}_{\theta}(\mathbf{z}))) \right]$$

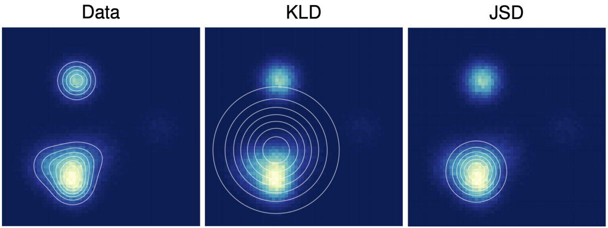


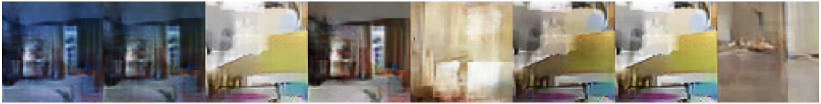
- $\mathbf{z} \sim p(\mathbf{z})$ is a latent variable.
- $p_{\theta}(\mathbf{x}|\mathbf{z}) = \delta(\mathbf{x} - \mathbf{G}_{\theta}(\mathbf{z}))$ serves as a deterministic decoder (like normalizing flows).
- There is no encoder present.

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

Mode Collapse

Mode collapse refers to the phenomenon where the generator in a GAN produces only one or a few different modes of the distribution.

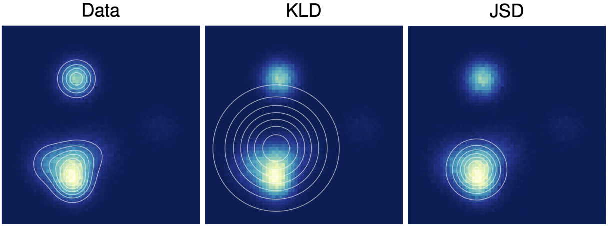


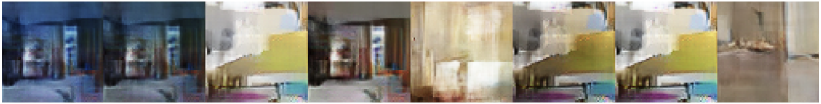


Goodfellow I. J. et al. Generative Adversarial Networks, 2014
Metz L. et al. Unrolled Generative Adversarial Networks, 2016

Mode Collapse

Mode collapse refers to the phenomenon where the generator in a GAN produces only one or a few different modes of the distribution.





Numerous methods have been proposed to tackle mode collapse: changing architectures, adding regularization terms, injecting noise.

Goodfellow I. J. et al. Generative Adversarial Networks, 2014
Metz L. et al. Unrolled Generative Adversarial Networks, 2016

Jensen-Shannon vs Kullback-Leibler Divergences

- $p_{\text{data}}(\mathbf{x})$ is a fixed mixture of two Gaussians.
- $p(\mathbf{x}|\mu,\sigma) = \mathcal{N}(\mu,\sigma^2)$.

Mode Covering vs. Mode Seeking

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

$$\mathrm{JSD}(\pi \parallel p) = \frac{1}{2}\Big[\mathrm{KL}\Big(\pi(\mathbf{x}) \parallel \frac{\pi(\mathbf{x}) + p(\mathbf{x})}{2}\Big) + \mathrm{KL}\Big(p(\mathbf{x}) \parallel \frac{\pi(\mathbf{x}) + p(\mathbf{x})}{2}\Big)\Big]$$

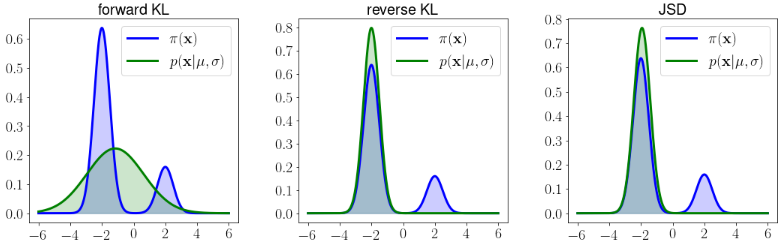
Jensen-Shannon vs Kullback-Leibler Divergences

- $p_{\text{data}}(\mathbf{x})$ is a fixed mixture of two Gaussians.
- $p(\mathbf{x}|\mu,\sigma) = \mathcal{N}(\mu,\sigma^2)$.

Mode Covering vs. Mode Seeking

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

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



Outline

1. Likelihood-Free Learning
2. Generative Adversarial Networks (GAN)
3. Wasserstein Distance
4. Wasserstein GAN

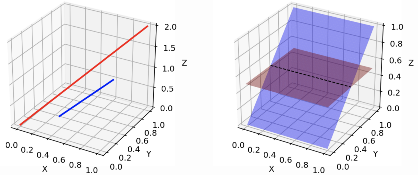
Theoretical Results

- The dimensionality of $\mathbf{z}$ is less than that of $\mathbf{x}$, so $p_{\boldsymbol{\theta}}(\mathbf{x})$ with $\mathbf{x} = \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$ lives on a low-dimensional manifold.

Weng L. From GAN to WGAN, 2019
Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017

Theoretical Results

- The dimensionality of $\mathbf{z}$ is less than that of $\mathbf{x}$, so $p_{\boldsymbol{\theta}}(\mathbf{x})$ with $\mathbf{x} = \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$ lives on a low-dimensional manifold.
- The true data distribution $p_{\text{data}}(\mathbf{x})$ is also supported on a low-dimensional manifold.

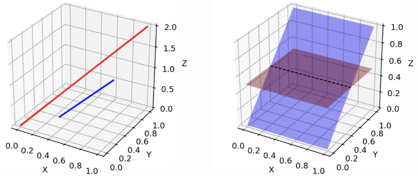


Weng L. *From GAN to WGAN*, 2019

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

Theoretical Results

- The dimensionality of $\mathbf{z}$ is less than that of $\mathbf{x}$, so $p_{\boldsymbol{\theta}}(\mathbf{x})$ with $\mathbf{x} = \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$ lives on a low-dimensional manifold.
- The true data distribution $p_{\text{data}}(\mathbf{x})$ is also supported on a low-dimensional manifold.

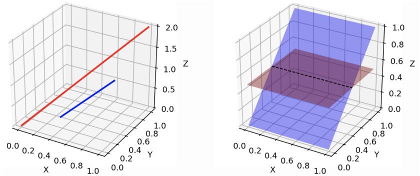


- If $p_{\text{data}}(\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{x})$ are disjoint, a smooth optimal discriminator can exist!

Weng L. From GAN to WGAN, 2019
Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017

Theoretical Results

- The dimensionality of $\mathbf{z}$ is less than that of $\mathbf{x}$, so $p_{\boldsymbol{\theta}}(\mathbf{x})$ with $\mathbf{x} = \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})$ lives on a low-dimensional manifold.
- The true data distribution $p_{\text{data}}(\mathbf{x})$ is also supported on a low-dimensional manifold.



- If $p_{\text{data}}(\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{x})$ are disjoint, a smooth optimal discriminator can exist!
- For such low-dimensional, disjoint manifolds:

$$\mathrm{KL}(p_{\text{data}} \,\|\, p_{\boldsymbol{\theta}}) = \mathrm{KL}(p_{\boldsymbol{\theta}} \,\|\, p_{\text{data}}) = \infty, \quad \mathrm{JSD}(p_{\text{data}} \,\|\, p_{\boldsymbol{\theta}}) = \log 2$$

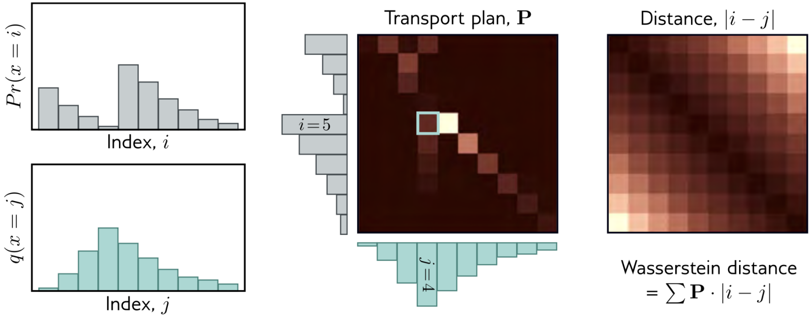
Weng L. From GAN to WGAN, 2019
Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017

Wasserstein Distance (Discrete)

Also known as the **Earth Mover's Distance**.

Optimal Transport Formulation

The minimum cost of moving and transforming a pile of "dirt" shaped like one probability distribution to match another.



Simon J.D. Prince. *Understanding Deep Learning*, 2023

Wasserstein Distance (Continuous)

$$W(\pi \| p) = \inf_{\gamma \in \Gamma(\pi, p)} \mathbb{E}_{(\mathbf{x}_1, \mathbf{x}_2) \sim \gamma} \|\mathbf{x}_1 - \mathbf{x}_2\| = \inf_{\gamma \in \Gamma(\pi, p)} \int \|\mathbf{x}_1 - \mathbf{x}_2\| \gamma(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2$$

- $\gamma(\mathbf{x}_1, \mathbf{x}_2)$ is the transport plan: the amount of "dirt" assigned from $\mathbf{x}_1$ to $\mathbf{x}_2$.

$$\int \gamma(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x} = p(\mathbf{x}_2); \quad \int \gamma(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2 = \pi(\mathbf{x}_1).$$

- $\Gamma(\pi, p)$ denotes the set of all joint distributions $\gamma(\mathbf{x}_1, \mathbf{x}_2)$ with marginals π and p.
- $\gamma(\mathbf{x}_1, \mathbf{x}_2)$ is the mass, $\|\mathbf{x}_1 - \mathbf{x}_2\|$ is the distance.

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein Distance (Continuous)

$$W(\pi \| p) = \inf_{\gamma \in \Gamma(\pi, p)} \mathbb{E}_{(\mathbf{x}_1, \mathbf{x}_2) \sim \gamma} \|\mathbf{x}_1 - \mathbf{x}_2\| = \inf_{\gamma \in \Gamma(\pi, p)} \int \|\mathbf{x}_1 - \mathbf{x}_2\| \, \gamma(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2$$

- $\gamma(\mathbf{x}_1, \mathbf{x}_2)$ is the transport plan: the amount of "dirt" assigned from $\mathbf{x}_1$ to $\mathbf{x}_2$.
$$\int \gamma(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x} = p(\mathbf{x}_2); \quad \int \gamma(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2 = \pi(\mathbf{x}_1).$$
- $\Gamma(\pi, p)$ denotes the set of all joint distributions $\gamma(\mathbf{x}_1, \mathbf{x}_2)$ with marginals π and p.
- $\gamma(\mathbf{x}_1, \mathbf{x}_2)$ is the mass, $\|\mathbf{x}_1 - \mathbf{x}_2\|$ is the distance.

Wasserstein Metric

$$W_s(\pi, p) = \inf_{\gamma \in \Gamma(\pi, p)} \left(\mathbb{E}_{(\mathbf{x}_1, \mathbf{x}_2) \sim \gamma} \|\mathbf{x}_1 - \mathbf{x}_2\|^s \right)^{1/s}$$

In our setting, $W(\pi \| p) = W_1(\pi, p)$, which is the transport cost using the ℓ_1 norm.

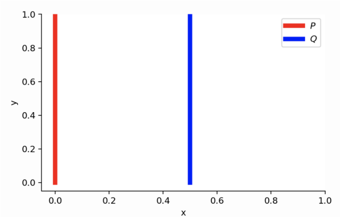
Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein Distance vs KL vs JSD

Consider two-dimensional distributions:

$$p_{\text{data}}(x, y) = (0, U[0, 1])$$

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



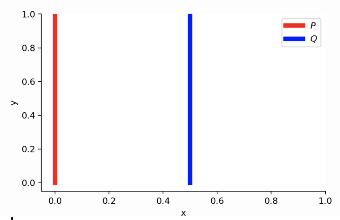
Weng L. From GAN to WGAN, 2019
Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein Distance vs KL vs JSD

Consider two-dimensional distributions:

$$p_{\text{data}}(x, y) = (0, U[0, 1])$$

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



- $\theta = 0$: Both distributions are identical.

$$\mathrm{KL}(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \mathrm{KL}(p_{\boldsymbol{\theta}} \| p_{\text{data}}) = \mathrm{JSD}(p_{\boldsymbol{\theta}} \| p_{\text{data}}) = W(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = 0$$

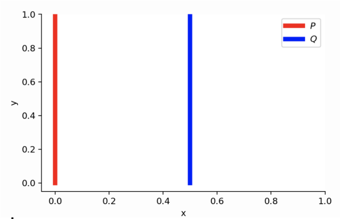
Weng L. From GAN to WGAN, 2019
Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein Distance vs KL vs JSD

Consider two-dimensional distributions:

$$p_{\text{data}}(x, y) = (0, U[0, 1])$$

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



- $\theta = 0$: Both distributions are identical.

$$\mathrm{KL}(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \mathrm{KL}(p_{\boldsymbol{\theta}} \| p_{\text{data}}) = \mathrm{JSD}(p_{\boldsymbol{\theta}} \| p_{\text{data}}) = W(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = 0$$

- $\theta \neq 0$:

$$\mathrm{KL}(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \int_{U[0,1]} 1 \log \frac{1}{0} \, dy = \infty = \mathrm{KL}(p_{\boldsymbol{\theta}} \| p_{\text{data}})$$

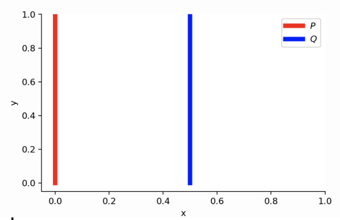
Weng L. From GAN to WGAN, 2019
Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein Distance vs KL vs JSD

Consider two-dimensional distributions:

$$p_{\text{data}}(x, y) = (0, U[0, 1])$$

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



- $\theta = 0$: Both distributions are identical.

$$\mathrm{KL}(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \mathrm{KL}(p_{\boldsymbol{\theta}} \| p_{\text{data}}) = \mathrm{JSD}(p_{\boldsymbol{\theta}} \| p_{\text{data}}) = W(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = 0$$

- $\theta \neq 0$:

$$\mathrm{KL}(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \int_{U[0,1]} 1 \log \frac{1}{0} \, dy = \infty = \mathrm{KL}(p_{\boldsymbol{\theta}} \| p_{\text{data}})$$

$$\mathrm{JSD}(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \frac{1}{2} \left(\int_{U[0,1]} 1 \log \frac{1}{1/2} dy + \int_{U[0,1]} 1 \log \frac{1}{1/2} dy \right) = \log 2$$

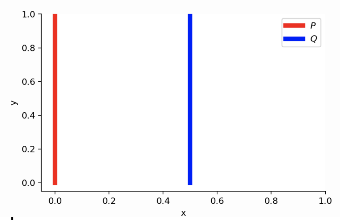
Weng L. From GAN to WGAN, 2019
Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein Distance vs KL vs JSD

Consider two-dimensional distributions:

$$p_{\text{data}}(x, y) = (0, U[0, 1])$$

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



- $\theta = 0$: Both distributions are identical.

$$\mathrm{KL}(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \mathrm{KL}(p_{\boldsymbol{\theta}} \| p_{\text{data}}) = \mathrm{JSD}(p_{\boldsymbol{\theta}} \| p_{\text{data}}) = W(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = 0$$

- $\theta \neq 0$:

$$\mathrm{KL}(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \int_{U[0,1]} 1 \log \frac{1}{0} \, dy = \infty = \mathrm{KL}(p_{\boldsymbol{\theta}} \| p_{\text{data}})$$

$$\mathrm{JSD}(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \frac{1}{2} \left(\int_{U[0,1]} 1 \log \frac{1}{1/2} dy + \int_{U[0,1]} 1 \log \frac{1}{1/2} dy \right) = \log 2$$

$$W(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = |\theta|$$

Weng L. From GAN to WGAN, 2019
Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein Distance vs KL vs JSD

Theorem 1

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

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein Distance vs KL vs JSD

Theorem 1

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

Theorem 2

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

$$\mathrm{KL}(\pi\|p_t) \to 0 \quad (\text{or } \mathrm{KL}(p_t \,\|\, \pi) \to 0) \tag{1}$$

$$\mathrm{JSD}(\pi\|p_t) \to 0 \tag{2}$$

$$W(\pi\|p_t) \to 0 \tag{3}$$

In summary, as $t \to \infty$, (1) $\Rightarrow$ (2), and (2) $\Rightarrow$ (3).

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Outline

1. Likelihood-Free Learning

2. Generative Adversarial Networks (GAN)

3. Wasserstein Distance

4. Wasserstein GAN

Wasserstein GAN

Wasserstein Distance

$$W(\pi \| p) = \inf_{\gamma \in \Gamma(\pi, p)} \mathbb{E}_{(\mathbf{x}_1, \mathbf{x}_2) \sim \gamma} \|\mathbf{x}_1 - \mathbf{x}_2\| = \inf_{\gamma \in \Gamma(\pi, p)} \int \|\mathbf{x}_1 - \mathbf{x}_2\| \gamma(\mathbf{x}_1, \mathbf{x}_2) \, d\mathbf{x}_1 \, d\mathbf{x}_2$$

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein GAN

Wasserstein Distance

$$W(\pi \| p) = \inf_{\gamma \in \Gamma(\pi, p)} \mathbb{E}_{(\mathbf{x}_1, \mathbf{x}_2) \sim \gamma} \|\mathbf{x}_1 - \mathbf{x}_2\| = \inf_{\gamma \in \Gamma(\pi, p)} \int \|\mathbf{x}_1 - \mathbf{x}_2\| \gamma(\mathbf{x}_1, \mathbf{x}_2) \, d\mathbf{x}_1 \, d\mathbf{x}_2$$

The infimum over all possible $\gamma \in \Gamma(\pi, p)$ is computationally intractable.

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein GAN

Wasserstein Distance

$$W(\pi \| p) = \inf_{\gamma \in \Gamma(\pi, p)} \mathbb{E}_{(\mathbf{x}_1, \mathbf{x}_2) \sim \gamma} \|\mathbf{x}_1 - \mathbf{x}_2\| = \inf_{\gamma \in \Gamma(\pi, p)} \int \|\mathbf{x}_1 - \mathbf{x}_2\| \gamma(\mathbf{x}_1, \mathbf{x}_2) \, d\mathbf{x}_1 \, d\mathbf{x}_2$$

The infimum over all possible $\gamma \in \Gamma(\pi, p)$ is computationally intractable.

Theorem (Kantorovich-Rubinstein Duality)

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

where $f : \mathbb{R}^m \to \mathbb{R}$ is K-Lipschitz ($\|f\|_L \leq K$):

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

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein GAN

Wasserstein Distance

$$W(\pi \| p) = \inf_{\gamma \in \Gamma(\pi, p)} \mathbb{E}_{(\mathbf{x}_1, \mathbf{x}_2) \sim \gamma} \|\mathbf{x}_1 - \mathbf{x}_2\| = \inf_{\gamma \in \Gamma(\pi, p)} \int \|\mathbf{x}_1 - \mathbf{x}_2\| \gamma(\mathbf{x}_1, \mathbf{x}_2) \, d\mathbf{x}_1 \, d\mathbf{x}_2$$

The infimum over all possible $\gamma \in \Gamma(\pi, p)$ is computationally intractable.

Theorem (Kantorovich-Rubinstein Duality)

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

where $f : \mathbb{R}^m \to \mathbb{R}$ is K-Lipschitz ($\|f\|_L \leq K$):

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

We can thus estimate $W(\pi \| p)$ using only samples and a function f.

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein GAN

Theorem (Kantorovich-Rubinstein Duality)

$$W(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \frac{1}{K} \max_{\|f\|_L \leq K} \Big[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} f(\mathbf{x}) \Big]$$

- We must ensure that f is K-Lipschitz continuous.
- Let $f_{\phi}(\mathbf{x})$ be a feedforward neural network parameterized by ϕ.
- If the weights ϕ are restricted to a compact set $\mathbf{\Phi}$, then f_{ϕ} is K-Lipschitz.

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein GAN

Theorem (Kantorovich-Rubinstein Duality)

$$W(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \frac{1}{K} \max_{\|f\|_L \leq K} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} f(\mathbf{x}) \right]$$

- We must ensure that f is K-Lipschitz continuous.
- Let $f_{\phi}(\mathbf{x})$ be a feedforward neural network parameterized by ϕ.
- If the weights ϕ are restricted to a compact set $\boldsymbol{\Phi}$, then f_{ϕ} is K-Lipschitz.
- Clamp weights within the box $\boldsymbol{\Phi} = [-c, c]^d$ (e.g. $c = 0.01$) after each update.

$$K \cdot W(p_{\text{data}} \| p_{\boldsymbol{\theta}}) = \max_{\|f\|_L \leq K} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} f(\mathbf{x}) \right] \geq$$
$$\geq \max_{\phi \in \boldsymbol{\Phi}} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} f_{\phi}(\mathbf{x}) - \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} f_{\phi}(\mathbf{x}) \right]$$

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein GAN

Standard GAN Objective

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D_{\boldsymbol{\phi}}(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D_{\boldsymbol{\phi}}(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})))$$

WGAN Objective

$$\min_{\boldsymbol{\theta}} W(p_{\text{data}} \| p_{\boldsymbol{\theta}}) \approx \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi} \in \boldsymbol{\Phi}} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} f_{\boldsymbol{\phi}}(\mathbf{x}) - \mathbb{E}_{p(\mathbf{z})} f_{\boldsymbol{\phi}}(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})) \right]$$

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein GAN

Standard GAN Objective

$$\min_{\boldsymbol{\theta}} \max_{\phi} \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log D_\phi(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D_\phi(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})))$$

WGAN Objective

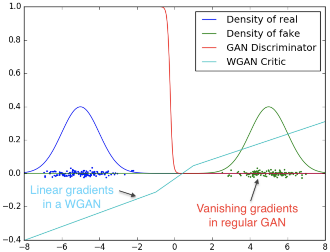
$$\min_{\boldsymbol{\theta}} W(p_{\text{data}} \| p_{\boldsymbol{\theta}}) \approx \min_{\boldsymbol{\theta}} \max_{\phi \in \boldsymbol{\Phi}} \left[\mathbb{E}_{p_{\text{data}}(\mathbf{x})} f_\phi(\mathbf{x}) - \mathbb{E}_{p(\mathbf{z})} f_\phi(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z})) \right]$$

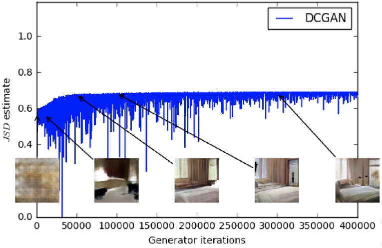
- The discriminator D is replaced by function f: in WGAN, it is known as the **critic**, which is *not* a classifier.
- *"Weight clipping is a clearly terrible way to enforce a Lipschitz constraint."*
  - If c is large, optimizing the critic is hard.
  - If c is small, gradients may vanish.

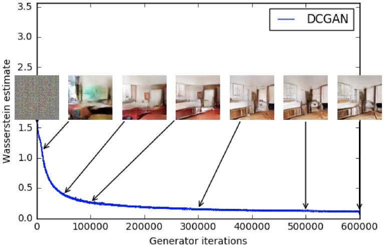
Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein GAN

- WGAN provides nonzero gradients even if distributions' supports are disjoint.
- $\mathrm{JSD}(p_{\text{data}} \| p_\theta)$ is poorly correlated with sample quality and remains near its maximum value $\log 2 \approx 0.69$.
- $W(p_{\text{data}} \| p_\theta)$ is tightly correlated with quality.







Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Summary

- Likelihood is not a reliable metric for generative model evaluation.
- Adversarial learning casts distribution matching as a minimax game.
- GANs, in theory, optimize the Jensen-Shannon divergence.
- KL and JS divergences fail as objectives when the model and data distributions are disjoint.
- The Earth Mover's (Wasserstein) distance provides a more meaningful loss for distribution matching.
- Kantorovich-Rubinstein duality allows us to compute the EM distance using only samples.
- Wasserstein GAN enforces the Lipschitz condition on the critic through weight clipping—although better alternatives exist.

Summary

-