# Deep Generative Models

## Lecture 7

Roman Isachenko

AI Masters

2024, Spring

# Recap of previous lecture

| | **VAE** | **NF** |
|---|---|---|
| **Objective** | ELBO $\mathcal{L}$ | Forward KL/MLE |
| **Encoder** | stochastic<br>$\mathbf{z} \sim q(\mathbf{z}\vert\mathbf{x}, \phi)$ | deterministic<br>$\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x})$<br>$q(\mathbf{z}\vert\mathbf{x}, \boldsymbol{\theta}) = \delta(\mathbf{z} - f_{\boldsymbol{\theta}}(\mathbf{x}))$ |
| **Decoder** | stochastic<br>$\mathbf{x} \sim p(\mathbf{x}\vert\mathbf{z}, \boldsymbol{\theta})$ | deterministic<br>$\mathbf{x} = g_{\boldsymbol{\theta}}(\mathbf{z})$<br>$p(\mathbf{x}\vert\mathbf{z}, \boldsymbol{\theta}) = \delta(\mathbf{x} - g_{\boldsymbol{\theta}}(\mathbf{z}))$ |
| **Parameters** | $\boldsymbol{\phi}, \boldsymbol{\theta}$ | $\boldsymbol{\theta} \equiv \boldsymbol{\phi}$ |

### Theorem

MLE for normalizing flow is equivalent to maximization of ELBO for VAE model with deterministic encoder and decoder:

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \delta(\mathbf{x} - f^{-1}(\mathbf{z}, \boldsymbol{\theta})) = \delta(\mathbf{x} - g_{\boldsymbol{\theta}}(\mathbf{z}));$$

$$q(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) = \delta(\mathbf{z} - f_{\boldsymbol{\theta}}(\mathbf{x})).$$

*Nielsen D., et al. SurVAE Flows: Surjections to Bridge the Gap between VAEs and Flows, 2020*

# Recap of previous lecture

## Theorem

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

## ELBO surgery

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

## Optimal prior

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

The optimal prior distribution $p(\mathbf{z})$ is aggregated posterior $q(\mathbf{z})$.

*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 $p(\mathbf{z}) = \mathcal{N}(0, I) \Rightarrow$ over-regularization;
- $p(\mathbf{z}) = q_{\text{agg}}(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} q(\mathbf{z}|\mathbf{x}_i) \Rightarrow$ overfitting and highly expensive.

## ELBO revisiting

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

It is Forward KL with respect to $p(\mathbf{z}|\boldsymbol{\lambda})$.

## ELBO with flow-based VAE prior

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\boldsymbol{\phi})}\left[\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \log p(\mathbf{z}|\boldsymbol{\lambda}) - \log q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})\right]$$

$$= \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\boldsymbol{\phi})}\Bigg[\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \underbrace{\Big(\log p(f_{\boldsymbol{\lambda}}(\mathbf{z})) + \log|\det(\mathbf{J}_f)|\Big)}_{\text{flow-based prior}} - \log q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})\Bigg]$$

$$\mathbf{z} = f_{\boldsymbol{\lambda}}^{-1}(\mathbf{z}^*) = g_{\boldsymbol{\lambda}}(\mathbf{z}^*), \quad \mathbf{z}^* \sim p(\mathbf{z}^*) = \mathcal{N}(0, 1)$$

*Chen X. et al. Variational Lossy Autoencoder, 2016*

# Recap of previous lecture

## Discrete VAE latents

- Define dictionary (word book) space $\{\mathbf{e}_k\}_{k=1}^K$, where $\mathbf{e}_k \in \mathbb{R}^C$, $K$ is the size of the dictionary.
- Our variational posterior $q(c|\mathbf{x}, \phi) = \text{Categorical}(\boldsymbol{\pi}_\phi(\mathbf{x}))$ (encoder) outputs discrete probabilities vector.
- We sample $c^*$ from $q(c|\mathbf{x}, \phi)$ (reparametrization trick analogue).
- Our generative distribution $p(\mathbf{x}|\mathbf{e}_{c^*}, \boldsymbol{\theta})$ (decoder).

## ELBO

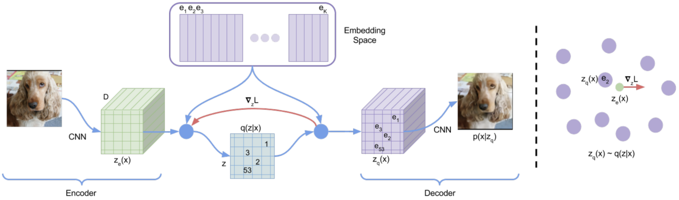
$$\mathcal{L}(\phi, \boldsymbol{\theta}) = \mathbb{E}_{q(c|\mathbf{x},\phi)} \log p(\mathbf{x}|c, \boldsymbol{\theta}) - KL(q(c|\mathbf{x}, \phi)||p(c)) \rightarrow \max_{\phi, \boldsymbol{\theta}}.$$

## KL term

$$KL(q(c|\mathbf{x}, \phi)||p(c)) = -H(q(c|\mathbf{x}, \phi)) + \log K.$$

Is it possible to make reparametrization trick? (we sample from discrete distribution now!).

# Recap of previous lecture



## Deterministic variational posterior

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

## ELBO

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

## Straight-through gradient estimation

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

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

# Recap of previous lecture

## Gumbel-max trick

Let $g_k \sim \text{Gumbel}(0, 1)$ for $k = 1, \dots, K$. Then

$$c = \arg\max_k [\log \pi_k + g_k]$$

has a categorical distribution $c \sim \text{Categorical}(\boldsymbol{\pi})$.

## Gumbel-softmax relaxation

Con**crete** distribution = **con**tinuous + dis**crete**

$$\hat{c}_k = \frac{\exp\left(\frac{\log q(k|\mathbf{x}, \phi) + g_k}{\tau}\right)}{\sum_{j=1}^K \exp\left(\frac{\log q(j|\mathbf{x}, \phi) + g_j}{\tau}\right)}, \quad k = 1, \dots, K.$$

## Reparametrization trick

$$\nabla_\phi \mathbb{E}_{q(c|\mathbf{x}, \phi)} \log p(\mathbf{x}|\mathbf{e}_c, \boldsymbol{\theta}) = \mathbb{E}_{\text{Gumbel}(0,1)} \nabla_\phi \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}),$$

where $\mathbf{z} = \sum_{k=1}^K \hat{c}_k \mathbf{e}_k$ (all operations are differentiable now).

*Maddison C. J., Mnih A., Teh Y. W. The Concrete distribution: A continuous relaxation of discrete random variables, 2016*

*Jang E., Gu S., Poole B. Categorical reparameterization with Gumbel-Softmax, 2016*

# Recap of previous lecture

## Likelihood-free learning

- Likelihood is not a perfect quality measure for generative model.
- Likelihood could be intractable.

Imagine we have two sets of samples

- $\mathcal{S}_1 = \{\mathbf{x}_i\}_{i=1}^{n_1} \sim \pi(\mathbf{x})$ – real samples;
- $\mathcal{S}_2 = \{\mathbf{x}_i\}_{i=1}^{n_2} \sim p(\mathbf{x}|\boldsymbol{\theta})$ – generated (or fake) samples.

Let define discriminative model (classifier):

$$p(y=1|\mathbf{x}) = P\bigl(\{\mathbf{x} \sim \pi(\mathbf{x})\}\bigr); \quad p(y=0|\mathbf{x}) = P\bigl(\{\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})\}\bigr)$$

## Assumption

Generative distribution $p(\mathbf{x}|\boldsymbol{\theta})$ equals to the true distribution $\pi(\mathbf{x})$ if we can not distinguish them using discriminative model $p(y|\mathbf{x})$. It means that $p(y=1|\mathbf{x}) = 0.5$ for each sample $\mathbf{x}$.

# Outline

1. Likelihood-free learning

2. Generative adversarial networks (GAN)

3. Wasserstein distance

# Outline

1. Likelihood-free learning
2. Generative adversarial networks (GAN)
3. Wasserstein distance

# Likelihood based models

## Poor likelihood
## Great samples

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

For small $\epsilon$ this model will generate samples with great quality, but likelihood of test sample will be very poor.

## Great likelihood
## Poor samples

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

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

Noisy irrelevant samples, but for high dimensions $\log p(\mathbf{x})$ becomes proportional to $m$.

- Likelihood is not a perfect quality measure for generative model.
- Likelihood could be intractable.

*Theis L., Oord A., Bethge M. A note on the evaluation of generative models, 2015*

# Likelihood-free learning

## Where did we start

We would like to approximate true data distribution $\pi(\mathbf{x})$. Instead of searching true $\pi(\mathbf{x})$ over all probability distributions, learn function approximation $p(\mathbf{x}|\boldsymbol{\theta}) \approx \pi(\mathbf{x})$.

Imagine we have two sets of samples

- $\mathcal{S}_1 = \{\mathbf{x}_i\}_{i=1}^{n_1} \sim \pi(\mathbf{x})$ – real samples;
- $\mathcal{S}_2 = \{\mathbf{x}_i\}_{i=1}^{n_2} \sim p(\mathbf{x}|\boldsymbol{\theta})$ – generated (or fake) samples.

Let define discriminative model (classifier):

$$p(y = 1|\mathbf{x}) = P\bigl(\{\mathbf{x} \sim \pi(\mathbf{x})\}\bigr); \quad p(y = 0|\mathbf{x}) = P\bigl(\{\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})\}\bigr)$$

## Assumption

Generative distribution $p(\mathbf{x}|\boldsymbol{\theta})$ equals to the true distribution $\pi(\mathbf{x})$ if we can not distinguish them using discriminative model $p(y|\mathbf{x})$. It means that $p(y = 1|\mathbf{x}) = 0.5$ for each sample $\mathbf{x}$.

# Generative adversarial networks (GAN)

The more powerful discriminative model we will have, the more likely we will get the "best" generative distribution $p(\mathbf{x}|\boldsymbol{\theta})$.
The most common way to learn a classifier is to minimize cross entropy loss.

- **Generator:** generative model $\mathbf{x} = G(\mathbf{z})$, which makes generated sample more realistic. Here $\mathbf{z}$ comes from the base (known) distribution $p(\mathbf{z})$ and $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$. Generator tries to **maximize** cross entropy.
- **Discriminator:** a classifier $p(y = 1|\mathbf{x}) = D(\mathbf{x}) \in [0, 1]$, which distinguishes real samples from generated samples. Discriminator tries to **minimize** cross entropy (tries to enhance discriminative model).

## Objective

$$\min_G \max_D \left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \log(1 - D(\mathbf{x}))\right]$$

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

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

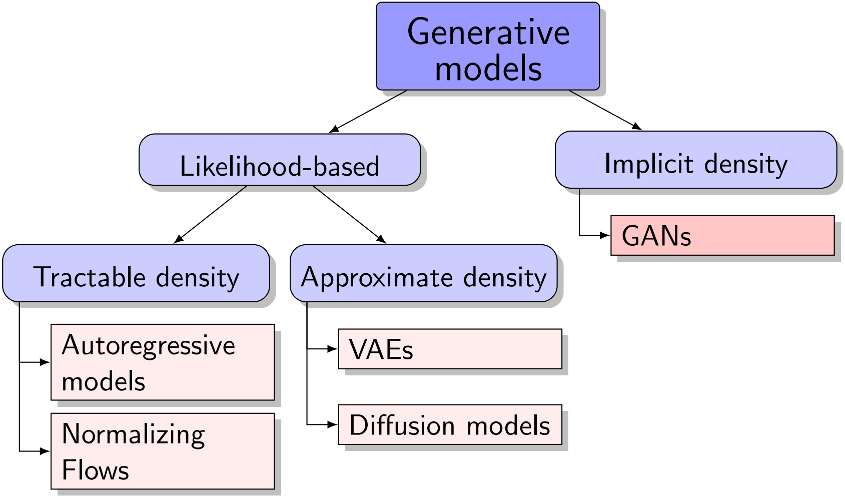
# Outline

1. Likelihood-free learning

2. Generative adversarial networks (GAN)

3. Wasserstein distance

# Generative models zoo



# GAN optimality

## Theorem

The minimax game

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

has the global optimum $\pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta})$, in this case $D^*(\mathbf{x}) = 0.5$.

## Proof (fixed $G$)

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

$$\frac{dy(D)}{dD} = \frac{\pi(\mathbf{x})}{D(\mathbf{x})} - \frac{p(\mathbf{x}|\boldsymbol{\theta})}{1 - D(\mathbf{x})} = 0 \quad \Rightarrow \quad D^*(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}$$

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

# GAN optimality

## Proof continued (fixed $D = D^*$)

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

$$= KL \left( \pi(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2} \right) + KL \left( p(\mathbf{x}|\boldsymbol{\theta}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2} \right) - 2 \log 2$$

$$= 2JSD(\pi(\mathbf{x}) || p(\mathbf{x}|\boldsymbol{\theta})) - 2 \log 2.$$

## Jensen-Shannon divergence (symmetric KL divergence)

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

Could be used as a distance measure!

$$V(G^*, D^*) = -2 \log 2, \quad \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}), \quad D^*(\mathbf{x}) = 0.5.$$

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

# GAN optimality

### Theorem

The minimax game

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

has the global optimum $\pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta})$, in this case $D^*(\mathbf{x}) = 0.5$.

### Expectations

If the generator could be **any** function and the discriminator is **optimal** at every step, then the generator is **guaranteed to converge** to the data distribution.

### Reality

- Generator updates are made in parameter space, discriminator is not optimal at every step.
- Generator and discriminator loss keeps oscillating during GAN training.

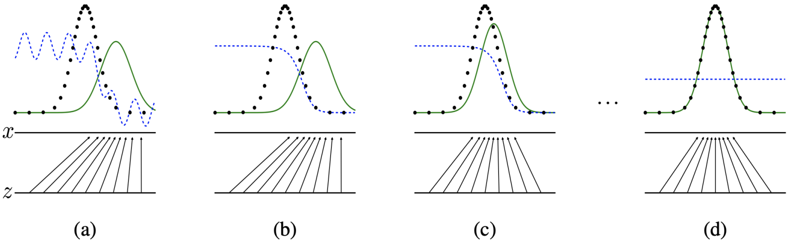
*Goodfellow I. J. et al. Generative Adversarial Networks, 2014*

# GAN training

Let further assume that generator and discriminator are parametric models: $D_{\boldsymbol{\phi}}(\mathbf{x})$ and $G_{\boldsymbol{\theta}}(\mathbf{z})$.

## Objective

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



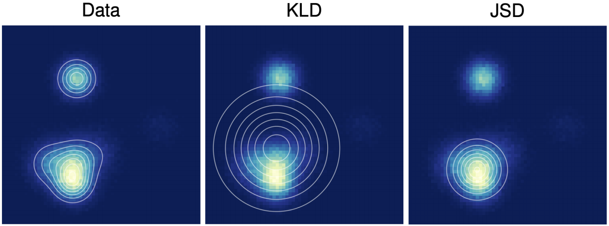
(a) (b) (c) (d)

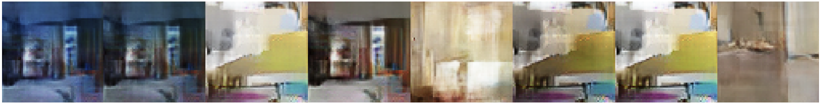
- $\mathbf{z} \sim p(\mathbf{z})$ is a latent variable.
- $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \delta(\mathbf{x} - G_{\boldsymbol{\theta}}(\mathbf{z}))$ is deterministic decoder (like NF).
- We do not have encoder at all.

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

# 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*

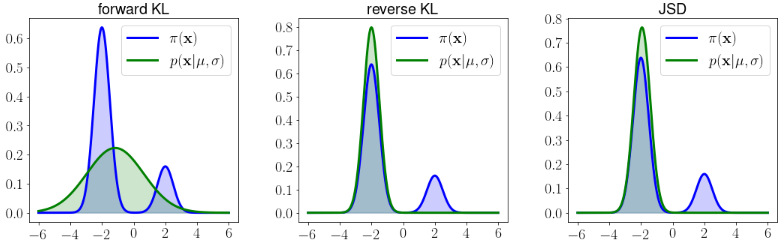
# Jensen-Shannon vs Kullback-Leibler

- $\pi(\mathbf{x})$ is a fixed mixture of 2 gaussians.
- $p(\mathbf{x}|\mu, \sigma) - \mathcal{N}(\mu, \sigma^2)$.

## Mode covering vs mode seeking

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

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

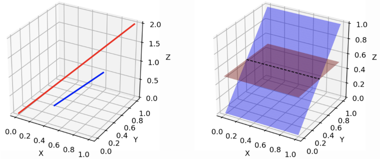


# Outline

1. Likelihood-free learning
2. Generative adversarial networks (GAN)
3. Wasserstein distance

# Informal theoretical results

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



- If $\pi(\mathbf{x})$ and $p(\mathbf{x}|\boldsymbol{\theta})$ 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.

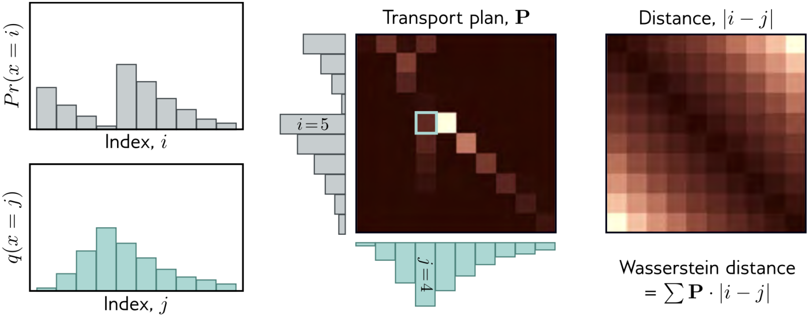
*Weng L. From GAN to WGAN, 2019*
*Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017*

# Wasserstein distance (discrete)

A.k.a. **Earth Mover's distance**.

## Optimal transport formulation

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.



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

# Wasserstein distance (continuous)

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

- $\gamma(\mathbf{x}, \mathbf{y})$ – transportation plan (the amount of "dirt" that should be transported from point $\mathbf{x}$ to point $\mathbf{y}$)

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

- $\Gamma(\pi, p)$ – the set of all joint distributions $\gamma(\mathbf{x}, \mathbf{y})$ with marginals $\pi$ and $p$.
- $\gamma(\mathbf{x}, \mathbf{y})$ – the amount, $\|\mathbf{x} - \mathbf{y}\|$– the distance.

## Wasserstein metric

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

Here we will use $W(\pi, p) = W_1(\pi, p)$ that corresponds to the optimal transport formulation.

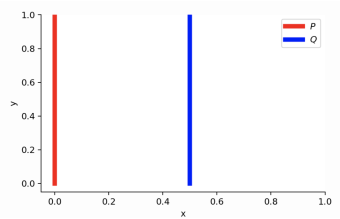
*Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017*

# 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

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

- $\theta \neq 0$

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

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

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

*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 $G_{\boldsymbol{\theta}}(\mathbf{z})$ 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 therefore $W(\pi, p)$ is continuous everywhere and differentiable almost everywhere.

### Theorem 2

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

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

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

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

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

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

# Wasserstein GAN

## Wasserstein distance

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

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

## Theorem (Kantorovich-Rubinstein duality)

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

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

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

Now we need only samples to get Monte Carlo estimate for $W(\pi||p)$.

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

# Summary

- Likelihood is not a perfect criteria to measure quality of generative model.
- Adversarial learning suggests to solve minimax problem to match the distributions.
- GAN tries to optimize Jensen-Shannon divergence (in theory).
- Mode collapse is one of the main problems of vanilla GAN. Lots of tips and tricks has to be used to make the GAN training is stable and scalable.
- KL and JS divergences work poorly as model objective in the case of disjoint supports.
- Earth-Mover distance is a more appropriate objective function for distribution matching problem.
- Kantorovich-Rubinstein duality gives the way to calculate the EM distance using only samples.