Deep Generative Models

Lecture 9

Roman Isachenko

AI Masters

Autumn, 2022

Recap of previous lecture

Standart ELBO

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

Expressive flow-based VAE posterior

$$\log q(\mathbf{z}^*|\mathbf{x}, \boldsymbol{\phi}, \boldsymbol{\lambda}) = \log q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) + \log \left| \det \left(\frac{\partial g(\mathbf{z}, \boldsymbol{\lambda})}{\partial \mathbf{z}} \right) \right|$$

ELBO with flow-based posterior

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \boldsymbol{\phi}, \boldsymbol{\lambda})} \bigl[\log p(\mathbf{x}, \mathbf{z}^*|\boldsymbol{\theta}) - \log q(\mathbf{z}^*|\mathbf{x}, \boldsymbol{\phi}, \boldsymbol{\lambda}) \bigr] =$$

$$= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})} \biggl[\log p(\mathbf{x}, g(\mathbf{z}, \boldsymbol{\lambda})|\boldsymbol{\theta}) - \log q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) + \log |\det(\mathbf{J}_g)| \biggr].$$

- Obtain samples $\mathbf{z}$ from the encoder $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$.
- Apply flow model $\mathbf{z}^* = g(\mathbf{z}, \boldsymbol{\lambda})$.
- Compute likelihood for $\mathbf{z}^*$ using the decoder, base distribution for $\mathbf{z}^*$ and the Jacobian.

Rezende D. J., Mohamed S. Variational Inference with Normalizing Flows, 2015

Recap of previous lecture

Expressive flow-based VAE posterior

$$\log q(\mathbf{z}^*|\mathbf{x}, \phi, \boldsymbol{\lambda}) = \log q(\mathbf{z}|\mathbf{x}, \phi) + \log \det \left| \frac{\partial g(\mathbf{z}, \boldsymbol{\lambda})}{\partial \mathbf{z}} \right|$$

Expressive flow-based VAE prior

$$\log p(\mathbf{z}|\boldsymbol{\lambda}) = \log p(\mathbf{z}^*) + \log \left| \det \left(\frac{d\mathbf{z}^*}{d\mathbf{z}} \right) \right|; \quad \mathbf{z} = f(\mathbf{z}^*, \boldsymbol{\lambda}) = g^{-1}(\mathbf{z}^*, \boldsymbol{\lambda})$$

Theorem

VAE with the flow-based prior for latent code $\mathbf{z}$ is equivalent to VAE with flow-based posterior for latent code $\mathbf{z}$.

$$\mathcal{L}(\phi, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \underbrace{KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}|\boldsymbol{\lambda}))}_{\text{flow-based prior}}$$

$$= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \underbrace{KL(q(\mathbf{z}|\mathbf{x}, \phi, \boldsymbol{\lambda}) || p(\mathbf{z}))}_{\text{flow-based posterior}}$$

Chen X. et al. Variational Lossy Autoencoder, 2016

Recap of previous lecture

Disentanglement learning

A disentangled representation is a one where single latent units are sensitive to changes in single generative factors, while being invariant to changes in other factors.

β-VAE

$$\mathcal{L}(q, \boldsymbol{\theta}, \beta) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \beta \cdot KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})).$$

Representations becomes disentangled by setting a stronger constraint with $\beta > 1$. However, it leads to poorer reconstructions and a loss of high frequency details.

ELBO surgery

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

Burgess C. P. et al. Understanding disentangling in β-VAE, 2018

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.

Two sample test

$$H_0 : \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}), \quad H_1 : \pi(\mathbf{x}) \neq p(\mathbf{x}|\boldsymbol{\theta})$$

If test statistic $T(\mathcal{S}_1, \mathcal{S}_2) < \alpha$, then accept H_0, else reject it.

- $p(\mathbf{x}|\boldsymbol{\theta})$ minimizes the value of test statistic $T(\mathcal{S}_1, \mathcal{S}_2)$.
- It is hard to find an appropriate test statistic in high dimensions. $T(\mathcal{S}_1, \mathcal{S}_2)$ could be learnable.

Recap of previous lecture

- **Generator:** generative model $\mathbf{x} = G(\mathbf{z})$, which makes generated sample more realistic.
- **Discriminator:** a classifier $D(\mathbf{x}) \in [0, 1]$, which distinguishes real samples from generated samples.

GAN optimality theorem

The minimax game

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

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

$$\min_G V(G, D^*) = \min_G [2JSD(\pi||p) - \log 4] = -\log 4, \quad \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}).$$

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.

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

Outline

1. Vanishing gradients and mode collapse
2. Adversatial variational Bayes
3. Wasserstein distance

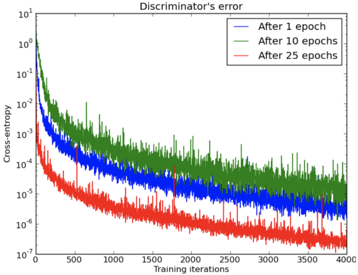
Outline

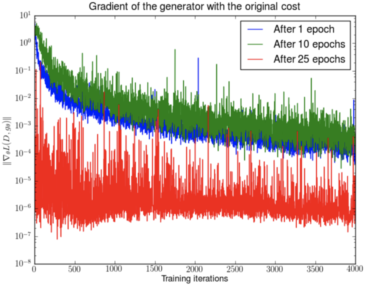
Vanishing gradients

Objective

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

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





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

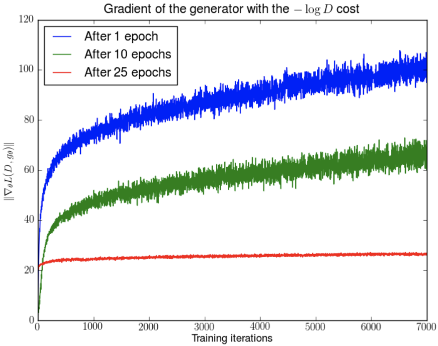
Vanishing gradients

Objective

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

Non-saturating GAN

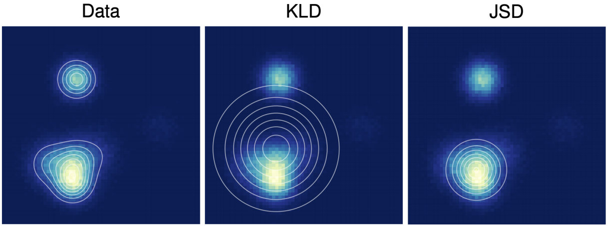
- Maximize $\log D(G(z))$ instead of minimizing $\log(1 - D(G(\mathbf{z})))$.
- Gradients are getting much stronger, but the training is unstable (with increasing mean and variance).

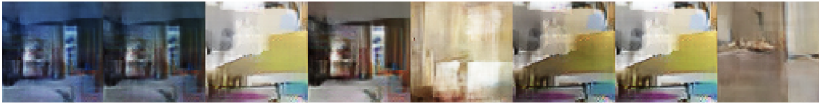


Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017

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.

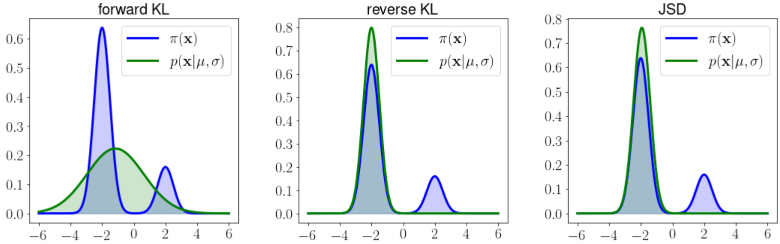
Goodfellow I. J. et al. Generative Adversarial Networks, 2014
Metz L. et al. Unrolled Generative Adversarial Networks, 2016

Jensen-Shannon vs Kullback-Leibler

Mode covering vs mode seeking

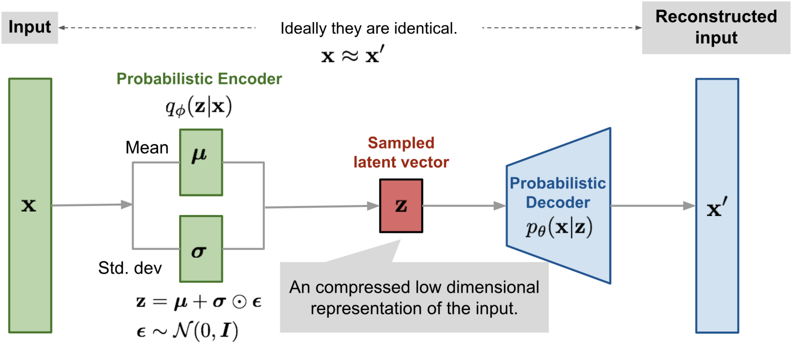
$$KL(\pi||p) = \int \pi(\mathbf{x}) \log \frac{\pi(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x}, \quad KL(p||\pi) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\pi(\mathbf{x})} d\mathbf{x}$$

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



Outline

VAE recap



- Encoder $q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi(\mathbf{x}))$.
- Variational posterior $q(\mathbf{z}|\mathbf{x}, \phi)$ originally approximates the true posterior $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$.
- Which methods are you already familiar with to make the posterior is more flexible?

image credit: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Adversarial Variational Bayes

ELBO objective

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

What is the problem to make the variational posterior model an implicit model?

- The first term is reconstruction loss that needs only samples from $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ to evaluate.
- Reparametrization trick allows to get gradients of reconstruction loss

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} \int q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) f(\mathbf{z}) d\mathbf{z} &= \nabla_{\boldsymbol{\phi}} \int r(\boldsymbol{\epsilon}) f(\mathbf{z}) d\boldsymbol{\epsilon} \\ &= \int r(\boldsymbol{\epsilon}) \nabla_{\boldsymbol{\phi}} f(g(\mathbf{x}, \boldsymbol{\epsilon}, \boldsymbol{\phi})) d\boldsymbol{\epsilon} \approx \nabla_{\boldsymbol{\phi}} f(g(\mathbf{x}, \boldsymbol{\epsilon}^*, \boldsymbol{\phi})),\end{aligned}$$

where $\boldsymbol{\epsilon}^* \sim r(\boldsymbol{\epsilon}), \quad \mathbf{z} = g(\mathbf{x}, \boldsymbol{\epsilon}, \boldsymbol{\phi}), \quad \mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$.

Mescheder L., Nowozin S., Geiger A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks, 2017

Adversarial Variational Bayes

ELBO objective

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

What is the problem to make the variational posterior model an implicit model?

- The third term requires the explicit the value of $q(\mathbf{z}|\mathbf{x}, \phi)$.
- We could join second and third terms:

$$KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z})) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\phi)} \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z})} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\phi)} \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)\pi(\mathbf{x})}{p(\mathbf{z})\pi(\mathbf{x})}.$$

- We have to estimate density ratio

$$r(\mathbf{x}, \mathbf{z}) = \frac{q_1(\mathbf{x}, \mathbf{z})}{q_2(\mathbf{x}, \mathbf{z})} = \frac{q(\mathbf{z}|\mathbf{x}, \phi)\pi(\mathbf{x})}{p(\mathbf{z})\pi(\mathbf{x})}.$$

Mescheder L., Nowozin S., Geiger A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks, 2017

Density ratio trick

Consider two distributions $q_1(\mathbf{x})$, $q_2(\mathbf{x})$ and probabilistic model

$$p(\mathbf{x}|y) = \begin{cases} q_1(\mathbf{x}), & \text{if } y = 1, \\ q_2(\mathbf{x}), & \text{if } y = 0, \end{cases} \qquad y \sim \text{Bern}(0.5).$$

Density ratio

$$\frac{q_1(\mathbf{x})}{q_2(\mathbf{x})} = \frac{p(\mathbf{x}|y=1)}{p(\mathbf{x}|y=0)} = \frac{p(y=1|\mathbf{x})p(\mathbf{x})}{p(y=1)} \Bigg/ \frac{p(y=0|\mathbf{x})p(\mathbf{x})}{p(y=0)} =$$
$$= \frac{p(y=1|\mathbf{x})}{p(y=0|\mathbf{x})} = \frac{p(y=1|\mathbf{x})}{1 - p(y=1|\mathbf{x})} = \frac{D(\mathbf{x})}{1 - D(\mathbf{x})}$$

Here $D(\mathbf{x})$ is a discriminator model the output of which is a probability that $\mathbf{x}$ is a sample from $q_1(\mathbf{x})$ rather than from $q_2(\mathbf{x})$.

Adversarial Variational Bayes

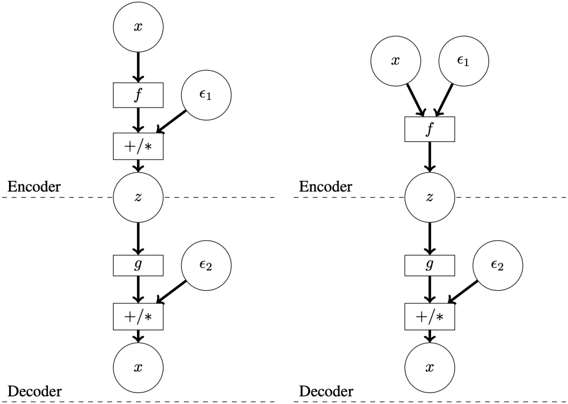
$$\max_D \left[\mathbb{E}_{\pi(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\phi)} \log D(\mathbf{x},\mathbf{z}) + \mathbb{E}_{\pi(\mathbf{x})}\mathbb{E}_{p(\mathbf{z})} \log(1 - D(\mathbf{x},\mathbf{z}))\right]$$

Mescheder L., Nowozin S., Geiger A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks, 2017

Adversarial Variational Bayes

ELBO objective

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

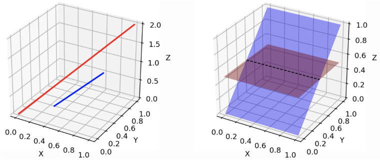


Mescheder L., Nowozin S., Geiger A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks, 2017

Outline

Informal theoretical results

- Since $\mathbf{z}$ usually has lower dimensionality compared to $\mathbf{x}$, manifold $G(\mathbf{z}, \boldsymbol{\theta})$ 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

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

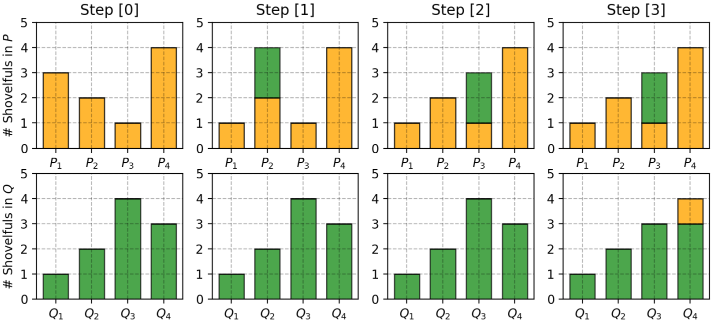
- Adding continuous noise to the inputs of the discriminator smoothes the distributions of the probability mass.

Weng L. From GAN to WGAN, 2019

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

Wasserstein distance (discrete)

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.



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

Weng L. From GAN to WGAN, 2019

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 $\mathbf{y}$)

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

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

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

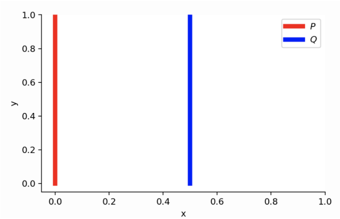
Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

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 $G(\mathbf{z}, \boldsymbol{\theta})$ 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 π be a distribution on a compact space $\mathcal{X}$ and $\{p_t\}_{t=1}^{\infty}$ be a sequence of distributions on $\mathcal{X}$.

$$KL(\pi || p_t) \to 0 \,(\text{or } KL(p_t || \pi) \to 0) \tag{1}$$
$$JSD(\pi || p_t) \to 0 \tag{2}$$
$$W(\pi || p_t) \to 0 \tag{3}$$

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 $(f : \mathcal{X} \to \mathbb{R})$

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

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

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.
- Adversarial Variational Bayes uses density ratio trick to get more powerful variational posterior.
- 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.