Deep Generative Models

Lecture 9

Roman Isachenko

Ozon Masters

Spring, 2022

Recap of previous lecture

Images are discrete data, flow is a continuous model. We need to convert a discrete data distribution to a continuous one.

Uniform dequantization bound

$$\mathbf{x} \sim \text{Categorical}(\boldsymbol{\pi}), \quad \mathbf{u} \sim U[0,1], \quad \mathbf{y} = \mathbf{x} + \mathbf{u} \sim \text{Continuous}$$
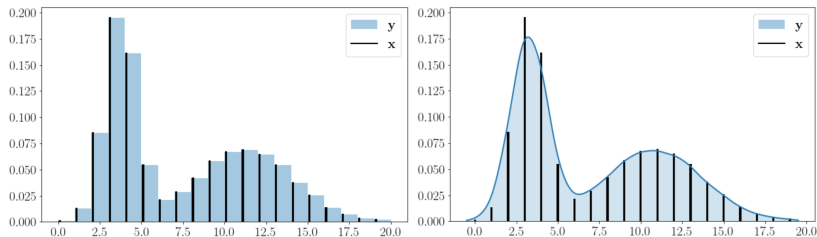
$$\log P(\mathbf{x}|\boldsymbol{\theta}) \geq \int_{U[0,1]} \log p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u}.$$

Variational dequantization bound

Introduce variational dequantization noise distribution $q(\mathbf{u}|\mathbf{x})$ and treat it as an approximate posterior.

$$\log P(\mathbf{x}|\boldsymbol{\theta}) \geq \int q(\mathbf{u}|\mathbf{x}) \log \frac{p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta})}{q(\mathbf{u}|\mathbf{x})} d\mathbf{u} = \mathcal{L}(q, \boldsymbol{\theta}).$$

Ho J. et al. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design, 2019

Recap of previous lecture

Flow model for dequantization

$$q(\mathbf{u}|\mathbf{x}) = p(h^{-1}(\mathbf{u}, \phi)) \cdot \left| \det \frac{\partial h^{-1}(\mathbf{u}, \phi)}{\partial \mathbf{u}} \right|.$$

Variational dequantization bound

$$\mathcal{L}(q, \boldsymbol{\theta}) = \int q(\mathbf{u}|\mathbf{x}) \log \frac{p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta})}{q(\mathbf{u}|\mathbf{x})} d\mathbf{u}.$$

Ho J. et al. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design, 2019

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

$$\mathcal{L}(q, \boldsymbol{\theta}, \beta) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \beta \cdot KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})).$$
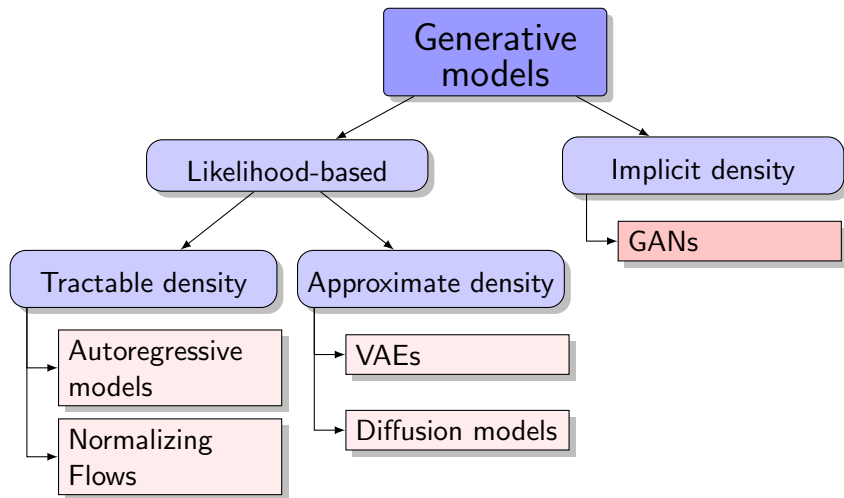
Representations becomes disentangled by setting a stronger constraint with $\beta > 1$. However, it leads to poorer reconstructions and a loss of high frequency details.

ELBO surgery

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

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

Outline

Generative models zoo

Likelihood based models

Is likelihood a good measure of model quality?

Poor likelihood
Great samples

Great likelihood
Poor samples

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

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

For small ϵ this model will generate samples with great quality, but likelihood will be very poor.

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

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

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

Likelihood-free learning

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

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.

Two sample test

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

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

Likelihood-free learning

Two sample test

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

Desired behaviour

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

GAN objective

- **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.

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

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

Vanilla 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 $D^*(\mathbf{x}) = 0.5$.

Proof (fixed G)

$$V(G, D) = \mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \log(1 - D(\mathbf{x}))$$

$$= \int \underbrace{\left[\pi(\mathbf{x}) \log D(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta}) \log(1 - D(\mathbf{x})) \right]}_{y(D)} d\mathbf{x}$$

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

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

Vanilla GAN optimality

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

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

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

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

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!

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

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

Vanilla 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 $D^*(\mathbf{x}) = 0.5$.

Proof

for fixed G:

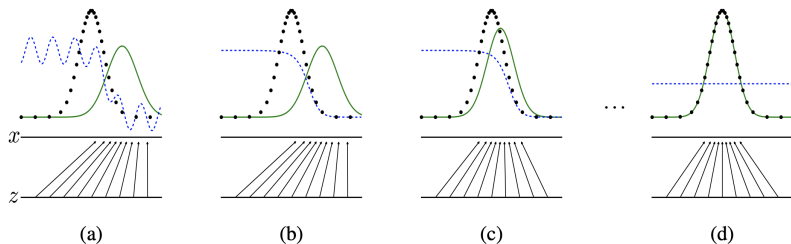$$D^*(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}$$

for fixed $D = D^*$:

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

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

Vanilla GAN

Objective

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


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

▶ 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

Vanishing gradients

Objective

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

Early in learning, G is poor, D can reject samples with high confidence. In this case, $\log(1 - D(G(\mathbf{z})))$ saturates.
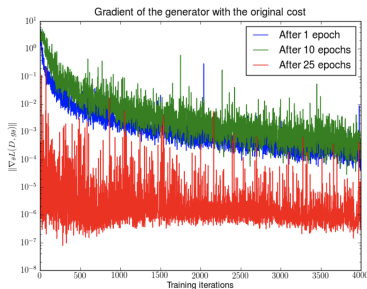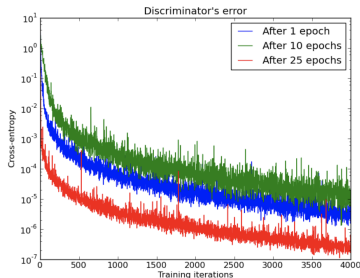
Arjovsky M., Bottou L. *Towards Principled Methods for Training Generative Adversarial Networks*, 2017

Vanishing gradients

Objective

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

## 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).

Gradient of the generator with the $-\log D$ cost

After 1 epoch
After 10 epochs
After 25 epochs

$\|\nabla_\theta L(D, g_\theta)\|$

Training iterations

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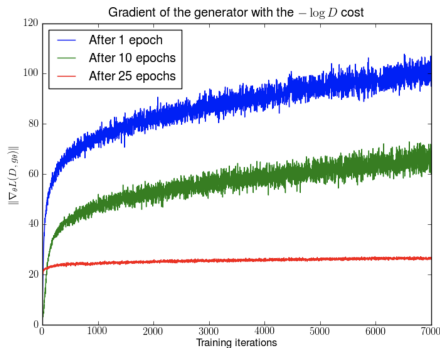
Mode collapse

The phenomena where the generator of a GAN collapses to one or few distribution modes.
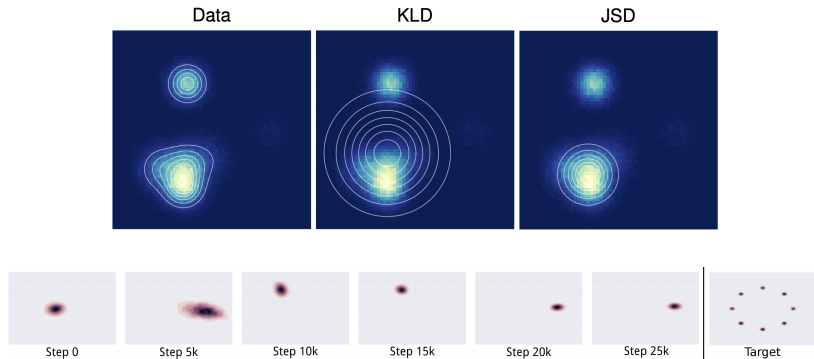
Alternate architectures, adding regularization terms, injecting small noise perturbations and other millions bags and tricks are used to avoid the mode collapse.

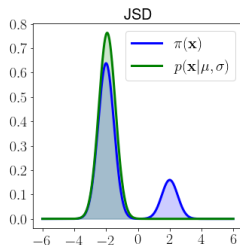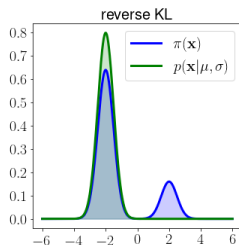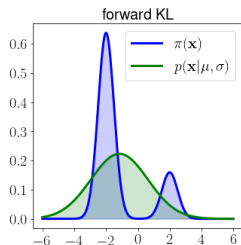*Goodfellow I. J. et al. Generative Adversarial Networks, 2014*
Metz L. et al. Unrolled Generative Adversarial Networks, 2016

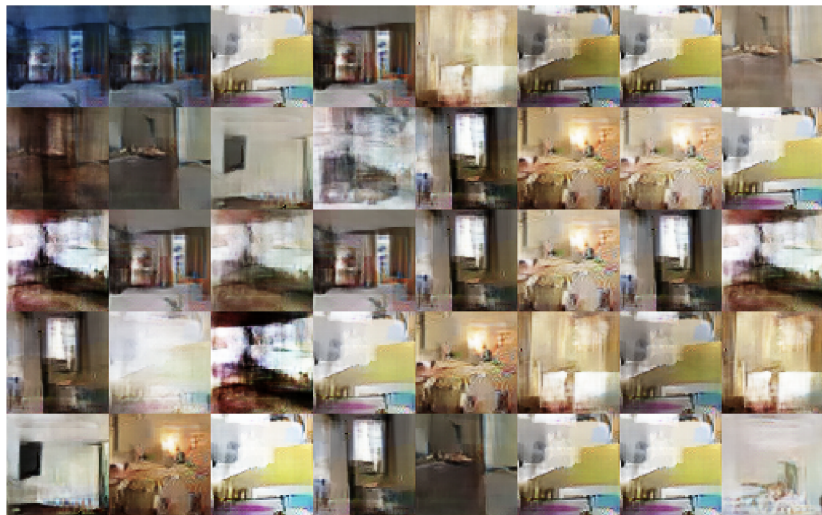# Jensen-Shannon vs Kullback-Leibler

Mode covering vs mode seeking

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

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

Mode collapse: Deep Convolutional GAN

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.

- ▶ Vanilla GAN tries to optimize Jensen-Shannon divergence (in theory).

- ▶ 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.

- ▶ KL and JS divergences work poorly as model objective in the case of disjoint supports.