

Deep Generative Models

Lecture 4

Roman Isachenko

Moscow Institute of Physics and Technology
Yandex School of Data Analysis

2024, Autumn

Recap of previous lecture

Forward KL for flow model

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})) + \log |\det(\mathbf{J}_{\mathbf{f}})|$$

Reverse KL for flow model

$$KL(p||\pi) = \mathbb{E}_{p(\mathbf{z})} [\log p(\mathbf{z}) - \log |\det(\mathbf{J}_{\mathbf{g}})| - \log \pi(\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z}))]$$

Flow KL duality

$$\arg \min_{\boldsymbol{\theta}} KL(\pi(\mathbf{x})||p(\mathbf{x}|\boldsymbol{\theta})) = \arg \min_{\boldsymbol{\theta}} KL(p(\mathbf{z}|\boldsymbol{\theta})||p(\mathbf{z}))$$

- ▶ $p(\mathbf{z})$ is a base distribution; $\pi(\mathbf{x})$ is a data distribution;
- ▶ $\mathbf{z} \sim p(\mathbf{z})$, $\mathbf{x} = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z})$, $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$;
- ▶ $\mathbf{x} \sim \pi(\mathbf{x})$, $\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$, $\mathbf{z} \sim p(\mathbf{z}|\boldsymbol{\theta})$.

Recap of previous lecture

Bayes theorem

$$p(\mathbf{t}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{t})p(\mathbf{t})}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathbf{t})p(\mathbf{t})}{\int p(\mathbf{x}|\mathbf{t})p(\mathbf{t})d\mathbf{t}}$$

- ▶ \mathbf{x} – observed variables, \mathbf{t} – unobserved variables (latent variables/parameters);
- ▶ $p(\mathbf{x}|\mathbf{t})$ – likelihood;
- ▶ $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{t})p(\mathbf{t})d\mathbf{t}$ – evidence;
- ▶ $p(\mathbf{t})$ – prior distribution, $p(\mathbf{t}|\mathbf{x})$ – posterior distribution.

Posterior distribution

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\theta)p(\theta)}{\int p(\mathbf{X}|\theta)p(\theta)d\theta}$$

Recap of previous lecture

Latent variable models (LVM)

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})d\mathbf{z} = \int p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})d\mathbf{z}.$$

MLE problem for LVM

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg \max_{\boldsymbol{\theta}} \log p(\mathbf{X}|\boldsymbol{\theta}) = \arg \max_{\boldsymbol{\theta}} \sum_{i=1}^n \log p(\mathbf{x}_i|\boldsymbol{\theta}) = \\ &= \arg \max_{\boldsymbol{\theta}} \sum_{i=1}^n \log \int p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta})p(\mathbf{z}_i)d\mathbf{z}_i.\end{aligned}$$

Naive Monte-Carlo estimation

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})d\mathbf{z} = \mathbb{E}_{p(\mathbf{z})}p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) \approx \frac{1}{K} \sum_{k=1}^K p(\mathbf{x}|\mathbf{z}_k, \boldsymbol{\theta}),$$

where $\mathbf{z}_k \sim p(\mathbf{z})$.

Recap of previous lecture

ELBO derivation 1 (inequality)

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} \geq \mathbb{E}_q \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} = \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x})$$

ELBO derivation 2 (equality)

$$\begin{aligned}\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) &= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \int q(\mathbf{z}) \log \frac{p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \\ &= \log p(\mathbf{x}|\boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}))\end{aligned}$$

Variational decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}).$$

Recap of previous lecture

Variational lower Bound (ELBO)

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}).$$

$$\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) = \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z}))$$

Log-likelihood decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z})) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

- Instead of maximizing incomplete likelihood, maximize ELBO

$$\max_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}) \rightarrow \max_{q,\boldsymbol{\theta}} \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x})$$

- Maximization of ELBO by variational distribution q is equivalent to minimization of KL

$$\arg \max_q \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) \equiv \arg \min_q KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

Outline

1. ELBO gradients, reparametrization trick
2. Variational autoencoder (VAE)
3. Normalizing flows as VAE model
4. Discrete VAE latent representations

Outline

1. ELBO gradients, reparametrization trick
2. Variational autoencoder (VAE)
3. Normalizing flows as VAE model
4. Discrete VAE latent representations

ELBO gradients, (M-step, $\nabla_{\theta} \mathcal{L}_{\phi, \theta}(\mathbf{x})$)

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

M-step: $\nabla_{\theta} \mathcal{L}_{\phi, \theta}(\mathbf{x})$

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{\phi, \theta}(\mathbf{x}) &= \int q(\mathbf{z}|\mathbf{x}, \phi) \nabla_{\theta} \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} \approx \\ &\approx \nabla_{\theta} \log p(\mathbf{x}|\mathbf{z}^*, \theta), \quad \mathbf{z}^* \sim q(\mathbf{z}|\mathbf{x}, \phi). \end{aligned}$$

Naive Monte-Carlo estimation

$$p(\mathbf{x}|\theta) = \int p(\mathbf{x}|\mathbf{z}, \theta) p(\mathbf{z}) d\mathbf{z} \approx \frac{1}{K} \sum_{k=1}^K p(\mathbf{x}|\mathbf{z}_k, \theta), \quad \mathbf{z}_k \sim p(\mathbf{z}).$$

The variational posterior $q(\mathbf{z}|\mathbf{x}, \phi)$ assigns typically more probability mass in a smaller region than the prior $p(\mathbf{z})$.

ELBO gradients, (E-step, $\nabla_{\phi} \mathcal{L}_{\phi, \theta}(\mathbf{x})$)

E-step: $\nabla_{\phi} \mathcal{L}_{\phi, \theta}(\mathbf{x})$

Difference from M-step: density function $q(\mathbf{z}|\mathbf{x}, \phi)$ depends on the parameters ϕ , it is impossible to use the Monte-Carlo estimation:

$$\begin{aligned}\nabla_{\phi} \mathcal{L}_{\phi, \theta}(\mathbf{x}) &= \nabla_{\phi} \int q(\mathbf{z}|\mathbf{x}, \phi) \left[\log p(\mathbf{x}|\mathbf{z}, \theta) - \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z})} \right] d\mathbf{z} \\ &\neq \int q(\mathbf{z}|\mathbf{x}, \phi) \nabla_{\phi} \left[\log p(\mathbf{x}|\mathbf{z}, \theta) - \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z})} \right] d\mathbf{z}\end{aligned}$$

Reparametrization trick (LOTUS trick)

- ▶ $r(x) = \mathcal{N}(0, 1)$, $y = \sigma \cdot x + \mu$, $p(y|\theta) = \mathcal{N}(\mu, \sigma^2)$, $\theta = [\mu, \sigma]$.
- ▶ $\epsilon^* \sim r(\epsilon)$, $\mathbf{z} = \mathbf{g}_{\phi}(\mathbf{x}, \epsilon)$, $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi)$

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

ELBO gradient (E-step, $\nabla_{\phi} \mathcal{L}_{\phi, \theta}(\mathbf{x})$)

$$\begin{aligned}\nabla_{\phi} \mathcal{L}_{\phi, \theta}(\mathbf{x}) &= \nabla_{\phi} \int q(\mathbf{z}|\mathbf{x}, \phi) \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} - \nabla_{\phi} \text{KL}(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z})) \\ &= \int r(\epsilon) \nabla_{\phi} \log p(\mathbf{x}|\mathbf{g}_{\phi}(\mathbf{x}, \epsilon), \theta) d\epsilon - \nabla_{\phi} \text{KL}(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z})) \\ &\approx \nabla_{\phi} \log p(\mathbf{x}|\mathbf{g}_{\phi}(\mathbf{x}, \epsilon^*), \theta) - \nabla_{\phi} \text{KL}(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}))\end{aligned}$$

Variational assumption

$$r(\epsilon) = \mathcal{N}(0, \mathbf{I}); \quad q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mu_{\phi}(\mathbf{x}), \sigma_{\phi}^2(\mathbf{x})).$$

$$\mathbf{z} = \mathbf{g}_{\phi}(\mathbf{x}, \epsilon) = \sigma_{\phi}(\mathbf{x}) \odot \epsilon + \mu_{\phi}(\mathbf{x}).$$

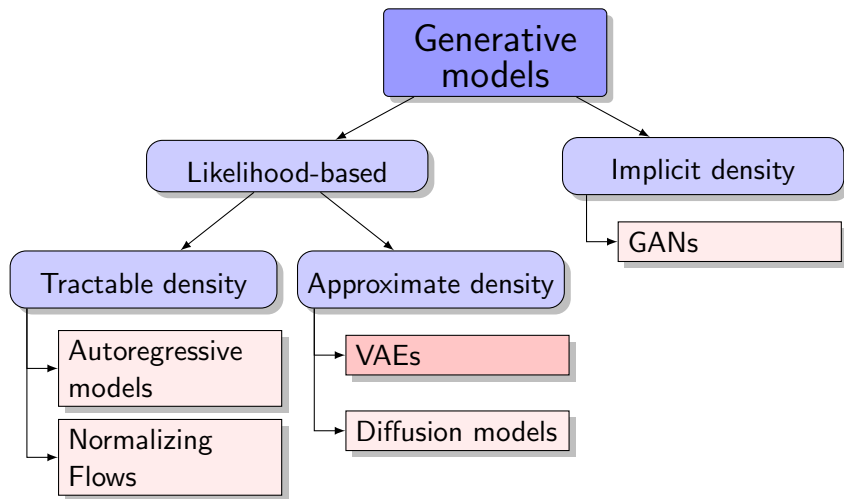
Here $\mu_{\phi}(\cdot), \sigma_{\phi}(\cdot)$ are parameterized functions (outputs of neural network).

- ▶ $p(\mathbf{z})$ – prior distribution on latent variables \mathbf{z} . We could specify any distribution that we want. Let say $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$.
- ▶ $p(\mathbf{x}|\mathbf{z}, \theta)$ – generative distribution. Since it is a parameterized function let it be neural network with parameters θ .

Outline

1. ELBO gradients, reparametrization trick
2. Variational autoencoder (VAE)
3. Normalizing flows as VAE model
4. Discrete VAE latent representations

Generative models zoo



Variational autoencoder (VAE)

Final EM-algorithm

- ▶ pick random sample $\mathbf{x}_i, i \sim U[1, n]$.
- ▶ compute the objective:

$$\epsilon^* \sim r(\epsilon); \quad \mathbf{z}^* = \mathbf{g}_\phi(\mathbf{x}, \epsilon^*);$$

$$\mathcal{L}_{\phi, \theta}(\mathbf{x}) \approx \log p(\mathbf{x}|\mathbf{z}^*, \theta) - KL(q(\mathbf{z}^*|\mathbf{x}, \phi) || p(\mathbf{z}^*)).$$

- ▶ compute a stochastic gradients w.r.t. ϕ and θ

$$\nabla_\phi \mathcal{L}_{\phi, \theta}(\mathbf{x}) \approx \nabla_\phi \log p(\mathbf{x}|\mathbf{g}_\phi(\mathbf{x}, \epsilon^*), \theta) - \nabla_\phi KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}));$$

$$\nabla_\theta \mathcal{L}_{\phi, \theta}(\mathbf{x}) \approx \nabla_\theta \log p(\mathbf{x}|\mathbf{z}^*, \theta).$$

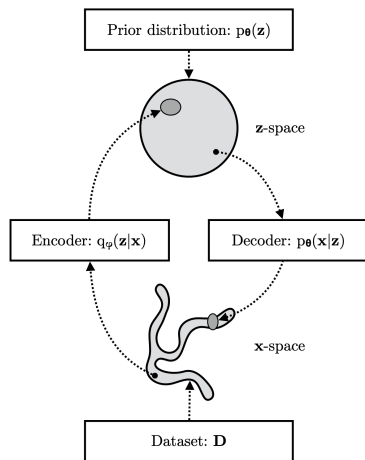
- ▶ update θ, ϕ according to the selected optimization method (SGD, Adam, etc):

$$\phi := \phi + \eta \cdot \nabla_\phi \mathcal{L}_{\phi, \theta}(\mathbf{x}),$$

$$\theta := \theta + \eta \cdot \nabla_\theta \mathcal{L}_{\phi, \theta}(\mathbf{x}).$$

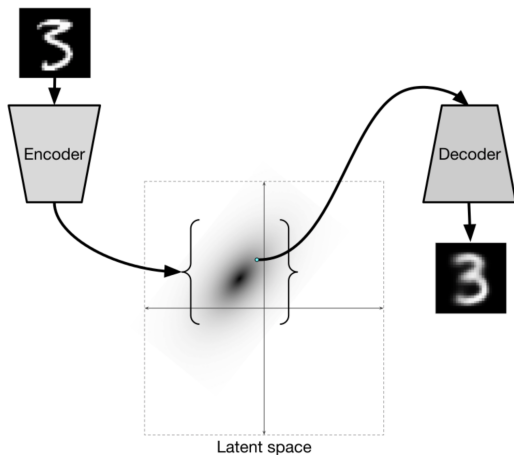
Variational autoencoder (VAE)

- ▶ VAE learns stochastic mapping between \mathbf{x} -space, from complicated distribution $\pi(\mathbf{x})$, and a latent \mathbf{z} -space, with simple distribution.
- ▶ The generative model learns a joint distribution $p(\mathbf{x}, \mathbf{z}|\theta) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}, \theta)$, with a prior distribution $p(\mathbf{z})$, and a stochastic decoder $p(\mathbf{x}|\mathbf{z}, \theta)$.
- ▶ The stochastic encoder $q(\mathbf{z}|\mathbf{x}, \phi)$ (inference model), approximates the true but intractable posterior $p(\mathbf{z}|\mathbf{x}, \theta)$ of the generative model.



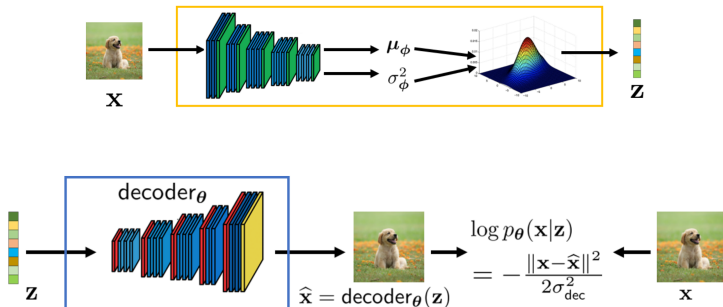
Variational Autoencoder

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



Variational autoencoder (VAE)

- ▶ Encoder $q(\mathbf{z}|\mathbf{x}, \phi) = \text{NN}_e(\mathbf{x}, \phi)$ outputs $\mu_\phi(\mathbf{x})$ and $\sigma_\phi(\mathbf{x})$.
- ▶ Decoder $p(\mathbf{x}|\mathbf{z}, \theta) = \text{NN}_d(\mathbf{z}, \theta)$ outputs parameters of the sample distribution.



Outline

1. ELBO gradients, reparametrization trick
2. Variational autoencoder (VAE)
3. Normalizing flows as VAE model
4. Discrete VAE latent representations

VAE vs Normalizing flows

	VAE	NF
Objective	ELBO \mathcal{L}	Forward KL/MLE
Encoder	stochastic $\mathbf{z} \sim q(\mathbf{z} \mathbf{x}, \phi)$	deterministic $\mathbf{z} = \mathbf{f}_{\theta}(\mathbf{x})$ $q(\mathbf{z} \mathbf{x}, \theta) = \delta(\mathbf{z} - \mathbf{f}_{\theta}(\mathbf{x}))$
Decoder	stochastic $\mathbf{x} \sim p(\mathbf{x} \mathbf{z}, \theta)$	deterministic $\mathbf{x} = \mathbf{g}_{\theta}(\mathbf{z})$ $p(\mathbf{x} \mathbf{z}, \theta) = \delta(\mathbf{x} - \mathbf{g}_{\theta}(\mathbf{z}))$
Parameters	ϕ, θ	$\theta \equiv \phi$

Theorem

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

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

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

Normalizing flow as VAE

Proof

1. Dirac delta function property

$$\mathbb{E}_{\delta(\mathbf{x}-\mathbf{y})}\mathbf{f}(\mathbf{x}) = \int \delta(\mathbf{x}-\mathbf{y})\mathbf{f}(\mathbf{x})d\mathbf{x} = \mathbf{f}(\mathbf{y}).$$

2. CoV theorem and Bayes theorem:

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{z})|\det(\mathbf{J}_{\mathbf{f}})|;$$

$$p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})}{p(\mathbf{x}|\boldsymbol{\theta})}; \quad \Rightarrow \quad p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})|\det(\mathbf{J}_{\mathbf{f}})|.$$

3. Log-likelihood decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}) + KL(q(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) = \mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}).$$

Normalizing flow as VAE

Proof

ELBO objective:

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\theta)} \left[\log p(\mathbf{x}|\mathbf{z}, \theta) - \log \frac{q(\mathbf{z}|\mathbf{x}, \theta)}{p(\mathbf{z})} \right] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\theta)} \left[\log \frac{p(\mathbf{x}|\mathbf{z}, \theta)}{q(\mathbf{z}|\mathbf{x}, \theta)} + \log p(\mathbf{z}) \right].\end{aligned}$$

1. Dirac delta function property:

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\theta)} \log p(\mathbf{z}) = \int \delta(\mathbf{z} - \mathbf{f}_\theta(\mathbf{x})) \log p(\mathbf{z}) d\mathbf{z} = \log p(\mathbf{f}_\theta(\mathbf{x})).$$

2. CoV theorem and Bayes theorem:

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\theta)} \log \frac{p(\mathbf{x}|\mathbf{z}, \theta)}{q(\mathbf{z}|\mathbf{x}, \theta)} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\theta)} \log \frac{p(\mathbf{z}|\mathbf{x}, \theta) |\det(\mathbf{J}_\mathbf{f})|}{q(\mathbf{z}|\mathbf{x}, \theta)} = \log |\det \mathbf{J}_\mathbf{f}|.$$

3. Log-likelihood decomposition

$$\log p(\mathbf{x}|\theta) = \mathcal{L}_\theta(\mathbf{x}) = \log p(\mathbf{f}_\theta(\mathbf{x})) + \log |\det \mathbf{J}_\mathbf{f}|.$$

Outline

1. ELBO gradients, reparametrization trick
2. Variational autoencoder (VAE)
3. Normalizing flows as VAE model
4. Discrete VAE latent representations

Discrete VAE latents

Motivation

- ▶ Previous VAE models had **continuous** latent variables \mathbf{z} .
- ▶ **Discrete** representations \mathbf{z} are potentially a more natural fit for many of the modalities.
- ▶ Powerful autoregressive models (like PixelCNN) have been developed for modelling distributions over discrete variables.
- ▶ All cool transformer-like models work with discrete tokens.

ELBO

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

- ▶ Reparametrization trick to get unbiased gradients.
- ▶ Normal assumptions for $q(\mathbf{z}|\mathbf{x}, \phi)$ and $p(\mathbf{z})$ to compute KL analytically.

Discrete VAE latents

Assumptions

- ▶ Let $c \sim \text{Categorical}(\boldsymbol{\pi})$, where

$$\boldsymbol{\pi} = (\pi_1, \dots, \pi_K), \quad \pi_k = P(c = k), \quad \sum_{k=1}^K \pi_k = 1.$$

- ▶ Let VAE model has discrete latent representation c with prior $p(c) = \text{Uniform}\{1, \dots, K\}$.

ELBO

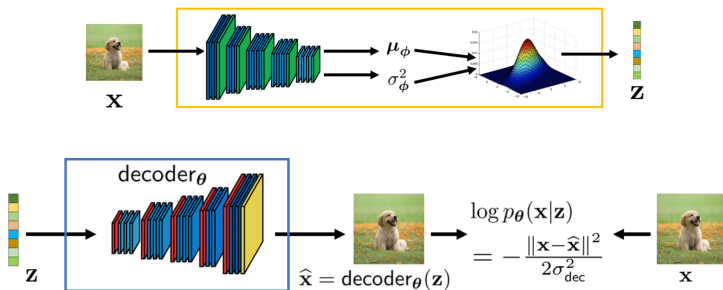
$$\mathcal{L}_{\phi, \theta}(\mathbf{x}) = \mathbb{E}_{q(c|\mathbf{x}, \phi)} \log p(\mathbf{x}|c, \theta) - \textcolor{brown}{KL}(q(c|\mathbf{x}, \phi) || p(c)) \rightarrow \max_{\phi, \theta}.$$

$$\begin{aligned} \textcolor{brown}{KL}(q(c|\mathbf{x}, \phi) || p(c)) &= \sum_{k=1}^K q(k|\mathbf{x}, \phi) \log \frac{q(k|\mathbf{x}, \phi)}{p(k)} = \\ &= \sum_{k=1}^K q(k|\mathbf{x}, \phi) \log q(k|\mathbf{x}, \phi) - \sum_{k=1}^K q(k|\mathbf{x}, \phi) \log p(k) = \\ &= -H(q(c|\mathbf{x}, \phi)) + \log K. \end{aligned}$$

Discrete VAE latents

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

- ▶ Our encoder should output discrete distribution $q(c|\mathbf{x}, \phi)$.
- ▶ We need the analogue of the reparametrization trick for the discrete distribution $q(c|\mathbf{x}, \phi)$.
- ▶ Our decoder $p(\mathbf{x}|c, \theta)$ should input discrete random variable c .



Summary

- ▶ The reparametrization trick gets unbiased gradients w.r.t to the variational posterior distribution $q(\mathbf{z}|\mathbf{x}, \phi)$.
- ▶ The VAE model is an LVM with two neural network: stochastic encoder $q(\mathbf{z}|\mathbf{x}, \phi)$ and stochastic decoder $p(\mathbf{x}|\mathbf{z}, \theta)$.
- ▶ NF models could be treated as VAE model with deterministic encoder and decoder.
- ▶ Discrete VAE representations is a natural form of latent variables.