Deep Generative Models

Lecture 4

Roman Isachenko

AI Masters

2025, Spring

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

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

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;
- $\mathbf{z} \sim p(\mathbf{z})$, $\mathbf{x} = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z})$, $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$;
- $\mathbf{x} \sim \pi(\mathbf{x})$, $\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$, $\mathbf{z} \sim p(\mathbf{z}|\boldsymbol{\theta})$.

Papamakarios G. et al. Normalizing flows for probabilistic modeling and inference, 2019

Recap of previous lecture

Posterior distribuiton (Bayes theorem)

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

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

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)

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

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
$$\max_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}) \quad \rightarrow \quad \max_{q,\boldsymbol{\theta}} \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x})$$
- Maximization of ELBO by variational distribution q is equivalent to minimization of KL
$$\arg\max_{q} \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) \equiv \arg\min_{q} KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x},\boldsymbol{\theta})).$$

Recap of previous lecture

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

EM-algorithm (block-coordinate optimization)

- Initialize $\boldsymbol{\theta}^*$;
- **E-step** ($\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) \rightarrow \max_q$)

$$q^*(\mathbf{z}) = \arg\max_q \mathcal{L}_{q,\boldsymbol{\theta}^*}(\mathbf{x}) =$$
$$= \arg\min_q KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*)) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*);$$

- **M-step** ($\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) \rightarrow \max_{\boldsymbol{\theta}}$)

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{L}_{q^*,\boldsymbol{\theta}}(\mathbf{x});$$

- Repeat E-step and M-step until convergence.

Outline

1. EM-algorithm
   - Amortized inference
   - ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

3. Discrete VAE latent representations

Outline

1. EM-algorithm
   - Amortized inference
   - ELBO gradients, reparametrization trick
2. Variational autoencoder (VAE)
3. Discrete VAE latent representations

Outline

1. EM-algorithm

    Amortized inference

    ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

3. Discrete VAE latent representations

Amortized variational inference

E-step

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

$q(\mathbf{z})$ approximates true posterior distribution $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*)$, that is why it is called **variational posterior**.

- $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*)$ could be **intractable**;
- $q(\mathbf{z})$ is different for each object $\mathbf{x}$.

Variational Bayes

Restrict a family of all possible distributions $q(\mathbf{z})$ to a parametric class $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ conditioned on samples $\mathbf{x}$ with parameters $\boldsymbol{\phi}$.

- E-step

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\phi}} \mathcal{L}_{\boldsymbol{\phi}, \boldsymbol{\theta}_{k-1}}(\mathbf{x})\big|_{\boldsymbol{\phi}=\boldsymbol{\phi}_{k-1}}$$

- M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi}_k, \boldsymbol{\theta}}(\mathbf{x})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}}$$

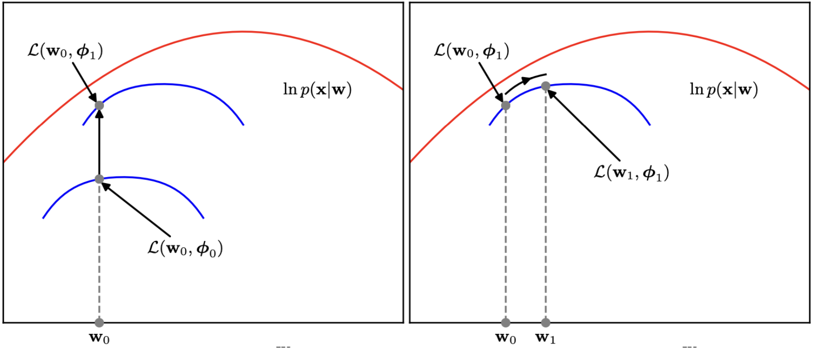
Variational EM illustration

- E-step

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\phi}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}_{k-1}}(\mathbf{x})\Big|_{\boldsymbol{\phi}=\boldsymbol{\phi}_{k-1}}$$

- M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi}_k,\boldsymbol{\theta}}(\mathbf{x})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}}$$


$\mathcal{L}(\mathbf{w}_0, \boldsymbol{\phi}_1)$
$\ln p(\mathbf{x}|\mathbf{w})$
$\mathcal{L}(\mathbf{w}_0, \boldsymbol{\phi}_0)$
$\mathbf{w}_0$

$\mathcal{L}(\mathbf{w}_0, \boldsymbol{\phi}_1)$
$\ln p(\mathbf{x}|\mathbf{w})$
$\mathcal{L}(\mathbf{w}_1, \boldsymbol{\phi}_1)$
$\mathbf{w}_0$
$\mathbf{w}_1$

Bishop C. Deep Learning: Foundations and Concepts, 2024

Variational EM-algorithm

ELBO

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

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

- **E-step**

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\phi}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}_{k-1}}(\mathbf{x})\big|_{\boldsymbol{\phi}=\boldsymbol{\phi}_{k-1}},$$

  where ϕ – parameters of the variational posterior distribution $q(\mathbf{z}|\mathbf{x}, \phi)$.

- **M-step**

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi}_k,\boldsymbol{\theta}}(\mathbf{x})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}},$$

  where $\boldsymbol{\theta}$ – parameters of the generative distribution $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

Now all that is left is to obtain **unbiased** Monte Carlo estimates of the gradients: $\nabla_{\boldsymbol{\phi}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x})$, $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x})$.

Outline

1. EM-algorithm
   - Amortized inference
   - ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

3. Discrete VAE latent representations

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

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

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

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

Naive Monte-Carlo estimation

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

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

image credit: https://jmtomczak.github.io/blog/4/4_VAE.html

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

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

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:

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

Reparametrization trick (LOTUS trick)

Suppose that $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x},\phi)$ is a random variable that is induced by the random variable $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ using the deterministic transform $\mathbf{z} = \mathbf{g}_{\phi}(\mathbf{x},\boldsymbol{\epsilon})$. Then

$$\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x},\phi)}\mathbf{f}(\mathbf{z}) = \mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})}\mathbf{f}(\mathbf{g}_{\phi}(\mathbf{x},\boldsymbol{\epsilon}))$$

Note that LHS takes the expectation by the parametric distribution $q(\mathbf{z}|\mathbf{x},\phi)$ and the RHS uses non-parametric distribution $p(\boldsymbol{\epsilon})$.

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

Reparametrization trick (LOTUS trick)

$$\nabla_{\phi} \int q(\mathbf{z}|\mathbf{x}, \phi)\mathbf{f}(\mathbf{z})d\mathbf{z} = \nabla_{\phi} \int p(\epsilon)\mathbf{f}(\mathbf{g}_{\phi}(\mathbf{x}, \epsilon))d\epsilon$$

$$= \int p(\epsilon)\nabla_{\phi}\mathbf{f}(\mathbf{g}_{\phi}(\mathbf{x}, \epsilon))d\epsilon \approx \nabla_{\phi}\mathbf{f}(\mathbf{g}_{\phi}(\mathbf{x}, \epsilon^*)),$$

where $\epsilon^* \sim p(\epsilon)$.

Variational assumption

$$p(\epsilon) = \mathcal{N}(0, \mathbf{I}); \quad \mathbf{z} = \mathbf{g}_{\phi}(\mathbf{x}, \epsilon) = \boldsymbol{\sigma}_{\phi}(\mathbf{x}) \odot \epsilon + \boldsymbol{\mu}_{\phi}(\mathbf{x});$$

$$q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\boldsymbol{\mu}_{\phi}(\mathbf{x}), \boldsymbol{\sigma}^2_{\phi}(\mathbf{x})).$$

Here $\boldsymbol{\mu}_{\phi}(\cdot), \boldsymbol{\sigma}_{\phi}(\cdot)$ are parameterized functions (outputs of neural network).
We will say that $q(\mathbf{z}|\mathbf{x}, \phi) = \mathrm{NN}_e(\mathbf{x}, \phi)$ is the **encoder**.

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

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

Reconstruction term

$$\nabla_\phi \int q(\mathbf{z}|\mathbf{x},\phi) \log p(\mathbf{x}|\mathbf{z},\boldsymbol{\theta}) d\mathbf{z} = \int p(\boldsymbol{\epsilon}) \nabla_\phi \log p(\mathbf{x}|\mathbf{g}_{\boldsymbol{\phi}}(\mathbf{x},\boldsymbol{\epsilon}),\boldsymbol{\theta}) d\boldsymbol{\epsilon} \approx$$
$$\approx \nabla_\phi \log p\left(\mathbf{x}|\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x}) \odot \boldsymbol{\epsilon}^* + \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\theta}\right), \quad \text{where } \boldsymbol{\epsilon}^* \sim \mathcal{N}(0, \mathbf{I})$$

Let the generative distibution $p(\mathbf{x}|\mathbf{z},\boldsymbol{\theta})$ be the neural network.
We will say that $p(\mathbf{x}|\mathbf{z},\boldsymbol{\theta}) = \text{NN}_d(\mathbf{z},\boldsymbol{\theta})$ is the **decoder**.

KL term

$p(\mathbf{z})$ is the prior distribution on the latent variables $\mathbf{z}$. Let assume $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$.

$$\nabla_\phi KL(q(\mathbf{z}|\mathbf{x},\phi)||p(\mathbf{z})) = \nabla_\phi KL\left(\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x}))||\mathcal{N}(0, \mathbf{I})\right)$$

This expression has analytical formula.

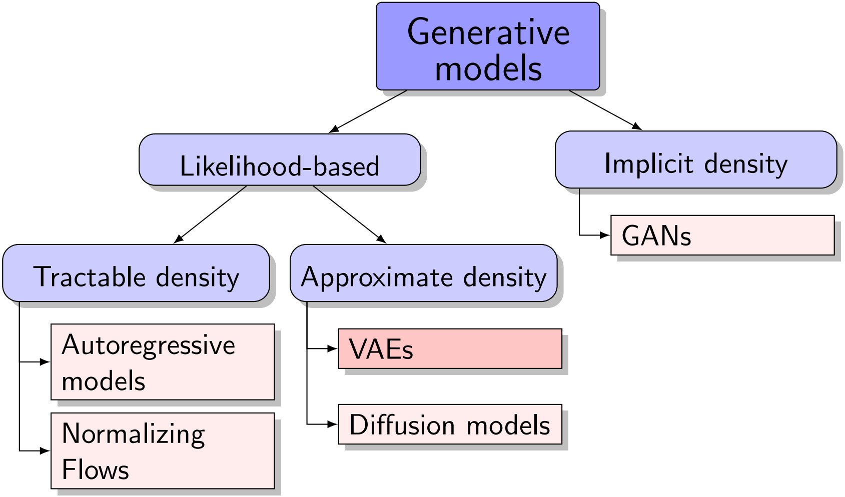
Outline

1. EM-algorithm
   - Amortized inference
   - ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

3. Discrete VAE latent representations

Generative models zoo



]

Variational autoencoder (VAE)

Training (EM-algorithm)

- pick random sample $\mathbf{x}_i$, $i \sim \text{Uniform}\{1, n\}$ (or batch).
- compute the objective (using reparametrization trick):

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

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

- make gradient step using stochastic gradients w.r.t. $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ via autograd

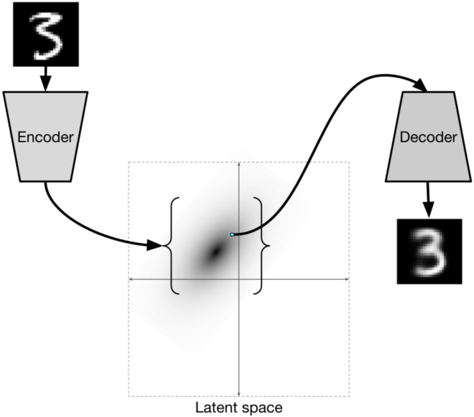
Inference

- sample $\mathbf{z}^*$ from the prior distribution $p(\mathbf{z})$ ($\mathcal{N}(0, \mathbf{I})$);
- sample from the decoder $p(\mathbf{x}|\mathbf{z}^*, \boldsymbol{\theta})$.

Note: you do not need the encoder $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ during the generation.

Variational Autoencoder

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



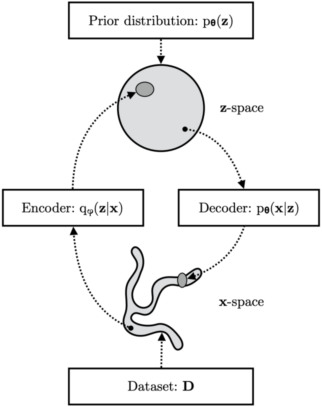
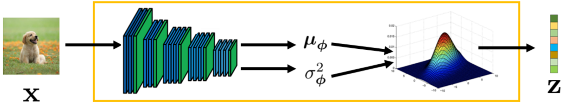
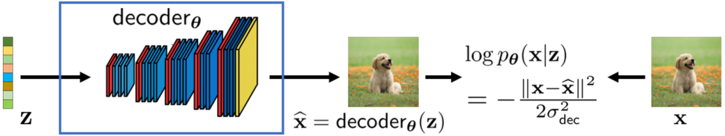

Prior distribution: $p_{\boldsymbol{\theta}}(\mathbf{z})$
z-space
Encoder: $q_{\varphi}(\mathbf{z}|\mathbf{x})$
Decoder: $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$
x-space
Dataset: **D**

image credit: http://ijdykeman.github.io/ml/2016/12/21/cvae.html
Kingma D. P., Welling M. An introduction to variational autoencoders, 2019

Variational autoencoder (VAE)

- Encoder $q(\mathbf{z}|\mathbf{x},\phi) = \mathsf{NN}_e(\mathbf{x},\phi)$ outputs $\boldsymbol{\mu}_\phi(\mathbf{x})$ and $\boldsymbol{\sigma}_\phi(\mathbf{x})$.
- Decoder $p(\mathbf{x}|\mathbf{z},\boldsymbol{\theta}) = \mathsf{NN}_d(\mathbf{z},\boldsymbol{\theta})$ outputs parameters of the sample distribution.




decoder$_{\boldsymbol{\theta}}$
$\mathbf{z}$
$\widehat{\mathbf{x}} = \text{decoder}_{\boldsymbol{\theta}}(\mathbf{z})$
$\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$
$= -\frac{\|\mathbf{x}-\widehat{\mathbf{x}}\|^2}{2\sigma^2_{\text{dec}}}$
$\mathbf{x}$

Chan S. Tutorial on Diffusion Models for Imaging and Vision, 2024

VAE vs Normalizing flows

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

Theorem

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

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

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

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

Outline

1. EM-algorithm
   - Amortized inference
   - ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

3. 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

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

- 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

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

$$KL(q(c|\mathbf{x}, \phi)||p(c)) = \sum_{k=1}^{K} q(k|\mathbf{x}, \phi) \log \frac{q(k|\mathbf{x}, \phi)}{p(k)} =$$

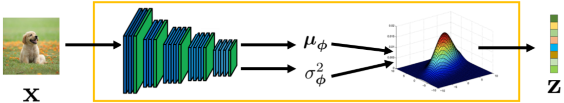
$$= \sum_{k=1}^{K} q(k|\mathbf{x}, \phi) \log q(k|\mathbf{x}, \phi) - \sum_{k=1}^{K} q(k|\mathbf{x}, \phi) \log p(k) =$$

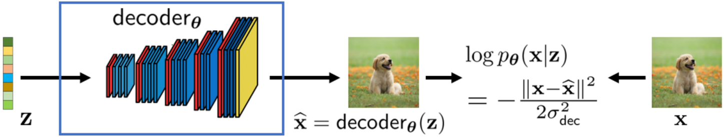
$$= -H(q(c|\mathbf{x}, \phi)) + \log K.$$

Discrete VAE latents

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

- 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 $p(\mathbf{x}|c,\boldsymbol{\theta})$ should input discrete random variable c.





$\mathbf{z}$
decoder$_{\boldsymbol{\theta}}$
$\widehat{\mathbf{x}} = \text{decoder}_{\boldsymbol{\theta}}(\mathbf{z})$
$\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = -\frac{\|\mathbf{x}-\widehat{\mathbf{x}}\|^2}{2\sigma^2_{\text{dec}}}$
$\mathbf{x}$

Chan S. Tutorial on Diffusion Models for Imaging and Vision, 2024

Summary

- Amortized variational inference allows to efficiently compute the stochastic gradients for ELBO using Monte-Carlo estimation.

- The reparametrization trick gets unbiased gradients w.r.t. 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 $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

- NF models could be treated as VAE model with deterministic encoder and decoder.

- Discrete VAE representations is a natural form of latent variables.