Deep Generative Models

Lecture 3

Roman Isachenko

Moscow Institute of Physics and Technology

2023, Autumn

Recap of previous lecture

Posterior distribution

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

Bayesian inference

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

Maximum a posteriori (MAP) estimation

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{X}) = \arg\max_{\boldsymbol{\theta}} \big(\log p(\mathbf{X}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\big)$$

MAP inference

$$p(\mathbf{x}|\mathbf{X}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta} \approx p(\mathbf{x}|\boldsymbol{\theta}^*).$$

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)

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

ELBO derivation 2 (equality)

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

Variational decomposition

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

Outline

1. ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

Outline

1. ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

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

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

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

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

$$\approx \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\mathbf{z}^*, \boldsymbol{\theta}), \quad \mathbf{z}^* \sim q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}).$$

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

The variational posterior $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ 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, $\nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta})$)

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

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:

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

Reparametrization trick (LOTUS trick)

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

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

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

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

$$= \int r(\epsilon) \nabla_{\phi} \log p(\mathbf{x}|g(\mathbf{x}, \epsilon, \phi), \boldsymbol{\theta}) d\epsilon - \nabla_{\phi} \mathrm{KL}(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}))$$

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

Variational assumption

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

$$\mathbf{z} = g(\mathbf{x}, \epsilon, \phi) = \boldsymbol{\sigma}_{\phi}(\mathbf{x}) \cdot \epsilon + \boldsymbol{\mu}_{\phi}(\mathbf{x}).$$

Here $\boldsymbol{\mu}_{\phi}(\cdot), \boldsymbol{\sigma}_{\phi}(\cdot)$ are parameterized functions (outputs of neural network).

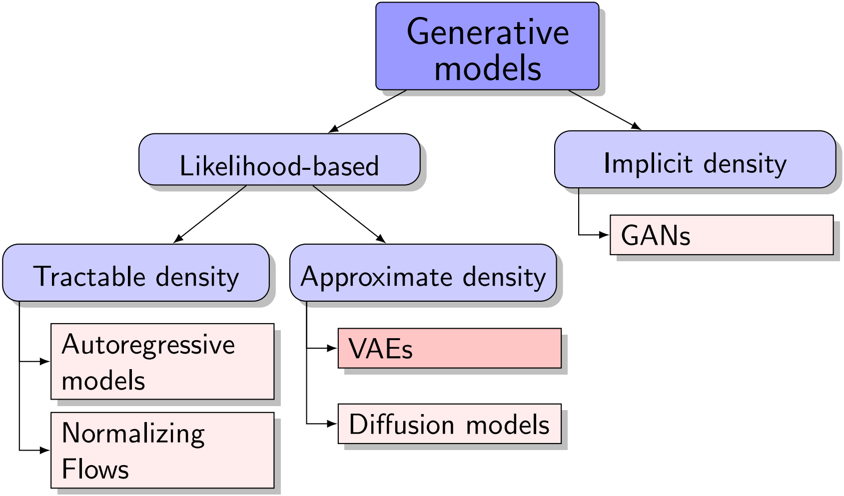
- $p(\mathbf{z})$ – prior distribution on latent variables $\mathbf{z}$. We could specify any distribution that we want. Let say $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$.
- $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ - generative distibution. Since it is a parameterized function let it be neural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

Outline

1. ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

Generative models zoo



Variational autoencoder (VAE)

Final EM-algorithm

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

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

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

- compute a stochastic gradients w.r.t. $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$

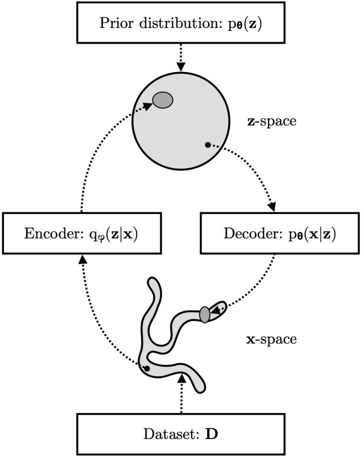
$$\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) \approx \nabla_{\boldsymbol{\phi}} \log p(\mathbf{x}|g(\mathbf{x}, \boldsymbol{\epsilon}^*, \boldsymbol{\phi}), \boldsymbol{\theta}) - \nabla_{\boldsymbol{\phi}}\mathrm{KL}(q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})||p(\mathbf{z}));$$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

- update $\boldsymbol{\theta}, \boldsymbol{\phi}$ according to the selected optimization method (SGD, Adam, RMSProp):

$$\boldsymbol{\phi} := \boldsymbol{\phi} + \eta\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}),$$
$$\boldsymbol{\theta} := \boldsymbol{\theta} + \eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}).$$

Variational autoencoder (VAE)

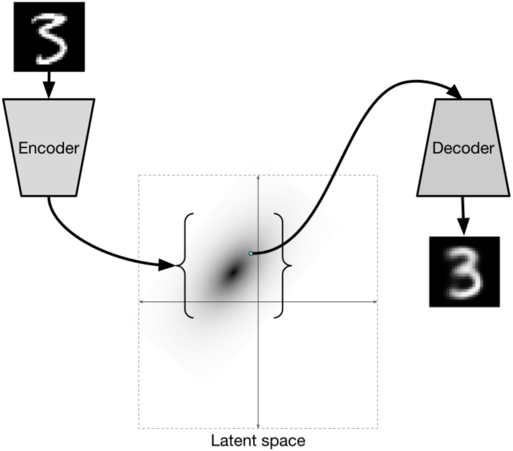
- VAE learns stochastic mapping between **x**-space, from complicated distribution $\pi(\mathbf{x})$, and a latent **z**-space, with simple distribution.
- The generative model learns a joint distribution $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$,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



Kingma D. P., Welling M. An introduction to variational autoencoders, 2019

Variational Autoencoder

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



---

image credit: http://ijdykeman.github.io/ml/2016/12/21/cvae.html

Variational autoencoder (VAE)

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

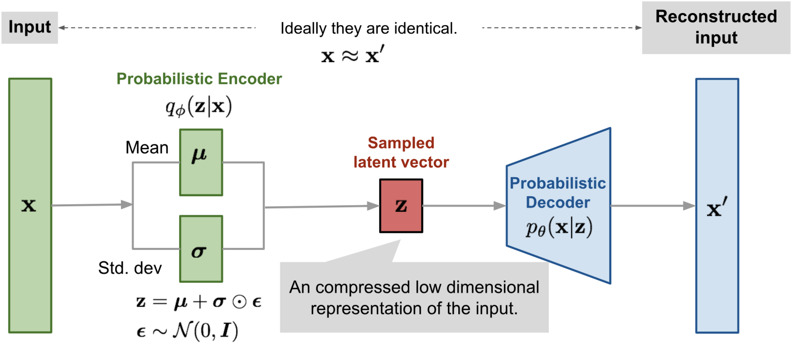


image credit: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Summary

- The general variational EM algorithm maximizes ELBO objective for LVM model to find MLE for parameters $\boldsymbol{\theta}$.
- 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 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 $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.