Deep Generative Models

Lecture 5

Roman Isachenko

AI Masters

2024, Spring

Recap of previous lecture

Forward pass (Loss function)

$$\mathbf{z} = \mathbf{x} + \int_{t_1}^{t_0} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t), t)dt, \quad L(\mathbf{z}) = -\log p(\mathbf{x}|\boldsymbol{\theta})$$

$$L(\mathbf{z}) = -\log p(\mathbf{z}) + \int_{t_0}^{t_1} \text{tr}\left(\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t), t)}{\partial \mathbf{z}(t)}\right) dt$$

Adjoint functions

$$\mathbf{a}_{\mathbf{z}}(t) = \frac{\partial L}{\partial \mathbf{z}(t)}; \quad \mathbf{a}_{\boldsymbol{\theta}}(t) = \frac{\partial L}{\partial \boldsymbol{\theta}(t)}.$$

These functions show how the gradient of the loss depends on the hidden state $\mathbf{z}(t)$ and parameters $\boldsymbol{\theta}$.

Theorem (Pontryagin)

$$\frac{d\mathbf{a}_{\mathbf{z}}(t)}{dt} = -\mathbf{a}_{\mathbf{z}}(t)^T \cdot \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t), t)}{\partial \mathbf{z}}; \quad \frac{d\mathbf{a}_{\boldsymbol{\theta}}(t)}{dt} = -\mathbf{a}_{\mathbf{z}}(t)^T \cdot \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t), t)}{\partial \boldsymbol{\theta}}.$$

Chen R. T. Q. et al. Neural Ordinary Differential Equations, 2018

Recap of previous lecture

Forward pass

$$\mathbf{z} = \mathbf{z}(t_0) = \int_{t_0}^{t_1} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t), t)dt + \mathbf{x} \quad \Rightarrow \quad \text{ODE Solver}$$

Backward pass

$$\left.\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{\theta}(t_1)} &= \mathbf{a}_{\boldsymbol{\theta}}(t_1) = -\int_{t_0}^{t_1} \mathbf{a}_{\mathbf{z}}(t)^T \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t), t)}{\partial \boldsymbol{\theta}(t)} dt + 0 \\
\frac{\partial L}{\partial \mathbf{z}(t_1)} &= \mathbf{a}_{\mathbf{z}}(t_1) = -\int_{t_0}^{t_1} \mathbf{a}_{\mathbf{z}}(t)^T \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t), t)}{\partial \mathbf{z}(t)} dt + \frac{\partial L}{\partial \mathbf{z}(t_0)} \\
\mathbf{z}(t_1) &= -\int_{t_1}^{t_0} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t), t)dt + \mathbf{z}_0.
\end{aligned}\right\} \Rightarrow \text{ODE Solver}$$

Note: These scary formulas are the standard backprop in the discrete case.

Chen R. T. Q. et al. Neural Ordinary Differential Equations, 2018

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

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

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

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{X}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(\mathbf{x}_i|\boldsymbol{\theta}) =$$

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

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

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

Recap of previous lecture

Variational lower Bound (ELBO)

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

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

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

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

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

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

Outline

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

2. Variational autoencoder (VAE)

3. Normalizing flows as VAE model

Outline

EM-algorithm

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

Block-coordinate optimization

- Initialize $\boldsymbol{\theta}^*$;

- **E-step** $(\mathcal{L}(q, \boldsymbol{\theta}) \to \max_q)$

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

- **M-step** $(\mathcal{L}(q, \boldsymbol{\theta}) \to \max_{\boldsymbol{\theta}})$

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

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

EM-algorithm illustration

Bishop C. Pattern Recognition and Machine Learning, 2006

Outline

Amortized variational inference

E-step

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

- ▶ $q(\mathbf{z})$ 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 $\mathbf{x}$.

Idea

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

Variational Bayes

- ▶ E-step

$$\phi_k = \phi_{k-1} + \eta \cdot \nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta}_{k-1})|_{\phi = \phi_{k-1}}$$

- ▶ M-step

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

Variational EM illustration

- E-step

$$\phi_k = \phi_{k-1} + \eta \nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta}_{k-1})|_{\phi=\phi_{k-1}}$$

- M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\phi_k, \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}}$$
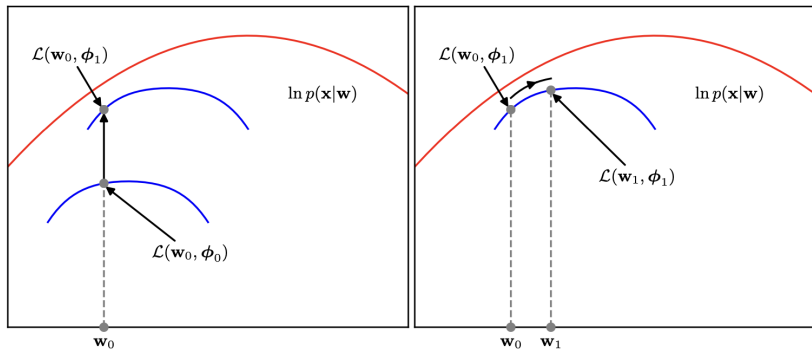

Bishop C. Deep Learning: Foundations and Concepts, 2024

Variational EM-algorithm

ELBO

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

▶ **E-step**

$$\phi_k = \phi_{k-1} + \eta \cdot \nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta}_{k-1})|_{\phi=\phi_{k-1}},$$

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

▶ **M-step**

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

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

Now all that is left is to obtain gradients: $\nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta})$, $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\phi, \boldsymbol{\theta})$.

Challenge: Number of samples n could be huge (we need derive the **unbiased** stochastic gradients).

Outline

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

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

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

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

Naive Monte-Carlo estimation

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

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, $\nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta})$)

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:

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

$$\neq \int q(\mathbf{z}|\mathbf{x}, \phi) \nabla_\phi \left[\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z})} \right] d\mathbf{z}$$

Reparametrization trick (LOTUS trick)

▶ $r(x) = \mathcal{N}(0, 1)$, $y = \sigma \cdot x + \mu$, $p(y|\boldsymbol{\theta}) = \mathcal{N}(\mu, \sigma^2)$, $\boldsymbol{\theta} = [\mu, \sigma]$.

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

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

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

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

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

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

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

Variational assumption

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

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

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 $\boldsymbol{\theta}$.
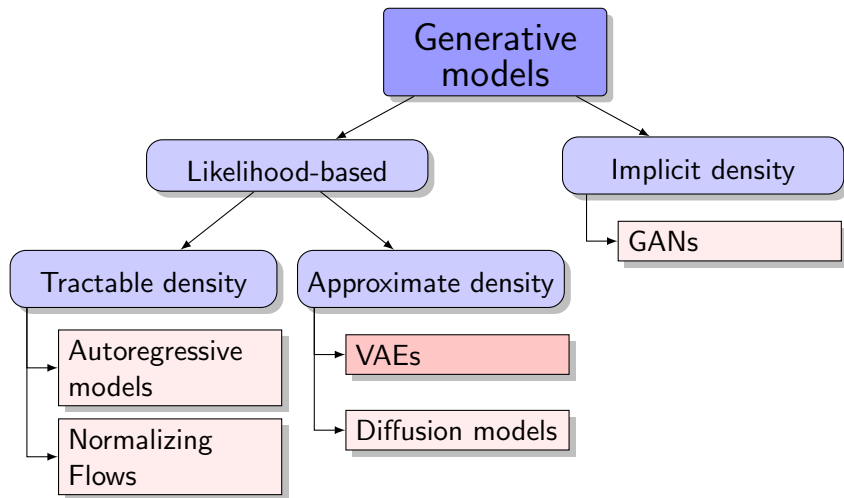
Outline

Generative models zoo

Variational autoencoder (VAE)

Final EM-algorithm

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

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

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

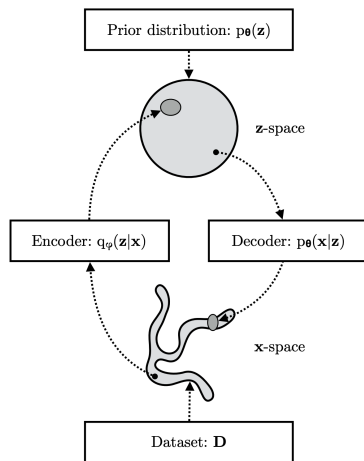- ▶ compute a stochastic gradients w.r.t. ϕ and $\boldsymbol{\theta}$

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

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

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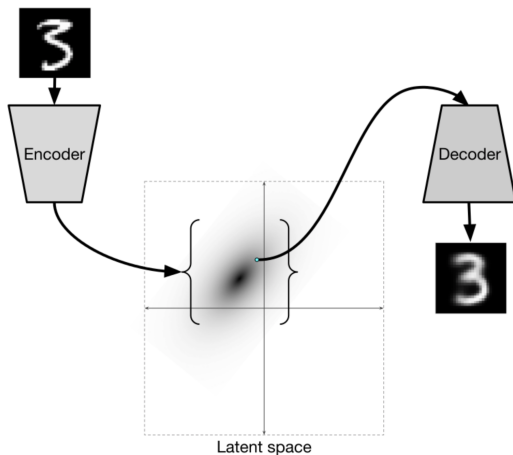
Variational autoencoder (VAE)

- ▶ 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 $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$, with a prior distribution $p(\mathbf{z})$, and a stochastic decoder $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

- ▶ The stochastic encoder $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ (inference model), approximates the true but intractable posterior $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$ of the generative model.

Prior distribution: $p_{\boldsymbol{\theta}}(\mathbf{z})$

$\mathbf{z}$-space

Encoder: $q_\varphi(\mathbf{z}|\mathbf{x})$

Decoder: $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$

$\mathbf{x}$-space

Dataset: $\mathbf{D}$

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

Variational Autoencoder

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

Latent space

Variational autoencoder (VAE)

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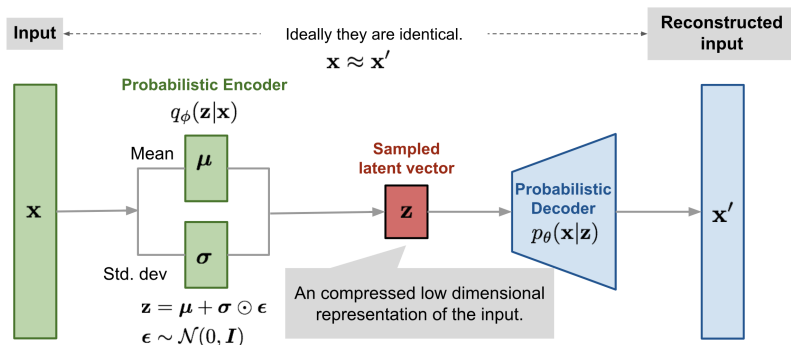

image credit:
https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Outline

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:

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

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

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

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:

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

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

3. Log-likelihood decomposition

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

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

Normalizing flow as VAE

Proof

ELBO objective:

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

$$= \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\boldsymbol{\theta})}\left[\log \frac{p(\mathbf{x}|\mathbf{z},\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x},\boldsymbol{\theta})} + \log p(\mathbf{z})\right].$$

1. Dirac delta function property:

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

2. CoV theorem and Bayes theorem:

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

3. Log-likelihood decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) = \log p(f_{\boldsymbol{\theta}}(\mathbf{x})) + \log|\det \mathbf{J_f}|.$$

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

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 $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$.

▶ The VAE model is an LVM with two neural network: stochastic encoder $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ and stochastic decoder $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

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