Deep Generative Models

Lecture 4

Roman Isachenko

 Ozon Masters

Spring, 2022

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

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

Recap of previous lecture

EM-algorithm

- E-step

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

- M-step

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

Amortized variational inference

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 $\boldsymbol{\phi}$.

Variational Bayes

- E-step

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

- M-step

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

ELBO gradients

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

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

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

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

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

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:

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

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

Reparametrization trick

Law of the unconscious statistician (LOTUS)

Let X be a random variable and let $Y = g(X)$. Then

$$\mathbb{E}_{p_Y} Y = \mathbb{E}_{p_X} g(X) = \int g(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}.$$

Examples

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

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

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

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

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

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

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

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 $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$.
- $p(\mathbf{x}|\mathbf{z}, \theta)$ - generative distibution. Since it is a parameterized function let it be neural network with parameters θ.

Variational autoencoder (VAE)

Final algorithm

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

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

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

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

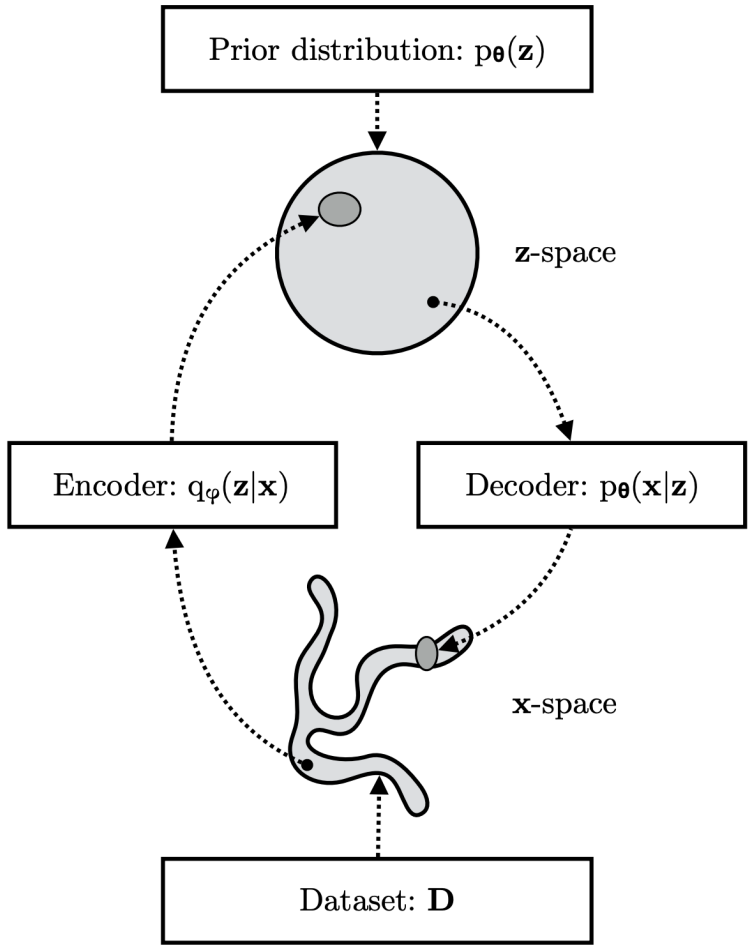
$$\nabla_\phi \mathcal{L}(\phi, \theta) \approx \nabla_\phi \log p(\mathbf{x}|g(\mathbf{x}, \epsilon^*, \phi), \theta) - \nabla_\phi \mathrm{KL}(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}));$$
$$\nabla_\theta \mathcal{L}(\phi, \theta) \approx \nabla_\theta \log p(\mathbf{x}|\mathbf{z}^*, \theta).$$

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

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

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.



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

Variational Autoencoder

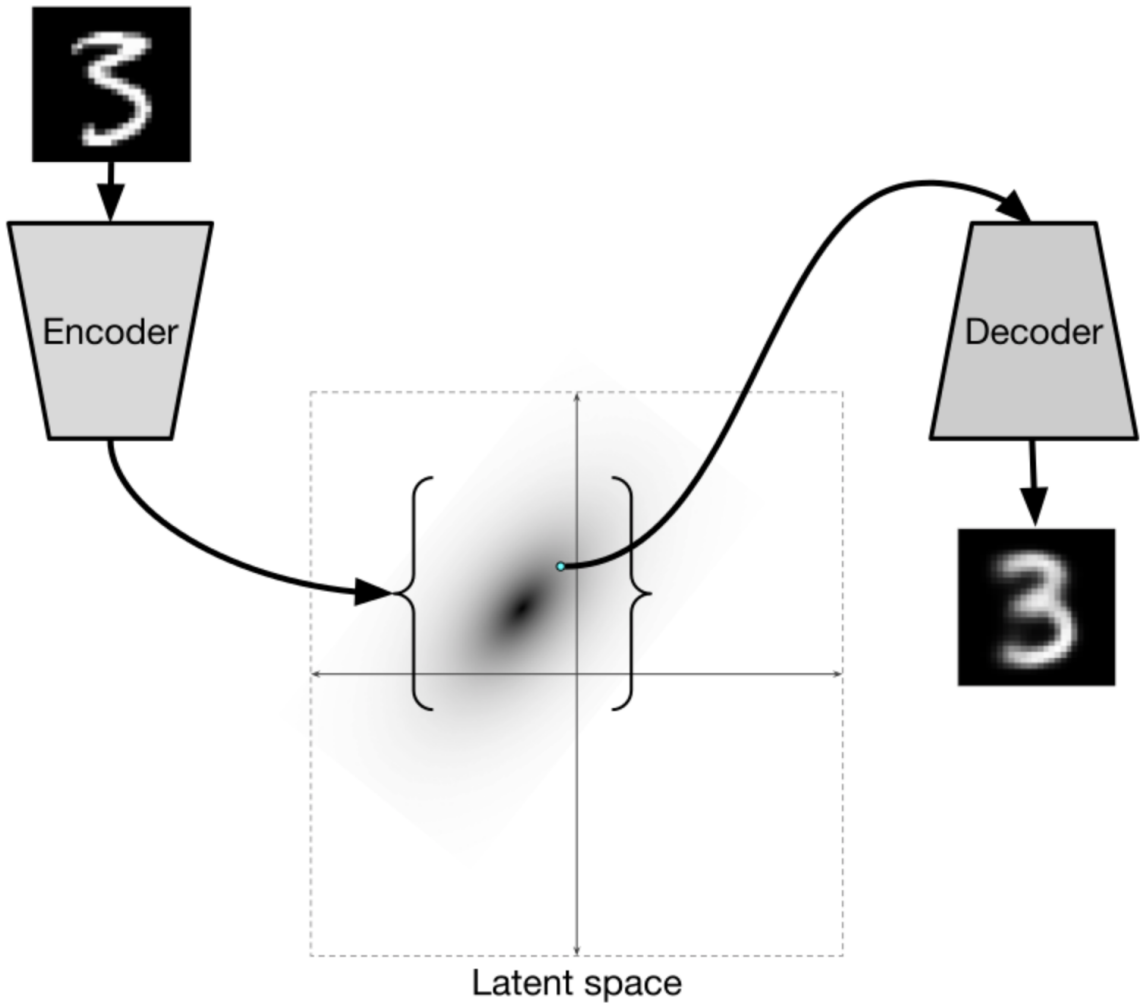


image credit: http://ijdykeman.github.io/ml/2016/12/21/cvae.html

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.

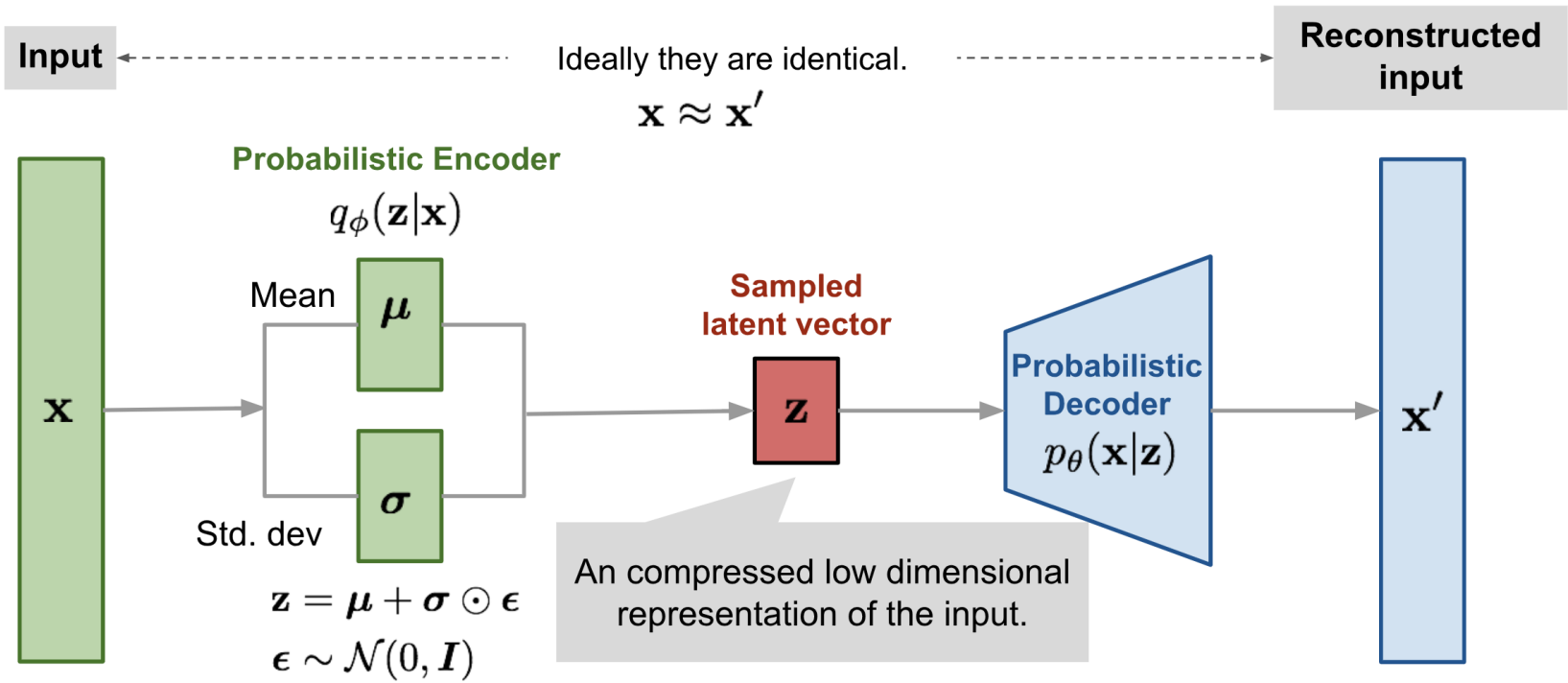


image credit:
https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Bayesian framework

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

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

MAP inference

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

VAE as Bayesian model

Posterior distribution

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

ELBO

$$\begin{aligned}\log p(\boldsymbol{\theta}|\mathbf{X}) &= \log p(\mathbf{X}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log p(\mathbf{X}) \\ &= \mathcal{L}(q, \boldsymbol{\theta}) + KL(q||p) + \log p(\boldsymbol{\theta}) - \log p(\mathbf{X}) \\ &\geq [\mathcal{L}(q, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})] - \log p(\mathbf{X}).\end{aligned}$$

EM-algorithm

- E-step

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

- M-step

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \left[\mathcal{L}(q, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right].$$

VAE limitations

- Poor variational posterior distribution (inference model encoder)

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

- Poor prior distribution

$$p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I}).$$

- Poor probabilistic model (generative model, decoder)

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

- Loose lower bound

$$\log p(\mathbf{x}|\boldsymbol{\theta}) - \mathcal{L}(q, \boldsymbol{\theta}) = (?).$$

Likelihood-based models so far...

Autoregressive models

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{i=1}^{m} p(x_i|\mathbf{x}_{1:i-1}, \boldsymbol{\theta})$$

- tractable likelihood,
- no inferred latent factors.

Latent variable models

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

- latent feature representation,
- intractable likelihood.

How to build model with latent variables and tractable likelihood?

Normalizing flows prerequisites

Change of variable theorem

Let $\mathbf{x}$ be a random variable with density function $p(\mathbf{x})$ and $f : \mathbb{R}^m \to \mathbb{R}^m$ is a differentiable, invertible function (diffeomorphism). If $\mathbf{z} = f(\mathbf{x})$, $\mathbf{x} = f^{-1}(\mathbf{z}) = g(\mathbf{z})$, then

$$p(\mathbf{x}) = p(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right| = p(f(\mathbf{x})) \left| \det \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right) \right|$$

$$p(\mathbf{z}) = p(\mathbf{x}) \left| \det \left(\frac{\partial \mathbf{x}}{\partial \mathbf{z}} \right) \right| = p(g(\mathbf{z})) \left| \det \left(\frac{\partial g(\mathbf{z})}{\partial \mathbf{z}} \right) \right|.$$

Inverse function theorem

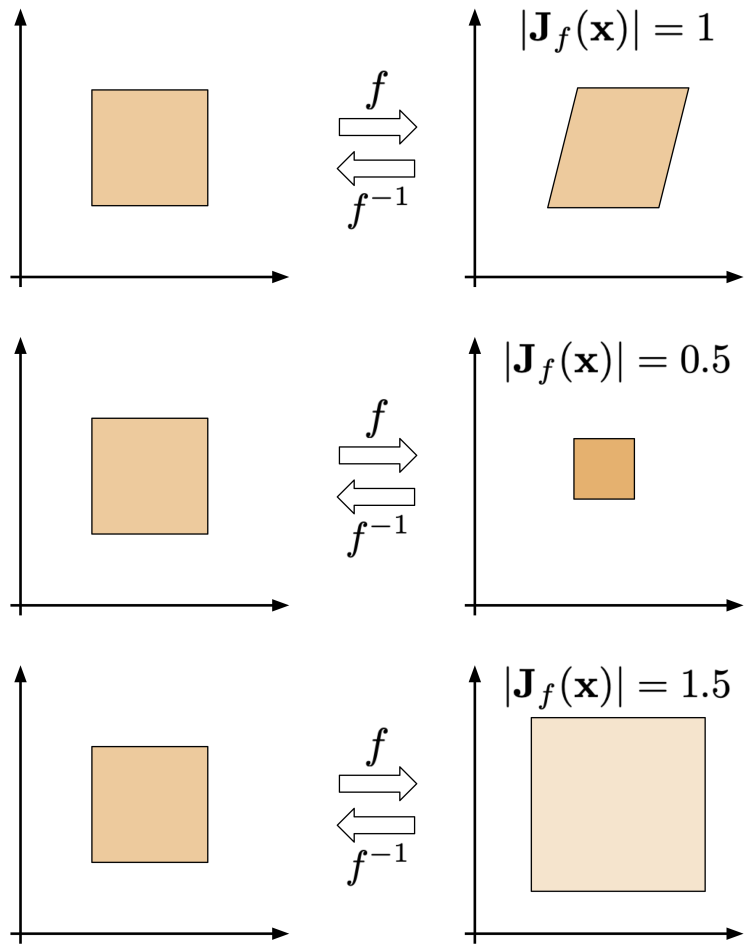
If function f is invertible and Jacobian is continuous and non-singular, then

$$\left| \det \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = \left| \det \left(\frac{\partial g^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = \left| \det \left(\frac{\partial g(\mathbf{z})}{\partial \mathbf{z}} \right) \right|^{-1}$$

Jacobian-determinant

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(f(\mathbf{x}, \boldsymbol{\theta})) \left| \det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} \right) \right|$$

- $\mathbf{x}$ and $\mathbf{z}$ have the same dimensionality ($\mathbb{R}^m$).
- $f(\mathbf{x}, \boldsymbol{\theta})$ could be parametric function.
- Determinant of Jacobian $\mathbf{J} = \frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}}$ shows how the volume changes under the transorfmation.



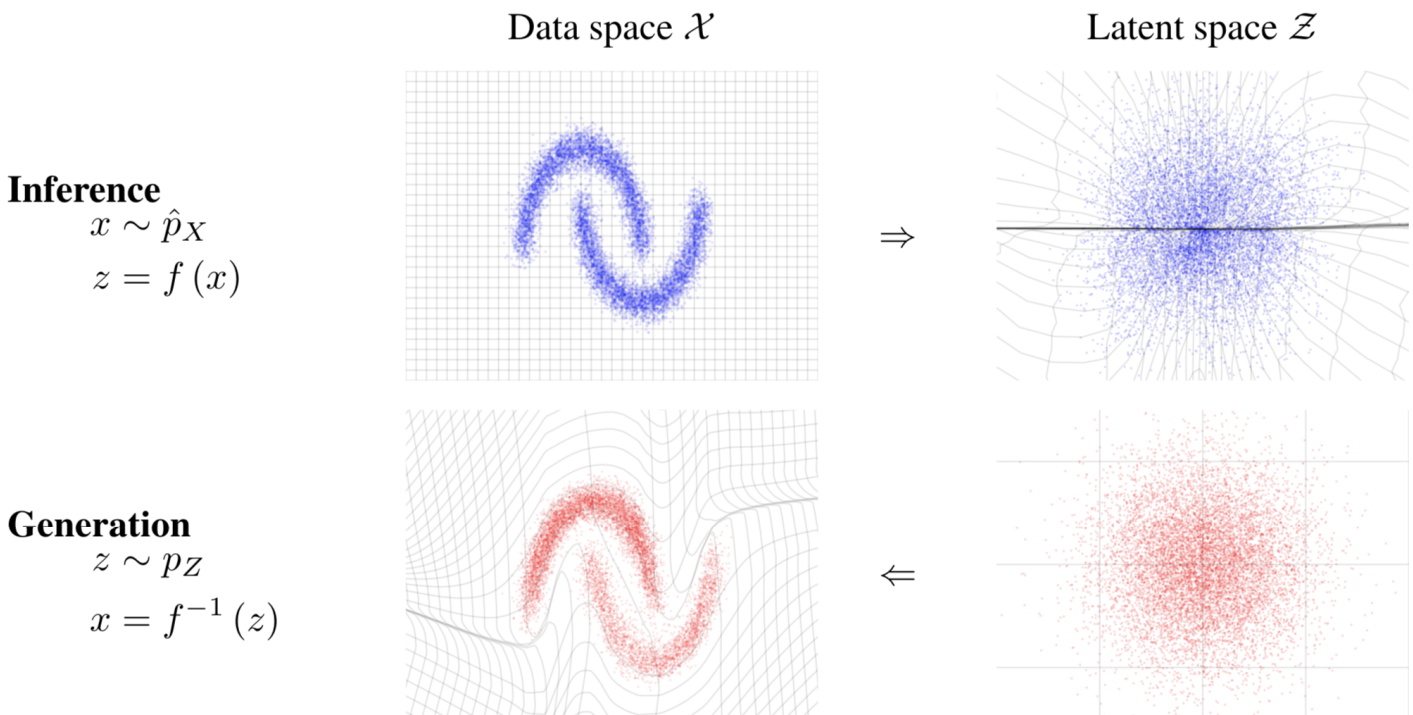
https://jmtomczak.github.io/blog/3/3_flows.html

Fitting flows

MLE problem

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{z}) \left|\det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right)\right| = p(f(\mathbf{x}, \boldsymbol{\theta})) \left|\det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}}\right)\right|$$

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log \left|\det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}}\right)\right| \to \max_{\boldsymbol{\theta}}$$



Dinh L., Sohl-Dickstein J., Bengio S. Density estimation using Real NVP, 2016

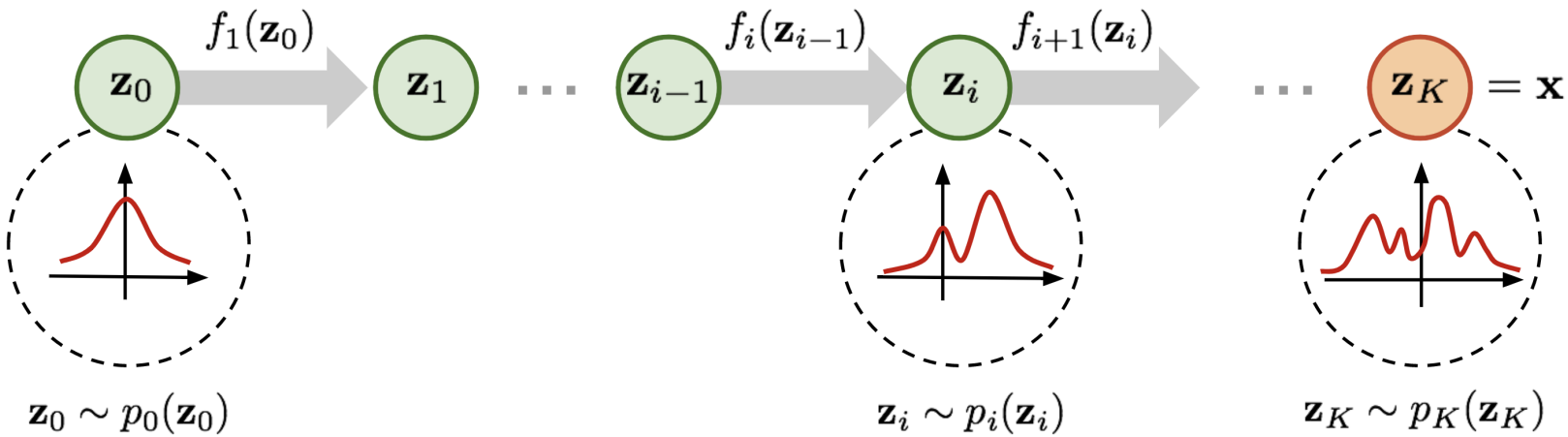
Composition of flows

Theorem

Diffeomorphisms are **composable** (If $\{f_k\}_{k=1}^K$ satisfy conditions of the change of variable theorem, then $\mathbf{z} = f(\mathbf{x}) = f_K \circ \cdots \circ f_1(\mathbf{x})$ also satisfies it).

$$p(\mathbf{x}) = p(f(\mathbf{x})) \left|\det \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right)\right| = p(f(\mathbf{x})) \left|\det \left(\frac{\partial f_K \circ \cdots \circ f_1(\mathbf{x})}{\partial \mathbf{x}}\right)\right| =$$

$$= p(f(\mathbf{x})) \left|\det \left(\frac{\partial \mathbf{f}_K}{\partial \mathbf{f}_{K-1}} \cdots \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}}\right)\right| = p(f(\mathbf{x})) \prod_{k=1}^{K} \left|\det \left(\frac{\partial \mathbf{f}_k}{\partial \mathbf{f}_{k-1}}\right)\right|$$



https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html

Flows

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log \left| \det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} \right) \right|$$

Definition

Normalizing flow is a *differentiable*, *invertible* mapping from data $\mathbf{x}$ to the noise $\mathbf{z}$.

- **Normalizing** means that the inverse flow takes samples from $p(\mathbf{x})$ and normalizes them into samples from density $p(\mathbf{z})$.
- **Flow** refers to the trajectory followed by samples from $p(\mathbf{z})$ as they are transformed by the sequence of transformations

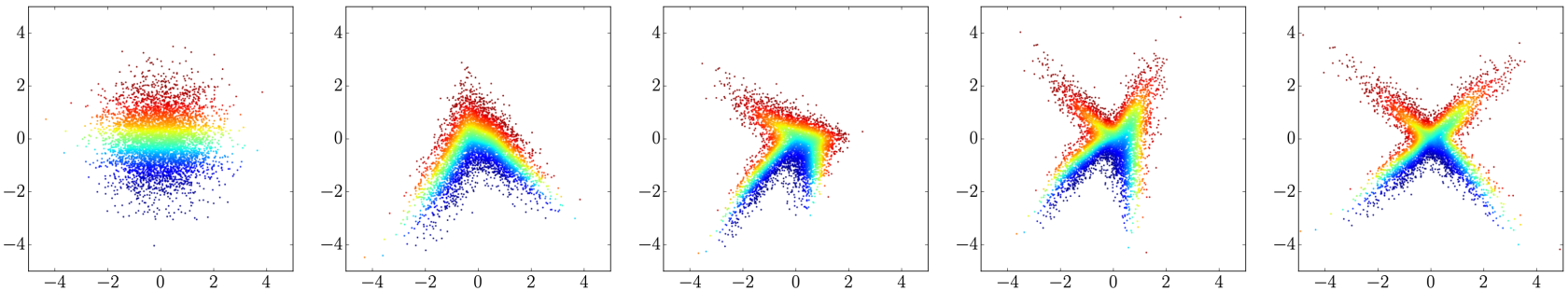
$$\mathbf{z} = f_K \circ \cdots \circ f_1(\mathbf{x}); \quad \mathbf{x} = f_1^{-1} \circ \cdots \circ f_K^{-1}(\mathbf{z}) = g_1 \circ \cdots \circ g_K(\mathbf{z})$$

$$p(\mathbf{x}) = p(f_K \circ \cdots \circ f_1(\mathbf{x})) \left| \det \left(\frac{\partial f_K \circ \cdots \circ f_1(\mathbf{x})}{\partial \mathbf{x}} \right) \right| =$$

$$= p(f_K \circ \cdots \circ f_1(\mathbf{x})) \prod_{k=1}^{K} \left| \det \left(\frac{\partial \mathbf{f}_k}{\partial \mathbf{f}_{k-1}} \right) \right|.$$

Flows

Example of a 4-step flow



Flow likelihood

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log \left| \det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} \right) \right|$$

What is the complexity of the determinant computation?

What we want

- Efficient computation of Jacobian $\frac{\partial f(\mathbf{x},\boldsymbol{\theta})}{\partial \mathbf{x}}$;
- Efficient sampling from the base distribution $p(\mathbf{z})$;
- Efficient inversion of $f(\mathbf{x}, \boldsymbol{\theta})$.

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

Forward KL vs Reverse KL

Forward KL

$$KL(\pi||p) = \int \pi(\mathbf{x}) \log \frac{\pi(\mathbf{x})}{p(\mathbf{x}|\boldsymbol{\theta})} d\mathbf{x}$$
$$= -\mathbb{E}_{\pi(\mathbf{x})} \log p(\mathbf{x}|\boldsymbol{\theta}) + \text{const} \to \min_{\boldsymbol{\theta}}$$

Maximum likelihood estimation is equivalent to minimization of the Monte-Carlo estimation of forward KL.

Forward KL for flow model

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log \left| \det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} \right) \right|$$

- We need to be able to compute $f(\mathbf{x}, \boldsymbol{\theta})$ and its Jacobian.
- We need to be able to compute the density $p(\mathbf{z})$.
- We don't need to think about computing the function $g(\mathbf{z}, \boldsymbol{\theta}) = f^{-1}(\mathbf{z}, \boldsymbol{\theta})$ until we want to sample from the flow.

Forward KL vs Reverse KL

Reverse KL

$$KL(p||\pi) = \int p(\mathbf{x}|\boldsymbol{\theta}) \log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{\pi(\mathbf{x})} d\mathbf{x}$$
$$= \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \left[\log p(\mathbf{x}|\boldsymbol{\theta}) - \log \pi(\mathbf{x})\right] \to \min_{\boldsymbol{\theta}}$$

Reverse KL for flow model

$$\log p(\mathbf{z}) = \log p(\mathbf{x}|\boldsymbol{\theta}) + \log \left|\det \left(\frac{\partial g(\mathbf{z}, \boldsymbol{\theta})}{\partial \mathbf{z}}\right)\right|$$

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

- We need to be able to compute $g(\mathbf{z}, \boldsymbol{\theta})$ and its Jacobian.
- We need to be able to sample from the density $p(\mathbf{z})$ (do not need to evaluate it).
- We don't need to think about computing the function $f(\mathbf{x}, \boldsymbol{\theta})$.

Flow KL duality

Theorem

Fitting flow model $p(\mathbf{x}|\boldsymbol{\theta})$ to the target distribution $\pi(\mathbf{x})$ using forward KL (MLE) is equivalent to fitting the induced distribution $p(\mathbf{z}|\boldsymbol{\theta})$ to the base $p(\mathbf{z})$ using reverse KL:

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

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

$$\log p(\mathbf{z}|\boldsymbol{\theta}) = \log \pi(g(\mathbf{z}, \boldsymbol{\theta})) + \log \left| \det \left(\frac{\partial g(\mathbf{z}, \boldsymbol{\theta})}{\partial \mathbf{z}} \right) \right|;$$

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log \left| \det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} \right) \right|.$$

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

Flow KL duality

Theorem

Fitting flow model $p(\mathbf{x}|\boldsymbol{\theta})$ to the target distribution $\pi(\mathbf{x})$ using forward KL (MLE) is equivalent to fitting the induced distribution $p(\mathbf{z}|\boldsymbol{\theta})$ to the base $p(\mathbf{z})$ using reverse KL:

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

Proof

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

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

$$= \mathbb{E}_{\pi(\mathbf{x})}\left[\log \pi(\mathbf{x}) - \log\left|\det\left(\frac{\partial f(\mathbf{x},\boldsymbol{\theta})}{\partial \mathbf{x}}\right)\right| - \log p(f(\mathbf{x},\boldsymbol{\theta}))\right] =$$

$$= \mathbb{E}_{\pi(\mathbf{x})}\left[\log \pi(\mathbf{x}) - \log p(\mathbf{x}|\boldsymbol{\theta})\right] = KL(\pi(\mathbf{x})||p(\mathbf{x}|\boldsymbol{\theta})).$$

Papamakarios G., Pavlakou T., Murray I. Masked Autoregressive Flow for Density Estimation, 2017

Summary

- The reparametrization trick gets unbiased gradients w.r.t to a variational posterior distribution.
- The VAE model is an LVM with two neural network: stochastic encoder $q(\mathbf{z}|\mathbf{x}, \phi)$ and stochastic decoder $p(\mathbf{x}|\mathbf{z}, \theta)$.
- VAE is not a "true" bayesian model since parameters θ do not have a prior distribution.
- Standart VAE has several limitations that we will address later in the course.
- Flow models transform a simple base distribution to a complex one via a sequence of invertible transformations.
- Flow models have a tractable likelihood that is given by the change of variable theorem.
- Flows could be fitted using forward and reverse KL minimization. We will consider each of the scenarios later in the course.