Deep Generative Models

Lecture 3

Roman Isachenko

Moscow Institute of Physics and Technology
Yandex School of Data Analysis

2025, Autumn

Recap of Previous Lecture

Jacobian Matrix

Let $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^m$ be a differentiable function.

$$\mathbf{z} = \mathbf{f}(\mathbf{x}), \quad \mathbf{J} = \frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \cdots & \frac{\partial z_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_m}{\partial x_1} & \cdots & \frac{\partial z_m}{\partial x_m} \end{pmatrix} \in \mathbb{R}^{m \times m}$$

Change of Variables Theorem (CoV)

Let $\mathbf{x} \in \mathbb{R}^m$ be a random vector with density $p(\mathbf{x})$, and let $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^m$ be a C^1-diffeomorphism ($\mathbf{f}$ and $\mathbf{f}^{-1}$ are continuously differentiable mappings). If $\mathbf{z} = \mathbf{f}(\mathbf{x})$, then

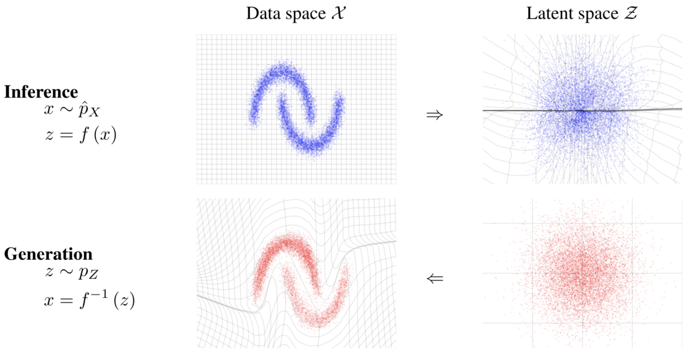
$$p(\mathbf{x}) = p(\mathbf{z}) |\det(\mathbf{J}_\mathbf{f})| = p(\mathbf{z}) \left| \det\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right) \right| = p(\mathbf{f}(\mathbf{x})) \left| \det\left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right) \right|$$

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

Recap of Previous Lecture

Definition

A normalizing flow is a C^1-diffeomorphism that transforms data $\mathbf{x}$ to noise $\mathbf{z}$.



Log-Likelihood

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log p(\mathbf{f}_K \circ \cdots \circ \mathbf{f}_1(\mathbf{x})) + \sum_{k=1}^{K} \log |\det(\mathbf{J}_{\mathbf{f}_k})|$$

Dinh L., Sohl-Dickstein J., Bengio S. Density Estimation Using Real NVP, 2016

Recap of Previous Lecture

Flow Log-Likelihood

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

One significant challenge is efficiently computing the Jacobian determinant.

Linear Flows

$$\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}\mathbf{x}, \quad \mathbf{W} \in \mathbb{R}^{m \times m}, \quad \boldsymbol{\theta} = \mathbf{W}, \quad \mathbf{J}_{\mathbf{f}} = \mathbf{W}^T$$

- LU Decomposition:
$$\mathbf{W} = \mathbf{P}\mathbf{L}\mathbf{U}.$$
- QR Decomposition:
$$\mathbf{W} = \mathbf{Q}\mathbf{R}.$$

Decomposition is performed only once during initialization. Then the decomposed matrices ($\mathbf{P}, \mathbf{L}, \mathbf{U}$ or $\mathbf{Q}, \mathbf{R}$) are optimized.

Kingma D. P., et al. Glow: Generative Flow With Invertible 1x1 Convolutions, 2018
Hoogeboom E., et al. Emerging Convolutions for Generative Normalizing Flows, 2019

Recap of Previous Lecture

Consider an autoregressive model:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \prod_{j=1}^{m} p_{\boldsymbol{\theta}}(x_j|\mathbf{x}_{1:j-1}), \quad p_{\boldsymbol{\theta}}(x_j|\mathbf{x}_{1:j-1}) = \mathcal{N}\left(\mu_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1}), \sigma^2_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1})\right).$$

Gaussian Autoregressive Normalizing Flow

$$\mathbf{x} = \mathbf{f}_{\boldsymbol{\theta}}^{-1}(\mathbf{z}) \quad \Rightarrow \quad x_j = \sigma_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1}) \cdot z_j + \mu_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1}).$$

$$\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) \quad \Rightarrow \quad z_j = (x_j - \mu_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1})) \cdot \frac{1}{\sigma_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1})}.$$

- This transformation is both C^1-**diffeomorphism**, mapping $p(\mathbf{z})$ to $p_{\boldsymbol{\theta}}(\mathbf{x})$.
- The Jacobian matrix for this transformation is triangular.

The generative function $\mathbf{f}_{\boldsymbol{\theta}}^{-1}(\mathbf{z})$ is **sequential**, while the inference function $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ is **not sequential**.

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

Recap of Previous Lecture

Let us partition $\mathbf{x}$ and $\mathbf{z}$ into two groups:

$$\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2] = [\mathbf{x}_{1:d}, \mathbf{x}_{d+1:m}]; \quad \mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2] = [\mathbf{z}_{1:d}, \mathbf{z}_{d+1:m}].$$

Coupling Layer

$$\begin{cases} \mathbf{x}_1 = \mathbf{z}_1; \\ \mathbf{x}_2 = \mathbf{z}_2 \odot \boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{z}_1) + \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_1). \end{cases} \qquad \begin{cases} \mathbf{z}_1 = \mathbf{x}_1; \\ \mathbf{z}_2 = (\mathbf{x}_2 - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_1)) \odot \frac{1}{\boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{x}_1)}. \end{cases}$$

Both density estimation and sampling require just a single pass!

Jacobian

$$\det\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right) = \det\begin{pmatrix} \mathbf{I}_d & 0_{d\times(m-d)} \\ \frac{\partial \mathbf{z}_2}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{z}_2}{\partial \mathbf{x}_2} \end{pmatrix} = \prod_{j=1}^{m-d} \frac{1}{\sigma_{j,\boldsymbol{\theta}}(\mathbf{x}_1)}.$$

A coupling layer is a special instance of an gaussian autoregressive normalizing flow.

Dinh L., Sohl-Dickstein J., Bengio S. Density Estimation Using Real NVP, 2016

Recap of Previous Lecture

Posterior Distribution (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 parameters);
- $p_{\boldsymbol{\theta}}(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta})$ – 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.

Outline

Outline

Latent Variable Models (LVM)

Maximum Likelihood Extimation (MLE) Problem

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

Latent Variable Models (LVM)

Maximum Likelihood Extimation (MLE) Problem

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

The distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ can be highly complex and often intractable (just like the true data distribution $p_{\text{data}}(\mathbf{x})$).

Latent Variable Models (LVM)

Maximum Likelihood Extimation (MLE) Problem

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

The distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ can be highly complex and often intractable (just like the true data distribution $p_{\text{data}}(\mathbf{x})$).

Extended Probabilistic Model

Introduce a latent variable $\mathbf{z}$ for each observed sample $\mathbf{x}$:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z}); \quad \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) + \log p(\mathbf{z}).$$

Latent Variable Models (LVM)

Maximum Likelihood Extimation (MLE) Problem

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

The distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ can be highly complex and often intractable (just like the true data distribution $p_{\text{data}}(\mathbf{x})$).

Extended Probabilistic Model

Introduce a latent variable $\mathbf{z}$ for each observed sample $\mathbf{x}$:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z}); \quad \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) + \log p(\mathbf{z}).$$

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

Latent Variable Models (LVM)

Maximum Likelihood Extimation (MLE) Problem

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

The distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ can be highly complex and often intractable (just like the true data distribution $p_{\text{data}}(\mathbf{x})$).

Extended Probabilistic Model

Introduce a latent variable $\mathbf{z}$ for each observed sample $\mathbf{x}$:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z}); \quad \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) + \log p(\mathbf{z}).$$

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

Motivation

Both $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ and $p(\mathbf{z})$ are usually much simpler than $p_{\boldsymbol{\theta}}(\mathbf{x})$.

Latent Variable Models (LVM)

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \rightarrow \max_{\boldsymbol{\theta}}$$

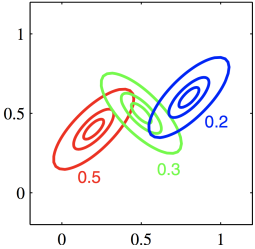
Bishop C. Pattern Recognition and Machine Learning, 2006

Latent Variable Models (LVM)

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \to \max_{\boldsymbol{\theta}}$$

Examples

Mixture of Gaussians



- $p_{\boldsymbol{\theta}}(\mathbf{x}|z) = \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$
- $p(z) = \text{Categorical}(\boldsymbol{\pi})$

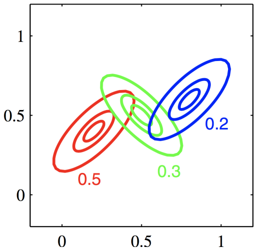
Bishop C. Pattern Recognition and Machine Learning, 2006

Latent Variable Models (LVM)

$$\log p_{\theta}(\mathbf{x}) = \log \int p_{\theta}(\mathbf{x}|\mathbf{z})p(\mathbf{z})dz \to \max_{\theta}$$

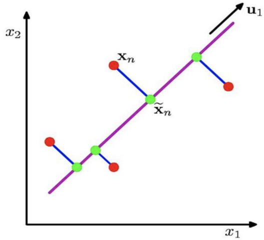
Examples

Mixture of Gaussians



- $p_{\theta}(\mathbf{x}|z) = \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$
- $p(z) = \text{Categorical}(\boldsymbol{\pi})$

PCA Model



- $p_{\theta}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$
- $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$

Bishop C. Pattern Recognition and Machine Learning, 2006

MLE for LVM

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

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

MLE for LVM

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

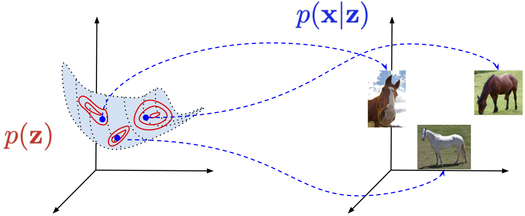
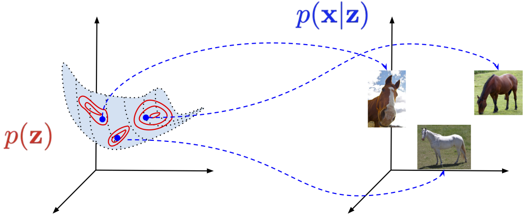


image credit: https://jmtomczak.github.io/blog/4/4_VAE.html

MLE for LVM

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



Naive Approach

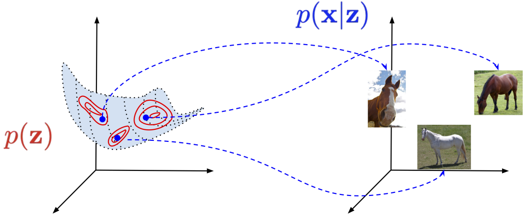
$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} = \mathbb{E}_{p(\mathbf{z})} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) \approx \frac{1}{K} \sum_{k=1}^{K} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_k),$$

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

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

MLE for LVM

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



Naive Approach

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

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

Challenge: As the dimensionality of $\mathbf{z}$ increases, the number of samples needed to adequately cover the latent space grows exponentially.

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

Outline

ELBO Derivation I

Inequality Derivation

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z}$$

ELBO Derivation I

Inequality Derivation

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int \frac{q(\mathbf{z})}{q(\mathbf{z})} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z}$$

ELBO Derivation I

Inequality Derivation

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int \frac{q(\mathbf{z})}{q(\mathbf{z})} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} =$$

$$= \log \mathbb{E}_q \left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right]$$

ELBO Derivation I

Inequality Derivation

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int \frac{q(\mathbf{z})}{q(\mathbf{z})} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} =$$

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

ELBO Derivation I

Inequality Derivation

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int \frac{q(\mathbf{z})}{q(\mathbf{z})} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} =$$

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

Here, $q(\mathbf{z})$ is any distribution such that $\int q(\mathbf{z}) d\mathbf{z} = 1$.

ELBO Derivation I

Inequality Derivation

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int \frac{q(\mathbf{z})}{q(\mathbf{z})} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} =$$

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

Here, $q(\mathbf{z})$ is any distribution such that $\int q(\mathbf{z}) d\mathbf{z} = 1$.

Variational Evidence Lower Bound (ELBO)

$$\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_q \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \leq \log p_{\boldsymbol{\theta}}(\mathbf{x})$$

ELBO Derivation I

Inequality Derivation

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int \frac{q(\mathbf{z})}{q(\mathbf{z})} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} =$$

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

Here, $q(\mathbf{z})$ is any distribution such that $\int q(\mathbf{z}) d\mathbf{z} = 1$.

Variational Evidence Lower Bound (ELBO)

$$\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_q \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \leq \log p_{\boldsymbol{\theta}}(\mathbf{x})$$

This inequality holds for any choice of $q(\mathbf{z})$.

ELBO Derivation II

$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}$$

Equality Derivation

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

ELBO Derivation II

$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}$$

Equality Derivation

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

$$= \int q(\mathbf{z}) \log \frac{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) p_{\boldsymbol{\theta}}(\mathbf{x})}{q(\mathbf{z})} d\mathbf{z}$$

ELBO Derivation II

$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}$$

Equality Derivation

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

ELBO Derivation II

$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}$$

Equality Derivation

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

ELBO Derivation II

$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}$$

Equality Derivation

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

Variational Decomposition

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

ELBO Derivation II

$$p_{\theta}(\mathbf{z}|\mathbf{x}) = \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{p_{\theta}(\mathbf{x})}$$

Equality Derivation

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

Variational Decomposition

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

Here, $\mathrm{KL}(q(\mathbf{z}) \| p_{\theta}(\mathbf{z}|\mathbf{x})) \geq 0$.

Variational Evidence Lower Bound (ELBO)

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

Variational Evidence Lower Bound (ELBO)

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

Variational Evidence Lower Bound (ELBO)

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

Variational Evidence Lower Bound (ELBO)

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

Log-Likelihood Decomposition

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

Variational Evidence Lower Bound (ELBO)

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

Log-Likelihood Decomposition

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) + \mathrm{KL}(q(\mathbf{z}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})) = \\ = \mathbb{E}_q \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) - \mathrm{KL}(q(\mathbf{z}) \| p(\mathbf{z})) + \mathrm{KL}(q(\mathbf{z}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})).\end{aligned}$$

Variational Evidence Lower Bound (ELBO)

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

Log-Likelihood Decomposition

$$
\begin{aligned}
\log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) + \mathrm{KL}(q(\mathbf{z}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})) = \\
&= \mathbb{E}_q \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) - \mathrm{KL}(q(\mathbf{z}) \| p(\mathbf{z})) + \mathrm{KL}(q(\mathbf{z}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})).
\end{aligned}
$$

- Instead of maximizing the likelihood, maximize the ELBO:

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

Variational Evidence Lower Bound (ELBO)

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

Log-Likelihood Decomposition

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) + \mathrm{KL}(q(\mathbf{z}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})) = \\ &= \mathbb{E}_q \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) - \mathrm{KL}(q(\mathbf{z}) \| p(\mathbf{z})) + \mathrm{KL}(q(\mathbf{z}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})).\end{aligned}$$

- Instead of maximizing the likelihood, maximize the ELBO:

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

- Maximizing the ELBO with respect to the **variational** distribution q is equivalent to minimizing the KL divergence:

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

Variational Posterior

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

What is the optimal distribution $q^*(\mathbf{z})$ given fixed $\boldsymbol{\theta}^*$?

Bishop C. Pattern Recognition and Machine Learning, 2006

Variational Posterior

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

What is the optimal distribution $q^*(\mathbf{z})$ given fixed $\boldsymbol{\theta}^*$?

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

$$= \arg\min_q \mathrm{KL}(q(\mathbf{z})\|p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{x})) = p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{x});$$

Here we got the intuition about $q(\mathbf{z})$ – it estimates the posterior $p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{x})$.

Bishop C. Pattern Recognition and Machine Learning, 2006

Variational Posterior

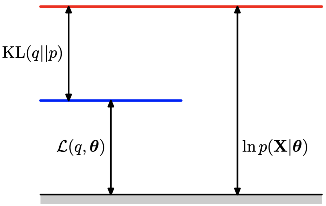
$$\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_q \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) - \mathrm{KL}(q(\mathbf{z})\|p(\mathbf{z})) \to \max_{q,\boldsymbol{\theta}}.$$

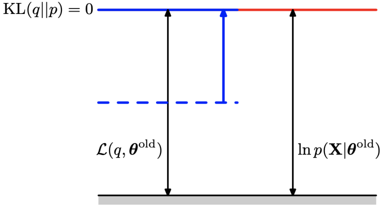
What is the optimal distribution $q^*(\mathbf{z})$ given fixed $\boldsymbol{\theta}^*$?

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

$$= \arg\min_q \mathrm{KL}(q(\mathbf{z})\|p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{x})) = p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{x});$$

Here we got the intuition about $q(\mathbf{z})$ – it estimates the posterior $p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{x})$.





Bishop C. Pattern Recognition and Machine Learning, 2006

Outline

Parametric Variable Posterior

Variational Posterior

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

Parametric Variable Posterior

Variational Posterior

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

- $p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{x})$ may be **intractable**;
- $q(\mathbf{z})$ is individual for each data point $\mathbf{x}$.

Parametric Variable Posterior

Variational Posterior

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

- $p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{x})$ may be **intractable**;
- $q(\mathbf{z})$ is individual for each data point $\mathbf{x}$.

Amortized Variational Inference

We restrict the family of possible distributions $q(\mathbf{z})$ to a parametric class $q_{\phi}(\mathbf{z}|\mathbf{x})$, conditioned on data $\mathbf{x}$ and parameterized by ϕ.

Parametric Variable Posterior

Variational Posterior

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

- $p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{x})$ may be **intractable**;
- $q(\mathbf{z})$ is individual for each data point $\mathbf{x}$.

Amortized Variational Inference

We restrict the family of possible distributions $q(\mathbf{z})$ to a parametric class $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, conditioned on data $\mathbf{x}$ and parameterized by $\boldsymbol{\phi}$.

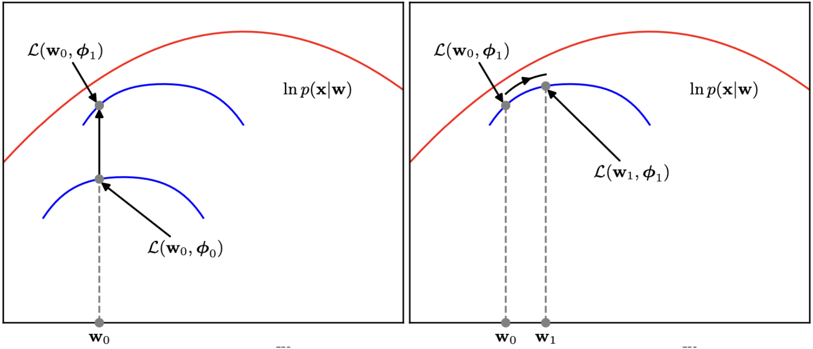
Gradient Update

$$\begin{bmatrix} \boldsymbol{\phi}_k \\ \boldsymbol{\theta}_k \end{bmatrix} = \left.\begin{bmatrix} \boldsymbol{\phi}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\phi}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}) \\ \boldsymbol{\theta}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}) \end{bmatrix}\right|_{(\boldsymbol{\phi}_{k-1},\boldsymbol{\theta}_{k-1})}$$

ELBO optimization

Gradient Update

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



Bishop C., Deep Learning: Foundations and Concepts, 2024

ELBO optimization

ELBO

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

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

ELBO optimization

ELBO

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

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

Gradient Update

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

- $\boldsymbol{\phi}$ denotes the parameters of the variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$.
- $\boldsymbol{\theta}$ represents the parameters of the generative model $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$.

ELBO optimization

ELBO

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

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

Gradient Update

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

- $\boldsymbol{\phi}$ denotes the parameters of the variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$.
- $\boldsymbol{\theta}$ represents the parameters of the generative model $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$.

The remaining step is to obtain **unbiased** Monte Carlo estimates of the gradients: $\nabla_{\boldsymbol{\phi}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x})$ and $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x})$.

Outline

ELBO Gradients: $\nabla_{\theta}\mathcal{L}_{\phi,\theta}(\mathbf{x})$

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

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

ELBO Gradients: $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x})$

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

Gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x})$

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}) = \nabla_{\boldsymbol{\theta}} \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) d\mathbf{z}$$

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

ELBO Gradients: $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x})$

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

Gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x})$

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x}) &= \nabla_{\boldsymbol{\theta}} \int q_{\phi}(\mathbf{z}|\mathbf{x}) \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) d\mathbf{z} \\
&= \int q_{\phi}(\mathbf{z}|\mathbf{x}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) d\mathbf{z}
\end{aligned}$$

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

ELBO Gradients: $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x})$

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

Gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x})$

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

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

ELBO Gradients: $\nabla_{\theta}\mathcal{L}_{\phi,\theta}(\mathbf{x})$

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

Gradient $\nabla_{\theta}\mathcal{L}_{\phi,\theta}(\mathbf{x})$

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

Naive Monte Carlo Estimation

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

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

ELBO Gradients: $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x})$

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

Gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x})$

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

Naive Monte Carlo Estimation

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

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

---

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

ELBO Gradients: $\nabla_\phi \mathcal{L}_{\phi,\theta}(\mathbf{x})$

Gradient $\nabla_\phi \mathcal{L}_{\phi,\theta}(\mathbf{x})$

Unlike the $\boldsymbol{\theta}$-gradient, the density $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ now depends on ϕ, so standard Monte Carlo estimation can't be applied:

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

ELBO Gradients: $\nabla_\phi \mathcal{L}_{\phi,\theta}(\mathbf{x})$

Gradient $\nabla_\phi \mathcal{L}_{\phi,\theta}(\mathbf{x})$

Unlike the $\boldsymbol{\theta}$-gradient, the density $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ now depends on ϕ, so standard Monte Carlo estimation can't be applied:

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

ELBO Gradients: $\nabla_\phi \mathcal{L}_{\phi,\theta}(\mathbf{x})$

Gradient $\nabla_\phi \mathcal{L}_{\phi,\theta}(\mathbf{x})$

Unlike the $\boldsymbol{\theta}$-gradient, the density $q_\phi(\mathbf{z}|\mathbf{x})$ now depends on ϕ, so standard Monte Carlo estimation can't be applied:

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

$$\neq \int q_\phi(\mathbf{z}|\mathbf{x}) \nabla_\phi \log p_\theta(\mathbf{x}|\mathbf{z}) d\mathbf{z} - \nabla_\phi \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}))$$

Reparametrization Trick (LOTUS Trick)

Assume $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ is generated by a random variable $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ via a deterministic mapping $\mathbf{z} = \mathbf{g}_\phi(\mathbf{x}, \boldsymbol{\epsilon})$. Then,

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

ELBO Gradients: $\nabla_\phi \mathcal{L}_{\phi,\theta}(\mathbf{x})$

Gradient $\nabla_\phi \mathcal{L}_{\phi,\theta}(\mathbf{x})$

Unlike the θ-gradient, the density $q_\phi(\mathbf{z}|\mathbf{x})$ now depends on ϕ, so standard Monte Carlo estimation can't be applied:

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

Reparametrization Trick (LOTUS Trick)

Assume $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ is generated by a random variable $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ via a deterministic mapping $\mathbf{z} = \mathbf{g}_\phi(\mathbf{x}, \boldsymbol{\epsilon})$. Then,

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

Note: The LHS expectation is with respect to the parametric distribution $q_\phi(\mathbf{z}|\mathbf{x})$, while the RHS is for the non-parametric $p(\boldsymbol{\epsilon})$.

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

Reparametrization Trick (LOTUS Trick)

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

,

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

Reparametrization Trick (LOTUS Trick)

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

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

where $\boldsymbol{\epsilon}^* \sim p(\boldsymbol{\epsilon})$.

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

Reparametrization Trick (LOTUS Trick)

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

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

Variational Assumption

$$p(\epsilon) = \mathcal{N}(0, \mathbf{I}); \quad \mathbf{z} = \mathbf{g}_{\phi}(\mathbf{x}, \epsilon) = \boldsymbol{\sigma}_{\phi}(\mathbf{x}) \odot \epsilon + \boldsymbol{\mu}_{\phi}(\mathbf{x});$$
$$q_{\phi}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\phi}(\mathbf{x}), \boldsymbol{\sigma}^2_{\phi}(\mathbf{x})).$$

Here, $\boldsymbol{\mu}_{\phi}(\cdot)$ and $\boldsymbol{\sigma}_{\phi}(\cdot)$ are parameterized functions (outputs of a neural network).
Thus, we can write $q_{\phi}(\mathbf{z}|\mathbf{x}) = \text{NN}_{e,\phi}(\mathbf{x})$, the **encoder**.

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

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

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

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

Reconstruction Term

$$\nabla_\phi \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) d\mathbf{z} = \int p(\boldsymbol{\epsilon}) \nabla_\phi \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{g}_\phi(\mathbf{x}, \boldsymbol{\epsilon})) d\boldsymbol{\epsilon} \approx$$

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

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

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

Reconstruction Term

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

The generative distribution $p_\theta(\mathbf{x}|\mathbf{z})$ can be implemented as a neural network.
We may write $p_\theta(\mathbf{x}|\mathbf{z}) = \mathrm{NN}_{d,\theta}(\mathbf{z})$, called the **decoder**.

KL Term

$p(\mathbf{z})$ is the prior over latents $\mathbf{z}$, typically $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$.

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

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

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

Reconstruction Term

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

The generative distribution $p_\theta(\mathbf{x}|\mathbf{z})$ can be implemented as a neural network.
We may write $p_\theta(\mathbf{x}|\mathbf{z}) = \mathrm{NN}_{d,\theta}(\mathbf{z})$, called the **decoder**.

KL Term

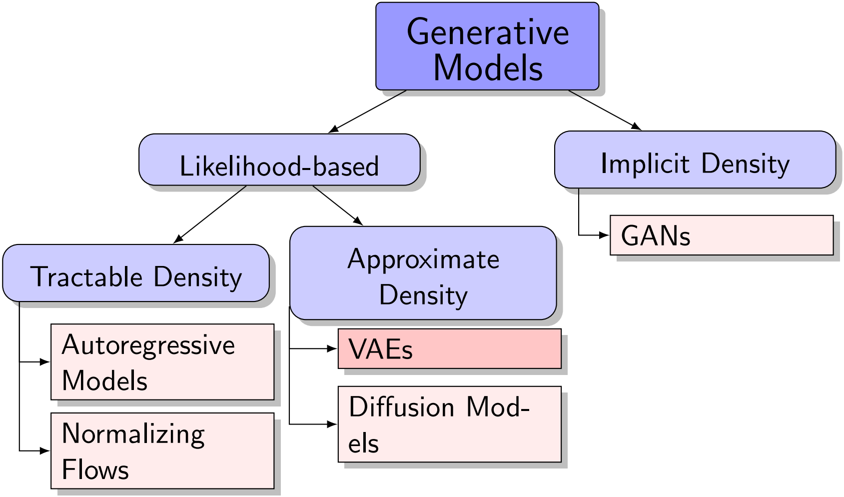
$p(\mathbf{z})$ is the prior over latents $\mathbf{z}$, typically $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$.

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

This expression admits a closed-form analytic solution.

Outline

Generative Models Zoo



Variational Autoencoder (VAE)

Training

- Pick a batch of samples $\{\mathbf{x}_i\}_{i=1^B}$ (here we use Monte Carlo technique).

Variational Autoencoder (VAE)

Training

- Pick a batch of samples $\{\mathbf{x}_i\}_{i=1^B}$ (here we use Monte Carlo technique).
- Compute the objective for each sample (apply the reparametrization trick):

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

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

Variational Autoencoder (VAE)

Training

- Pick a batch of samples $\{\mathbf{x}_i\}_{i=1^B}$ (here we use Monte Carlo technique).
- Compute the objective for each sample (apply the reparametrization trick):

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

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

- Update parameters via stochastic gradient steps with respect to $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$.

Variational Autoencoder (VAE)

Training

- Pick a batch of samples $\{\mathbf{x}_i\}_{i=1^B}$ (here we use Monte Carlo technique).
- Compute the objective for each sample (apply the reparametrization trick):

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

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

- Update parameters via stochastic gradient steps with respect to $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$.

Inference

- Sample $\mathbf{z}^*$ from the prior $p(\mathbf{z})$ ($\mathcal{N}(0, \mathbf{I})$);

Variational Autoencoder (VAE)

Training

- Pick a batch of samples $\{\mathbf{x}_i\}_{i=1^B}$ (here we use Monte Carlo technique).
- Compute the objective for each sample (apply the reparametrization trick):

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

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

- Update parameters via stochastic gradient steps with respect to $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$.

Inference

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

Variational Autoencoder (VAE)

Training

- Pick a batch of samples $\{\mathbf{x}_i\}_{i=1^B}$ (here we use Monte Carlo technique).
- Compute the objective for each sample (apply the reparametrization trick):

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

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

- Update parameters via stochastic gradient steps with respect to $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$.

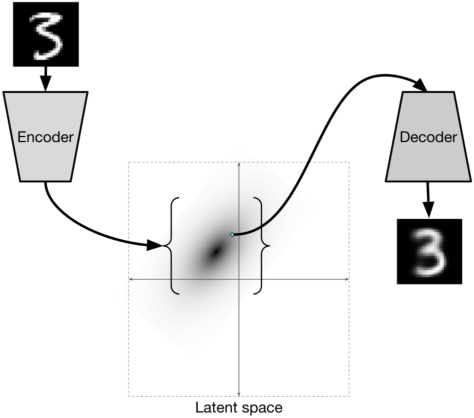
Inference

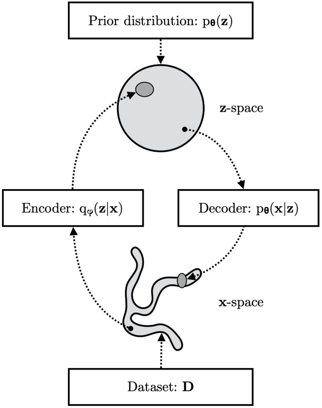
- Sample $\mathbf{z}^*$ from the prior $p(\mathbf{z})$ ($\mathcal{N}(0, \mathbf{I})$);
- Generate data from the decoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}^*)$.

Note: The encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ isn't needed during generation.

Variational Autoencoder

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



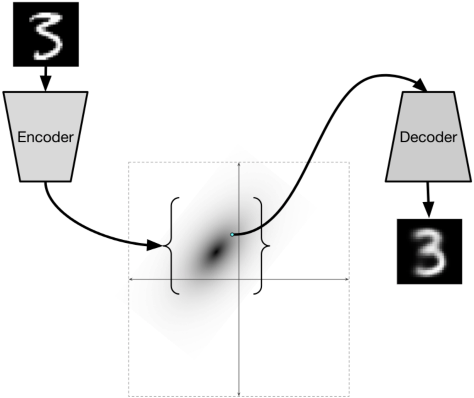


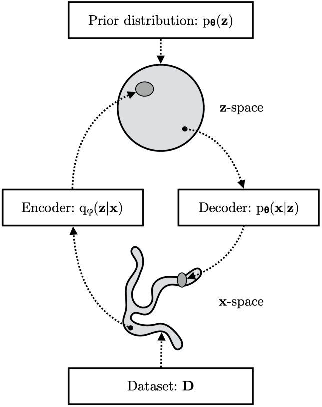
---

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

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





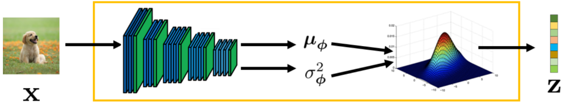
VAEs are widely used as a preliminary stage of projecting data onto low-dimensional space.

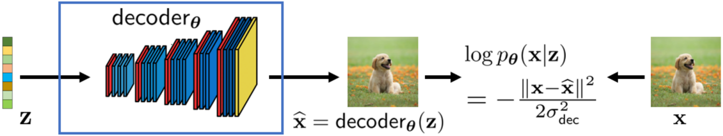
---

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

- The encoder $q_\phi(\mathbf{z}|\mathbf{x}) = \text{NN}_{e,\phi}(\mathbf{x})$ outputs $\boldsymbol{\mu}_\phi(\mathbf{x})$ and $\boldsymbol{\sigma}_\phi(\mathbf{x})$.
- The decoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \text{NN}_{d,\boldsymbol{\theta}}(\mathbf{z})$ outputs parameters of the observed data distribution.





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 $\mathbf{z} \sim q_{\phi}(\mathbf{z}\|\mathbf{x})$	deterministic $\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ $q_{\boldsymbol{\theta}}(\mathbf{z}\|\mathbf{x}) = \delta(\mathbf{z} - \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}))$
Decoder	stochastic $\mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x}\|\mathbf{z})$	deterministic $\mathbf{x} = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z})$ $p_{\boldsymbol{\theta}}(\mathbf{x}\|\mathbf{z}) = \delta(\mathbf{x} - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z}))$
Parameters	$\phi, \boldsymbol{\theta}$	$\boldsymbol{\theta} \equiv \phi$

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

VAE vs Normalizing Flows

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

Theorem

MLE for a normalizing flow is equivalent to maximizing the ELBO for a VAE where:

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

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

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

Summary

- LVMs introduce latent representations for observed data, enabling more interpretable models.
- LVMs maximize the variational evidence lower bound (ELBO) to obtain maximum likelihood estimates for the parameters.
- Parametric posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ makes the method scalable.
- The reparametrization trick provides unbiased gradients with respect to the variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$.
- The VAE model is a latent variable model parameterized by two neural networks: a stochastic encoder $q_\phi(\mathbf{z}|\mathbf{x})$ and a stochastic decoder $p_\theta(\mathbf{x}|\mathbf{z})$.
- Nowadays, the main role of VAEs is to project data into low-dimensional latent space.