CS-E4891 Deep Generative Models
Lecture 2: Latent variable models, variational inference and variational autoencoders

Harri Lähdesmäki

Department of Computer Science
Aalto University

March 24, 2025

Outline

- Latent variable models (Section 6.2 from (Murphy, 2023))
- Linear latent variable models (Sec. 28.3)
- Deep latent variable models (Sec. 21.1)
- Variational inference: amortized, stochastic, gradient-based, reparameterized (Sec. 10.1-10.2)
- Variational autoencoders (Sec. 21)

- Reading: parts of Sections 6.2, 10.1-10.2, 21.1, 28.3 from (Murphy, 2023)

Different types of latent variable models

Latent variable model

A latent variable model (LVM) is a probabilistic model that contains variables that are always unobserved

Latent variables are random variable

Latent variables can also represent model parameters

Three types of latent variable models

1. Global latent variables
2. Local latent variables
3. Global and local latent variables

Global latent variable models

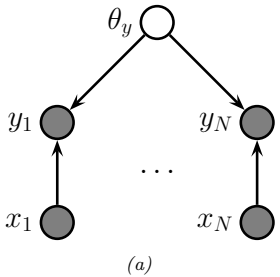
- A model that contains global latent variables $\boldsymbol{\theta}$ that are shared across all N observations: e.g. model parameters
- E.g. the usual supervised learning setting

$$p(\boldsymbol{y}_{1:N}, \boldsymbol{\theta} \mid \boldsymbol{x}_{1:N}) = p(\boldsymbol{\theta}) \prod_{n=1}^{N} p(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})$$

  where $\boldsymbol{y}_{1:N}$ is a shorthand for $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$
- In Bayesian setting, the goal is to obtain the posterior

$$p(\boldsymbol{\theta} \mid \boldsymbol{x}_{1:N}, \boldsymbol{y}_{1:N})$$

- In maximum likelihood setting, the goal is to maximize the likelihood $\prod_{n=1}^{N} p(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})$



(a)

Figure 7.1a from (Murphy, 2023)

Local latent variable models

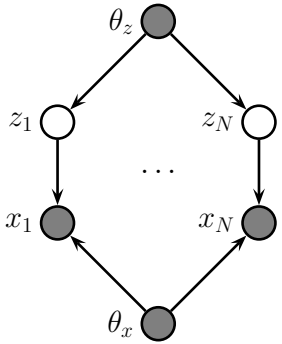
- A model that contains local latent variables $\boldsymbol{z}_n$ ($n \in \{1, \ldots, N\}$) that are associated with individual observations $\boldsymbol{x}_n$
- Model parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_z, \boldsymbol{\theta}_x)$ are assumed to be deterministic or known
- The joint distribution typically factorizes

$$p(\boldsymbol{x}_{1:N}, \boldsymbol{z}_{1:N} \mid \boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}_x) p(\boldsymbol{z}_n \mid \boldsymbol{\theta}_z)$$

- The goal is to obtain the posterior for each local latent

$$p(\boldsymbol{z}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})$$

- If parameters are not known, they can be treated as hyperparameters and optimized using e.g. the maximum likelihood



(b)

Figure 7.1b from (Murphy, 2023)

Global and local latent variable models

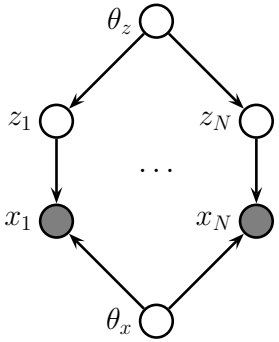
- A model that contains both
  - global latent variables $\boldsymbol{\theta} = (\boldsymbol{\theta}_z, \boldsymbol{\theta}_x)$
  - local latent variables $\boldsymbol{z}_n$ ($n \in \{1, \ldots, N\}$), that are associated with individual observations $\boldsymbol{x}_n$
- The joint distribution typically factorizes

$$p(\boldsymbol{x}_{1:N}, \boldsymbol{z}_{1:N}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}_z)p(\boldsymbol{\theta}_x) \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}_x)p(\boldsymbol{z}_n \mid \boldsymbol{\theta}_z)$$

- The goal is to obtain the posterior for unknowns

$$p(\boldsymbol{z}_{1:N}, \boldsymbol{\theta} \mid \boldsymbol{x}_{1:N})$$

- Benefits in considering $\boldsymbol{\theta}$ as a random variable?



(c)

Figure 7.1c from (Murphy, 2023)

Marginalization of latent variables

An indirect approach to define a probability distribution $p(\boldsymbol{x} \mid \boldsymbol{\theta})$ via a joint distribution $p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})$

Target distribution $p(\boldsymbol{x} \mid \boldsymbol{\theta})$ is obtained via marginalization

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \int p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}) d\boldsymbol{z} = \int p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\theta}_x) p(\boldsymbol{z} \mid \boldsymbol{\theta}_z) d\boldsymbol{z}$$

Marginalization of latent variables

An indirect approach to define a probability distribution $p(\boldsymbol{x} \mid \boldsymbol{\theta})$ via a joint distribution $p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})$

Target distribution $p(\boldsymbol{x} \mid \boldsymbol{\theta})$ is obtained via marginalization

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \int p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}) d\boldsymbol{z} = \int p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\theta}_x) p(\boldsymbol{z} \mid \boldsymbol{\theta}_z) d\boldsymbol{z}$$

- Example: 1-D mixture of three Gaussians with

$$p(z = k) = p_k, \ k \in \{1, 2, 3\}$$
$$p(x \mid z = k) = \mathcal{N}(x \mid \mu_k, \sigma_k^2)$$

and

$$p(x) = \sum_{k=1}^{3} p(x \mid z_k) p(z_k)$$
$$= \sum_{k=1}^{3} p_k \cdot p(x \mid \mu_k, \sigma_k^2)$$

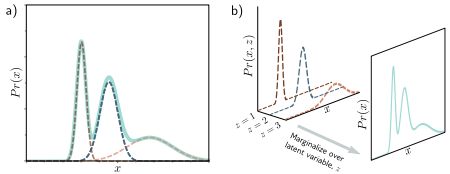


Figure 17.1 from (Prince, 2023)

Generation

A new data point $\boldsymbol{x}^*$ can be generated from a latent variable model using the ancestral sampling

$$\boldsymbol{z}^* \sim p(\boldsymbol{z} \mid \boldsymbol{\theta}_z)$$
$$\boldsymbol{x}^* \mid \boldsymbol{z}^* \sim p(\boldsymbol{x} \mid \boldsymbol{z}^*, \boldsymbol{\theta}_x)$$

This is straightforward as long as distributions $p(\boldsymbol{z} \mid \boldsymbol{\theta}_z)$ and $p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\theta}_x)$ are easy to sample

Probabilistic factor analysis

Probabilistic factor analysis (FA) model is a linear (local) latent variable model defined as

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \mathbf{0}, \mathbf{I})$$
$$p(\boldsymbol{x} \mid \boldsymbol{z}) = \mathcal{N}(\boldsymbol{x} \mid \mathbf{W}\boldsymbol{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}),$$

where

- $\boldsymbol{z} \in \mathbb{R}^L$ is a latent variable
- $\boldsymbol{x} \in \mathbb{R}^D$ denotes observed data
- $\mathbf{W} \in \mathbb{R}^{D \times L}$ is the factor loading matrix
- $\boldsymbol{\mu}$ is a mean offset
- $\boldsymbol{\Psi}$ is D-by-D covariance matrix
- parameters are deterministic (not r.v.s)

Probabilistic factor analysis

Probabilistic factor analysis (FA) model is a linear (local) latent variable model defined as

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \mathbf{0}, \mathbf{I})$$
$$p(\boldsymbol{x} \mid \boldsymbol{z}) = \mathcal{N}(\boldsymbol{x} \mid \mathbf{W}\boldsymbol{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}),$$

where

- $\boldsymbol{z} \in \mathbb{R}^L$ is a latent variable
- $\boldsymbol{x} \in \mathbb{R}^D$ denotes observed data
- $\mathbf{W} \in \mathbb{R}^{D \times L}$ is the factor loading matrix
- $\boldsymbol{\mu}$ is a mean offset
- $\boldsymbol{\Psi}$ is D-by-D covariance matrix
- parameters are deterministic (not r.v.s)

We can marginalize out the latent variable $\boldsymbol{z}$ from $p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})$ to get marginal likelihood

$$\begin{aligned} p(\boldsymbol{x}) &= \int p(\boldsymbol{x} \mid \boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z} \\ &= \int \mathcal{N}(\boldsymbol{x} \mid \mathbf{W}\boldsymbol{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}) \mathcal{N}(\boldsymbol{z} \mid \mathbf{0}, \mathbf{I}) d\boldsymbol{z} \\ &= \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi}) \end{aligned}$$

and

$$p(\boldsymbol{x}_{1:N}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi})$$

Probabilistic factor analysis: illustration

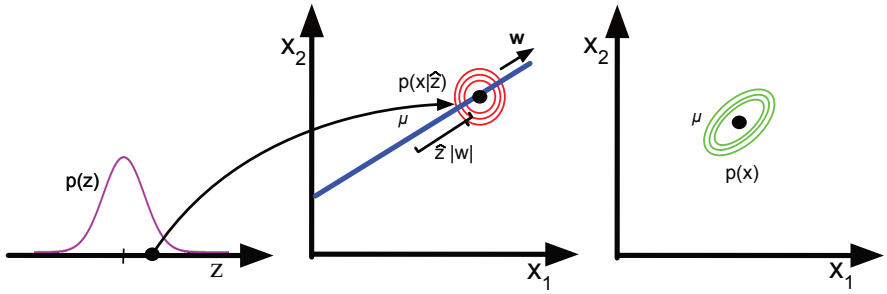


Figure 28.6: Illustration of the FA generative process, where we have $L = 1$ latent dimension generating $D = 2$ observed dimensions; we assume $\boldsymbol{\Psi} = \sigma^2\mathbf{I}$. The latent factor has value $z \in \mathbb{R}$, sampled from $p(z)$; this gets mapped to a 2d offset $\boldsymbol{\delta} = z\boldsymbol{w}$, where $\boldsymbol{w} \in \mathbb{R}^2$, which gets added to $\boldsymbol{\mu}$ to define a Gaussian $p(\boldsymbol{x}|z) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu} + \boldsymbol{\delta}, \sigma^2\mathbf{I})$. By integrating over z, we "slide" this circular Gaussian "spray can" along the principal component axis $\boldsymbol{w}$, which induces elliptical Gaussian contours in $\boldsymbol{x}$ space centered on $\boldsymbol{\mu}$. Adapted from Figure 12.9 of [Bis06].

Figure 28.3 from (Murphy, 2023)

Probabilistic principle component analysis

Probabilistic principle component analysis (PPCA) is a special case of the probabilistic FA model where

- columns of $\mathbf{W}$ are orthogonal
- $\mathbf{\Psi} = \sigma^2\mathbf{I}$,

and the generative model is

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

- The marginal likelihood

$$\begin{aligned} & p(\boldsymbol{x} \mid \mathbf{W}, \boldsymbol{\mu}, \sigma^2) \\ &= \int \mathcal{N}(\boldsymbol{x} \mid \mathbf{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})\mathcal{N}(\boldsymbol{z} \mid \mathbf{0}, \mathbf{I})d\boldsymbol{z} \\ &= \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}) \end{aligned}$$

Probabilistic principle component analysis: parameter learning

Given N observations $\mathcal{D} = \boldsymbol{x}_{1:N}$, we can estimate the model parameters by maximizing the marginal likelihood w.r.t. $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2

$$L(\mathbf{W}, \boldsymbol{\mu}, \sigma^2 \mid \mathcal{D}) = p(\boldsymbol{x}_{1:N} \mid \mathbf{W}, \boldsymbol{\mu}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I})$$

- Compute sample mean $\boldsymbol{\mu} = \bar{\boldsymbol{x}}$ and covariance matrix $\mathbf{S} = \frac{1}{N} \sum_{i=1}^{n} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$
- Rewrite $\mathbf{S}$ using the eigenvector-eigenvalue decomposition $\mathbf{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$
- ML parameters are obtained using the L largest eigenvalues-vectors

$$\mathbf{W} = \mathbf{U}_L (\boldsymbol{\Lambda}_L - \sigma^2 \mathbf{I})^{\frac{1}{2}} \text{ (upto arbitrary rotation)}$$

$$\sigma^2 = \frac{1}{D - L} \sum_{i=L+1}^{D} \lambda_i$$

$$\boldsymbol{\mu} = \bar{\boldsymbol{x}}$$

Probabilistic principle component analysis: posterior

- Given parameters $\mathbf{W}$, $\boldsymbol{\mu}$, σ^2 (fixed or optimized) and an observation $\boldsymbol{x}$, we want to know the distribution of the latent variable

$$p(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x})}$$

- The posterior of the latent variable can be shown to have normal distribution

$$p(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z} \mid \mathbf{M}^{-1}\mathbf{W}^\top(\boldsymbol{x} - \boldsymbol{\mu}), \sigma^2\mathbf{M}^{-1}),$$

where $\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}$

Probabilistic principle component analysis: posterior

- Given parameters $\mathbf{W}$, $\boldsymbol{\mu}$, σ^2 (fixed or optimized) and an observation $\boldsymbol{x}$, we want to know the distribution of the latent variable

$$p(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x})}$$

- The posterior of the latent variable can be shown to have normal distribution

$$p(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z} \mid \mathbf{M}^{-1}\mathbf{W}^\top(\boldsymbol{x} - \boldsymbol{\mu}), \sigma^2\mathbf{M}^{-1}),$$

  where $\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}$

- In the noise-free case of $\sigma^2 = 0$ the PPCA and PCA are directly comparable

Deep latent variable model (also called as non-linear latent variable models)

Deep latent variable model is a generative model

$$\boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$$
$$\boldsymbol{x} \mid \boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) \triangleq p(\boldsymbol{x} \mid d_{\boldsymbol{\theta}}(\boldsymbol{z})),$$

where

- $\boldsymbol{z} \in \mathcal{Z} = \mathbb{R}^L$ is a latent variable
- $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is a prior distribution, typically Gaussian
- $\boldsymbol{x} \in \mathcal{X}$ denotes observed data, e.g. $\mathcal{X} = \mathbb{R}^D$, where typically $D \gg L$
- $p(\boldsymbol{x} \mid \boldsymbol{\upsilon})$ is a distribution from the exponential family with parameters $\boldsymbol{\upsilon} \in \boldsymbol{\Upsilon}$ (e.g. $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ for Gaussian, or $\boldsymbol{p} = (p_1, \ldots, p_D)$ for Bernoulli)
- $d_{\boldsymbol{\theta}}(\cdot) : \mathcal{Z} \to \boldsymbol{\Upsilon}$ is a non-linear function called decoder that maps the latent variable to parameters of the observation likelihood (e.g. $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = d_{\boldsymbol{\theta}}(\boldsymbol{z})$)

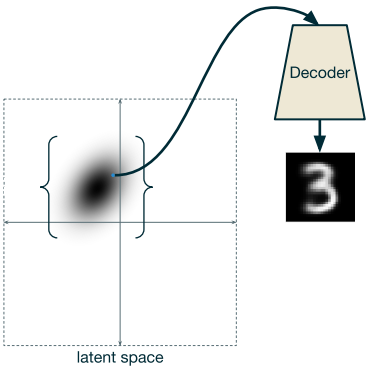


Figure adapted from https://ijdykeman.github.io/ml/2016/12/21/cvae.html

Deep latent variable model (also called as non-linear latent variable models)

Deep latent variable model is a generative model

$$\boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$$
$$\boldsymbol{x} \mid \boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) \triangleq p(\boldsymbol{x} \mid d_{\boldsymbol{\theta}}(\boldsymbol{z})),$$

where

- $\boldsymbol{z} \in \mathcal{Z} = \mathbb{R}^L$ is a latent variable
- $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is a prior distribution, typically Gaussian
- $\boldsymbol{x} \in \mathcal{X}$ denotes observed data, e.g. $\mathcal{X} = \mathbb{R}^D$, where typically $D \gg L$
- $p(\boldsymbol{x} \mid \boldsymbol{\upsilon})$ is a distribution from the exponential family with parameters $\boldsymbol{\upsilon} \in \boldsymbol{\Upsilon}$ (e.g. $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ for Gaussian, or $\boldsymbol{p} = (p_1, \ldots, p_D)$ for Bernoulli)
- $d_{\boldsymbol{\theta}}(\cdot) : \mathcal{Z} \to \boldsymbol{\Upsilon}$ is a non-linear function called decoder that maps the latent variable to parameters of the observation likelihood (e.g. $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = d_{\boldsymbol{\theta}}(\boldsymbol{z})$)

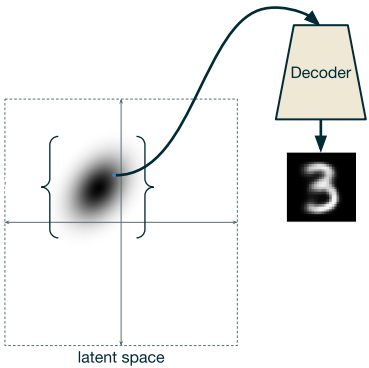


Figure adapted from https://ijdykeman.github.io/ml/2016/12/21/cvae.html

Marg. likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p(\boldsymbol{x} \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}))p_{\boldsymbol{\theta}}(\boldsymbol{z})d\boldsymbol{z}$ is generally intractable

Variational inference

Assume a probabilistic model with

- latent variables $\boldsymbol{z}$
- observed variables $\boldsymbol{x}$
- fixed parameters $\boldsymbol{\theta}$ (if random variables, add $\boldsymbol{\theta}$ to $\boldsymbol{z}$)

Interested in the posterior

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

where the evidence $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z}$ is intractable

Variational inference

Assume a probabilistic model with

- latent variables $\boldsymbol{z}$
- observed variables $\boldsymbol{x}$
- fixed parameters $\boldsymbol{\theta}$ (if random variables, add $\boldsymbol{\theta}$ to $\boldsymbol{z}$)

Interested in the posterior

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

where the evidence $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z}$ is intractable

Variational inference

Variational inference (VI) approximates $p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})$ with a variational approximation $q(\boldsymbol{z})$ by minimizing the reverse KL divergence over a variational family of distributions $\mathcal{Q}$

$$q^* = \arg\min_{q \in \mathcal{Q}} D_{\mathbb{KL}}(q(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})),$$

Variational inference

Assume a probabilistic model with

- latent variables $\boldsymbol{z}$
- observed variables $\boldsymbol{x}$
- fixed parameters $\boldsymbol{\theta}$ (if random variables, add $\boldsymbol{\theta}$ to $\boldsymbol{z}$)

Interested in the posterior

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

where the evidence $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z}$ is intractable

Variational inference

Variational inference (VI) approximates $p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})$ with a variational approximation $q(\boldsymbol{z})$ by minimizing the reverse KL divergence over a variational family of distributions $\mathcal{Q}$

$$q^* = \arg\min_{q \in \mathcal{Q}} D_{\mathbb{KL}}(q(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})),$$

In practice we use a parametric family $\mathcal{Q}$ with variational parameters $\boldsymbol{\psi}$, $q_{\boldsymbol{\psi}}(\boldsymbol{z})$

- e.g.: $q_{\boldsymbol{\psi}}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\boldsymbol{\psi} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Variational inference converts posterior inference to optimization

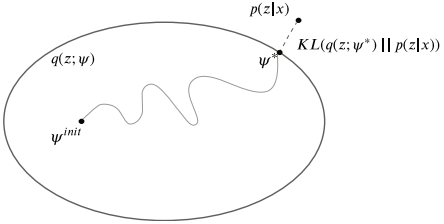


Figure 10.1 from (Murphy, 2023)

Variational inference: objective function

The variational objective can be written as

$$
\begin{aligned}
D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})) &= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log\left(\frac{q_{\boldsymbol{\psi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})}\right)\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log q_{\boldsymbol{\psi}}(\boldsymbol{z}) - \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})p_{\boldsymbol{\theta}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})}\right)\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log q_{\boldsymbol{\psi}}(\boldsymbol{z}) - \log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) - \log p_{\boldsymbol{\theta}}(\boldsymbol{z})\right] + \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \\
&= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log q_{\boldsymbol{\psi}}(\boldsymbol{z}) - \log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})\right] + \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \\
&= \underbrace{\mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log\left(\frac{q_{\boldsymbol{\psi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}\right)\right]}_{\tilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x})} + \log p_{\boldsymbol{\theta}}(\boldsymbol{x}),
\end{aligned}
$$

where $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is intrabtable but independent of $\boldsymbol{\psi}$

We can minimize $\tilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x})$ w.r.t. $\boldsymbol{\psi}$

Variational inference: evidence lower bound

Alternatively, we can derive a lower bound for the evidence (marg. likelihood)

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \log \left(\int p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} \right) \\
&= \log \left(\int q_{\boldsymbol{\psi}}(\boldsymbol{z}) \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})} d\boldsymbol{z} \right) \\
&= \log \left(\mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})} \left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})} \right] \right) \\
&\overset{\text{Jensen}}{\geq} \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})} \left[\log \left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})} \right) \right] \\
&= \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x})
\end{aligned}$$

$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x})$ is called the evidence lower bound (ELBO): optimize w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$

Variational inference: evidence lower bound

Alternatively, we can derive a lower bound for the evidence (marg. likelihood)

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \log \left(\int p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} \right) \\
&= \log \left(\int q_{\boldsymbol{\psi}}(\boldsymbol{z}) \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})} d\boldsymbol{z} \right) \\
&= \log \left(\mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})} \left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})} \right] \right) \\
&\overset{\text{Jensen}}{\geq} \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})} \left[\log \left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})} \right) \right] \\
&= \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x})
\end{aligned}$$

$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x})$ is called the evidence lower bound (ELBO): optimize w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$

Notice that $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x}) = -\tilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x})$

$\rightarrow$ Maximizing $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x})$ equals minimizing $\tilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x})$

$\rightarrow$ Minimizing $D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x}))$ equals maximizing a lower bound for $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$

ELBO as reconstruction loss minus KL divergence

We can also write the ELBO as

$$
\begin{aligned}
\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &\geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x}) \\
&= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\right)\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) + \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\right)\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\right] - \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log\left(\frac{q_{\boldsymbol{\psi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})}\right)\right] \\
&= \underbrace{\mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\right]}_{\text{expected log likelihood}} - \underbrace{D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z}))}_{\text{KL of the posterior from prior}}
\end{aligned}
$$

The first term measures the reconstruction loss

The second term can be interpreted as prior regularization

ELBO as reconstruction loss minus KL divergence

We can also write the ELBO as

$$
\begin{aligned}
\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &\geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x}) \\
&= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\right)\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) + \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\right)\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\right] - \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log\left(\frac{q_{\boldsymbol{\psi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})}\right)\right] \\
&= \underbrace{\mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\right]}_{\text{expected log likelihood}} - \underbrace{D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z}))}_{\text{KL of the posterior from prior}}
\end{aligned}
$$

The first term measures the reconstruction loss

The second term can be interpreted as prior regularization

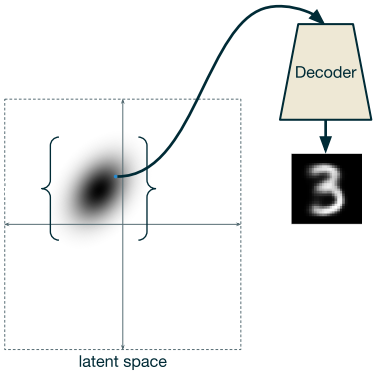


Figure adapted from

`https://ijdykeman.github.io/ml/2016/12/21/cvae.html`

ELBO: Monte Carlo estimate

ELBO involves expectations that generally cannot be computed analytically but can be estimated using Monte Carlo

For the standard ELBO formula

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &\geq \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\right)\right] \\ &\approx \frac{1}{S}\left(\sum_{s=1}^{S}\log p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}_s) - \log q_{\boldsymbol{\psi}}(\boldsymbol{z}_s)\right) \\ &= \frac{1}{S}\left(\sum_{s=1}^{S}\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}_s) + \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_s)\right. \\ &\quad \left. - \log q_{\boldsymbol{\psi}}(\boldsymbol{z}_s)\right)\end{aligned}$$

where $\boldsymbol{z}_s \overset{\text{i.i.d}}{\sim} q_{\boldsymbol{\psi}}(\boldsymbol{z})$

ELBO: Monte Carlo estimate

ELBO involves expectations that generally cannot be computed analytically but can be estimated using Monte Carlo

For the standard ELBO formula

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &\geq \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\right)\right] \\ &\approx \frac{1}{S}\left(\sum_{s=1}^{S}\log p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}_s) - \log q_{\boldsymbol{\psi}}(\boldsymbol{z}_s)\right) \\ &= \frac{1}{S}\left(\sum_{s=1}^{S}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z}_s) + \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_s) \right. \\ &\qquad \left. - \log q_{\boldsymbol{\psi}}(\boldsymbol{z}_s)\right)\end{aligned}$$

where $\boldsymbol{z}_s \overset{\text{i.i.d}}{\sim} q_{\boldsymbol{\psi}}(\boldsymbol{z})$

For the reconstruction minus KL formula

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &\geq \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z})\right] - D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\ &\approx \frac{1}{S}\sum_{s=1}^{S}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z}_s) - D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z}))\end{aligned}$$

where $\boldsymbol{z}_s \overset{\text{i.i.d}}{\sim} q_{\boldsymbol{\psi}}(\boldsymbol{z})$

KL term has a closed-form solution for a pair of distributions from the same exponential family

Tightness of the ELBO

From the variational bound

$$D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})) = \tilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

we get

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= -\tilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x}) + D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})) \\ &= \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x}) + D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})) \\ &\geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x}),\end{aligned}$$

because KL divergence is non-negative

Tightness of the ELBO

From the variational bound

$$D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})) = \tilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

we get

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= -\tilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x}) + D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})) \\ &= \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x}) + D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})) \\ &\geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{x}),\end{aligned}$$

because KL divergence is non-negative

KL divergence $D_{\mathbb{KL}}(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x}))$ defines us two "distances"

1. The KL divergence of the variational approximation from the true posterior (i.e., the VI objective)
2. The tightness of the bound, i.e., the gap between the ELBO and $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$

Variational inference: mean field model

For a standard local (deep) latent variable model with N observations $\mathcal{D} = \boldsymbol{x}_{1:N}$ and deterministic parameters $\boldsymbol{\theta}$

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:N}, \boldsymbol{z}_{1:N}) &= p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:N} \mid \boldsymbol{z}_{1:N}) p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:N}) \\ &= \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}_n)) p_{\boldsymbol{\theta}}(\boldsymbol{z}_n) \end{aligned}$$

The posterior factorize accordingly

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:N} \mid \boldsymbol{x}_{1:N}) &\propto p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:N} \mid \boldsymbol{z}_{1:N}) p(\boldsymbol{z}_{1:N}) \\ &= \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}_n)) p_{\boldsymbol{\theta}}(\boldsymbol{z}_n) \end{aligned}$$

Variational inference: mean field model

For a standard local (deep) latent variable model with N observations $\mathcal{D} = \boldsymbol{x}_{1:N}$ and deterministic parameters $\boldsymbol{\theta}$

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:N}, \boldsymbol{z}_{1:N}) &= p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:N} \mid \boldsymbol{z}_{1:N}) p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:N}) \\ &= \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}_n)) p_{\boldsymbol{\theta}}(\boldsymbol{z}_n) \end{aligned}$$

The posterior factorize accordingly

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:N} \mid \boldsymbol{x}_{1:N}) &\propto p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:N} \mid \boldsymbol{z}_{1:N}) p(\boldsymbol{z}_{1:N}) \\ &= \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}_n)) p_{\boldsymbol{\theta}}(\boldsymbol{z}_n) \end{aligned}$$

It is common to use the so-called mean field variational approximation that factorizes accordingly

$$q(\boldsymbol{z}_{1:N} \mid \boldsymbol{\psi}_{1:N}) = \prod_{n=1}^{N} q_{\boldsymbol{\psi}_n}(\boldsymbol{z}_n)$$

Note: each latent variable $\boldsymbol{z}_n$ has its own variational parameters $\boldsymbol{\psi}_n$

Variational inference: mean field model

For a standard local (deep) latent variable model with N observations $\mathcal{D} = \boldsymbol{x}_{1:N}$ and deterministic parameters $\boldsymbol{\theta}$

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:N}, \boldsymbol{z}_{1:N}) &= p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:N} \mid \boldsymbol{z}_{1:N}) p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:N}) \\ &= \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}_n)) p_{\boldsymbol{\theta}}(\boldsymbol{z}_n) \end{aligned}$$

The posterior factorize accordingly

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:N} \mid \boldsymbol{x}_{1:N}) &\propto p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:N} \mid \boldsymbol{z}_{1:N}) p(\boldsymbol{z}_{1:N}) \\ &= \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}_n)) p_{\boldsymbol{\theta}}(\boldsymbol{z}_n) \end{aligned}$$

It is common to use the so-called mean field variational approximation that factorizes accordingly

$$q(\boldsymbol{z}_{1:N} \mid \boldsymbol{\psi}_{1:N}) = \prod_{n=1}^{N} q_{\boldsymbol{\psi}_n}(\boldsymbol{z}_n)$$

Note: each latent variable $\boldsymbol{z}_n$ has its own variational parameters $\boldsymbol{\psi}_n$

Because

- the above mean field approximation factorizes
- log likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:n} \mid \boldsymbol{z}_{1:N})$ factorizes
- KL for independent r.v.s. factorizes

the ELBO for $\log p_{\boldsymbol{\theta}}(\mathcal{D})$ factorizes as well

$$\log p_{\boldsymbol{\theta}}(\mathcal{D}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}_{1:N} \mid \mathcal{D}) = \sum_{n=1}^{N} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}_n \mid \boldsymbol{x}_n)$$

Variational inference and parameter estimation

- If generative model parameter $\boldsymbol{\theta}$ are unknown, we generally optimize them by maximizing the (intractable) log marginal likelihood $\log p_{\boldsymbol{\theta}}(\mathcal{D})$
- Again, the ELBO provides a lower bound: $\log p_{\boldsymbol{\theta}}(\mathcal{D}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}_{1:N} \mid \mathcal{D})$
- Variational EM algorithm iteratively optimizes $\boldsymbol{\psi}_{1:N}$ and $\boldsymbol{\theta}$
  - E step: maximize $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}_{1:N} \mid \mathcal{D})$ w.r.t. $\boldsymbol{\psi}_{1:N}$
  - M step: maximize $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}_{1:N} \mid \mathcal{D})$ w.r.t. $\boldsymbol{\theta}$
- Similar approach, called variational Bayes, if generative model parameters $\boldsymbol{\theta}$ are random variables $p(\boldsymbol{\theta} \mid \boldsymbol{\xi})$

Variational inference: stochastic VI

- Computing and minimizing the ELBO for large datasets (e.g. millions of data points) can be slow
- An unbiased stochastic approximation with a minibatch $\mathcal{B} \subseteq \mathcal{D}$ of size $B = |\mathcal{B}|$

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}_{1:N} \mid \mathcal{D}) &= \sum_{n=1}^{N} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}_n \mid \boldsymbol{x}_n) \\ &\approx \frac{N}{B} \sum_{\boldsymbol{x}_n \in \mathcal{B}} \left[\mathbb{E}_{q_{\boldsymbol{\psi}_n}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_n \mid \boldsymbol{z}_n) \right] - D_{\mathbb{KL}}(q_{\boldsymbol{\psi}_n}(\boldsymbol{z}_n) \mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{z}_n)) \right]\end{aligned}$$

- This corresponds to stochastic gradient descent (SGD) optimization algorithm and scales VI to large datasets

Variational inference: amortized VI

- For large datasets, optimizing N local variational parameters $\boldsymbol{\psi}_{1:N}$ can be slow
- Amortized VI trains a separate deterministic mapping, so-called encoder or inference network, that predicts $\boldsymbol{\psi}_n$ from $\boldsymbol{x}_n$

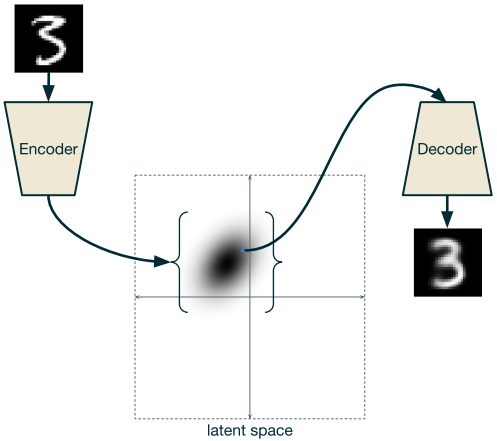
$$\boldsymbol{\psi}_n = f_{\boldsymbol{\phi}}(\boldsymbol{x}_n)$$

- Parameters $\boldsymbol{\phi}$ are shared across samples N
- Example: assume

$$q_{\boldsymbol{\psi}_n}(\boldsymbol{z}_n) = \mathcal{N}(\boldsymbol{z}_n \mid \boldsymbol{\mu}_n, \mathrm{diag}(\boldsymbol{\sigma}_n^2))$$
$$\boldsymbol{\psi}_n = (\boldsymbol{\mu}_n, \log \boldsymbol{\sigma}_n^2)$$

then

$$\boldsymbol{\psi}_n = (\boldsymbol{\mu}_n, \log \boldsymbol{\sigma}^2) = f_{\boldsymbol{\phi}}(\boldsymbol{x}_n)$$



https://ijdykeman.github.io/ml/2016/12/21/cvae.html

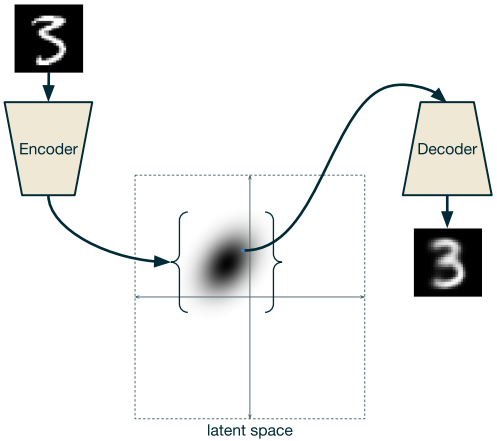
Variational inference: amortized VI

- The amortized version of the standard variational approximation $q_{\psi_n}(\boldsymbol{z}_n)$ is denoted as

$$q(\boldsymbol{z}_n \mid f_{\boldsymbol{\phi}}(\boldsymbol{x}_n)) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_n \mid \boldsymbol{x}_n)$$

- The ELBO objective becomes

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathcal{D}) = \sum_{n=1}^{N} \left[\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_n \mid \boldsymbol{x}_n)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_n \mid \boldsymbol{z}_n) \right] - D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}_n \mid \boldsymbol{x}_n) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z}_n)) \right]$$



https://ijdykeman.github.io/ml/2016/12/21/cvae.html

Variational inference: gradient-based VI

- The gradient of the ELBO w.r.t. generative model parameters

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}\mid\boldsymbol{x}) &= \nabla_{\boldsymbol{\theta}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z})\right] - \nabla_{\boldsymbol{\theta}}D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})\mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z})\right] - \nabla_{\boldsymbol{\theta}}D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})\mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\ &\overset{\text{MC}}{\approx} \sum_{s=1}^{S}\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z}_s) - \nabla_{\boldsymbol{\theta}}D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})\mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{z}))\end{aligned}$$

where $\boldsymbol{z}_s \overset{\text{i.i.d.}}{\sim} q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})$ and assuming the KL divergence has a closed-form solution (computation does not involve any random variables)

Variational inference: gradient-based VI

- The gradient of the ELBO w.r.t. generative model parameters

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}\mid\boldsymbol{x}) &= \nabla_{\boldsymbol{\theta}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z})\right] - \nabla_{\boldsymbol{\theta}}D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})\mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})}\left[\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z})\right] - \nabla_{\boldsymbol{\theta}}D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})\mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\ &\overset{\text{MC}}{\approx} \sum_{s=1}^{S}\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z}_s) - \nabla_{\boldsymbol{\theta}}D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})\mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{z}))\end{aligned}$$

  where $\boldsymbol{z}_s \overset{\text{i.i.d.}}{\sim} q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})$ and assuming the KL divergence has a closed-form solution (computation does not involve any random variables)
- The gradient of the ELBO w.r.t. the inference network parameters $\boldsymbol{\phi}$ is trickier

$$\begin{aligned}\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}\mid\boldsymbol{x}) &= \nabla_{\boldsymbol{\phi}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z})\right] - \nabla_{\boldsymbol{\phi}}D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})\mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\ &\neq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})}\left[\nabla_{\boldsymbol{\phi}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z})\right] - \nabla_{\boldsymbol{\phi}}D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\mid\boldsymbol{x})\mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{z}))\end{aligned}$$

Reparametrization trick

The reparametrization trick is to rewrite a random variable $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$ using a differentiable and invertible transformation g of another random variable $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ that does not depend on $\boldsymbol{\phi}$, i.e.,

$$\begin{aligned} \boldsymbol{\epsilon} &\sim p(\boldsymbol{\epsilon}) \\ \boldsymbol{z} &= g(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{z}) \end{aligned}$$

An example: if $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$ where $(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2) = f_{\boldsymbol{\phi}}(\boldsymbol{x})$, we can use

$$\begin{aligned} \boldsymbol{\epsilon} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \boldsymbol{z} &= \boldsymbol{\mu} + \mathrm{diag}(\boldsymbol{\sigma})\boldsymbol{\epsilon} \end{aligned}$$

Reparametrization trick

The reparametrization trick is to rewrite a random variable $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$ using a differentiable and invertible transformation g of another random variable $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ that does not depend on $\boldsymbol{\phi}$, i.e.,

$$\begin{aligned} \boldsymbol{\epsilon} &\sim p(\boldsymbol{\epsilon}) \\ \boldsymbol{z} &= g(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{z}) \end{aligned}$$

An example: if $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$ where $(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2) = f_{\boldsymbol{\phi}}(\boldsymbol{x})$, we can use

$$\begin{aligned} \boldsymbol{\epsilon} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \boldsymbol{z} &= \boldsymbol{\mu} + \mathrm{diag}(\boldsymbol{\sigma})\boldsymbol{\epsilon} \end{aligned}$$

Expectation of a function $h(\boldsymbol{z})$ w.r.t. $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$ can be written as

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[h(\boldsymbol{z})\right] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[h(\boldsymbol{z}')\right] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[h(g(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon}))\right]$$

where $\boldsymbol{z}' = g(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon})$, and gradient accordingly

$$\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[h(\boldsymbol{z})\right] = \nabla_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[h(\boldsymbol{z}')\right] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\phi}} h(\boldsymbol{z}')\right]$$

Reparametrization: illustration

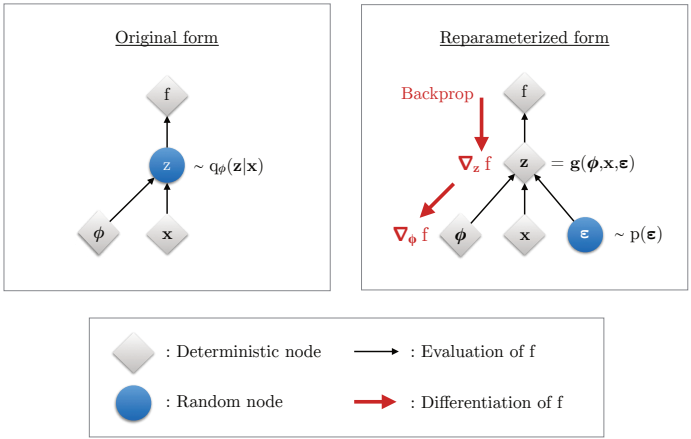


Figure 10.4 from (Murphy, 2023), originally from (Kingma & Welling, 2019)

Variational inference: reparameterized VI

Using reparametrization, the gradient of the ELBO w.r.t. the inference network parameters $\boldsymbol{\phi}$ is

$$\begin{aligned}
\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) &= \nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) \right] - \nabla_{\boldsymbol{\phi}} D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\
&\overset{\text{reparam.}}{=} \nabla_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}') \right] - \nabla_{\boldsymbol{\phi}} D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\
&= \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\nabla_{\boldsymbol{\phi}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}') \right] - \nabla_{\boldsymbol{\phi}} D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\
&\overset{\text{MC}}{\approx} \sum_{s=1}^{S} \nabla_{\boldsymbol{\phi}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}'_s) - \nabla_{\boldsymbol{\phi}} D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z}))
\end{aligned}$$

where $\boldsymbol{z}' = g(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon})$, $\boldsymbol{z}'_s = g(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon}_s)$, and $\boldsymbol{\epsilon}_s \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{\epsilon})$

Variational inference: reparameterized VI

Using reparametrization, the gradient of the ELBO w.r.t. the inference network parameters $\boldsymbol{\phi}$ is

$$\begin{aligned}
\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) &= \nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) \right] - \nabla_{\boldsymbol{\phi}} D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\
&\overset{\text{reparam.}}{=} \nabla_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}') \right] - \nabla_{\boldsymbol{\phi}} D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\
&= \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\nabla_{\boldsymbol{\phi}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}') \right] - \nabla_{\boldsymbol{\phi}} D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z})) \\
&\overset{\text{MC}}{\approx} \sum_{s=1}^{S} \nabla_{\boldsymbol{\phi}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}'_s) - \nabla_{\boldsymbol{\phi}} D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z}))
\end{aligned}$$

where $\boldsymbol{z}' = g(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon})$, $\boldsymbol{z}'_s = g(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon}_s)$, and $\boldsymbol{\epsilon}_s \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{\epsilon})$

Reparameterized gradient is an unbiased estimate of the exact gradient

If the KL does not have a closed form solution, reparameterized gradients may be computed from

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) &= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \right] \\
\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) &= \nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \right]
\end{aligned}$$

Variational autoencoder

Variational autoencoder (VAE) is a neural architecture that consists of a deep latent variable model (deep generative model) of the form

$$\boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$$
$$\boldsymbol{x} \mid \boldsymbol{z} \sim p(\boldsymbol{x} \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}))$$

that is trained using (reparameterized) amortized variational inference with an inference model $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$

- Data $\boldsymbol{x}$ can be continuous or discrete, or both, typically modeled using distribution from the exponential family
- Decoder $d_{\boldsymbol{\theta}}(\cdot)$ and encoder (inference network) $f_{\boldsymbol{\phi}}(\cdot)$ are commonly neural networks
- For general statistical latent variable models, this can be called as autoencoding variational Bayes method
- Variational approximation $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$ needs to admit reparametrization

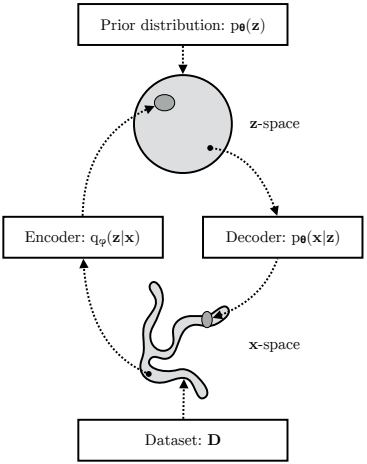
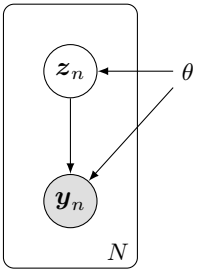


Figure from (Kingma & Welling, 2019)

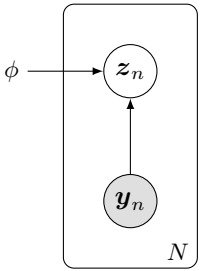
Variational autoencoder: Probabilistic graphical model

Probabilistic graphical models for VAE

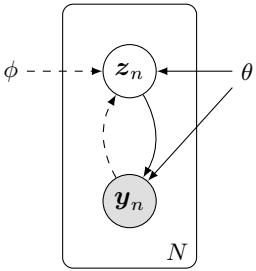
- Circled symbols denote random variables
- Non-circled symbols are hyperparameters
- White nodes (circles) are latent
- Shaded nodes are observed



(a) Generative model



(b) Inference model



(c) Combined

Learning algorithm for VAE

Algorithm for obtaining maximum likelihood estimate for deep latent variable model using reparameterized, stochastic and amortized variational inference, here with

- a minibatch sample size of one
- Monte Carlo estimate with sample size one

Algorithm 10.3: Reparameterized amortized SVI for MLE of an LVM

Initialize $\boldsymbol{\theta}$, $\boldsymbol{\phi}$
repeat
  Sample $\boldsymbol{x}_n \sim p_{\mathcal{D}}$
  Sample $\boldsymbol{\epsilon}_n \sim q_0$
  Compute $\boldsymbol{z}_n = g(\boldsymbol{\phi}, \boldsymbol{x}_n, \boldsymbol{\epsilon}_n)$
  Compute $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{x}_n, \boldsymbol{z}_n) = -\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_n, \boldsymbol{z}_n) + \log q_{\boldsymbol{\phi}}(\boldsymbol{z}_n|\boldsymbol{x}_n)$
  Update $\boldsymbol{\theta} := \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}|\boldsymbol{x}_n, \boldsymbol{z}_n)$
  Update $\boldsymbol{\phi} := \boldsymbol{\phi} - \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}|\boldsymbol{x}_n, \boldsymbol{z}_n)$
until converged

Algorithm 10.3 from (Murphy, 2023)
Note notation $\tilde{\mathcal{L}}$ vs. $\mathcal{L}$

Variational autoencoder: neural architecture training pipeline

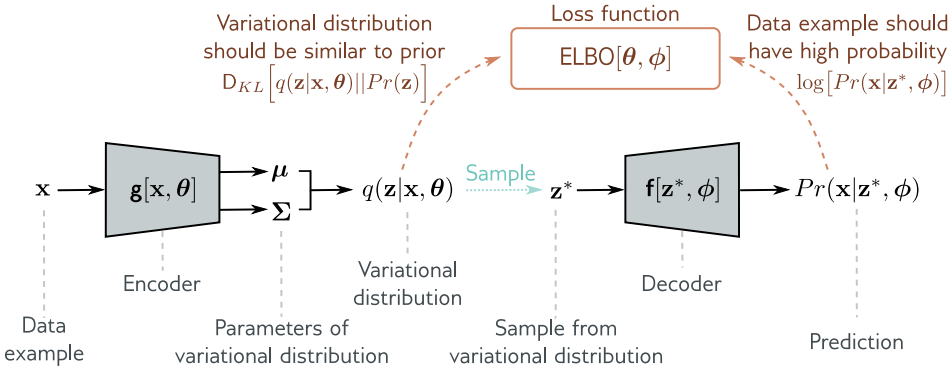


Figure 17.9 from (Prince, 2023). Note that notation differs.

Variational autoencoder: parametrization, NN architectures

In standard VAE models:

- Prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is typically Gaussian
- Likelihood $p(\boldsymbol{x} \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}))$ depends on application: typically Gaussian, Bernoulli, etc.
- Variational approximation $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$ is typically Gaussian (but can be more expressive, such as normalizing flows etc.)

Decoder $d_{\boldsymbol{\theta}}(\cdot)$ and encoder $f_{\boldsymbol{\phi}}(\cdot)$ neural network architectures are application dependent:

- MLPs for tabular data
- CNNs for images
- RNNs and attention for sequential data

Autoencoder vs. variational autoencoder

Autoencoder (AE) is a deterministic neural network architecture that

- Maps input $\boldsymbol{x}$ into a lower dimensional representation $\boldsymbol{z} = f_{\mathsf{e}}(\boldsymbol{x})$
- Attempts to reconstruct the original input by $f_{\mathsf{d}}(\boldsymbol{z})$, i.e., $x \approx \hat{\boldsymbol{x}} = f_{\mathsf{d}}(f_{\mathsf{e}}(\boldsymbol{x}))$

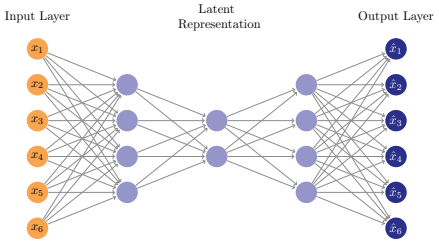


Figure 16.13 from (Murphy, 2023)

Autoencoder vs. variational autoencoder

Autoencoder (AE) is a deterministic neural network architecture that

- Maps input $\boldsymbol{x}$ into a lower dimensional representation $\boldsymbol{z} = f_{\mathsf{e}}(\boldsymbol{x})$
- Attempts to reconstruct the original input by $f_{\mathsf{d}}(\boldsymbol{z})$, i.e., $x \approx \hat{\boldsymbol{x}} = f_{\mathsf{d}}(f_{\mathsf{e}}(\boldsymbol{x}))$

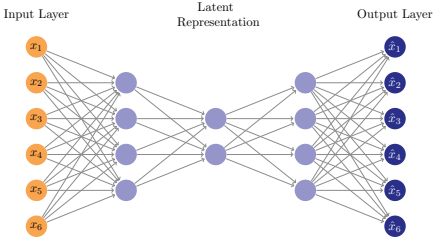
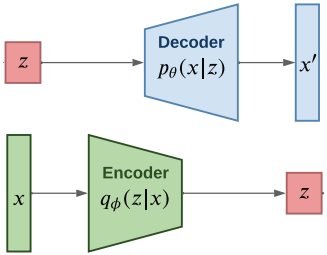


Figure 16.13 from (Murphy, 2023)

Variational autoencoder (VAE) is a neural architecture that consists of a deep generative model that is trained using an inference model



Autoencoder vs. variational autoencoder

Autoencoder (AE) is a deterministic neural network architecture that

- Maps input $\boldsymbol{x}$ into a lower dimensional representation $\boldsymbol{z} = f_{\mathsf{e}}(\boldsymbol{x})$
- Attempts to reconstruct the original input by $f_{\mathsf{d}}(\boldsymbol{z})$, i.e., $x \approx \hat{\boldsymbol{x}} = f_{\mathsf{d}}(f_{\mathsf{e}}(\boldsymbol{x}))$

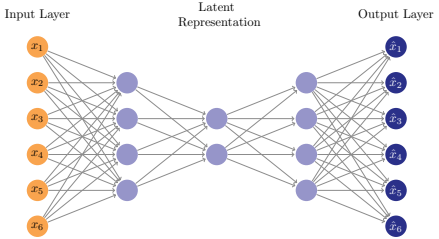
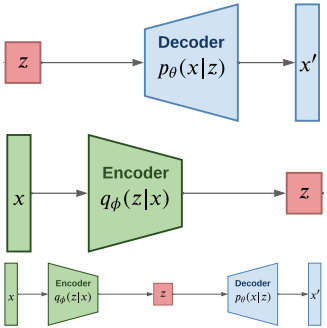


Figure 16.13 from (Murphy, 2023)

Variational autoencoder (VAE) is a neural architecture that consists of a deep generative model that is trained using an inference model



Adapted from Figure 20.1 from (Murphy, 2023)

Applications of variational autoencoders

Variational autoencoders can be used for various tasks

- Statistical modeling[a]
- Generation[a], conditional generation[b]
- Amortized variational inference[a]
- Representation learning[a]
- Reconstruction, missing value estimation[b]
- Latent space interpolation[b]
- Out-of-distribution detection[b]

[a] In this lecture.
[b] In the next lecture.

Variational autoencoder: MNIST image example

MNIST is a large image dataset $\mathcal{D} = \boldsymbol{x}_{1:N}$ of handwritten digits

https://riccardo-cantini.netlify.app/post/cnn_vae_mnist/

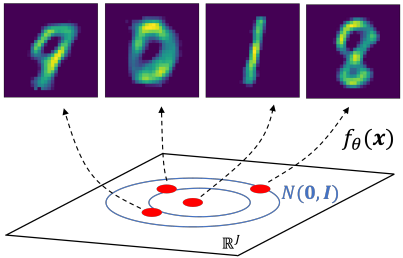
Variational autoencoder: MNIST image example

MNIST is a large image dataset $\mathcal{D} = \boldsymbol{x}_{1:N}$ of handwritten digits

https://riccardo-cantini.netlify.app/post/cnn_vae_mnist/

Train a VAE model and sample from the trained generative model to generate new images

$$\boldsymbol{z}^* \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$$
$$\boldsymbol{x}^* \mid \boldsymbol{z}^* \sim p(\boldsymbol{x} \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}^*))$$



https://mbernste.github.io/posts/vae/

Sample generation from VAE

As before, a new data point $\boldsymbol{x}^*$ can be generated from a latent variable model using the ancestral sampling

$$\boldsymbol{z}^* \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$$
$$\boldsymbol{x}^* \sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}^*)$$

For example in image tasks, so-called decoded mean samples can have higher quality

$$\boldsymbol{z}^* \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$$
$$\hat{\boldsymbol{x}}^* = d_{\boldsymbol{\theta}}(\boldsymbol{z}^*)$$

Sample generation from VAE

As before, a new data point $\boldsymbol{x}^*$ can be generated from a latent variable model using the ancestral sampling

$$\boldsymbol{z}^* \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$$
$$\boldsymbol{x}^* \sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}^*)$$

For example in image tasks, so-called decoded mean samples can have higher quality

$$\boldsymbol{z}^* \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$$
$$\hat{\boldsymbol{x}}^* = d_{\boldsymbol{\theta}}(\boldsymbol{z}^*)$$

Another trick is to define a joint inference distribution

$$q_{\mathcal{D},\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) = p_{\mathcal{D}}(\boldsymbol{x}) q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}),$$

where

$$p_{\mathcal{D}}(\boldsymbol{x}) = \frac{1}{N} \sum_{n=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_n),$$

and then sample latent variables from the aggregated posterior (marginal inference distribution) $q_{\mathcal{D},\boldsymbol{\phi}}(\boldsymbol{z})$

$$\begin{aligned} q_{\mathcal{D},\boldsymbol{\phi}}(\boldsymbol{z}) &= \int q_{\mathcal{D},\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{x} \\ &= \int \frac{1}{N} \sum_{n=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_n) q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) d\boldsymbol{x} \\ &= \frac{1}{N} \sum_{n=1}^{N} q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}_n) \end{aligned}$$

Variational autoencoder: CELEBA image example

Samples from VAEs that are trained on CELEBA dataset

a) Random samples:

$$\boldsymbol{z}^* \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$$
$$\boldsymbol{x}^* \mid \boldsymbol{z}^* \sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}^*))$$

b) Decoded mean samples:

$$\boldsymbol{z}^* \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$$
$$\hat{\boldsymbol{x}}^* \mid \boldsymbol{z}^* = d_{\boldsymbol{\theta}}(\boldsymbol{z}^*)$$

c) "Noise": $\hat{\boldsymbol{\epsilon}} = \boldsymbol{x}^* - \hat{\boldsymbol{x}}^*$

d) High-quality generation from a VAE with hierarchical prior and specialized architectures (details later)

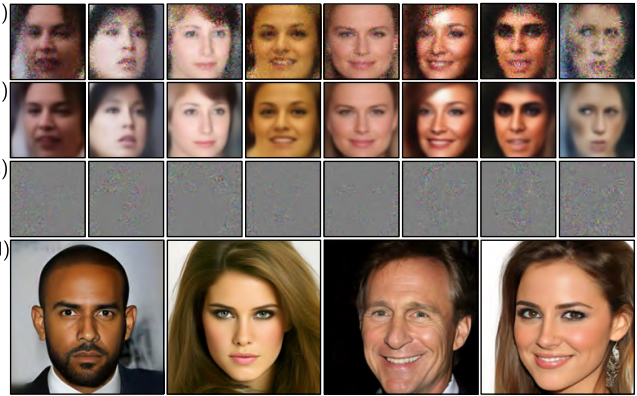


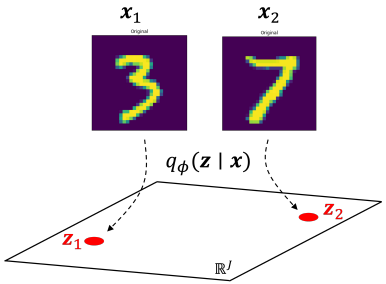
Figure 17.12 from (Prince, 2023).

Variational autoencoder: dimension reduction, visualization, clustering

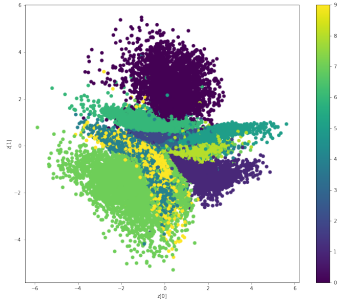
Example: Train a VAE model on the MNIST images

Use the encoder (amortized VI) for dimension reduction and visualization by plotting the approximative posteriors $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ (or their means)

It is common to use the learned representations $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ for clustering or other downstream tasks



https://mbernste.github.io/posts/vae/



https://riccardo-cantini.netlify.app/post/cnn_vae_mnist/

Variational autoencoder: modern molecular biology example

Single-cell sequencing is a modern technology that allows quantifying transcriptomic profiles of individual cells

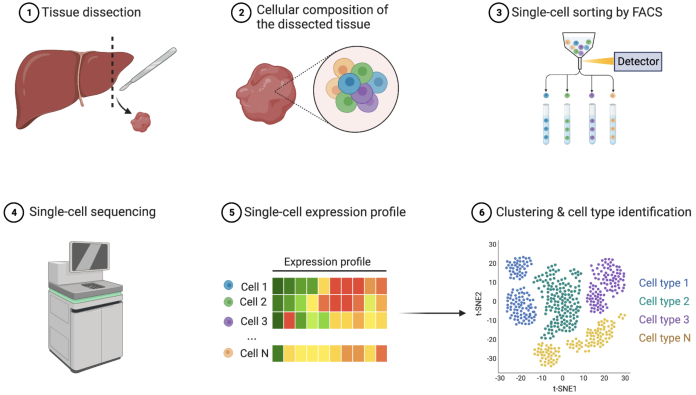


Figure from bioRender (https://www.biorender.com/template/single-cell-sequencing)

Variational autoencoder: modern molecular biology example

Challenging to specify a well-motivated probabilistic model for the data

$\rightarrow$ Incorporate biological inductive bias into neural network parameterized generative model

$$
\begin{aligned}
z_n &\sim \text{Normal}(0, I) \\
\ell_n &\sim \text{LogNormal}(\ell_\mu, \ell_\sigma^2) \\
\rho_n &= f_w(z_n, s_n) \\
w_{ng} &\sim \text{Gamma}(\rho_n^g, \theta) \\
y_{ng} &\sim \text{Poisson}(\ell_n w_{ng}) \\
h_{ng} &\sim \text{Bernoulli}(f_h^g(z_n, s_n)) \\
x_{ng} &= \begin{cases} y_{ng} & \text{if } h_{ng} = 0, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
$$

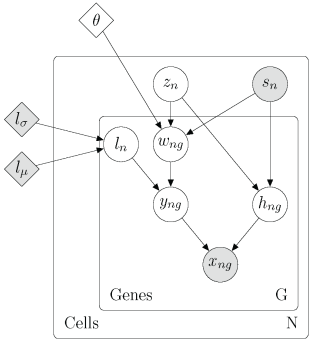


Figure from (Lopez et al, 2019)

Variational autoencoder: modern molecular biology example

Variational autoencoder architecture (generative model and amortized VI) with deep neural networks

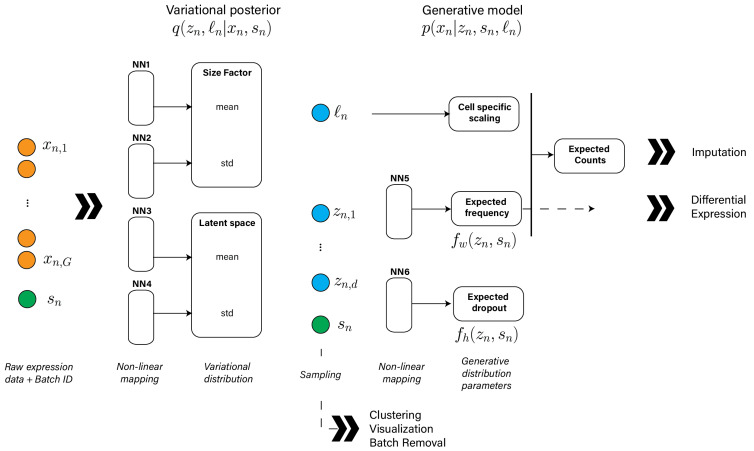


Figure from (Lopez et al, 2019)

Variational autoencoder: modern molecular biology example

Encoder (amortized VI) provides representations for visualization, clustering, and other downstream tasks

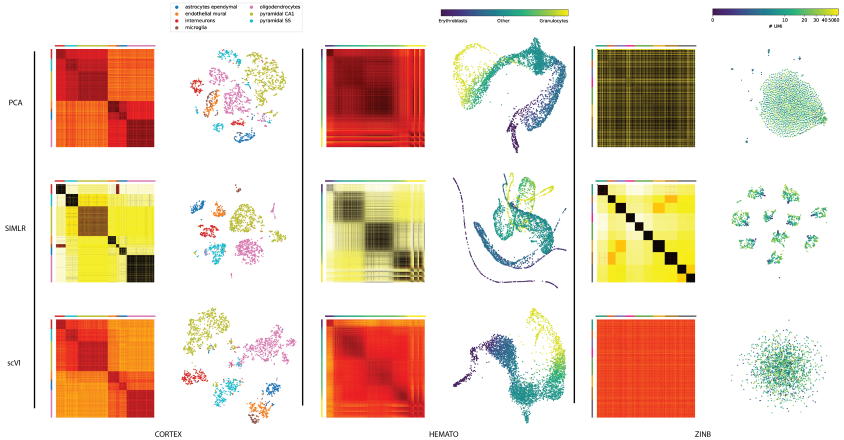


Figure from (Lopez et al, 2019)

References

- Kingma DP and Welling M, An Introduction to Variational Autoencoders, Foundations and Trends in Machine Learning, Vol. 12, No. 4, pp. 307-392.
- R. Lopez, J. Regier, MB. Cole, M. Jordan, N. Yosef, Deep generative modeling for single-cell transcriptomics, *Nature Methods*, 2019.
- Murphy K, Probabilistic Machine Learning: Advanced Topics, The MIT Press, 2023.
- Prince SJD, Understanding Deep Learning, The MIT Press, 2023.