CS-E4891 Deep Generative Models
Lecture 3: Variational autoencoders

Harri Lähdesmäki

Department of Computer Science
Aalto University

March 31, 2025

Outline

- Variational autoencoders (recap)
- Applications of VAE (continued)
- VAE generalizations
- Identifiability of VAEs
- Amortization gap

- Reading: parts of Sec. 20-21 from (Murphy, 2023), Sec. 17 from (Prince, 2023), and other literature (see references at the end)

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

- 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

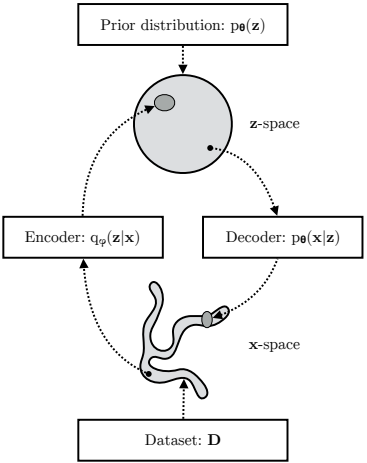


Figure from (Kingma & Welling, 2019)

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 the previous lecture.
[b] In this lecture.

Variational autoencoder for reconstruction and data imputation

Consider a test data point $\boldsymbol{x} = (\boldsymbol{x}_\text{o}, \boldsymbol{x}_\text{m})$, where $\boldsymbol{x}_\text{o}$ denotes observed features and $\boldsymbol{x}_\text{m}$ denotes unobserved (missing) features

Observed and missing features can vary between different test data points $\boldsymbol{x}$ and $\boldsymbol{x}'$

For a VAE model $\boldsymbol{x}_\text{o}$ and $\boldsymbol{x}_\text{m}$ are conditionally independent given $\boldsymbol{z}$ (assuming diagonal likelihood covariance), therefore

$$\begin{aligned} p(\boldsymbol{x}_\text{m} \mid \boldsymbol{x}_\text{o}) &= \int p(\boldsymbol{x}_\text{m} \mid \boldsymbol{z}) p(\boldsymbol{z} \mid \boldsymbol{x}_\text{o}) d\boldsymbol{z} \\ &\approx \int p_{\boldsymbol{\theta}}(\boldsymbol{x}_\text{m} \mid \boldsymbol{z}) q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}_\text{o}) d\boldsymbol{z} \end{aligned}$$

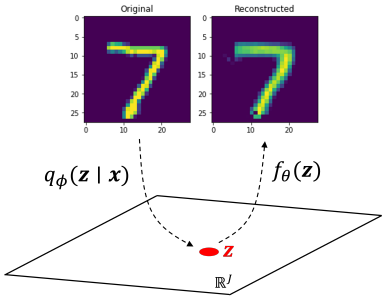
Variational autoencoder for reconstruction and data imputation

Consider a test data point $\boldsymbol{x} = (\boldsymbol{x}_\text{o}, \boldsymbol{x}_\text{m})$, where $\boldsymbol{x}_\text{o}$ denotes observed features and $\boldsymbol{x}_\text{m}$ denotes unobserved (missing) features

Observed and missing features can vary between different test data points $\boldsymbol{x}$ and $\boldsymbol{x}'$

For a VAE model $\boldsymbol{x}_\text{o}$ and $\boldsymbol{x}_\text{m}$ are conditionally independent given $\boldsymbol{z}$ (assuming diagonal likelihood covariance), therefore

$$p(\boldsymbol{x}_\text{m} \mid \boldsymbol{x}_\text{o}) = \int p(\boldsymbol{x}_\text{m} \mid \boldsymbol{z}) p(\boldsymbol{z} \mid \boldsymbol{x}_\text{o}) d\boldsymbol{z}$$
$$\approx \int p_{\boldsymbol{\theta}}(\boldsymbol{x}_\text{m} \mid \boldsymbol{z}) q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}_\text{o}) d\boldsymbol{z}$$



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

→ VAE architecture automatically implements a statistical data imputation method

Encoder model $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$ needs to be able to handle missing values in $\boldsymbol{x}$

Variational autoencoder for reconstruction and data imputation

Posterior matching technique (Strauss and Oliva, 2022)

- Artificially mask complete training data points as $\boldsymbol{x} = (\boldsymbol{x}_\text{o}, \boldsymbol{x}_\text{m})$
- Train an additional encoder model with parameters $\boldsymbol{\phi}'$ such that $q_{\boldsymbol{\phi}'}(\boldsymbol{z} \mid \boldsymbol{x}_\text{o}) \approx q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$ by maximizing

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log q_{\boldsymbol{\phi}'}(\boldsymbol{z} \mid \boldsymbol{x}_\text{o})]$$

Variational autoencoder for reconstruction and data imputation

Posterior matching technique (Strauss and Oliva, 2022)

- Artificially mask complete training data points as $\boldsymbol{x} = (\boldsymbol{x}_\text{o}, \boldsymbol{x}_\text{m})$
- Train an additional encoder model with parameters $\boldsymbol{\phi}'$ such that $q_{\boldsymbol{\phi}'}(\boldsymbol{z} \mid \boldsymbol{x}_\text{o}) \approx q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$ by maximizing

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log q_{\boldsymbol{\phi}'}(\boldsymbol{z} \mid \boldsymbol{x}_\text{o})]$$

Variationally optimal approach: use non-amortized variational inference to maximize (Agarwal et al., 2023)

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

w.r.t. $\boldsymbol{\psi}$ and estimate with previously trained decoder

$$p(\boldsymbol{x}_\text{m} \mid \boldsymbol{x}_\text{o}) \approx \int p_{\boldsymbol{\theta}}(\boldsymbol{x}_\text{m} \mid \boldsymbol{z}) q_{\boldsymbol{\psi}}(\boldsymbol{z}) d\boldsymbol{z}$$

Variational autoencoder for reconstruction and data imputation

Posterior matching technique (Strauss and Oliva, 2022)

- Artificially mask complete training data points as $\boldsymbol{x} = (\boldsymbol{x}_\text{o}, \boldsymbol{x}_\text{m})$
- Train an additional encoder model with parameters $\boldsymbol{\phi}'$ such that $q_{\boldsymbol{\phi}'}(\boldsymbol{z} \mid \boldsymbol{x}_\text{o}) \approx q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$ by maximizing

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log q_{\boldsymbol{\phi}'}(\boldsymbol{z} \mid \boldsymbol{x}_\text{o})]$$

Variationally optimal approach: use non-amortized variational inference to maximize (Agarwal et al., 2023)

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

w.r.t. $\boldsymbol{\psi}$ and estimate with previously trained decoder

$$p(\boldsymbol{x}_\text{m} \mid \boldsymbol{x}_\text{o}) \approx \int p_{\boldsymbol{\theta}}(\boldsymbol{x}_\text{m} \mid \boldsymbol{z}) q_{\boldsymbol{\psi}}(\boldsymbol{z}) d\boldsymbol{z}$$

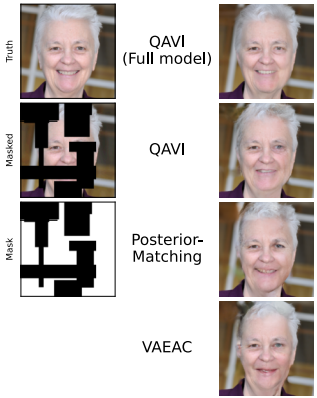


Figure from (Agarval et al., 2023).

Note: above results use additional tricks, such as deep/hierarchical VAEs, more expressive posteriors

Variational autoencoder for reconstruction and data imputation

Some VAE variants can explicitly model missingness together with $\boldsymbol{x}$, e.g., by also modeling missingness mask $\boldsymbol{m}$

Can model

- missing-completely-at-random (MCAR)
- missing-at-random (MAR)
- missing-not-at-random (MNAR)

Latent space interpolation

Explore the generative model via latent space interpolation

Consider two data points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and the mean of their embeddings $\boldsymbol{z}_1 = f_\phi(\boldsymbol{x}_1)$ and $\boldsymbol{z}_2 = f_\phi(\boldsymbol{x}_2)$

Generate new images via latent space interpolation

$$\boldsymbol{z} = \lambda \boldsymbol{z}_1 + (1 - \lambda)\boldsymbol{z}_2, \quad 0 \leq \lambda \leq 1$$

and decoding $\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})$

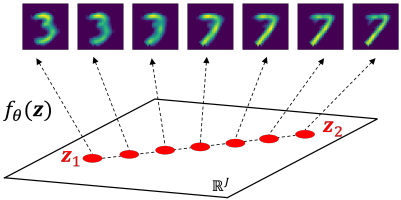


Figure from `https://mbernste.github.io/posts/vae/`

Note that the notation differs.

Latent space interpolation

Explore the generative model via latent space interpolation

Consider two data points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and the mean of their embeddings $\boldsymbol{z}_1 = f_\phi(\boldsymbol{x}_1)$ and $\boldsymbol{z}_2 = f_\phi(\boldsymbol{x}_2)$

Generate new images via latent space interpolation

$$\boldsymbol{z} = \lambda \boldsymbol{z}_1 + (1-\lambda)\boldsymbol{z}_2, \quad 0 \leq \lambda \leq 1$$

and decoding $\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})$

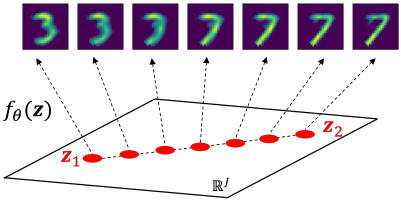


Figure from `https://mbernste.github.io/posts/vae/`

Note that the notation differs.

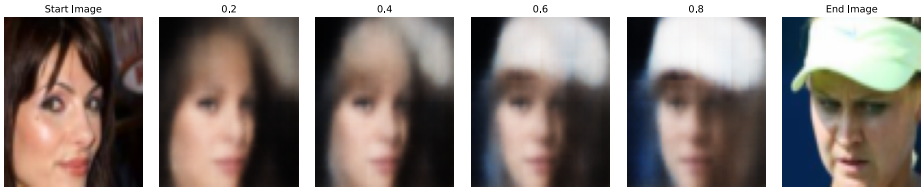


Figure 20.9 from (Murphy, 2023)

Latent space interpolation (and extrapolation)

Example: consider groups of images labeled as

- "neutral": $\{\boldsymbol{x}_i : i \in \mathcal{I}_{\text{neutral}}\}$
- "smiling": $\{\boldsymbol{x}_j : j \in \mathcal{I}_{\text{smiling}}\}$
- "mouth closed": $\{\boldsymbol{x}_k : k \in \mathcal{I}_{\text{closed}}\}$
- "mouth open": $\{\boldsymbol{x}_l : l \in \mathcal{I}_{\text{open}}\}$

and define

$$\boldsymbol{\Delta}_1 = \sum_{j \in \mathcal{I}_{\text{smiling}}} f_{\boldsymbol{\phi}}(\boldsymbol{x}_j) - \sum_{i \in \mathcal{I}_{\text{neutral}}} f_{\boldsymbol{\phi}}(\boldsymbol{x}_i)$$

$$\boldsymbol{\Delta}_2 = \sum_{l \in \mathcal{I}_{\text{open}}} f_{\boldsymbol{\phi}}(\boldsymbol{x}_l) - \sum_{k \in \mathcal{I}_{\text{closed}}} f_{\boldsymbol{\phi}}(\boldsymbol{x}_k)$$

A given image $\boldsymbol{x}$ can be manipulated via interpolation as

$$\begin{aligned} \boldsymbol{z} &= f_{\phi}(\boldsymbol{x}) \\ \boldsymbol{z}' &= \boldsymbol{z} + \lambda_1 \boldsymbol{\Delta}_1 + \lambda_2 \boldsymbol{\Delta}_2, \quad \lambda_1, \lambda_2 \in \mathbb{R} \\ \boldsymbol{x}' &\sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}') \end{aligned}$$

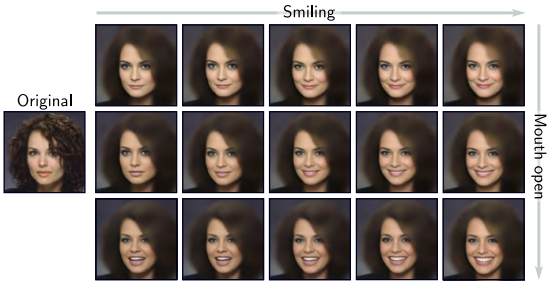


Figure 17.13 from (Prince, 2023)

Note: interpolation above uses non-linear interpolation in spherical coordinates.

Approximating sample probability

Use a trained VAE to approximate sample probability for test data to

- Estimate the quality of the model
- Detect out-of-distribution samples

Approximating sample probability

Use a trained VAE to approximate sample probability for test data to

- Estimate the quality of the model
- Detect out-of-distribution samples

Instead of using the ELBO, we can directly estimate probability for a given data point $\boldsymbol{x}$ using Monte Carlo

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \int p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z} \\ &= \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z})}[p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})] \\ &\approx \frac{1}{S} \sum_{s=1}^{S} p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}_s) \end{aligned}$$

where

$$\boldsymbol{z}_s \overset{\text{i.i.d}}{\sim} p_{\boldsymbol{\theta}}(\boldsymbol{z})$$

Approximating sample probability

Use a trained VAE to approximate sample probability for test data to

- Estimate the quality of the model
- Detect out-of-distribution samples

Instead of using the ELBO, we can directly estimate probability for a given data point $\boldsymbol{x}$ using Monte Carlo

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \int p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z} \\ &= \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z})}[p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})] \\ &\approx \frac{1}{S} \sum_{s=1}^{S} p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}_s) \end{aligned}$$

where

$$\boldsymbol{z}_s \overset{\text{i.i.d}}{\sim} p_{\boldsymbol{\theta}}(\boldsymbol{z})$$

This can be improved by importance sampling estimate

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \int p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z} \\ &= \int p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) d\boldsymbol{z} \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \left[p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \right] \\ &\approx \frac{1}{S} \sum_{s=1}^{S} \tilde{w}_s p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}_s), \end{aligned}$$

where

$$\boldsymbol{z}_s \overset{\text{i.i.d}}{\sim} q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}), \quad \tilde{w}_s = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}_s)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_s \mid \boldsymbol{x})}, \quad \sup p_{\boldsymbol{\theta}} \subseteq \sup q_{\boldsymbol{\phi}}$$

Posterior collapse

- Consider a powerful decoder (a "super" deep neural net of some sort) such that for every value of latent variable $\boldsymbol{z} \in \mathcal{Z}$ decoder outputs one of the data points
- Then approx. $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{z})$ because generative model does not really use the latent variable
- We can find inference network parameters $\boldsymbol{\phi}^*$ such that $q_{\boldsymbol{\phi}^*}(\boldsymbol{z} \mid \boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{z})$

Posterior collapse

- Consider a powerful decoder (a "super" deep neural net of some sort) such that for every value of latent variable $\boldsymbol{z} \in \mathcal{Z}$ decoder outputs one of the data points
- Then approx. $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{z})$ because generative model does not really use the latent variable
- We can find inference network parameters $\boldsymbol{\phi}^*$ such that $q_{\boldsymbol{\phi}^*}(\boldsymbol{z} \mid \boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{z})$
- In this setting, ELBO is maximized because $D_{\mathbb{KL}}(q_{\boldsymbol{\phi}^*}(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})) = 0$ but the model has not learned any meaningful latent representations
- Then also $D_{\mathbb{KL}}(q_{\boldsymbol{\phi}^*}(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z})) = 0$
- This is known as posterior collapse
- Posterior collapse is also due to other aspects, such local maxima in the complex objective function

β-VAE

VAE models can generate samples that may appear too smooth (e.g. blurry images)

β-VAE can improve sample quality (as well as help with posterior collapse) by reducing the penalty on the KL term

$$\mathcal{L}_\beta(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\right] - \beta D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z})), \quad \beta \geq 0$$

If we set

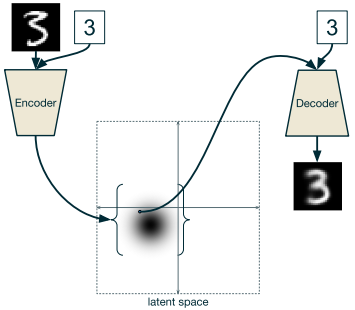
- $\beta = 1$ we obtain the standard objective for VAEs
- $\beta < 1$ we can reconstruct data samples better
- $\beta > 1$ we get more compressed representations (in terms of rate distortion, see Murphy (2023))
- $\beta = 0$ we obtain an objective that is analogous for autoencoder models

Conditional variational autoencoder

Conditional variational autoencoder (CVAE) is a neural architecture that consists of a conditional deep latent variable model of the form

$$\begin{aligned} \boldsymbol{z} &\sim p_{\boldsymbol{\theta}}(\boldsymbol{z}) \\ \boldsymbol{x} \mid \boldsymbol{z} &\sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{c}) = p(\boldsymbol{x} \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c})), \end{aligned}$$

where $\boldsymbol{c}$ contains the explanatory covariates and the model is trained using (reparameterized) amortized variational inference with an inference model $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{c})$.



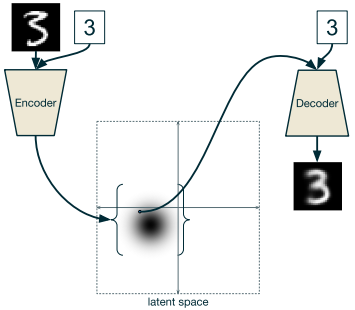
https://ijdykeman.github.io/ml/2016/12/21/cvae.html

Conditional variational autoencoder

Conditional variational autoencoder (CVAE) is a neural architecture that consists of a conditional deep latent variable model of the form

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

where $\boldsymbol{c}$ contains the explanatory covariates and the model is trained using (reparameterized) amortized variational inference with an inference model $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{c})$.



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

Alternative CVAE model definitions:

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

or

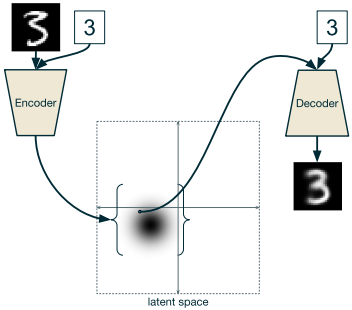
$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{c}) = p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{c})$$

Conditional variational autoencoder

Conditional variational autoencoder (CVAE) is a neural architecture that consists of a conditional deep latent variable model of the form

$$\begin{aligned} \boldsymbol{z} &\sim p_{\boldsymbol{\theta}}(\boldsymbol{z}) \\ \boldsymbol{x} \mid \boldsymbol{z} &\sim p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{c}) = p(\boldsymbol{x} \mid d_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c})), \end{aligned}$$

where $\boldsymbol{c}$ contains the explanatory covariates and the model is trained using (reparameterized) amortized variational inference with an inference model $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{c})$.



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

Alternative CVAE model definitions:

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

or

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

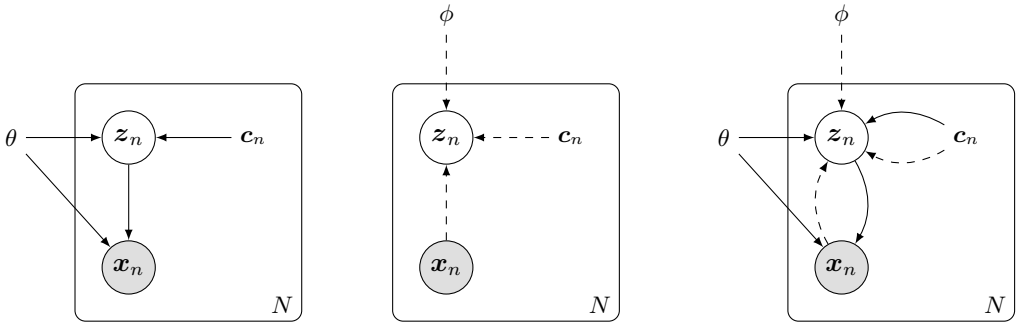
All methodological details remain essentially unchanged, including the ELBO objective

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

Conditional variational autoencoder: Probabilistic graphical model

Probabilistic graphical models for CVAE with $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{c}) = p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{c})$

- 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

Multimodal variational autoencoder

- Multimodal VAE allows modeling joint distributions over multiple data modalities, e.g. image and text
- Assuming M modalities $\boldsymbol{x}_1 \in \mathcal{X}_1, \ldots, \boldsymbol{x}_M \in \mathcal{X}_M$ and conditional independence given $\boldsymbol{z}$

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M, \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}) \prod_{m=1}^{M} p_{\boldsymbol{\theta}}(\boldsymbol{x}_m \mid \boldsymbol{z})$$

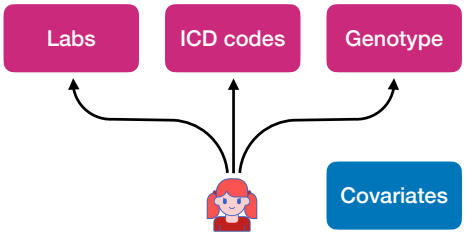


Illustration of multimodal data

Multimodal variational autoencoder

- Multimodal VAE allows modeling joint distributions over multiple data modalities, e.g. image and text
- Assuming M modalities $\boldsymbol{x}_1 \in \mathcal{X}_1, \ldots, \boldsymbol{x}_M \in \mathcal{X}_M$ and conditional independence given $\boldsymbol{z}$

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M, \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}) \prod_{m=1}^{M} p_{\boldsymbol{\theta}}(\boldsymbol{x}_m \mid \boldsymbol{z})$$

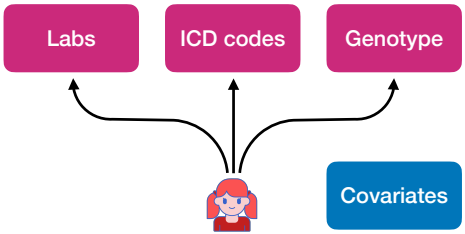


Illustration of multimodal data

- The ELBO for $\mathbf{X} = \{\boldsymbol{x}_1, \ldots \boldsymbol{x}_M\}$ is

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathbf{X}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})} \left[\sum_{m=1}^{M} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_m \mid \boldsymbol{z}) \right] - D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \mathbf{X}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z}))$$

- Likelihood terms may have different ranges
- Weighted ELBO with $\lambda_m \geq 0$

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathbf{X}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})} \left[\sum_{m=1}^{M} \lambda_m \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_m \mid \boldsymbol{z}) \right] - D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \mathbf{X}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z}))$$

Multimodal variational autoencoder for unpaired data

- Availability of paired multimodal data across all modalities is often limited
- For example, we may have lots of images, lots of text, but only few image-text pairs
- Let $O_m = 1$ denote that data modality m, $\boldsymbol{x}_m$, is observed, and $O_m = 0$ that modality m is not observed
- ELBO objective for partially paired data is

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathbf{X}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})} \left[\sum_{m:O_m=1} \lambda_m \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_m \mid \boldsymbol{z}) \right] - D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \mathbf{X}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z}))$$

Multimodal variational autoencoder for unpaired data

- Availability of paired multimodal data across all modalities is often limited
- For example, we may have lots of images, lots of text, but only few image-text pairs
- Let $O_m = 1$ denote that data modality m, $\boldsymbol{x}_m$, is observed, and $O_m = 0$ that modality m is not observed
- ELBO objective for partially paired data is

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathbf{X}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{X})}\left[\sum_{m:O_m=1} \lambda_m \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] - D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \mathbf{X}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z}))$$

- How to define the amortized VI such that it works for unpaired data?
- For example, if encoder $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}_1, \boldsymbol{x}_2)$ is trained on image-text pairs, how to compute posterior approximation using only images $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}_1)$ or text $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}_2)$, and vice versa?

Multimodal variational autoencoder with product of experts

From the conditional independence we get

$$
\begin{aligned}
p(\boldsymbol{z} \mid \mathbf{X}) &= \frac{p(\boldsymbol{z}) p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \mid \boldsymbol{z})}{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)} \\
&= \frac{p(\boldsymbol{z})}{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)} \prod_{m=1}^{M} p(\boldsymbol{x}_m \mid \boldsymbol{z}) \\
&= \frac{p(\boldsymbol{z})}{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)} \prod_{m=1}^{M} \frac{p(\boldsymbol{z} \mid \boldsymbol{x}_m) p(\boldsymbol{x}_m)}{p(\boldsymbol{z})} \\
&\propto p(\boldsymbol{z}) \prod_{m=1}^{M} \frac{p(\boldsymbol{z} \mid \boldsymbol{x}_m)}{p(\boldsymbol{z})} \\
&\approx p(\boldsymbol{z}) \prod_{m=1}^{M} q_m(\boldsymbol{z} \mid \boldsymbol{x}_m),
\end{aligned}
$$

where $q_m(\boldsymbol{z} \mid \boldsymbol{x}_m)$ is the "expert" for data modality m

Multimodal variational autoencoder with product of experts

From the conditional independence we get

$$
\begin{aligned}
p(\boldsymbol{z} \mid \mathbf{X}) &= \frac{p(\boldsymbol{z}) p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \mid \boldsymbol{z})}{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)} \\
&= \frac{p(\boldsymbol{z})}{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)} \prod_{m=1}^{M} p(\boldsymbol{x}_m \mid \boldsymbol{z}) \\
&= \frac{p(\boldsymbol{z})}{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)} \prod_{m=1}^{M} \frac{p(\boldsymbol{z} \mid \boldsymbol{x}_m) p(\boldsymbol{x}_m)}{p(\boldsymbol{z})} \\
&\propto p(\boldsymbol{z}) \prod_{m=1}^{M} \frac{p(\boldsymbol{z} \mid \boldsymbol{x}_m)}{p(\boldsymbol{z})} \\
&\approx p(\boldsymbol{z}) \prod_{m=1}^{M} q_m(\boldsymbol{z} \mid \boldsymbol{x}_m),
\end{aligned}
$$

where $q_m(\boldsymbol{z} \mid \boldsymbol{x}_m)$ is the "expert" for data modality m

This is known as the product of experts

Posterior approximation can be computed for any subset of modalities which are observed

If prior and experts are Gaussians

$$
\begin{aligned}
p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0^{-1}) \\
q_m(\boldsymbol{z} \mid \boldsymbol{x}_m) &= \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m^{-1}),
\end{aligned}
$$

then the approximative posterior is also Gaussian

$$
\prod_{m=0}^{M} \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m^{-1}) \propto \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}, \boldsymbol{\Lambda})
$$

$$
\boldsymbol{\Sigma} = \left(\sum_{m=0}^{M} \boldsymbol{\Lambda}_m \right)^{-1} \quad \text{and} \quad \boldsymbol{\mu} = \boldsymbol{\Sigma} \left(\sum_{m=0}^{M} \boldsymbol{\Lambda}_m \boldsymbol{\mu}_m \right)
$$

Multimodal variational autoencoder with product of experts: illustration

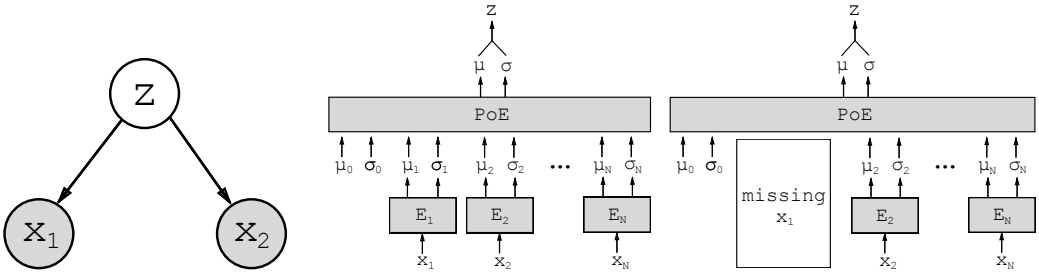


Figure 21.5 from (Murphy, 2023)

Semi-supervised VAEs

Semi-supervised learning methods can be useful when data consists of large amounts of unlabeled data $\mathcal{D}_U = \{(\boldsymbol{x}_n)\}$ and few labeled data points $\mathcal{D}_L = \{(\boldsymbol{x}_n, y_n)\}$

One of the earliest models, called M2, is

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, y) = p_{\boldsymbol{\theta}}(y)p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid y) = p_{\boldsymbol{\theta}}(y) \int p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid y, \boldsymbol{z})p_{\boldsymbol{\theta}}(\boldsymbol{z})d\boldsymbol{z},$$

where $p_{\boldsymbol{\theta}}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \mathbf{0}, \mathbf{I})$, $p_{\boldsymbol{\theta}}(y) = \mathrm{Cat}(y \mid \boldsymbol{\pi})$ is trainable label prior, and $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid y, \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid d_{\boldsymbol{\theta}}(y, \boldsymbol{z}))$

Comparison to CVAE:

- $\boldsymbol{c}$ is a fixed covariate/predictor
- here label y is a random variable

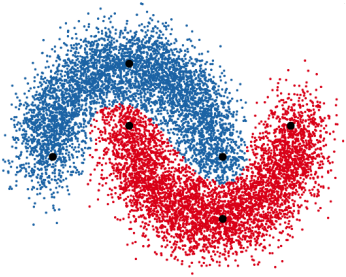


Figure from (Maaloe et al, 2016)

Semi-supervised VAEs

Variational approximation for labeled data

$$\begin{aligned} q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}, y) &= q(\boldsymbol{z} \mid f_{\boldsymbol{\phi}}(\boldsymbol{x}, y)) \\ &= \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}, y), \operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x}, y))) \end{aligned}$$

Standard ELBO for labeled data (by marginalizing $\boldsymbol{z}$)

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

Semi-supervised VAEs

Variational approximation for labeled data

$$\begin{aligned} q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}, y) &= q(\boldsymbol{z} \mid f_{\boldsymbol{\phi}}(\boldsymbol{x}, y)) \\ &= \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}, y), \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x}, y))) \end{aligned}$$

Standard ELBO for labeled data (by marginalizing $\boldsymbol{z}$)

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

Variational approximation for unlabeled data

$$\begin{aligned} q_{\boldsymbol{\phi}}(\boldsymbol{z}, y \mid \boldsymbol{x}) &= q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) q_{\boldsymbol{\phi}}(y \mid \boldsymbol{x}) \\ q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) &= \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x}))) \\ q_{\boldsymbol{\phi}}(y \mid \boldsymbol{x}) &= \mathrm{Cat}(y \mid \boldsymbol{\pi}_{\boldsymbol{\phi}}(\boldsymbol{x})) \end{aligned}$$

ELBO for unlabeled data by (marginalizing $\boldsymbol{z}$ and y)

$$\begin{aligned} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &\geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}, y \mid \boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, y, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}, y \mid \boldsymbol{x})} \right] \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}, y \mid \boldsymbol{x})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid y, \boldsymbol{z}) + \log p_{\boldsymbol{\theta}}(y) \\ &\quad + \log p_{\boldsymbol{\theta}}(\boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}, y \mid \boldsymbol{x})] \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(y \mid \boldsymbol{x})} [\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}, y)] + \mathbb{H}(q_{\boldsymbol{\phi}}(y \mid \boldsymbol{x})) \\ &= \mathcal{U}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) \end{aligned}$$

Semi-supervised VAEs

Variational approximation $q_{\boldsymbol{\phi}}(y \mid \boldsymbol{x})$ can be considered as discriminative classifier

Because classifier is only used to compute log-likelihood of unlabeled data, a modified objective

$$\mathcal{L}'(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathcal{D}_L, \mathcal{D}_U) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathcal{D}_L) + \mathcal{U}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathcal{D}_U) + \mathbb{E}_{(\boldsymbol{x}_n, y_n) \sim \mathcal{D}_L} \log q_{\boldsymbol{\phi}}(y \mid \boldsymbol{x}),$$

where $(\boldsymbol{x}_n, y_n) \sim \mathcal{D}_L$ denotes labeled data distribution

N	NN	CNN	TSVM	CAE	MTC	AtlasRBF	M1+TSVM	M2	M1+M2
100	25.81	22.98	16.81	13.47	12.03	8.10 ($\pm$ 0.95)	11.82 ($\pm$ 0.25)	11.97 ($\pm$ 1.71)	**3.33** ($\pm$ 0.14)
600	11.44	7.68	6.16	6.3	5.13	–	5.72 ($\pm$ 0.049)	4.94 ($\pm$ 0.13)	**2.59** ($\pm$ 0.05)
1000	10.7	6.45	5.38	4.77	3.64	3.68 ($\pm$ 0.12)	4.24 ($\pm$ 0.07)	3.60 ($\pm$ 0.56)	**2.40** ($\pm$ 0.02)
3000	6.04	3.35	3.45	3.22	2.57	–	3.49 ($\pm$ 0.04)	3.92 ($\pm$ 0.63)	**2.18** ($\pm$ 0.04)

Figure from (Kingma et al, 2014)

Semi-supervised VAEs

Variational approximation $q_\phi(y \mid \boldsymbol{x})$ can be considered as discriminative classifier

Because classifier is only used to compute log-likelihood of unlabeled data, a modified objective

$$\mathcal{L}'(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathcal{D}_L, \mathcal{D}_U) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathcal{D}_L) + \mathcal{U}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathcal{D}_U) + \mathbb{E}_{(\boldsymbol{x}_n, y_n) \sim \mathcal{D}_L} \log q_\phi(y \mid \boldsymbol{x}),$$

where $(\boldsymbol{x}_n, y_n) \sim \mathcal{D}_L$ denotes labeled data distribution

Methods have been further developed e.g. by adding additional latent variables (e.g. Maaloe et al, 2016) among others

N	NN	CNN	TSVM	CAE	MTC	AtlasRBF	M1+TSVM	M2	M1+M2
100	25.81	22.98	16.81	13.47	12.03	8.10 ($\pm$ 0.95)	11.82 ($\pm$ 0.25)	11.97 ($\pm$ 1.71)	**3.33** ($\pm$ 0.14)
600	11.44	7.68	6.16	6.3	5.13	–	5.72 ($\pm$ 0.049)	4.94 ($\pm$ 0.13)	**2.59** ($\pm$ 0.05)
1000	10.7	6.45	5.38	4.77	3.64	3.68 ($\pm$ 0.12)	4.24 ($\pm$ 0.07)	3.60 ($\pm$ 0.56)	**2.40** ($\pm$ 0.02)
3000	6.04	3.35	3.45	3.22	2.57	–	3.49 ($\pm$ 0.04)	3.92 ($\pm$ 0.63)	**2.18** ($\pm$ 0.04)

Figure from (Kingma et al, 2014)

Hierarchical VAEs

We can define hierarchical VAEs with K stochastic layers $\boldsymbol{z}_{1:K} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_K)$ as follows

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}_{1:K}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}_K) \prod_{k=1}^{K} p_{\boldsymbol{\theta}}(\boldsymbol{z}_k \mid \boldsymbol{z}_{k+1}) p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}_1)$$

Generative model can be extended into non-Markovian model

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}_{1:K}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}_K) \prod_{k=1}^{K} p_{\boldsymbol{\theta}}(\boldsymbol{z}_k \mid \boldsymbol{z}_{k+1:K}) p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}_{1:K})$$

Similar to using skip connection architecture

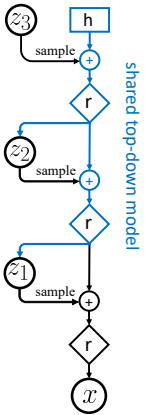


Figure 21.11 from (Murphy et al, 2023)

Hierarchical VAEs: inference model

Variational inference model can be define as bottom-up

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K}) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_1 \mid \boldsymbol{x}) \prod_{k=2}^{K} q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x}, \boldsymbol{z}_{1:k-1})$$

Hierarchical VAEs: inference model

Variational inference model can be define as bottom-up

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K}) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_1 \mid \boldsymbol{x}) \prod_{k=2}^{K} q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x}, \boldsymbol{z}_{1:k-1})$$

or as top-down

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K}) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_K \mid \boldsymbol{x}) \prod_{k=K-1}^{1} q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x}, \boldsymbol{z}_{k+1:K})$$

Hierarchical VAEs: inference model

Variational inference model can be define as bottom-up

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K}) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_1 \mid \boldsymbol{x}) \prod_{k=2}^{K} q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x}, \boldsymbol{z}_{1:k-1})$$

or as top-down

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K}) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_K \mid \boldsymbol{x}) \prod_{k=K-1}^{1} q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x}, \boldsymbol{z}_{k+1:K})$$

or both

Top-down inference model is generally considered better because it can be shown to more closely approximate the true posterior of a given layer

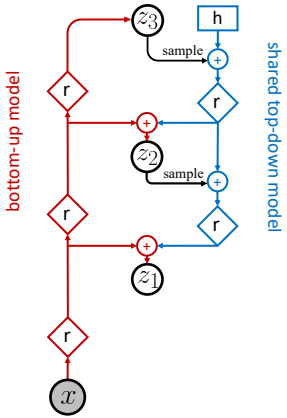


Figure 21.11 from (Murphy et al, 2023)

Hierarchical VAEs: ELBO

Evidence lower-bound for hierarchical VAE with top-down inference model is

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K} \mid \boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid z_{1:K})] - D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}_K \mid \boldsymbol{x}) \mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{z}_K)) - \sum_{k=K-1}^{1} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{k+1:K} \mid \boldsymbol{x})} D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x}, \boldsymbol{z}_{k+1:K}) \mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{z}_k \mid \boldsymbol{z}_{k+1:K})),$$

where $q_{\boldsymbol{\phi}}(\boldsymbol{z}_{k+1:K} \mid \boldsymbol{x}) = \prod_{i=k+1}^{K} q_{\boldsymbol{\phi}}(\boldsymbol{z}_i \mid \boldsymbol{x}, \boldsymbol{z}_{i+1:K})$

Hierarchical VAE example

So-called very deep VAE (VDVAE) model uses hierarchical VAE with bidirectional CNN architecture and nearest neighbour upsampling

Low-resolution (top) latents capture global structure, while the bottom of the hierarchy fills in details

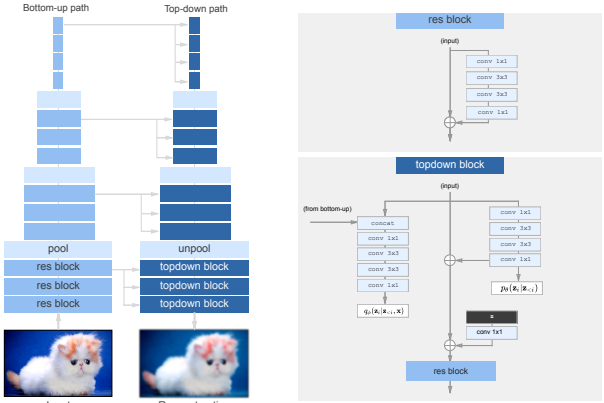


Figure 21.12 from (Murphy et al, 2023)

Hierarchical VAE example

So-called very deep VAE (VDVAE) model uses hierarchical VAE with bidirectional CNN architecture and nearest neighbour upsampling

Low-resolution (top) latents capture global structure, while the bottom of the hierarchy fills in details

Figure 21.13 from (Murphy et al, 2023)

Recap: Marginal likelihood, ELBO, importance sampling

Marginal likelihood and ELBO

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \log \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})}\right] \overset{\text{Jensen}}{\geq} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})}\right)\right] = \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi} \mid \boldsymbol{x})$$

and Monte Carlo estimate

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi} \mid \boldsymbol{x}) \approx \frac{1}{S}\left(\sum_{s=1}^{S} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}_s)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_s \mid \boldsymbol{x})}\right) \quad \text{where} \quad \boldsymbol{z}_s \overset{\text{i.i.d}}{\sim} q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$$

Recap: Marginal likelihood, ELBO, importance sampling

Marginal likelihood and ELBO

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \log \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})}\right] \overset{\text{Jensen}}{\geq} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})}\right)\right] = \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi} \mid \boldsymbol{x})$$

and Monte Carlo estimate

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi} \mid \boldsymbol{x}) \approx \frac{1}{S}\left(\sum_{s=1}^{S} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}_s)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_s \mid \boldsymbol{x})}\right) \quad \text{where} \quad \boldsymbol{z}_s \overset{\text{i.i.d}}{\sim} q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$$

Importance sampling estimate of $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})}\right] \approx \frac{1}{S}\sum_{s=1}^{S} \tilde{w}_s \quad \text{and} \quad \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \approx \log\left(\frac{1}{S}\sum_{s=1}^{S} \tilde{w}_s\right)$$

where

$$\tilde{w}_s = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}_s)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_s \mid \boldsymbol{x})}, \quad \boldsymbol{z}_s \overset{\text{i.i.d}}{\sim} q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}), \quad \sup p_{\boldsymbol{\theta}} \subseteq \sup q_{\boldsymbol{\phi}}$$

Importance weighted variational autoencoder

Empirical estimate of the importance weighted variational autoencoder (IWAE) objective

$$\tilde{\mathcal{L}}_K^{\text{IWAE}}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \log \left(\frac{1}{K} \sum_{k=1}^{K} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}_k)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x})} \right) = \log \left(\frac{1}{K} \sum_{k=1}^{K} \tilde{w}_k \right) \approx \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

where

$$\boldsymbol{z}_k \overset{\text{i.i.d}}{\sim} q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$$

Importance weighted variational autoencoder

Empirical estimate of the importance weighted variational autoencoder (IWAE) objective

$$\tilde{\mathcal{L}}_K^{\text{IWAE}}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \log\left(\frac{1}{K}\sum_{k=1}^{K}\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}_k)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x})}\right) = \log\left(\frac{1}{K}\sum_{k=1}^{K}\tilde{w}_k\right) \approx \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

where

$$\boldsymbol{z}_k \overset{\text{i.i.d}}{\sim} q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$$

By defining $q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K} \mid \boldsymbol{x}) = \prod_{k=1}^{K} q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x})$, IWAE objective (also called as the multiple-sample ELBO) is

$$\mathcal{L}_K^{\text{IWAE}}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K} \mid \boldsymbol{x})}\left[\log\frac{1}{K}\sum_{k=1}^{K}\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}_k)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x})}\right] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K} \mid \boldsymbol{x})}\left[\log\frac{1}{K}\sum_{k=1}^{K}\tilde{w}_k\right],$$

where

$$\tilde{w}_k = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}_k)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x})}$$

Importance weighted variational autoencoder: properties

1. IWAE objective reduces to the standard single sample ELBO when $K = 1$

$$\mathcal{L}_1^{\text{IWAE}}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:1} \mid \boldsymbol{x})} \left[\log \frac{1}{1} \sum_{k=1}^{1} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}_k)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x})} \right] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \right] = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x})$$

Importance weighted variational autoencoder: properties

1. IWAE objective reduces to the standard single sample ELBO when $K = 1$

$$\mathcal{L}_1^{\text{IWAE}}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:1} \mid \boldsymbol{x})} \left[\log \frac{1}{1} \sum_{k=1}^{1} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}_k)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x})} \right] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \right] = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x})$$

2. IWAE objective is a lower bound for $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$

$$\mathcal{L}_K^{\text{IWAE}}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K} \mid \boldsymbol{x})} \left[\log \frac{1}{K} \sum_{k=1}^{K} \tilde{w}_k \right] \overset{\text{Jensen}}{\leq} \log \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K} \mid \boldsymbol{x})} \left[\frac{1}{K} \sum_{k=1}^{K} \tilde{w}_k \right] = \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Importance weighted variational autoencoder: properties

1. IWAE objective reduces to the standard single sample ELBO when $K = 1$

$$\mathcal{L}_1^{\text{IWAE}}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:1} \mid \boldsymbol{x})} \left[\log \frac{1}{1} \sum_{k=1}^{1} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}_k)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x})} \right] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \right] = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x})$$

2. IWAE objective is a lower bound for $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$

$$\mathcal{L}_K^{\text{IWAE}}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K} \mid \boldsymbol{x})} \left[\log \frac{1}{K} \sum_{k=1}^{K} \tilde{w}_k \right] \overset{\text{Jensen}}{\leq} \log \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K} \mid \boldsymbol{x})} \left[\frac{1}{K} \sum_{k=1}^{K} \tilde{w}_k \right] = \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

3. For any $K \geq 1$, IWAE lower bound satisfies

$$\mathcal{L}_K^{\text{IWAE}} \leq \mathcal{L}_{K+1}^{\text{IWAE}} \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Importance weighted variational autoencoder: properties

1. IWAE objective reduces to the standard single sample ELBO when $K = 1$

$$\mathcal{L}_1^{\text{IWAE}}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:1} \mid \boldsymbol{x})} \left[\log \frac{1}{1} \sum_{k=1}^{1} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}_k)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k \mid \boldsymbol{x})} \right] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})} \right] = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x})$$

2. IWAE objective is a lower bound for $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$

$$\mathcal{L}_K^{\text{IWAE}}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K} \mid \boldsymbol{x})} \left[\log \frac{1}{K} \sum_{k=1}^{K} \tilde{w}_k \right] \overset{\text{Jensen}}{\leq} \log \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:K} \mid \boldsymbol{x})} \left[\frac{1}{K} \sum_{k=1}^{K} \tilde{w}_k \right] = \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

3. For any $K \geq 1$, IWAE lower bound satisfies

$$\mathcal{L}_K^{\text{IWAE}} \leq \mathcal{L}_{K+1}^{\text{IWAE}} \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

4. IWAE lower bound approaches the true log marginal likelihood in the limit

$$\lim_{K \to \infty} \mathcal{L}_K^{\text{IWAE}} = \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Identifiability of VAEs

Consider VAE model with linear parametrization

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

where $\mathbf{D}$ is diagonal and all parameters are shared with all samples

VAE models are trained via ELBO $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$

Probabilistic principle component analysis (PPCA) is equivalent with the above decoder model

- Marginal likelihood $\log p(\boldsymbol{x})$ has closed-form
- Posterior $p(\boldsymbol{z} \mid \boldsymbol{x})$ has closed-form
- Maximum likelihood parameter estimates have also closed form (up to rotation), i.e., identifiable

Identifiability of VAEs

Consider VAE model with linear parametrization

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

where $\mathbf{D}$ is diagonal and all parameters are shared with all samples

VAE models are trained via ELBO $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{x}) \leq \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$

Probabilistic principle component analysis (PPCA) is equivalent with the above decoder model

- Marginal likelihood $\log p(\boldsymbol{x})$ has closed-form
- Posterior $p(\boldsymbol{z} \mid \boldsymbol{x})$ has closed-form
- Maximum likelihood parameter estimates have also closed form (up to rotation), i.e., identifiable

Identifiability of linear VAE (Lucas et al., 2019)

The global maximum of the ELBO for the linear VAE

- Is identical to the global maximum for the log marginal likelihood of PPCA
- Has the scaled principal components as the columns of the decoder network

Identifiability of VAEs

An unconstrained deep latent variable model $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is not in general identifiable

Model identifiability can be proved by assuming specific sparsity in $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})$ and additional technical assumptions (Moran et al., 2022)

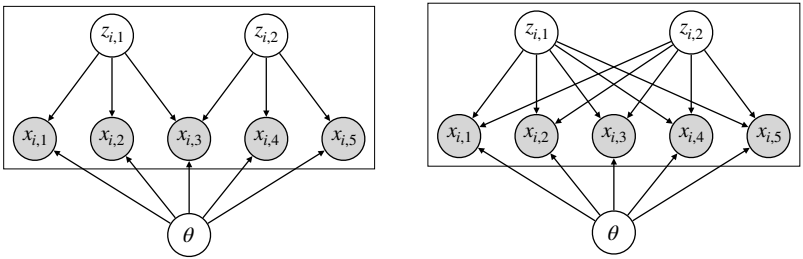


Figure from (Moran et al, 2022)

Identifiability of CVAEs

Consider a CVAE with conditional generative model $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{c}) = p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{c})$

- Assume prior $p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{c})$ is mean field with exponential family of distributions
- Covariate $\boldsymbol{c}$ has at least $\dim(\boldsymbol{z}) \cdot k + 1$ distinct values, where k denotes the number of sufficient statistics of the prior
- Amortized variational approximation $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{c})$ includes the true posterior
- $+$ Additional technical assumptions (Khemakem et al., 2020)

Then the parameters of the CVAE are asymptotically identifiable (Khemakem et al., 2020)

Amortized VI: amortization gap

References

- Agarwal S, Hope G, Younis A, Sudderth EB, A decoder suffices for query-adaptive variational inference, *Uncertainty in Artificial Intelligence*, 2023.
- Khemakhem I, Kingma D, Monti R, Hyvärinen A, Variational autoencoders and nonlinear ICA: A unifying framework, *International Conference on Artificial Intelligence and Statistics*, 2020.
- Kingma DP and Welling M, An Introduction to variational autoencoders, *Foundations and Trends in Machine Learning*, Vol. 12, No. 4, pp. 307-392.
- Kingma DP, Rezende DJ, Mohamed S, Welling M, Semi-supervised learning with deep generative models, *Neural information processing systems*, 2014.
- Lucas J, Tucker G, Grosse R, Norouzi M, Don't blame the ELBO! A linear VAE perspective on posterior collapse, *Neural Information Processing Systems*, 2019.
- Maaloe L, Sonderby CK, Sønderby SK, Winther O, Auxiliary deep generative models, *International Conference on Machine Learning*, PMLR 48:1445-1453, 2016.
- Moran GE, Sridhar D, Wang Y, Blei DM, Identifiable deep generative models via sparse decoding, *Transactions on Machine Learning Research*, 2022.
- Murphy K, Probabilistic Machine Learning: Advanced Topics, 2023.
- Prince SJD, Understanding Deep Learning, The MIT Press, 2023.
- Strauss R, Oliva J, Posterior matching for arbitrary conditioning, *Advances in Neural Information Processing Systems*, 2022.