Statistical Models & Computing Methods

Lecture 18: Variational Autoencoder

Cheng Zhang

School of Mathematical Sciences, Peking University

November 24, 2022

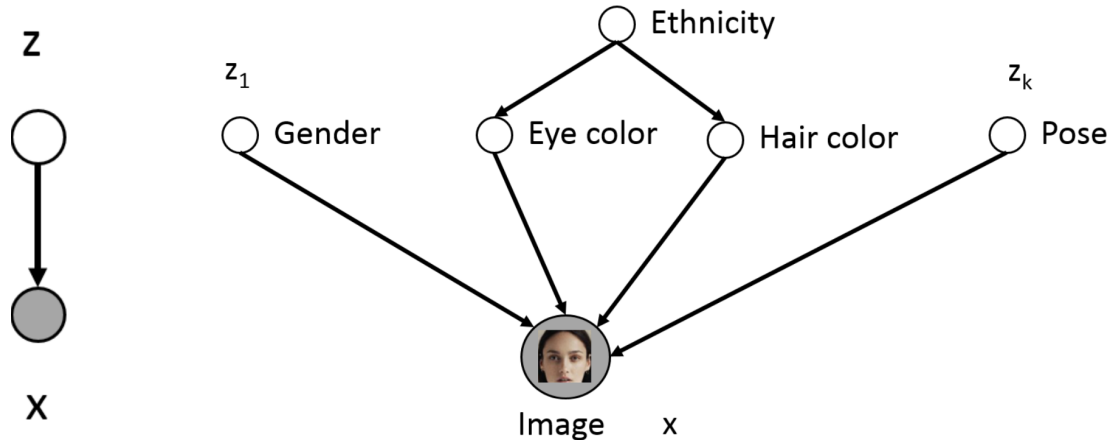
Recap of Autoregressive Models

- Autoregressive models:
  - Chain rule based factorization is fully general
  - Compact representation via conditional independence and /or neural parameterization
- Pros:
  - Easy to evaluate likelihoods
  - Easy to train
- Cons:
  - Requires an ordering
  - Generation is sequential
  - Cannot learn features in an unsupervised way

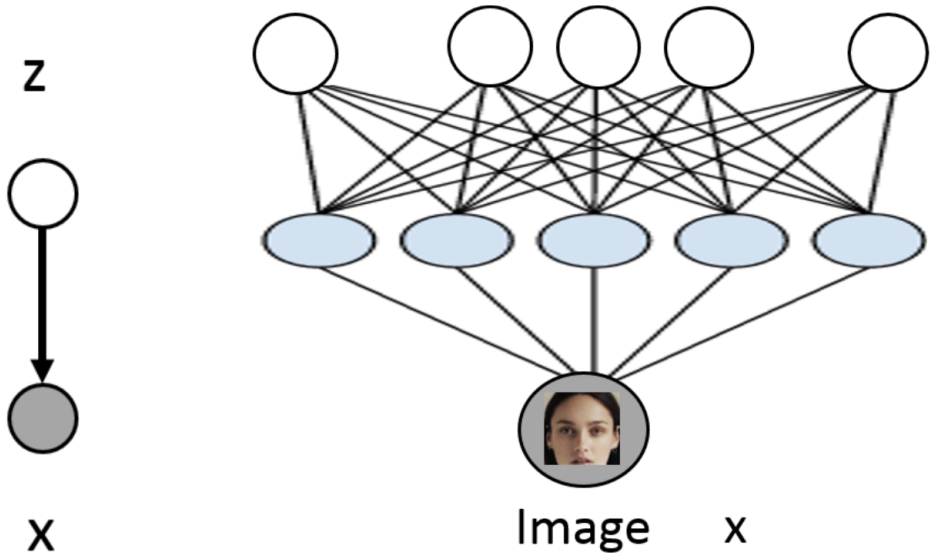
Latent Variable Models: Motivation

- Lots of variability in images x due to gender, eye color, hair color, pose, etc. However, unless images are annotated, these factors of variation are not explicitly available (latent)
- Idea: explicitly model these factors using latent variables z

Latent Variable Models: Motivation

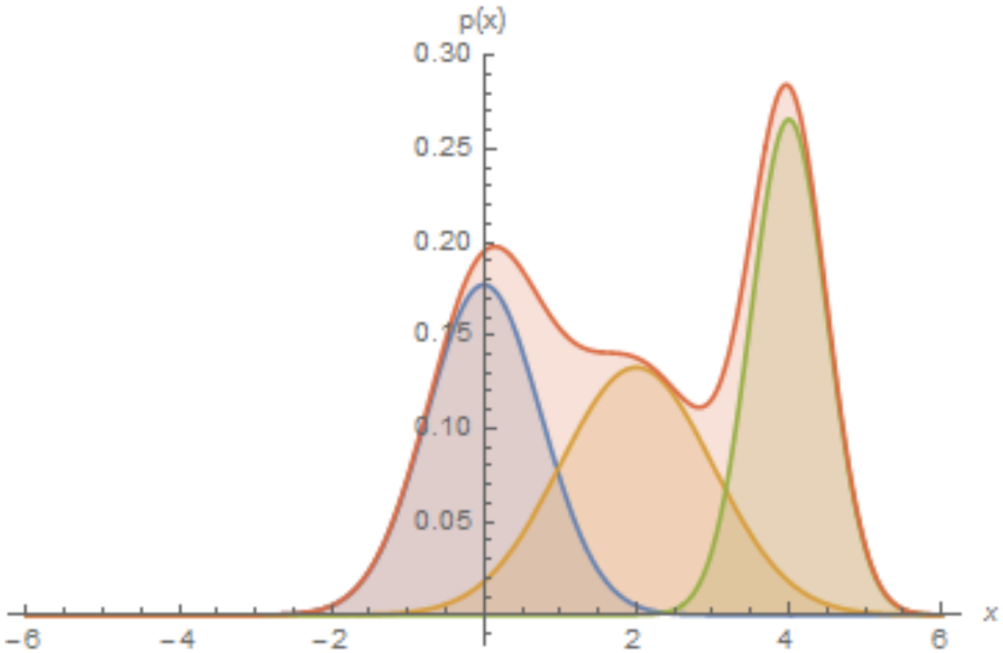


- Only shaded variables x are observed in the data (pixel values)
- Latent variables z correspond to high level features
  - If z chosen properly, $p(x|z)$ could be much simpler than $p(x)$
  - If we had trained this model, then we could identify features via $p(z|x)$, e.g., $p(\text{EyeColor} = \text{Blue}|x)$
- **Challenge**: Very difficult to specify these conditionals by hand



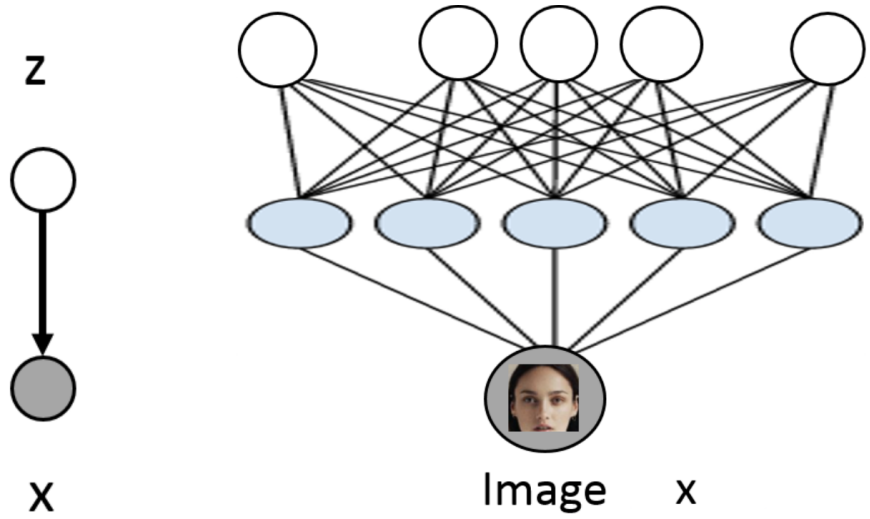
- $z \sim \mathcal{N}(0, I)$
- $p(x|z) = \mathcal{N}(\mu_\theta(z), \Sigma_\theta(z))$ where $\mu_\theta, \Sigma_\theta$ are neural networks
- Hope that after training, z will correspond to meaningful latent factors of variation (features). Unsupervised representation learning
- As before, features can be computed via $p(z|x)$

Combine simple models into a more complex and expressive one



$$p(x) = \sum_z p(x, z) = \sum_z p(z)p(x|z) = \sum_{k=1}^{K} p(z = k)\mathcal{N}(x; \mu_k, \Sigma_k)$$

Variational Autoencoder: Marginal Likelihood



A mixture of infinite many Gaussians

- $z \sim \mathcal{N}(0, I)$
- $p(x|z) = \mathcal{N}(\mu_\theta(z), \Sigma_\theta(z))$ where $\mu_\theta, \Sigma_\theta$ are neural networks
- Even though $p(x|z)$ is simple, the marginal $p(x)$ could be very complex/flexible

$$p_\theta(x) = \int_z p_\theta(x, z)dz = \int_z p_\theta(x|z)p(z)dz$$

- Allow us to define complex models $p(x)$ in terms of simple building blocks $p(x|z)$
- Natural for unsupervised learning tasks (clustering, unsupervised representation learning, etc)
- No free lunch: much more difficult to learn compared to fully observed autoregressive models

$$p_\theta(x) = \mathbb{E}_{z\sim p(z)} p_\theta(x|z), \quad \nabla_\theta p_\theta(x) = \mathbb{E}_{z\sim p(z)} \nabla_\theta p_\theta(x|z)$$

We can use Monte Carlo estimate for the marginal likelihood and its gradient

- Sample $z^{(1)}, \cdots, z^{(k)}$ from the prior $p(z)$
- Approximate expectation with sample average

$$p_\theta(x) \approx \frac{1}{k}\sum_{i=1}^{k} p_\theta(x|z^{(i)}), \quad \nabla_\theta p_\theta(x) \approx \frac{1}{k}\sum_{i=1}^{k} \nabla_\theta p_\theta(x|z^{(i)})$$

Remark: work in theory but not in practice. For most $z \sim p(z)$, $p_\theta(x|z)$ is very low, i.e., mismatch between the prior and posterior. This leads to large variance for the Monte Carlo estimates. We need a clever way to select $z^{(i)}$ to reduce the variance of the estimator.

Second Attempt: Importance Sampling

We can use importance sampling to reduce the variance

$$p_\theta(x) = \int_z p_\theta(x|z)p(z)dz = \int_z q(z)\frac{p_\theta(x,z)}{q(z)}dz = \mathbb{E}_{z\sim q(z)}\frac{p_\theta(x,z)}{q(z)}$$

Similarly, we can use Monte Carlo estimate

- Sample $z^{(1)}, \cdots, z^{(k)}$ from the important distribution $q(z)$
- Approximate expectation with sample average

$$p_\theta(x) \approx \frac{1}{k}\sum_{i=1}^{k}\frac{p_\theta(x,z^{(i)})}{q(z^{(i)})}$$

Remark: What is a good choice for $q(z)$?

- Evidence Lower Bound (ELBO)

$$\begin{aligned}\log p_\theta(x) &\geq \mathbb{E}_{z\sim q(z)} \log \frac{p_\theta(x,z)}{q(z)} \\ &= \mathbb{E}_{z\sim q(z)} \log p_\theta(x,z) - \mathbb{E}_{z\sim q(z)} \log q(z) \\ &= \mathbb{E}_{z\sim q(z)} \log p_\theta(x,z) + H(q)\end{aligned}$$

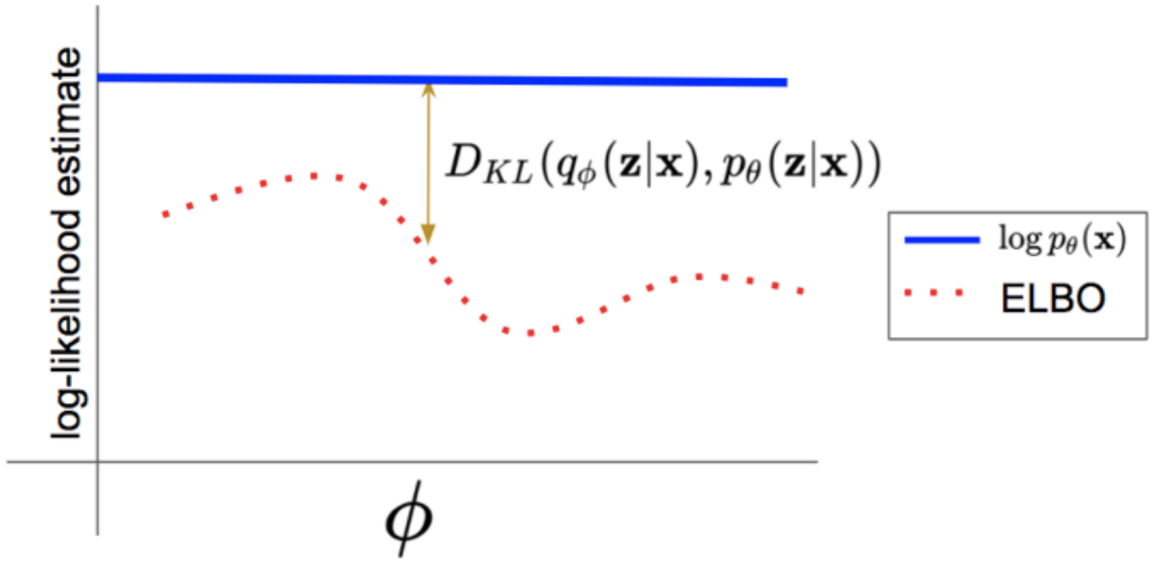
- Equality holds when $q(z) = p(z|x;\theta)$

$$\log p_\theta(x) = \mathbb{E}_{z\sim p(z|x;\theta)} \log p_\theta(x,z) + H(p(z|x;\theta))$$

This is the E-step in EM!

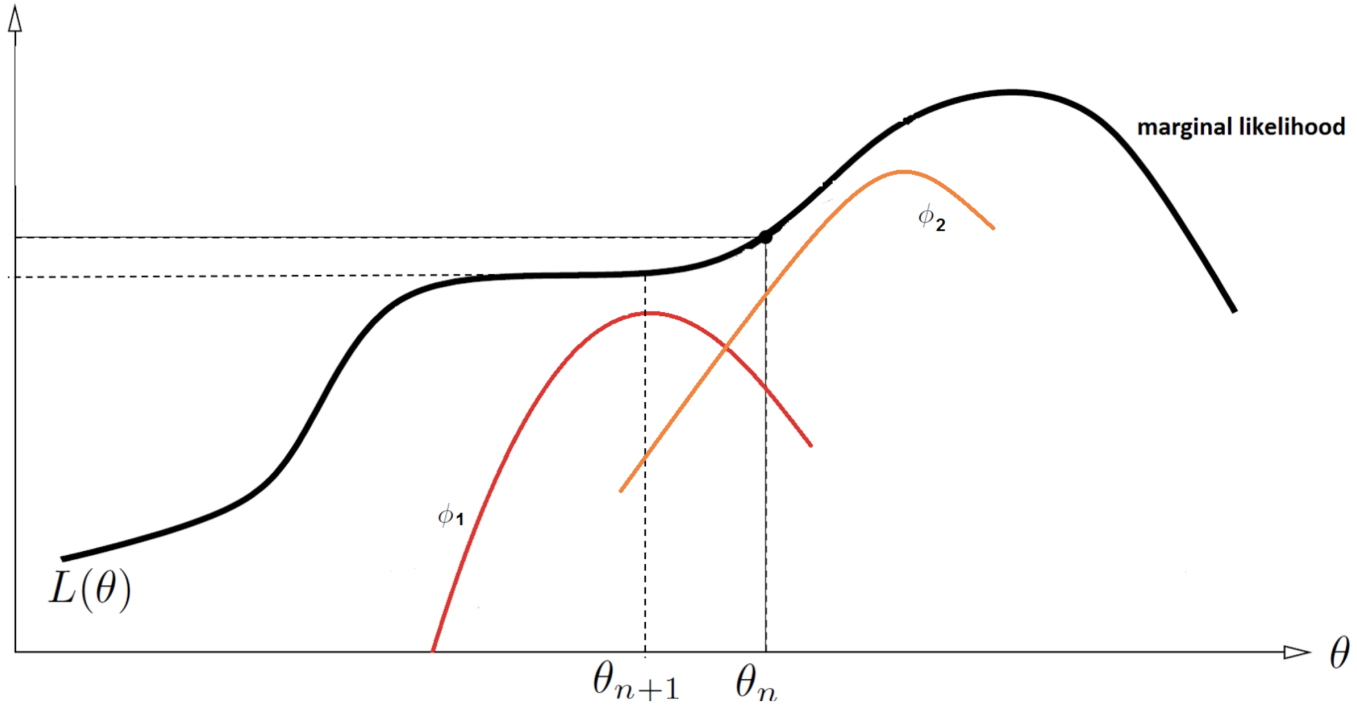
- In practice, $p(z|x,\theta)$ is usually intractable. We can find the "best" $q(z)$ by maximizing the ELBO in a parameterized family of $\{q_\phi(z) : \phi \in \Phi\}$

The Evidence Lower Bound



$$\log p_\theta(x) \geq \int_z q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} = \mathcal{L}(x;\theta,\phi)$$
$$= \mathcal{L}(x;\theta,\phi) + \mathrm{KL}(q_\phi(z|x) \| p(z|x;\theta))$$

The better $q_\phi(z|x)$ can approximate the posterior $p(z|x;\theta)$, the closer ELBO will be to the $\log p_\theta(x)$. We then jointly optimize over θ and ϕ to maximize the ELBO over a dataset.



$\mathcal{L}(x;\theta,\phi_1)$ and $\mathcal{L}(x;\theta,\phi_2)$ are both lower bounds, we want to jointly optimize θ and ϕ.

ELBO for The Entire Dataset

- For each data point x, ELBO holds

$$\log p_\theta(x) \geq \int_z q_\phi(z|x) \log p_\theta(x, z) + H(q_\phi(z|x)) = \mathcal{L}(x; \theta, \phi)$$

- Maximum likelihood learning over the entire dataset

$$\ell(\theta; \mathcal{D}) = \sum_{x^i \in \mathcal{D}} \log p_\theta(x^i) \geq \sum_{x^i \in \mathcal{D}} \mathcal{L}(x^i; \theta, \phi^i)$$

- Therefore

$$\max_\theta \ell(\theta; \mathcal{D}) \geq \max_{\theta, \phi^1, \cdots, \phi^M} \sum_{i=1}^{M} \mathcal{L}(x^i; \theta, \phi^i)$$

- Note that we use different *variational parameters* ϕ^i for every data point x^i, because the true posterior $p_\theta(z|x^i)$ is different across data points x^i

- Assume $p_\theta(z, x^i)$ is close to $p_{\text{data}}(z, x^i)$. Suppose z captures information such as digit identity (label), style, etc. For simplicity, assume $z \in \{0, 1, \ldots, 9\}$
- Suppose $q_{\phi^i}(z)$ is a probability distribution over the hidden variable z parameterized by $\phi^i = (p_0, \ldots, p_9)$
- If $\phi^i = (0, 0, 0, 1, \ldots, 0)$, is $q_{\phi^i}(z)$ a good approximation of $p_\theta(z|x^1)$(x^1 is the leftmost datapoint)? Yes
- If $\phi^i = (0, 0, 0, 1, \ldots, 0)$, is $q_{\phi^i}(z)$ a good approximation of $p_\theta(z|x^3)$(x^3 is the rightmost datapoint)? No
- For each x^i, need to find a good $\phi^{i,*}$ via optimization, can be expensive

- Optimizing $\sum_{x^i \in \mathcal{D}} \mathcal{L}(x^i; \theta, \phi^i)$ as a function of $\theta, \phi^1, \ldots, \phi^M$ using stochastic gradient ascent

$$L(\mathcal{D}; \theta, \phi^{1:M}) = \sum_{i=1}^{M} \mathbb{E}_{q_{\phi^i}(z^i)} \left(\log p_\theta(x^i, z) - \log q_{\phi^i}(z^i) \right)$$

1. Initialize $\theta, \phi^1, \cdots, \phi^M$
2. Randomly sample a data point x^i from $\mathcal{D}$
3. Optimize $\mathcal{L}(x^i; \theta, \phi^i)$ as a function of ϕ^i, e.g., local gradient update
4. Compute $\nabla_\theta \mathcal{L}(x^i; \theta, \phi^{i,*})$
5. Update θ in the gradient direction. Go to step 2

- How to compute the gradients? Often no close form solution for the expectations. Use Monte Carlo estimates!

$$\mathcal{L}(x;\theta,\phi) = \mathbb{E}_{q_\phi(z)}\left(\log p_\theta(x,z) - \log q_\phi(z)\right)$$

- Similarly as in VI, we assume $q_\phi(z)$ is tractable, i.e., easy to sample from and evaluate
- Suppose $z^1, \ldots, z^k$ are samples from $q_\phi(z)$
- The gradient with respect to θ is easy

$$\begin{aligned}\nabla_\theta \mathcal{L}(x;\theta,\phi) &= \nabla_\theta \mathbb{E}_{q_\phi(z)}\left(\log p_\theta(x,z) - \log q_\phi(z)\right) \\ &= \mathbb{E}_{q_\phi(z)} \nabla_\theta \log p_\theta(x,z) \\ &\approx \frac{1}{k}\sum_{i=1}^{k} \nabla_\theta \log p_\theta(x,z^i)\end{aligned}$$

- The gradient with respect to ϕ is more complicated because the expectation depends on ϕ
- We can use score function estimator (or REINFORCE) with *control variates*. When $q_\phi(z)$ is reparameterizable, we can also use the reparameterization trick.
- If these exists g_ϕ and q_ϵ, s.t. $z = g_\phi(\epsilon), \epsilon \sim q_\epsilon \Rightarrow z \sim q_\phi(z)$

$$\begin{aligned}\nabla_\phi \mathcal{L}(x;\theta,\phi) &= \nabla_\phi \mathbb{E}_{q_\epsilon(\epsilon)} \left(\log p_\theta(x, g_\phi(\epsilon)) - \log q_\phi(g_\phi(\epsilon))\right) \\ &= \mathbb{E}_{q_\epsilon(\epsilon)} \left(\nabla_\phi \log p_\theta(x, g_\phi(\epsilon)) - \nabla_\phi \log q_\phi(g_\phi(\epsilon))\right) \\ &\approx \frac{1}{k} \sum_{i=1}^{k} \left(\nabla_\phi \log p_\theta(x, g_\phi(\epsilon^i)) - \nabla_\phi \log q_\phi(g_\phi(\epsilon^i))\right)\end{aligned}$$

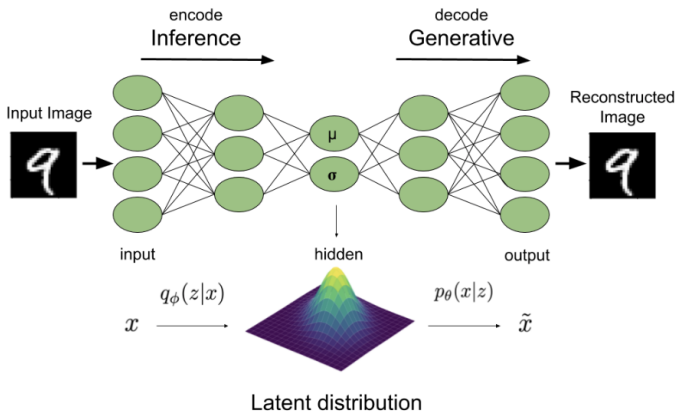
  where $\epsilon^i \sim q_\epsilon(\epsilon), i = 1, \ldots, k$
- Example: $z = \mu + \sigma\epsilon, \epsilon \sim \mathcal{N}(0,1) \Leftrightarrow z \sim \mathcal{N}(\mu, \sigma^2) = q_\phi(z)$

$$\max_{\theta} \ell(\theta; \mathcal{D}) \geq \max_{\theta, \phi^{1:M}} \sum_{i=1}^{M} \mathcal{L}(x^i; \theta, \phi^i)$$

- So far we have used a set of variational parameters ϕ^i for each data point x^i. Unfortunately, this does not scale to large datasets.
- **Amortization**: Learn a single parameteric function f_λ that maps each x to a set of variational parameters. Like doing regression $x^i \mapsto \phi^{i,*}$
  - For example, if $q(z|x^i)$ are Gaussians with different means $\mu^1, \ldots, \mu^m$, we learn a single neural network f_λ mapping x^i to μ^i
- We approximate the posteriors $q(z|x^i)$ using this distribution $q_\lambda(z|x^i)$

- Assume $p_\theta(z, x^i)$ is close to $p_{\text{data}}(z, x^i)$. Suppose z captures information such as digit identity (label), style, etc.
- Suppose $q_{\phi^i}(z)$ is a probability distribution over the hidden variable z parameterized by ϕ^i
- For each x^i, need to find a good $\phi^{i,*}$ via optimization, expensive for large dataset
- Amortized Inference: learn how to map x^i to a good set of parameters ϕ^i via $q(z; f_\lambda(x^i))$. f_λ learns how to solve the optimization problem for you, jointly across all datapoints.
- In the literature, $q(z; f_\lambda(x^i))$ often denoted as $q_\phi(z|x^i)$

Autoencoder Perspective

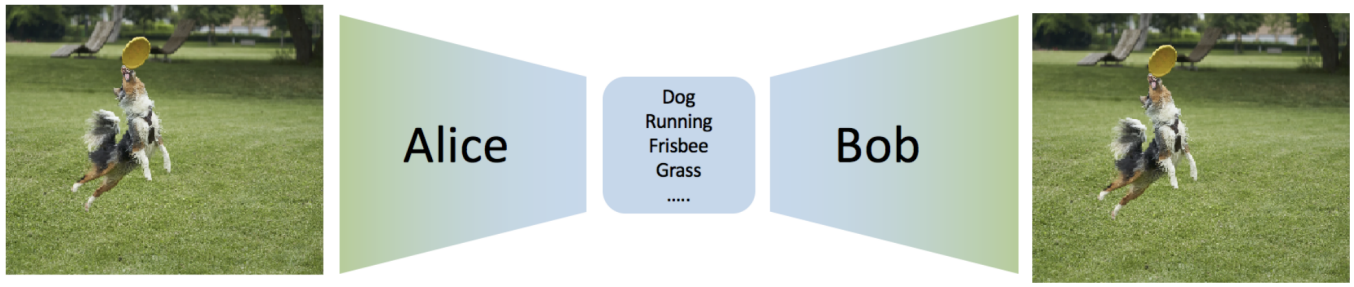


$$\begin{aligned}\mathcal{L}(x;\theta,\phi) &= \mathbb{E}_{q_\phi(z|x)}\left(\log p_\theta(x,z) - \log q_\phi(z|x)\right)\\ &= \mathbb{E}_{q_\phi(z|x)}\left(\log p_\theta(x|z) + \log p(z) - \log q_\phi(z|x)\right)\\ &= \mathbb{E}_{q_\phi(z|x)} \log p(x|z;\theta) - \mathrm{KL}\left(q_\phi(z|x)\|p(z)\right)\end{aligned}$$

Take a data point $x^i \rightarrow$ Map it to $\hat{z}$ by sampling from $q_\phi(z|x^i)$ (encoder) $\rightarrow$ Reconstruct $\hat{x}$ by sampling from $p(x|\hat{z};\theta)$ (decoder)

What does the training objective $\mathcal{L}(x;\theta,\phi)$ do?

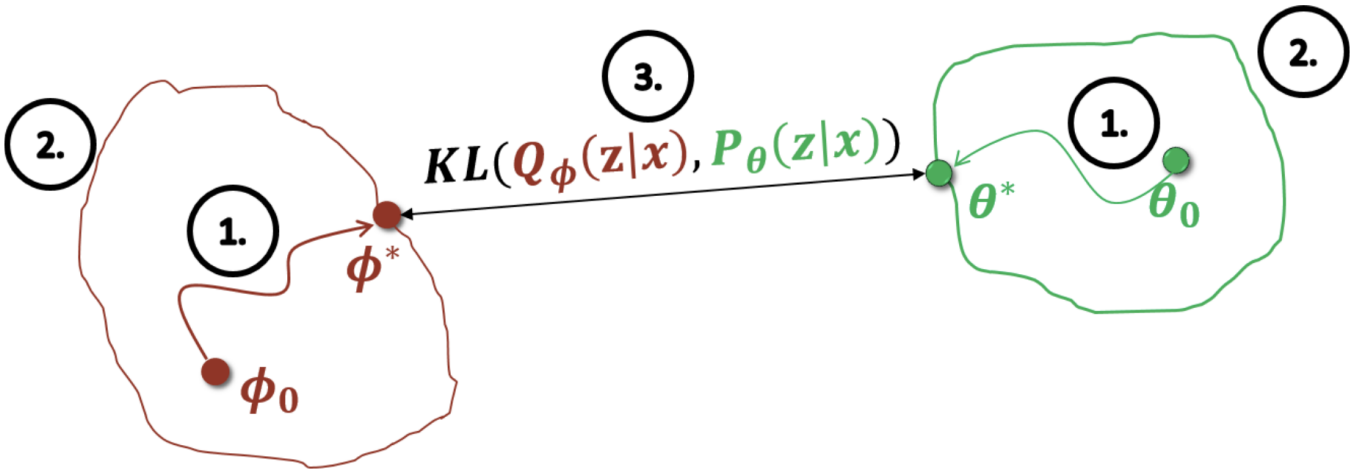
- First term encourages $\hat{x} \approx x^i$ (x^i likely under $p(x|\hat{z};\theta)$)
- Second term encourages $\hat{z}$ to be likely under the prior $p(z)$



- Alice goes on a space mission and needs to send images to Bob. Given an image x^i, she (stochastically) compress it using $\hat{z} \sim q_\phi(z|x^i)$ obtaining a message $\hat{z}$. Alice sends the message $\hat{z}$ to Bob
- Given $\hat{z}$, Bob tries to reconstruct the image using $p_\theta(x|\hat{z})$
  - This scheme works well if $\mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z)$ is large
  - The term $\mathrm{KL}\left(q_\phi(z|x)||p(z)\right)$ forces the distribution over messages to have a specific shape $p(z)$. If Bob knows $p(z)$, he can generate realistic messages $\hat{z} \sim p(z)$ and the corresponding image, as if he had received them from Alice!

Summary on Latent Variable Models

- Combine simple models to get a more flexible one (e.g., mixture of Gaussians)
- Directed model permits ancestral sampling (efficient generation): $z \sim p(z), \quad x \sim p_\theta(x|z)$
- However, log-likelihood is generally intractable, hence learning is difficult (compared to autoregressive models)
- Joint learning of a model (θ) and an amortized inference component ϕ to achieve tractability via ELBO optimization
- Latent representations for any x can be inferred via $q_\phi(z|x)$



Improving variational learning via

- Better optimization techniques
- More expressive approximating families
- Alternate loss functions

Model Families

Amortization (Gershman & Goodman, 2015; Kingma; Rezende;..)

- Scalability: efficient learning and inference on massive datasets
- Regularization effect: due to joint training, it also implicitly regularizes the model θ (Shu et al., 2018)

Augmenting variational posteriors

- Monte Carlo methods: Importance sampling (Burda et al., 2015), MCMC (Salimans et al., 2015, Hoffman, 2017, Levy et al., 2018), Sequential Monte Carlo (Maddison et al., 2017, Le et al., 2018, Naesseth et al., 2018), Rejection sampling (Grover et al., 2018)
- Normalizing flows (Rezende & Mohammed, 2015, Kingma et al., 2016)

Variational Objective

Tighter ELBO does not imply:

- Better samples: sample quality and likelihoods are uncorrelated (Theis et al., 2016)
- Informative latent codes: powerful decoders can ignore latent codes due to tradeoff in minimizing reconstruction error vs KL prior penalty (Bowman et al., 2015, Chen et al., 2016, Zhao et al., 2017, Alemi et al., 2018)

Alternatives to the KL divergence:

- Renyi's alpha-divergences (Li & Turner, 2016)
- Integral probability metrics such as maximum mean discrepancy, Wasserstein distance (Dziugaite et al., 2015, Zhao et al., 2017, Tolstikhin et al., 2018)

References

- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.
- Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In Proceedings of the Cognitive Science Society, volume 36, 2014.
- R. Shu, H. H. Bui, S. Zhao, M. J. Kochenderfer, and S. Ermon. Amortized inference regularization. In Advances in Neural Information Processing Systems, pages 4393–4402, 2018.
- Naesseth, C. A., Linderman, S. W., Ranganath, R., and Blei, D. M. Variational sequential Monte Carlo. In International Conference on Artificial Intelligence and Statistics, 2018.

References

- C.J. Maddison, D. Lawson, G. Tucker, N. Heess, M. Norouzi, A. Mnih, A. Doucet, and Y. Whye Teh. Filtering variational objectives. In Advances in Neural Information Processing Systems, 2017.
- Le, T. A., Igl, M., Rainforth, T., Jin, T., and Wood, F. (2018). Auto-Encoding Sequential Monte Carlo. International Conference on Learning Representations.
- L. Theis, A. v. d. Oord, and M. Bethge. A note on the evaluation of generative models. International Conference on Learning Representations, 2016.
- Zhao, S., Song, J., and Ermon, S. Infovae: Information maximizing variational autoencoders. arXiv preprint arXiv:1706.02262, 2017.