Deep Generative Models

Lecture 7

Roman Isachenko

AI Masters

Autumn, 2022

Recap of previous lecture

Gaussian AR NF

$$\mathbf{x} = g(\mathbf{z}, \boldsymbol{\theta}) \quad \Rightarrow \quad x_j = \sigma_j(\mathbf{x}_{1:j-1}) \cdot z_j + \mu_j(\mathbf{x}_{1:j-1}).$$

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

▶ Sampling is sequential, density estimation is parallel.

▶ Forward KL is a natural loss.

Inverse gaussian AR NF

$$\mathbf{x} = g(\mathbf{z}, \boldsymbol{\theta}) \quad \Rightarrow \quad x_j = \tilde{\sigma}_j(\mathbf{z}_{1:j-1}) \cdot z_j + \tilde{\mu}_j(\mathbf{z}_{1:j-1})$$

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

▶ Sampling is parallel, density estimation is sequential.

▶ Reverse KL is a natural loss.

Kingma D. P. et al. *Improving Variational Inference with Inverse Autoregressive Flow*, 2016

Recap of previous lecture

Let split $\mathbf{x}$ and $\mathbf{z}$ in two parts:

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

Coupling layer

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

Estimating the density takes 1 pass, sampling takes 1 pass!

Jacobian

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

Coupling layer is a special case of autoregressive flow.

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

Recap of previous lecture

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

Theorem

MLE for normalizing flow is equivalent to maximization of ELBO for VAE model with deterministic encoder and decoder:

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

$$q(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) = \delta(\mathbf{z} - f(\mathbf{x}, \boldsymbol{\theta})).$$

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

Outline

Outline

Discrete data vs continuous model

Let our data **y** comes from discrete distribution $\Pi(\mathbf{y})$ and we have continuous model $p(\mathbf{x}|\boldsymbol{\theta}) = \text{NN}(\mathbf{x}, \boldsymbol{\theta})$.

- ▶ Images (and not only images) are discrete data, pixels lie in the integer domain ($\{0, 255\}$).
- ▶ By fitting a continuous density model $p(\mathbf{x}|\boldsymbol{\theta})$ to discrete data $\Pi(\mathbf{y})$, one can produce a degenerate solution with all probability mass on discrete values.

Discrete model

- ▶ Use **discrete** model (e.x. $P(\mathbf{y}|\boldsymbol{\theta}) = \text{Cat}(\boldsymbol{\pi}(\boldsymbol{\theta}))$).
- ▶ Minimize any suitable divergence measure $D(\Pi, P)$.
- ▶ NF works only with continuous data **x** (there are discrete NF, see papers below).
- ▶ If pixel value is not presented in the train data, it won't be predicted.

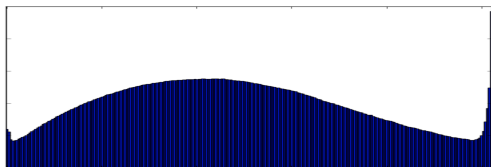Hoogeboom E. et al. Integer discrete flows and lossless compression, 2019
Tran D. et al. Discrete flows: Invertible generative models of discrete data, 2019

Discrete data vs continuous model

Continuous model

▶ Use **continuous** model (e.x. $p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu_\theta}(\mathbf{x}), \boldsymbol{\sigma}_{\boldsymbol{\theta}}^2(\mathbf{x})))$, but
  ▶ **discretize** model (make the model outputs discrete): transform $p(\mathbf{x}|\boldsymbol{\theta})$ to $P(\mathbf{y}|\boldsymbol{\theta})$;
  ▶ **dequantize** data (make the data continuous): transform $\Pi(\mathbf{y})$ to $\pi(\mathbf{x})$.
▶ Continuous distribution know numerical relationships.

CIFAR-10 pixel values distribution

Salimans T. et al. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications, 2017

Outline

Discretization of continuous distribution

Model discretization through CDF

$$F(\mathbf{x}|\boldsymbol{\theta}) = \int_{-\infty}^{\mathbf{x}} p(\mathbf{x}'|\boldsymbol{\theta})d\mathbf{x}'; \quad P(\mathbf{y}|\boldsymbol{\theta}) = F(\mathbf{y}+0.5|\boldsymbol{\theta}) - F(\mathbf{y}-0.5|\boldsymbol{\theta})$$

Mixture of logistic distributions

$$p(x|\mu,s) = \frac{\exp^{-(x-\mu)/s}}{s(1+\exp^{-(x-\mu)/s})^2}; \quad p(x|\boldsymbol{\pi},\boldsymbol{\mu},\mathbf{s}) = \sum_{k=1}^{K} \pi_k p(x|\mu_k,s_k).$$

PixelCNN++

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{j=1}^{m} p(x_j|\mathbf{x}_{1:j-1},\boldsymbol{\theta}); \quad p(x_j|\mathbf{x}_{1:j-1},\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p(x|\mu_k,s_k).$$

Here, $\pi_k = \pi_{k,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1})$, $\mu_k = \mu_{k,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1})$, $s_k = s_{k,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1})$.

For the pixel edge cases of 0, replace $x - 0.5$ by $-\infty$, and for 255 replace $x + 0.5$ by $+\infty$.

Salimans T. et al. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications, 2017
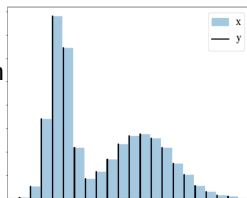
Outline

Uniform discretization

Let dequantize discrete distribution $\Pi(\mathbf{y})$ to continuous distribution $\pi(\mathbf{x})$ in the following way: $\mathbf{x} = \mathbf{y} + \mathbf{u}$, where $\mathbf{u} \sim U[0,1]$.

Theorem

Fitting continuous model $p(\mathbf{x}|\boldsymbol{\theta})$ on uniformly dequantized data is equivalent to maximization of a lower bound on log-likelihood for a discrete model:
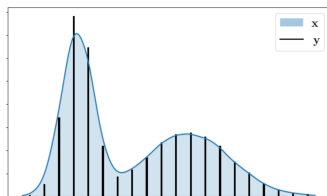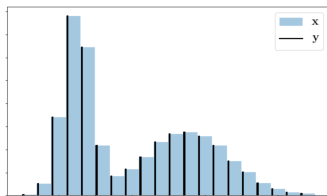

$$P(\mathbf{y}|\boldsymbol{\theta}) = \int_{U[0,1]} p(\mathbf{y} + \mathbf{u}|\boldsymbol{\theta})d\mathbf{u}$$

Proof

$$\mathbb{E}_\pi \log p(\mathbf{x}|\boldsymbol{\theta}) = \int \pi(\mathbf{x}) \log p(\mathbf{x}|\boldsymbol{\theta})d\mathbf{x} = \sum \Pi(\mathbf{y}) \int_{U[0,1]} \log p(\mathbf{y}+\mathbf{u}|\boldsymbol{\theta})d\mathbf{u} \leq$$

$$\leq \sum \Pi(\mathbf{y}) \log \int_{U[0,1]} p(\mathbf{y} + \mathbf{u}|\boldsymbol{\theta})d\mathbf{u} =$$

$$= \sum \Pi(\mathbf{y}) \log P(\mathbf{y}|\boldsymbol{\theta}) = \mathbb{E}_\Pi \log P(\mathbf{y}|\boldsymbol{\theta}).$$

Theis L., Oord A., Bethge M. A note on the evaluation of generative models. 2015

Variational dequantization

- $p(\mathbf{x}|\boldsymbol{\theta})$ assign uniform density to unit hypercubes $\mathbf{y} + U[0, 1]$ (left fig).
- Smooth dequantization is more natural (right fig).
- Neural network density models are smooth function approximators.

Introduce variational dequantization noise distribution $q(\mathbf{u}|\mathbf{y})$, which tells what kind of noise we have to add to our discrete data. Treat it as an approximate posterior as in VAE model.

Variational dequantization

Variational lower bound

$$\log P(\mathbf{y}|\boldsymbol{\theta}) = \left[\log \int q(\mathbf{u}|\mathbf{y}) \frac{p(\mathbf{y}+\mathbf{u}|\boldsymbol{\theta})}{q(\mathbf{u}|\mathbf{y})} d\mathbf{u}\right] \geq$$

$$\geq \int q(\mathbf{u}|\mathbf{y}) \log \frac{p(\mathbf{y}+\mathbf{u}|\boldsymbol{\theta})}{q(\mathbf{u}|\mathbf{y})} d\mathbf{u} = \mathcal{L}(q, \boldsymbol{\theta}).$$

Uniform dequantization is a special case of variational dequantization ($q(\mathbf{u}|\mathbf{y}) = U[0,1]$).

Flow++: flow-based variational dequantization

Let $\mathbf{u} = g(\boldsymbol{\epsilon}, \mathbf{y}, \boldsymbol{\lambda})$ is a flow model with base distribution $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$:

$$q(\mathbf{u}|\mathbf{y}) = p(f(\mathbf{u}, \mathbf{y}, \boldsymbol{\lambda})) \cdot \left|\det \frac{\partial f(\mathbf{u}, \mathbf{y}, \boldsymbol{\lambda})}{\partial \mathbf{u}}\right|.$$

$$\log P(\mathbf{y}|\boldsymbol{\theta}) \geq \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \int p(\boldsymbol{\epsilon}) \log \left(\frac{p(\mathbf{y} + g(\boldsymbol{\epsilon}, \mathbf{y}, \boldsymbol{\lambda})|\boldsymbol{\theta})}{p(\boldsymbol{\epsilon}) \cdot |\det \mathbf{J}_g|^{-1}}\right) d\boldsymbol{\epsilon}.$$

Ho J. et al. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design, 2019

Outline

ELBO surgery

$$\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i(q, \boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\Big[\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)}\log p(\mathbf{x}_i|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z}))\Big].$$

Theorem

$$\frac{1}{n}\sum_{i=1}^{n}KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z})) = KL(q_{\mathrm{agg}}(\mathbf{z})||p(\mathbf{z})) + \mathbb{I}_q[\mathbf{x}, \mathbf{z}];$$

▶ $q_{\mathrm{agg}}(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} q(\mathbf{z}|\mathbf{x}_i)$ – **aggregated** posterior distribution.

▶ $\mathbb{I}_q[\mathbf{x}, \mathbf{z}]$ – mutual information between $\mathbf{x}$ and $\mathbf{z}$ under empirical data distribution and distribution $q(\mathbf{z}|\mathbf{x})$.

▶ First term pushes $q_{\mathrm{agg}}(\mathbf{z})$ towards the prior $p(\mathbf{z})$.

▶ Second term reduces the amount of information about $\mathbf{x}$ stored in $\mathbf{z}$.

Hoffman M. D., Johnson M. J. ELBO surgery: yet another way to carve up the variational evidence lower bound, 2016

ELBO surgery

Theorem

$$\frac{1}{n}\sum_{i=1}^{n} KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z})) = KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) + \mathbb{I}_q[\mathbf{x}, \mathbf{z}].$$

Proof

$$\frac{1}{n}\sum_{i=1}^{n} KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z})) = \frac{1}{n}\sum_{i=1}^{n}\int q(\mathbf{z}|\mathbf{x}_i) \log \frac{q(\mathbf{z}|\mathbf{x}_i)}{p(\mathbf{z})} d\mathbf{z} =$$

$$= \frac{1}{n}\sum_{i=1}^{n}\int q(\mathbf{z}|\mathbf{x}_i) \log \frac{q_{\text{agg}}(\mathbf{z})q(\mathbf{z}|\mathbf{x}_i)}{p(\mathbf{z})q_{\text{agg}}(\mathbf{z})} d\mathbf{z} = \int \frac{1}{n}\sum_{i=1}^{n} q(\mathbf{z}|\mathbf{x}_i) \log \frac{q_{\text{agg}}(\mathbf{z})}{p(\mathbf{z})} d\mathbf{z} +$$

$$+ \frac{1}{n}\sum_{i=1}^{n}\int q(\mathbf{z}|\mathbf{x}_i) \log \frac{q(\mathbf{z}|\mathbf{x}_i)}{q_{\text{agg}}(\mathbf{z})} d\mathbf{z} = KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) + \frac{1}{n}\sum_{i=1}^{n} KL(q(\mathbf{z}|\mathbf{x}_i)||q_{\text{agg}}(\mathbf{z}))$$

Without proof:

$$\mathbb{I}_q[\mathbf{x}, \mathbf{z}] = \frac{1}{n}\sum_{i=1}^{n} KL(q(\mathbf{z}|\mathbf{x}_i)||q_{\text{agg}}(\mathbf{z})) \in [0, \log n].$$

Hoffman M. D., Johnson M. J. ELBO surgery: yet another way to carve up the variational evidence lower bound, 2016

ELBO surgery

ELBO revisiting

$$\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i(q,\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\left[\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)}\log p(\mathbf{x}_i|\mathbf{z},\boldsymbol{\theta}) - KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z}))\right] =$$

$$= \underbrace{\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)}\log p(\mathbf{x}_i|\mathbf{z},\boldsymbol{\theta})}_{\text{Reconstruction loss}} - \underbrace{\mathbb{I}_q[\mathbf{x},\mathbf{z}]}_{\text{MI}} - \underbrace{KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z}))}_{\text{Marginal KL}}$$

Prior distribution $p(\mathbf{z})$ is only in the last term.

Optimal VAE prior

$$KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) = 0 \quad \Leftrightarrow \quad p(\mathbf{z}) = q_{\text{agg}}(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n}q(\mathbf{z}|\mathbf{x}_i).$$

The optimal prior $p(\mathbf{z})$ is the aggregated posterior $q_{\text{agg}}(\mathbf{z})$!

Hoffman M. D., Johnson M. J. ELBO surgery: yet another way to carve up the variational evidence lower bound, 2016

Outline

Outline

VAE limitations

▶ Poor generative distribution (decoder)

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\sigma}_{\boldsymbol{\theta}}^2(\mathbf{z})) \quad \text{or} \; = \text{Softmax}(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{z})).$$

▶ Loose lower bound

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

▶ **Poor prior distribution**
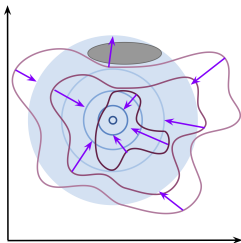
$$p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I}).$$

▶ Poor variational posterior distribution (encoder)

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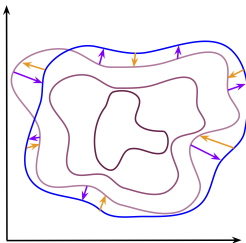
Optimal VAE prior

- Standard Gaussian $p(\mathbf{z}) = \mathcal{N}(0, I) \Rightarrow$ over-regularization;
- $p(\mathbf{z}) = q_{\text{agg}}(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} q(\mathbf{z}|\mathbf{x}_i) \Rightarrow$ overfitting and highly expensive.

Non learnable prior $p(\mathbf{z})$ Learnable prior $p(\mathbf{z}|\boldsymbol{\lambda})$

ELBO revisiting

$$\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_i(q, \boldsymbol{\theta}) = \text{RL} - \text{MI} - KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z}|\boldsymbol{\lambda}))$$

It is Forward KL with respect to $p(\mathbf{z}|\boldsymbol{\lambda})$.

---

image credit: https://jmtomczak.github.io/blog/7/7_priors.html

Flow-based VAE prior

Flow model in latent space

$$\log p(\mathbf{z}|\boldsymbol{\lambda}) = \log p(\mathbf{z}^*) + \log \left| \det \left(\frac{d\mathbf{z}^*}{d\mathbf{z}} \right) \right| = \log p(f(\mathbf{z}, \boldsymbol{\lambda})) + \log |\det(\mathbf{J}_f)|$$

$$\mathbf{z} = g(\mathbf{z}^*, \boldsymbol{\lambda}) = f^{-1}(\mathbf{z}^*, \boldsymbol{\lambda})$$

- ▶ RealNVP with coupling layers.
- ▶ Autoregressive flow (fast $f(\mathbf{z}, \boldsymbol{\lambda})$, slow $g(\mathbf{z}^*, \boldsymbol{\lambda})$).
- ▶ Is it OK to use IAF for VAE prior?

ELBO with flow-based VAE prior

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})} \left[\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \log p(\mathbf{z}|\boldsymbol{\lambda}) - \log q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) \right]$$

$$= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})} \left[\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \underbrace{\left(\log p(f(\mathbf{z}, \boldsymbol{\lambda})) + \log |\det(\mathbf{J}_f)| \right)}_{\text{flow-based prior}} - \log q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) \right]$$

Is it possible to use non-invertible model in VAE prior?

Chen X. et al. Variational Lossy Autoencoder, 2016

Outline

VAE limitations

▶ Poor generative distribution (decoder)

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu_\theta}(\mathbf{z}), \boldsymbol{\sigma}_\theta^2(\mathbf{z})) \quad \text{or} \ = \text{Softmax}(\boldsymbol{\pi_\theta}(\mathbf{z})).$$

▶ Loose lower bound

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

▶ Poor prior distribution

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

▶ **Poor variational posterior distribution (encoder)**

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

Variational posterior

ELBO decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

▶ E-step of EM-algorithm: $KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) = 0$.
  (In this case the lower bound is tight $\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta})$).

▶ $q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\phi}(\mathbf{x}), \boldsymbol{\sigma}^2_{\phi}(\mathbf{x}))$ is a unimodal distribution
  (not expressive enough).

▶ NF convert a simple distribution to a complex one. Let use
  NF in VAE posterior.

Apply a sequence of transformations to the random variable

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

Let $q(\mathbf{z}|\mathbf{x}, \phi)$ (VAE encoder) be a base distribution for a flow
model.

Summary

▶ Lots of data are discrete. We able to discretize the model or to dequantize our data to use continuous model.

▶ Uniform dequantization is the simplest form of dequantization. Variational dequantization is a more natural type that uses variational inference.

▶ The ELBO surgery reveals insights about a prior distribution in VAE. The optimal prior is the aggregated posterior.

▶ We could use flow-based prior in VAE (even autoregressive).

▶ We could use flows to make variational posterior more expressive.