

Deep Generative Models

Lecture 5

Roman Isachenko



AI Masters

2024, Spring

Recap of previous lecture

Forward pass (Loss function)

$$\mathbf{z} = \mathbf{x} + \int_{t_1}^{t_0} \mathbf{f}_{\theta}(\mathbf{z}(t), t) dt, \quad L(\mathbf{z}) = -\log p(\mathbf{x}|\theta)$$

$$L(\mathbf{z}) = -\log p(\mathbf{z}) + \int_{t_0}^{t_1} \text{tr} \left(\frac{\partial \mathbf{f}_{\theta}(\mathbf{z}(t), t)}{\partial \mathbf{z}(t)} \right) dt$$

Adjoint functions

$$\mathbf{a}_{\mathbf{z}}(t) = \frac{\partial L}{\partial \mathbf{z}(t)}; \quad \mathbf{a}_{\theta}(t) = \frac{\partial L}{\partial \theta(t)}.$$

These functions show how the gradient of the loss depends on the hidden state $\mathbf{z}(t)$ and parameters θ .

Theorem (Pontryagin)

$$\frac{d\mathbf{a}_{\mathbf{z}}(t)}{dt} = -\mathbf{a}_{\mathbf{z}}(t)^T \cdot \frac{\partial \mathbf{f}_{\theta}(\mathbf{z}(t), t)}{\partial \mathbf{z}}; \quad \frac{d\mathbf{a}_{\theta}(t)}{dt} = -\mathbf{a}_{\mathbf{z}}(t)^T \cdot \frac{\partial \mathbf{f}_{\theta}(\mathbf{z}(t), t)}{\partial \theta}.$$

Recap of previous lecture

Forward pass

$$\mathbf{z} = \mathbf{z}(t_0) = \int_{t_0}^{t_1} \mathbf{f}_{\theta}(\mathbf{z}(t), t) dt + \mathbf{x} \quad \Rightarrow \quad \text{ODE Solver}$$

Backward pass

$$\left. \begin{aligned} \frac{\partial L}{\partial \theta(t_1)} &= \mathbf{a}_{\theta}(t_1) = - \int_{t_0}^{t_1} \mathbf{a}_{\mathbf{z}}(t)^T \frac{\partial \mathbf{f}_{\theta}(\mathbf{z}(t), t)}{\partial \theta(t)} dt + 0 \\ \frac{\partial L}{\partial \mathbf{z}(t_1)} &= \mathbf{a}_{\mathbf{z}}(t_1) = - \int_{t_0}^{t_1} \mathbf{a}_{\mathbf{z}}(t)^T \frac{\partial \mathbf{f}_{\theta}(\mathbf{z}(t), t)}{\partial \mathbf{z}(t)} dt + \frac{\partial L}{\partial \mathbf{z}(t_0)} \\ \mathbf{z}(t_1) &= - \int_{t_1}^{t_0} \mathbf{f}_{\theta}(\mathbf{z}(t), t) dt + \mathbf{z}_0. \end{aligned} \right\} \Rightarrow \text{ODE Solver}$$

Note: These scary formulas are the standard backprop in the discrete case.

Recap of previous lecture

Bayes theorem

$$p(\mathbf{t}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{t})p(\mathbf{t})}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathbf{t})p(\mathbf{t})}{\int p(\mathbf{x}|\mathbf{t})p(\mathbf{t})d\mathbf{t}}$$

- ▶ \mathbf{x} – observed variables, \mathbf{t} – unobserved variables (latent variables/parameters);
- ▶ $p(\mathbf{x}|\mathbf{t})$ – likelihood;
- ▶ $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{t})p(\mathbf{t})d\mathbf{t}$ – evidence;
- ▶ $p(\mathbf{t})$ – prior distribution, $p(\mathbf{t}|\mathbf{x})$ – posterior distribution.

Posterior distribution

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\theta)p(\theta)}{\int p(\mathbf{X}|\theta)p(\theta)d\theta}$$

Recap of previous lecture

Latent variable models (LVM)

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

MLE problem for LVM

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg \max_{\boldsymbol{\theta}} \log p(\mathbf{X}|\boldsymbol{\theta}) = \arg \max_{\boldsymbol{\theta}} \sum_{i=1}^n \log p(\mathbf{x}_i|\boldsymbol{\theta}) = \\ &= \arg \max_{\boldsymbol{\theta}} \sum_{i=1}^n \log \int p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta})p(\mathbf{z}_i)d\mathbf{z}_i.\end{aligned}$$

Naive Monte-Carlo estimation

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

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

Recap of previous lecture

ELBO derivation 1 (inequality)

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} \geq \mathbb{E}_q \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} = \mathcal{L}(q, \boldsymbol{\theta})$$

ELBO derivation 2 (equality)

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

Variational decomposition

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

Recap of previous lecture

Variational lower Bound (ELBO)

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

$$\mathcal{L}(q, \boldsymbol{\theta}) = \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z}))$$

Log-likelihood decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z})) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

- ▶ Instead of maximizing incomplete likelihood, maximize ELBO

$$\max_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}) \rightarrow \max_{q, \boldsymbol{\theta}} \mathcal{L}(q, \boldsymbol{\theta})$$

- ▶ Maximization of ELBO by variational distribution q is equivalent to minimization of KL

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

Outline

1. EM-algorithm

Amortized inference

ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

3. Data dequantization

Outline

1. EM-algorithm

Amortized inference

ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

3. Data dequantization

EM-algorithm

$$\begin{aligned}\mathcal{L}(q, \theta) &= \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \theta) - KL(q(\mathbf{z})||p(\mathbf{z})) = \\ &= \mathbb{E}_q \left[\log p(\mathbf{x}|\mathbf{z}, \theta) - \log \frac{q(\mathbf{z})}{p(\mathbf{z})} \right] d\mathbf{z} \rightarrow \max_{q, \theta}.\end{aligned}$$

Block-coordinate optimization

- ▶ Initialize θ^* ;
- ▶ **E-step** ($\mathcal{L}(q, \theta) \rightarrow \max_q$)

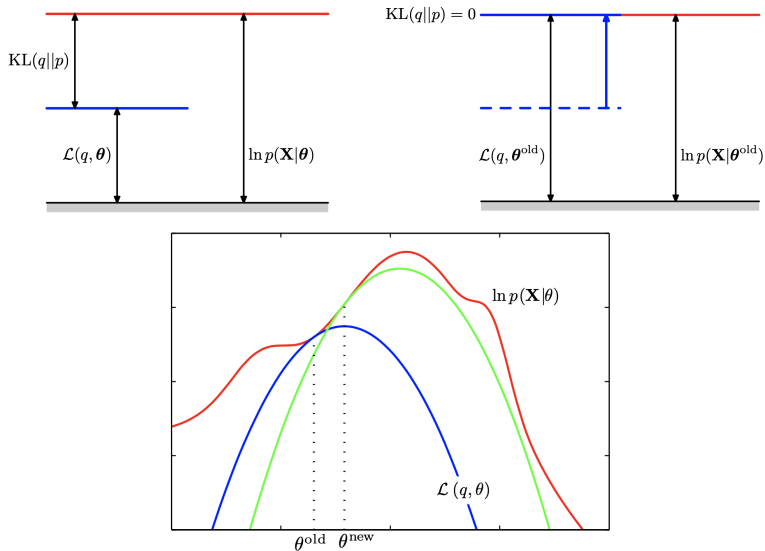
$$\begin{aligned}q^*(\mathbf{z}) &= \arg \max_q \mathcal{L}(q, \theta^*) = \\ &= \arg \min_q KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \theta^*)) = p(\mathbf{z}|\mathbf{x}, \theta^*);\end{aligned}$$

- ▶ **M-step** ($\mathcal{L}(q, \theta) \rightarrow \max_\theta$)

$$\theta^* = \arg \max_\theta \mathcal{L}(q^*, \theta);$$

- ▶ Repeat E-step and M-step until convergence.

EM-algorithm illustration



Outline

1. EM-algorithm

Amortized inference

ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

3. Data dequantization

Amortized variational inference

E-step

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

- ▶ $q(\mathbf{z})$ approximates true posterior distribution $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*)$, that is why it is called **variational posterior**;
- ▶ $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*)$ could be **intractable**;
- ▶ $q(\mathbf{z})$ is different for each object \mathbf{x} .

Idea

Restrict a family of all possible distributions $q(\mathbf{z})$ to a parametric class $q(\mathbf{z}|\mathbf{x}, \phi)$ conditioned on samples \mathbf{x} with parameters ϕ .

Variational Bayes

- ▶ E-step

$$\phi_k = \phi_{k-1} + \eta \cdot \nabla_{\phi} \mathcal{L}(\phi, \boldsymbol{\theta}_{k-1})|_{\phi=\phi_{k-1}}$$

- ▶ M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}(\phi_k, \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}}$$

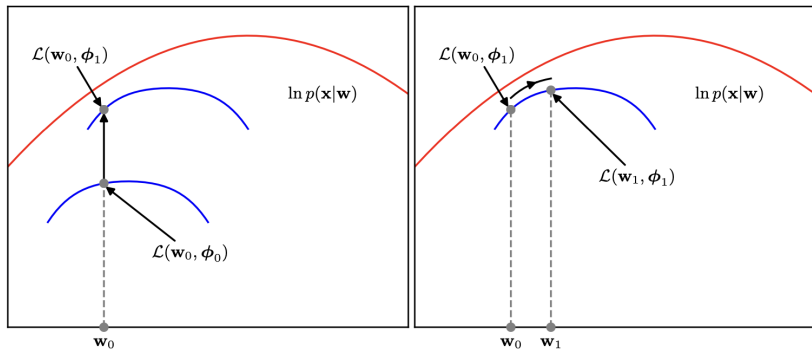
Variational EM illustration

► E-step

$$\phi_k = \phi_{k-1} + \eta \nabla_{\phi} \mathcal{L}(\phi, \theta_{k-1})|_{\phi=\phi_{k-1}}$$

► M-step

$$\theta_k = \theta_{k-1} + \eta \nabla_{\theta} \mathcal{L}(\phi_k, \theta)|_{\theta=\theta_{k-1}}$$



Variational EM-algorithm

ELBO

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

► E-step

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}_{k-1})|_{\boldsymbol{\phi}=\boldsymbol{\phi}_{k-1}},$$

where $\boldsymbol{\phi}$ – parameters of variational posterior distribution $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$.

► M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}_k, \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}},$$

where $\boldsymbol{\theta}$ – parameters of the generative distribution $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

Now all that is left is to obtain gradients: $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$, $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$.

Challenge: Number of samples n could be huge (we need derive the **unbiased** stochastic gradients).

Outline

1. EM-algorithm

Amortized inference

ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

3. Data dequantization

ELBO gradients, (M-step, $\nabla_{\theta} \mathcal{L}(\phi, \theta)$)

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \left[\log p(\mathbf{x}|\mathbf{z}, \theta) - \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z})} \right] \rightarrow \max_{\phi, \theta}.$$

M-step: $\nabla_{\theta} \mathcal{L}(\phi, \theta)$

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

Naive Monte-Carlo estimation

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

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

ELBO gradients, (E-step, $\nabla_{\phi} \mathcal{L}(\phi, \theta)$)

E-step: $\nabla_{\phi} \mathcal{L}(\phi, \theta)$

Difference from M-step: density function $q(\mathbf{z}|\mathbf{x}, \phi)$ depends on the parameters ϕ , it is impossible to use the Monte-Carlo estimation:

$$\begin{aligned}\nabla_{\phi} \mathcal{L}(\phi, \theta) &= \nabla_{\phi} \int q(\mathbf{z}|\mathbf{x}, \phi) \left[\log p(\mathbf{x}|\mathbf{z}, \theta) - \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z})} \right] d\mathbf{z} \\ &\neq \int q(\mathbf{z}|\mathbf{x}, \phi) \nabla_{\phi} \left[\log p(\mathbf{x}|\mathbf{z}, \theta) - \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z})} \right] d\mathbf{z}\end{aligned}$$

Reparametrization trick (LOTUS trick)

- ▶ $r(x) = \mathcal{N}(0, 1)$, $y = \sigma \cdot x + \mu$, $p(y|\theta) = \mathcal{N}(\mu, \sigma^2)$, $\theta = [\mu, \sigma]$.
- ▶ $\epsilon^* \sim r(\epsilon)$, $\mathbf{z} = \mathbf{g}_{\phi}(\mathbf{x}, \epsilon)$, $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi)$

$$\begin{aligned}\nabla_{\phi} \int q(\mathbf{z}|\mathbf{x}, \phi) \mathbf{f}(\mathbf{z}) d\mathbf{z} &= \nabla_{\phi} \int r(\epsilon) \mathbf{f}(\mathbf{z}) d\epsilon \Big|_{\mathbf{z}=\mathbf{g}_{\phi}(\mathbf{x}, \epsilon)} \\ &= \int r(\epsilon) \nabla_{\phi} \mathbf{f}(\mathbf{g}_{\phi}(\mathbf{x}, \epsilon)) d\epsilon \approx \nabla_{\phi} \mathbf{f}(\mathbf{g}_{\phi}(\mathbf{x}, \epsilon^*))\end{aligned}$$

ELBO gradient (E-step, $\nabla_{\phi} \mathcal{L}(\phi, \theta)$)

$$\begin{aligned}\nabla_{\phi} \mathcal{L}(\phi, \theta) &= \nabla_{\phi} \int q(\mathbf{z}|\mathbf{x}, \phi) \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} - \nabla_{\phi} \text{KL}(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z})) \\ &= \int r(\epsilon) \nabla_{\phi} \log p(\mathbf{x}|\mathbf{g}_{\phi}(\mathbf{x}, \epsilon), \theta) d\epsilon - \nabla_{\phi} \text{KL}(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z})) \\ &\approx \nabla_{\phi} \log p(\mathbf{x}|\mathbf{g}_{\phi}(\mathbf{x}, \epsilon^*), \theta) - \nabla_{\phi} \text{KL}(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}))\end{aligned}$$

Variational assumption

$$r(\epsilon) = \mathcal{N}(0, \mathbf{I}); \quad q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mu_{\phi}(\mathbf{x}), \sigma_{\phi}^2(\mathbf{x})).$$

$$\mathbf{z} = \mathbf{g}_{\phi}(\mathbf{x}, \epsilon) = \sigma_{\phi}(\mathbf{x}) \odot \epsilon + \mu_{\phi}(\mathbf{x}).$$

Here $\mu_{\phi}(\cdot), \sigma_{\phi}(\cdot)$ are parameterized functions (outputs of neural network).

- ▶ $p(\mathbf{z})$ – prior distribution on latent variables \mathbf{z} . We could specify any distribution that we want. Let say $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$.
- ▶ $p(\mathbf{x}|\mathbf{z}, \theta)$ – generative distribution. Since it is a parameterized function let it be neural network with parameters θ .

Outline

1. EM-algorithm

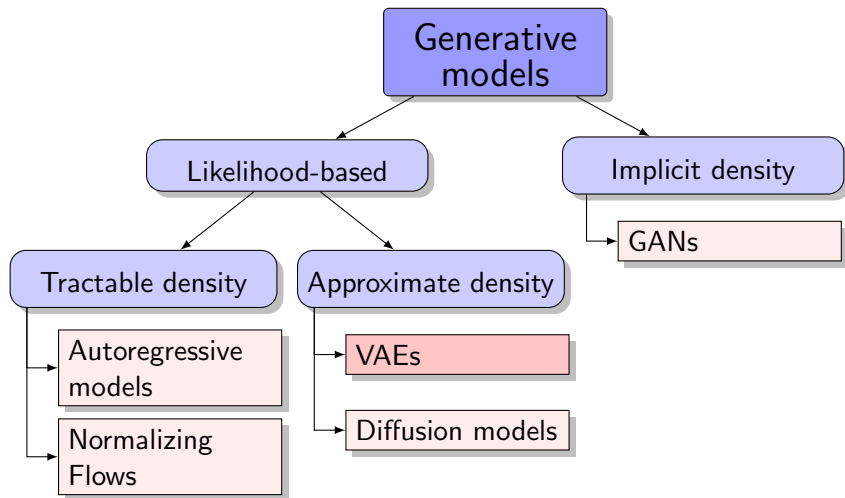
Amortized inference

ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

3. Data dequantization

Generative models zoo



Variational autoencoder (VAE)

Final EM-algorithm

- ▶ pick random sample $\mathbf{x}_i, i \sim U[1, n]$.
- ▶ compute the objective:

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

$$\mathcal{L}(\phi, \theta) \approx \log p(\mathbf{x}|\mathbf{z}^*, \theta) - KL(q(\mathbf{z}^*|\mathbf{x}, \phi)||p(\mathbf{z}^*)).$$

- ▶ compute a stochastic gradients w.r.t. ϕ and θ

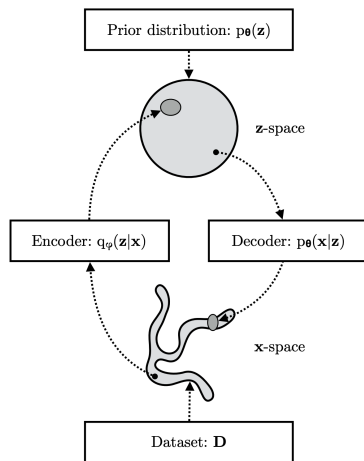
$$\begin{aligned}\nabla_\phi \mathcal{L}(\phi, \theta) &\approx \nabla_\phi \log p(\mathbf{x}|\mathbf{g}_\phi(\mathbf{x}, \epsilon^*), \theta) - \nabla_\phi KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z})); \\ \nabla_\theta \mathcal{L}(\phi, \theta) &\approx \nabla_\theta \log p(\mathbf{x}|\mathbf{z}^*, \theta).\end{aligned}$$

- ▶ update θ, ϕ according to the selected optimization method (SGD, Adam, etc):

$$\begin{aligned}\phi &:= \phi + \eta \cdot \nabla_\phi \mathcal{L}(\phi, \theta), \\ \theta &:= \theta + \eta \cdot \nabla_\theta \mathcal{L}(\phi, \theta).\end{aligned}$$

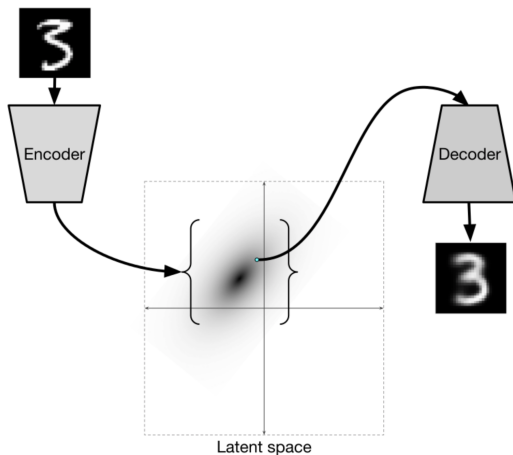
Variational autoencoder (VAE)

- ▶ VAE learns stochastic mapping between \mathbf{x} -space, from complicated distribution $\pi(\mathbf{x})$, and a latent \mathbf{z} -space, with simple distribution.
- ▶ The generative model learns a joint distribution $p(\mathbf{x}, \mathbf{z}|\theta) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}, \theta)$, with a prior distribution $p(\mathbf{z})$, and a stochastic decoder $p(\mathbf{x}|\mathbf{z}, \theta)$.
- ▶ The stochastic encoder $q(\mathbf{z}|\mathbf{x}, \phi)$ (inference model), approximates the true but intractable posterior $p(\mathbf{z}|\mathbf{x}, \theta)$ of the generative model.



Variational Autoencoder

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \left[\log p(\mathbf{x}|\mathbf{z}, \theta) - \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z})} \right] \rightarrow \max_{\phi, \theta}.$$



Variational autoencoder (VAE)

- ▶ Encoder $q(\mathbf{z}|\mathbf{x}, \phi) = \text{NN}_e(\mathbf{x}, \phi)$ outputs $\mu_\phi(\mathbf{x})$ and $\sigma_\phi(\mathbf{x})$.
- ▶ Decoder $p(\mathbf{x}|\mathbf{z}, \theta) = \text{NN}_d(\mathbf{z}, \theta)$ outputs parameters of the sample distribution.

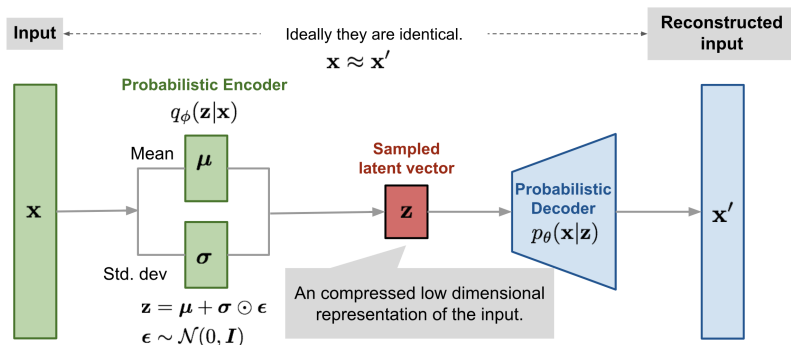


image credit:

<https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html>

Outline

1. EM-algorithm

Amortized inference

ELBO gradients, reparametrization trick

2. Variational autoencoder (VAE)

3. Data dequantization

Discrete data vs continuous model

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

- ▶ Images (and not only images) are discrete data, pixels lie in the integer domain $\{0, 255\}$.
- ▶ By fitting a continuous density model $p(\mathbf{x}|\theta)$ 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. $P(\mathbf{y}|\theta) = \text{Cat}(\pi(\theta))$).
- ▶ Minimize any suitable divergence measure $D(\Pi, P)$.
- ▶ NF works only with continuous data \mathbf{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

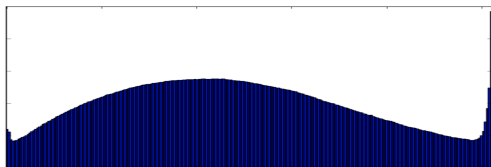
Tran D. et al. Discrete flows: Invertible generative models of discrete data, 2019

Discrete data vs continuous model

Continuous model

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

CIFAR-10 pixel values distribution



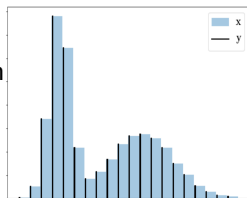
Uniform dequantization

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

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



Proof

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

Summary

- ▶ The general variational EM algorithm maximizes ELBO objective for LVM model to find MLE for parameters θ .
- ▶ Amortized variational inference allows to efficiently compute the stochastic gradients for ELBO using Monte-Carlo estimation.
- ▶ The reparametrization trick gets unbiased gradients w.r.t to the variational posterior distribution $q(\mathbf{z}|\mathbf{x}, \phi)$.
- ▶ The VAE model is an LVM with two neural network: stochastic encoder $q(\mathbf{z}|\mathbf{x}, \phi)$ and stochastic decoder $p(\mathbf{x}|\mathbf{z}, \theta)$.
- ▶ Lots of data are discrete. We able to discretize the model or to dequantize our data to use continuous model.
- ▶ Uniform dequantization helps to make discrete data continuous. It gives us lower bound on the log-likelihood.