

# Deep Generative Models

## Lecture 3

Roman Isachenko

Moscow Institute of Physics and Technology

Autumn, 2021

# Recap of previous lecture

## MLE problem for autoregressive model

$$\theta^* = \arg \max_{\theta} p(\mathbf{X}|\theta) = \arg \max_{\theta} \sum_{i=1}^n \sum_{j=1}^m \log p(x_{ij}|\mathbf{x}_{i,1:j-1}\theta).$$

## Sampling

$$\hat{x}_1 \sim p(x_1|\theta), \quad \hat{x}_2 \sim p(x_2|\hat{x}_1, \theta), \dots, \quad \hat{x}_m \sim p(x_m|\hat{\mathbf{x}}_{1:m-1}, \theta)$$

New generated object is  $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m)$ .

Masking helps to make neural network autoregressive.

- ▶ **MADE** - masked autoencoder (MLP).
- ▶ **WaveNet** - masked 1D convolutions.
- ▶ **PixelCNN** - masked 2D convolutions.

**PixelCNN++** uses discretized mixture of logistic distribution to make the output distribution more natural.

# Recap of previous lecture

## Posterior distribution

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

## Bayesian inference

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

## Maximum a posteriori (MAP) estimation

$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{X}) = \arg \max_{\boldsymbol{\theta}} (\log p(\mathbf{X}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}))$$

## MAP inference

$$p(\mathbf{x}|\mathbf{X}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta} \approx p(\mathbf{x}|\boldsymbol{\theta}^*).$$

# Latent variable models (LVM)

## MLE problem

$$\theta^* = \arg \max_{\theta} p(\mathbf{X}|\theta) = \arg \max_{\theta} \prod_{i=1}^n p(\mathbf{x}_i|\theta) = \arg \max_{\theta} \sum_{i=1}^n \log p(\mathbf{x}_i|\theta).$$

The distribution  $p(\mathbf{x}|\theta)$  could be very complex and intractable (as well as real distribution  $\pi(\mathbf{x})$ ).

## Extended probabilistic model

Introduce latent variable  $\mathbf{z}$  for each sample  $\mathbf{x}$

$$p(\mathbf{x}, \mathbf{z}|\theta) = p(\mathbf{x}|\mathbf{z}, \theta)p(\mathbf{z}); \quad \log p(\mathbf{x}, \mathbf{z}|\theta) = \log p(\mathbf{x}|\mathbf{z}, \theta) + \log p(\mathbf{z}).$$

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

## Motivation

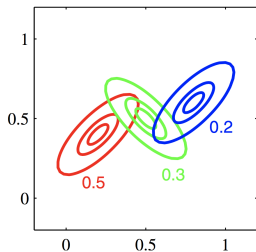
The distributions  $p(\mathbf{x}|\mathbf{z}, \theta)$  and  $p(\mathbf{z})$  could be quite simple.

# Latent variable models (LVM)

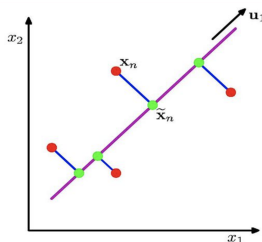
$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log \int p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})d\mathbf{z} \rightarrow \max_{\boldsymbol{\theta}}$$

## Examples

*Mixture of gaussians*



*PCA model*

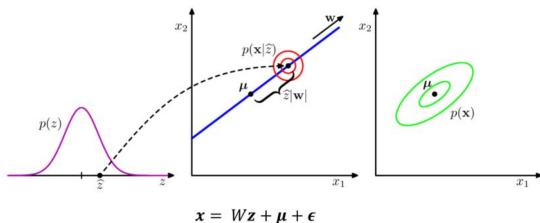


- ▶  $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{z}})$
- ▶  $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$
- ▶  $p(\mathbf{z}) = \text{Categorical}(\boldsymbol{\pi})$
- ▶  $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$

# Latent variable models (LVM)

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

**PCA** projects original data  $\mathbf{X}$  onto a low dimensional latent space while maximizing the variance of the projected data.



- ▶  $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$
- ▶  $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \mathbf{I})$
- ▶  $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})$
- ▶  $p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}), \sigma^2\mathbf{M}), \text{ where } \mathbf{M} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$

# Maximum likelihood estimation for LVM

## MLE for extended problem

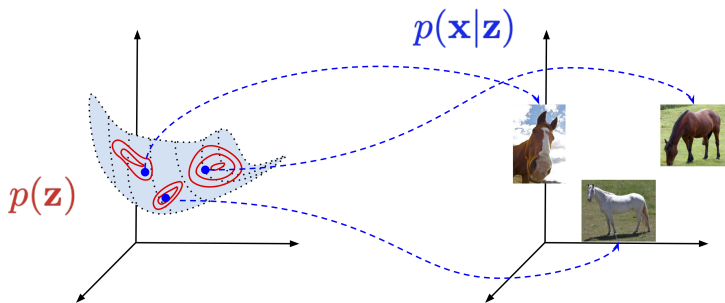
$$\begin{aligned}\theta^* &= \arg \max_{\theta} p(\mathbf{X}, \mathbf{Z} | \theta) = \arg \max_{\theta} \prod_{i=1}^n p(\mathbf{x}_i, \mathbf{z}_i | \theta) = \\ &= \arg \max_{\theta} \sum_{i=1}^n \log p(\mathbf{x}_i, \mathbf{z}_i | \theta).\end{aligned}$$

However,  $\mathbf{Z}$  is unknown.

## MLE for original problem

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

???



Naive approach: Monte-Carlo estimation

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



# Variational lower bound

## Derivation 1

$$\begin{aligned}\log p(\mathbf{x}|\boldsymbol{\theta}) &= \log \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} = \\ &= \log \int \frac{q(\mathbf{z})}{q(\mathbf{z})} p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} = \log \mathbb{E}_q \left[ \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} \right] \geq \\ &\geq \mathbb{E}_q \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} = \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \mathcal{L}(q, \boldsymbol{\theta})\end{aligned}$$

## Derivation 2

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

## Variational lower bound

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

## Evidence Lower Bound (ELBO)

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

Instead of maximizing incomplete likelihood, maximize ELBO (equivalently minimize KL)

$$\max_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}) \quad \rightarrow \quad \max_{q, \boldsymbol{\theta}} \mathcal{L}(q, \boldsymbol{\theta}) \equiv \min_{q, \boldsymbol{\theta}} KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

# EM-algorithm

$$\mathcal{L}(q, \theta) = \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} + \int q(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z}.$$

## Block-coordinate optimization

- ▶ Initialize  $\theta^*$ ;
- ▶ E-step

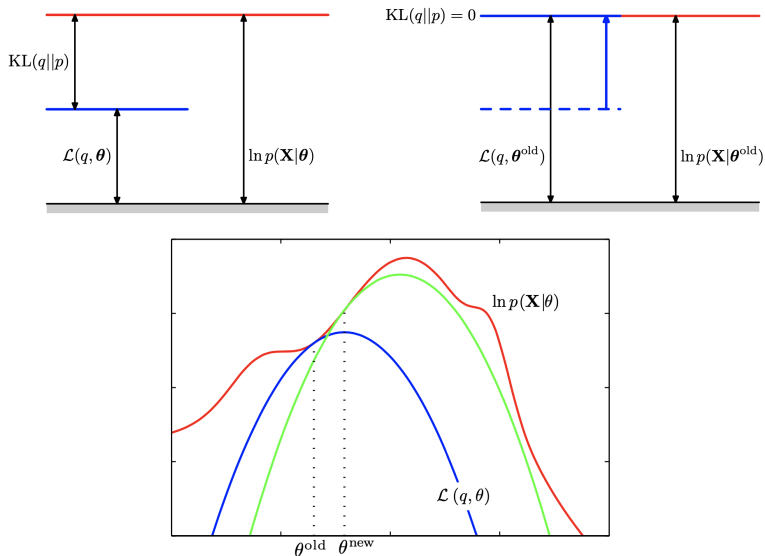
$$\begin{aligned} q(\mathbf{z}) &= \arg \max_q \mathcal{L}(q, \theta^*) = \\ &= \arg \min_q KL(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x}, \theta^*)) = p(\mathbf{z}|\mathbf{x}, \theta^*); \end{aligned}$$

- ▶ M-step

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

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

# EM illustration



# Amortized variational inference

## E-step

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

- ▶  $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*)$  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 particular parametric class  $q(\mathbf{z}|\mathbf{x}, \phi)$  conditioned on samples  $\mathbf{x}$  with parameters  $\phi$ .

## Variational Bayes

- ▶ E-step

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

- ▶ M-step

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

# 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

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

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

► M-step

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

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

Now all we have to do is to obtain two 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 to derive unbiased stochastic gradients).

## ELBO interpretations

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

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

- Evidence minus posterior KL

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

- Average negative energy plus entropy

$$\begin{aligned}\mathcal{L}(q, \boldsymbol{\theta}) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} [\log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) - \log q(\mathbf{z}|\mathbf{x}, \phi)] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) + \mathbb{H}[q(\mathbf{z}|\mathbf{x}, \phi)].\end{aligned}$$

- Average reconstruction minus KL to prior

$$\begin{aligned}\mathcal{L}(q, \boldsymbol{\theta}) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} [\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \log p(\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x}, \phi)] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z})).\end{aligned}$$

## Monte-Carlo estimation

$$\sum_{i=1}^n \mathbb{E}_q f(\mathbf{z}_i) \approx n \cdot \mathbb{E}_q f(\mathbf{z}) = n \cdot \int q(\mathbf{z}) f(\mathbf{z}) d\mathbf{z} \approx n \cdot f(\mathbf{z}^*), \text{ where } \mathbf{z}^* \sim q(\mathbf{z}).$$

## ELBO gradients

$$\nabla_{\theta} \sum_{i=1}^n \mathcal{L}_i(\phi, \theta) \approx n \cdot \nabla_{\theta} \mathcal{L}(\phi, \theta); \quad \nabla_{\phi} \sum_{i=1}^n \mathcal{L}_i(\phi, \theta) \approx n \cdot \nabla_{\phi} \mathcal{L}(\phi, \theta)$$

## ELBO

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

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

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



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

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

### Challenge

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

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

### Solution

Reparametrization trick for  $q(\mathbf{z}|\mathbf{x}, \phi)$  allows the expectation to become independent of parameters  $\phi$ .

## Reparametrization trick

$$f(\xi) = \mathbb{E}_{q(\eta|\xi)} h(\eta) = \int q(\eta|\xi) h(\eta) d\eta$$

Let  $\eta = g(\xi, \epsilon)$ , where  $g$  is a deterministic function,  $\epsilon$  is a random variable with a density function  $r(\epsilon)$ .

$$f(\xi) = \int q(\eta|\xi) h(\eta) d\eta = \int r(\epsilon) h(g(\xi, \epsilon)) d\epsilon \approx h(g(\xi, \epsilon^*)), \quad \epsilon^* \sim r(\epsilon).$$

### Examples

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

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

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

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

### Variational assumption

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

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

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

If we could calculate  $\log p(\mathbf{x}, \mathbf{z}|\theta)$  and  $\log q(\mathbf{z}|\mathbf{x}, \phi)$ , we are done. Could we?

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

$$\nabla_{\phi} \mathcal{L}(\phi, \theta) \approx \nabla_{\phi} [\log p(\mathbf{x}, g(\mathbf{x}, \epsilon^*, \phi) | \theta) - \log q(g(\mathbf{x}, \epsilon^*, \phi) | \mathbf{x}, \phi)]$$

### First term

$$\log p(\mathbf{x}, \mathbf{z} | \theta) = \log p(\mathbf{x} | \mathbf{z}, \theta) + \log p(\mathbf{z}).$$

- ▶  $p(\mathbf{z})$  – 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  $\theta$ .

### Second term

Function  $\mathbf{z} = g(\mathbf{x}, \epsilon, \phi) = \sigma_{\phi}(\mathbf{x}) \cdot \epsilon + \mu_{\phi}(\mathbf{x})$  is invertible.

$$q(\mathbf{z} | \mathbf{x}, \phi) = r(\epsilon) \cdot \left| \frac{\partial \epsilon}{\partial \mathbf{z}} \right| \Rightarrow \log q(\mathbf{z} | \mathbf{x}, \phi) = \log r(\epsilon) - \sum_{i=1}^d \log [\sigma_{\phi}(\mathbf{x})]_i$$

# Variational autoencoder (VAE)

## Final algorithm

- ▶ pick  $i \sim U[1, n]$ ;
- ▶ compute a stochastic gradient w.r.t.  $\phi$

$$\nabla_{\phi} \mathcal{L}(\phi, \theta) \approx \nabla_{\phi} [\log p(\mathbf{x}, g(\mathbf{x}, \epsilon^*, \phi) | \theta) - \log q(g(\mathbf{x}, \epsilon^*, \phi) | \mathbf{x}, \phi)], \quad \epsilon^* \sim r(\epsilon);$$

- ▶ compute a stochastic gradient w.r.t.  $\theta$

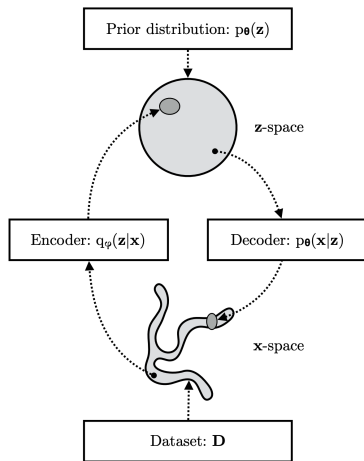
$$\nabla_{\theta} \mathcal{L}(\phi, \theta) \approx \nabla_{\theta} \log p(\mathbf{x} | \mathbf{z}^*, \theta), \quad \mathbf{z}^* \sim q(\mathbf{z} | \mathbf{x}, \phi);$$

- ▶ update  $\theta, \phi$  according to the selected optimization method (SGD, Adam, RMSProp):

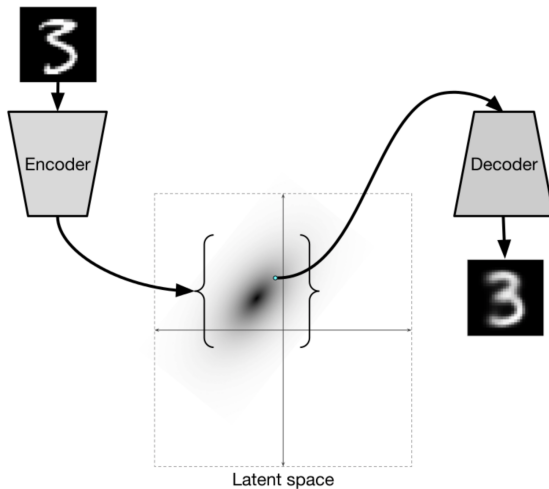
$$\begin{aligned}\phi &:= \phi + \eta \nabla_{\phi} \mathcal{L}(\phi, \theta), \\ \theta &:= \theta + \eta \nabla_{\theta} \mathcal{L}(\phi, \theta).\end{aligned}$$

# 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



# 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.

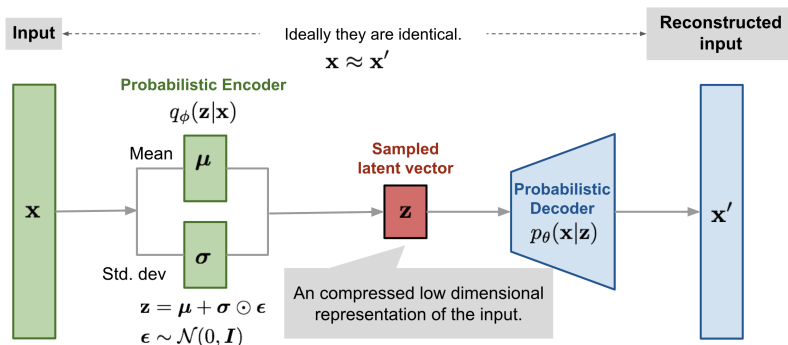


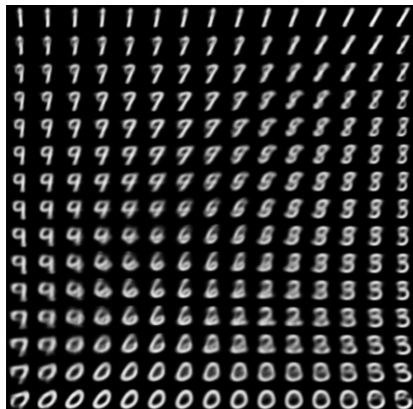
image credit:

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



# Variational Autoencoder

Generated images for latent objects  $\mathbf{z}$  sampled from prior  $\mathcal{N}(\mathbf{0}, \mathbf{I})$



# Bayesian framework

## Posterior distribution

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

## Bayesian inference

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

## Maximum a posteriori (MAP) estimation

$$\theta^* = \arg \max_{\theta} \log p(\theta|\mathbf{X}) = \arg \max_{\theta} (\log p(\mathbf{X}|\theta) + \log p(\theta))$$

## MAP inference

$$p(\mathbf{x}|\mathbf{X}) = \int p(\mathbf{x}|\theta)p(\theta|\mathbf{X})d\theta = \int p(\mathbf{x}|\theta)\delta(\theta - \theta^*)d\theta \approx p(\mathbf{x}|\theta^*).$$

# VAE as Bayesian model

## Posterior distribution

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

## ELBO

$$\begin{aligned}\log p(\theta|\mathbf{X}) &= \log p(\mathbf{X}|\theta) + \log p(\theta) - \log p(\mathbf{X}) \\ &= \mathcal{L}(q, \theta) + KL(q||p) + \log p(\theta) - \log p(\mathbf{X}) \\ &\geq [\mathcal{L}(q, \theta) + \log p(\theta)] - \log p(\mathbf{X}).\end{aligned}$$

## EM-algorithm

### ► E-step

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

### ► M-step

$$\theta^* = \arg \max_{\theta} [\mathcal{L}(q, \theta) + \log p(\theta)].$$

# VAE limitations

- ▶ Poor variational posterior distribution (inference model encoder)

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

- ▶ Poor prior distribution

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

- ▶ Poor probabilistic model (generative model, decoder)

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

- ▶ Loose lower bound

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

# Summary

- ▶ LVM introduce latent representation of observed samples to make model more interpretable.
- ▶ LVM maximizes variational evidence lower bound to find MLE of model parameters.
- ▶ ELBO maximization is performed by the general variational EM algorithm.
- ▶ Amortized inference allows to efficiently compute stochastic gradients for ELBO and to use deep neural networks for  $q(\mathbf{z}|\mathbf{x}, \phi)$  and  $p(\mathbf{x}|\mathbf{z}, \theta)$ .
- ▶ ELBO gradients are computed using Monte-Carlo estimation.
- ▶ The reparametrization trick allows to get unbiased gradients w.r.t to a variational posterior distribution.
- ▶ The VAE model is an LVM with an encoder network for  $q(\mathbf{z}|\mathbf{x}, \phi)$  and a decoder network for  $p(\mathbf{x}|\mathbf{z}, \theta)$ .
- ▶ VAE is not a "true" bayesian model since parameters  $\theta$  do not have a prior distribution.
- ▶ Standart VAE has several limitations that we will address later in the course.