

# Deep Generative Models

## Lecture 3

Roman Isachenko

 Ozon Masters

Spring, 2022

# 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

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

# 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

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

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

# Outline

1. Variational lower bound (ELBO)
2. EM-algorithm, amortized inference
3. ELBO gradients, reparametrization trick

# Outline

1. Variational lower bound (ELBO)
2. EM-algorithm, amortized inference
3. ELBO gradients, reparametrization trick

# Variational lower bound (ELBO)

## Derivation 1

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

## Derivation 2

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

## Variational lower bound (ELBO)

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

### Log-likelihood decomposition

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

- Instead of maximizing incomplete likelihood, maximize ELBO

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

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

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



# Outline

1. Variational lower bound (ELBO)
2. EM-algorithm, amortized inference
3. ELBO gradients, reparametrization trick

# 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

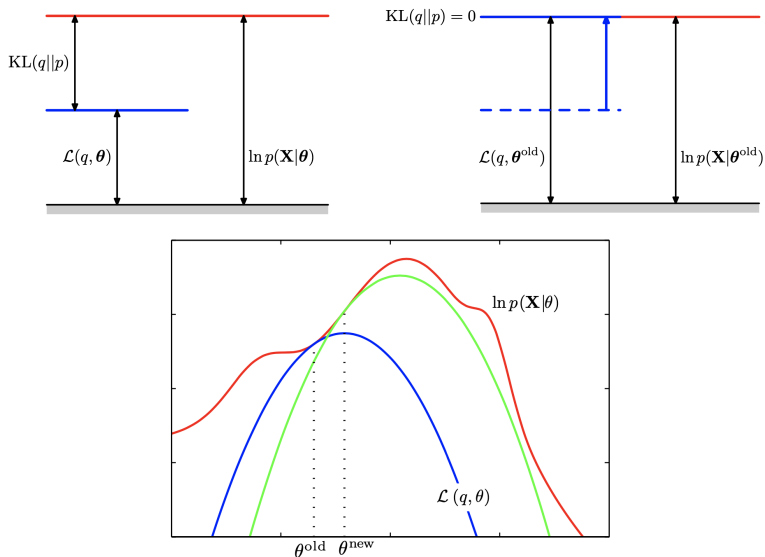
$$\begin{aligned} q^*(\mathbf{z}) &= \arg \max_q \mathcal{L}(q, \theta^*) = \\ &= \arg \min_q KL(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x}, \theta^*)) = p(\mathbf{z}|\mathbf{x}, \theta^*); \end{aligned}$$

- ▶ 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 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).

# Outline

1. Variational lower bound (ELBO)
2. EM-algorithm, amortized inference
3. ELBO gradients, reparametrization trick

# ELBO gradients

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

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

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

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

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:

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

## ELBO gradients

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

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

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

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

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:

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



# Reparametrization trick

## Law of the unconscious statistician (LOTUS)

Let  $X$  be a random variable and let  $Y = g(X)$ . Then

$$\mathbb{E}_{p_Y} Y = \mathbb{E}_{p_X} g(X) = \int g(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}.$$

## Examples

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

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

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

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

### Variational assumption

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

$$\mathbf{z} = g(\mathbf{x}, \epsilon, \phi) = \boldsymbol{\sigma}_{\phi}(\mathbf{x}) \cdot \epsilon + \boldsymbol{\mu}_{\phi}(\mathbf{x}).$$

Here  $\boldsymbol{\mu}_{\phi}(\cdot)$ ,  $\boldsymbol{\sigma}_{\phi}(\cdot)$  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  $\theta$ .

# Summary

- ▶ LVM maximizes variational evidence lower bound (ELBO) to find MLE for the parameters.
- ▶ The general variational EM algorithm maximizes ELBO objective.
- ▶ Amortized inference allows to efficiently compute stochastic gradients for ELBO using Monte-Carlo estimation.
- ▶ The reparametrization trick gets unbiased gradients w.r.t to a variational posterior distribution.