Deep Generative Models

Lecture 3

Roman Isachenko

Moscow Institute of Physics and Technology

2022

Recap of previous lecture

MLE problem for autoregressive model

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

Sampling

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

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

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

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. EM-algorithm, amortized inference

2. ELBO gradients, reparametrization trick

Outline

1. EM-algorithm, amortized inference
2. ELBO gradients, reparametrization trick

EM-algorithm

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

Block-coordinate optimization

- Initialize $\boldsymbol{\theta}^*$;
- E-step

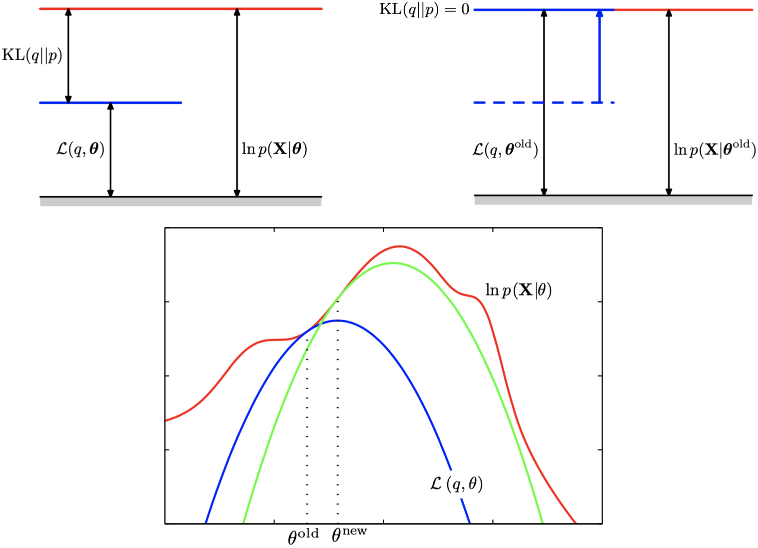
$$q^*(\mathbf{z}) = \arg\max_q \mathcal{L}(q, \boldsymbol{\theta}^*) =$$
$$= \arg\min_q KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*)) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*);$$

- M-step

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

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

EM illustration



Bishop C. Pattern Recognition and Machine Learning, 2006

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 $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ conditioned on samples $\mathbf{x}$ with parameters $\boldsymbol{\phi}$.

Variational Bayes

- E-step

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

- M-step

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

Variational EM-algorithm

ELBO

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

- **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 ϕ – parameters of variational posterior distribution $q(\mathbf{z}|\mathbf{x}, \phi)$.
- **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 $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\phi, \boldsymbol{\theta})$, $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\phi, \boldsymbol{\theta})$.
Challenge: Number of samples n could be huge (we heed to derive unbiased stochastic gradients).

Outline

1. EM-algorithm, amortized inference

2. ELBO gradients, reparametrization trick

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

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

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

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

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

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

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

---

image credit: https://jmtomczak.github.io/blog/4/4_VAE.html

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

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

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

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

Reparametrization trick

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

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

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

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

$$= \int r(\epsilon) \nabla_\phi \log p(\mathbf{x}|g(\mathbf{x}, \epsilon, \phi), \boldsymbol{\theta}) d\epsilon - \nabla_\phi \mathrm{KL}(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}))$$

$$\approx \nabla_\phi \log p(\mathbf{x}|g(\mathbf{x}, \epsilon^*, \phi), \boldsymbol{\theta}) - \nabla_\phi \mathrm{KL}(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}))$$

Variational assumption

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

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

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 $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$.
- $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ - generative distibution. Since it is a parameterized function let it be neural network with parameters $\boldsymbol{\theta}$.

Summary

- The general variational EM algorithm maximizes ELBO objective.
- 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.