Deep Generative Models

Lecture 4

Roman Isachenko

AI Masters

Autumn, 2022

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
$$\max_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}) \quad \rightarrow \quad \max_{q, \boldsymbol{\theta}} \mathcal{L}(q, \boldsymbol{\theta})$$
- Maximization of ELBO by variational distribution q is equivalent to minimization of KL
$$\arg\max_q \mathcal{L}(q, \boldsymbol{\theta}) \equiv \arg\min_q KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

Recap of previous lecture

EM-algorithm

- E-step

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

- M-step

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

Amortized variational inference

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

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

Recap of previous lecture

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

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

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

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

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

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

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

Variational assumption

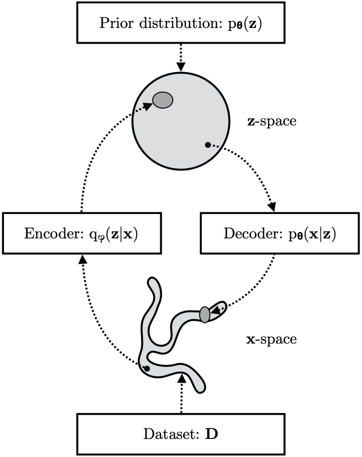
$$r(\boldsymbol{\epsilon}) = \mathcal{N}(0, \mathbf{I}); \quad q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\boldsymbol{\mu}_{\phi}(\mathbf{x}), \boldsymbol{\sigma}^2_{\phi}(\mathbf{x})).$$

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

Recap of previous lecture

Variational autoencoder (VAE)

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



Kingma D. P., Welling M. An introduction to variational autoencoders, 2019

Outline

1. VAE limitations
2. Posterior collapse and decoder weakening techniques
3. Tighter variational bound
4. Normalizing flows

Outline

1. VAE limitations

2. Posterior collapse and decoder weakening techniques

3. Tighter variational bound

4. Normalizing flows

VAE limitations

- Poor generative distribution (decoder)

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

- 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)

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

Outline

1. VAE limitations

2. Posterior collapse and decoder weakening techniques

3. Tighter variational bound

4. Normalizing flows

VAE limitations

- **Poor generative distribution (decoder)**

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

- 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)

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

Posterior collapse

LVM

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

ELBO objective

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

More powerful $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ leads to more powerful generative model $p(\mathbf{x}|\boldsymbol{\theta})$.

Extreme cast

$$p(\mathbf{x}|\boldsymbol{\theta}) \in \mathcal{P} = \{p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})| \, \forall \mathbf{z}, \boldsymbol{\theta}\}.$$

If the decoder $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ is too powerful (it could model $p(\mathbf{x}|\boldsymbol{\theta})$), then ELBO avoids paying any cost $KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}))$ ($q(\mathbf{z}|\mathbf{x}, \phi) \approx p(\mathbf{z})$), the variational posterior $q(\mathbf{z}|\mathbf{x}, \phi)$ will not carry any information about $\mathbf{x}$, the latent variables $\mathbf{z}$ becomes irrelevant.

Autoregressive VAE decoder

How to make the generative model $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ more powerful?

PixelVAE/VLAE

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \prod_{j=1}^{m} p(x_j|\mathbf{x}_{1:j-1}, \mathbf{z}, \boldsymbol{\theta})$$

- Global structure is captured by latent variables $\mathbf{z}$.
- Local statistics are captured by limited receptive field of autoregressive context $\mathbf{x}_{1:j-1}$.

PixelVAE/VLAE models use the autoregressive PixelCNN decoder model with small number of layers to limit receptive field.

---

https://arxiv.org/abs/1611.05013Gulrajani I. et al. PixelVAE: A Latent Variable Model for Natural Images, 2016,
https://arxiv.org/abs/1611.02731Chen X. et al. Variational Lossy Autoencoder, 2016

Decoder weakening techniques

How to force the model encode information about $\mathbf{x}$ into $\mathbf{z}$?

KL annealing

$$\mathcal{L}(\phi, \boldsymbol{\theta}, \beta) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\phi)} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \beta \cdot KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}))$$

Start training with $\beta = 0$, increase it until $\beta = 1$ during training.

Free bits

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

It ensures the use of less than λ bits of information and results in $KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z})) \geq \lambda$.

Bowman S. R. et al. Generating Sentences from a Continuous Space, 2015
Kingma D. P. et al. Improving Variational Inference with Inverse Autoregressive Flow, 2016

Outline

1. VAE limitations
2. Posterior collapse and decoder weakening techniques
3. Tighter variational bound
4. Normalizing flows

VAE limitations

- Poor generative distribution (decoder)

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

- **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)

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

Importance sampling

LVM

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} = \int \left[\frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x}, \phi)} \right] q(\mathbf{z}|\mathbf{x}, \phi) d\mathbf{z}$$
$$= \int f(\mathbf{x}, \mathbf{z}) q(\mathbf{z}|\mathbf{x}, \phi) d\mathbf{z} = \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi)} f(\mathbf{x}, \mathbf{z})$$

Here $f(\mathbf{x}, \mathbf{z}) = \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x}, \phi)}$.

ELBO: derivation 1

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi)} f(\mathbf{x}, \mathbf{z}) \geq \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi)} \log f(\mathbf{x}, \mathbf{z}) =$$
$$= \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi)} \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x}, \phi)} = \mathcal{L}(q, \boldsymbol{\theta}).$$

$f(\mathbf{x}, \mathbf{z})$ could be any function that satisfies $p(\mathbf{x}|\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z} \sim q} f(\mathbf{x}, \mathbf{z})$.
Could we choose better $f(\mathbf{x}, \mathbf{z})$?

Importance Weighted Autoencoders (IWAE)

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int \left[\frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x}, \phi)} \right] q(\mathbf{z}|\mathbf{x}, \phi) d\mathbf{z} = \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi)} f(\mathbf{x}, \mathbf{z})$$

Let define

$$f(\mathbf{x}, \mathbf{z}_1, \dots, \mathbf{z}_K) = \frac{1}{K} \sum_{k=1}^{K} \frac{p(\mathbf{x}, \mathbf{z}_k|\boldsymbol{\theta})}{q(\mathbf{z}_k|\mathbf{x}, \phi)}$$

$$\mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_K \sim q(\mathbf{z}|\mathbf{x}, \phi)} f(\mathbf{x}, \mathbf{z}_1, \dots, \mathbf{z}_K) = p(\mathbf{x}|\boldsymbol{\theta})$$

ELBO

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_K \sim q(\mathbf{z}|\mathbf{x})} f(\mathbf{x}, \mathbf{z}, \dots, \mathbf{z}_K) \geq$$

$$\geq \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_K \sim q(\mathbf{z}|\mathbf{x}, \phi)} \log f(\mathbf{x}, \mathbf{z}, \dots, \mathbf{z}_K) =$$

$$= \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_K \sim q(\mathbf{z}|\mathbf{x}, \phi)} \log \left[\frac{1}{K} \sum_{k=1}^{K} \frac{p(\mathbf{x}, \mathbf{z}_k|\boldsymbol{\theta})}{q(\mathbf{z}_k|\mathbf{x}, \phi)} \right] = \mathcal{L}_K(q, \boldsymbol{\theta}).$$

Burda Y., Grosse R., Salakhutdinov R. Importance Weighted Autoencoders, 2015

Importance Weighted Autoencoders (IWAE)

VAE objective

$$\log p(\mathbf{x}|\boldsymbol{\theta}) \geq \mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\phi)} \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x}, \phi)} \to \max_{q,\boldsymbol{\theta}}$$

$$\mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z}_1,\dots,\mathbf{z}_K \sim q(\mathbf{z}|\mathbf{x},\phi)} \left(\frac{1}{K} \sum_{k=1}^{K} \log \frac{p(\mathbf{x}, \mathbf{z}_k|\boldsymbol{\theta})}{q(\mathbf{z}_k|\mathbf{x}, \phi)} \right) \to \max_{q,\boldsymbol{\theta}}.$$

IWAE objective

$$\mathcal{L}_K(q, \boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z}_1,\dots,\mathbf{z}_K \sim q(\mathbf{z}|\mathbf{x},\phi)} \log \left(\frac{1}{K} \sum_{k=1}^{K} \frac{p(\mathbf{x}, \mathbf{z}_k|\boldsymbol{\theta})}{q(\mathbf{z}_k|\mathbf{x}, \phi)} \right) \to \max_{q,\boldsymbol{\theta}}.$$

If $K = 1$, these objectives coincide.

Burda Y., Grosse R., Salakhutdinov R. Importance Weighted Autoencoders, 2015

Importance Weighted Autoencoders (IWAE)

Theorem

1. $\log p(\mathbf{x}|\boldsymbol{\theta}) \geq \mathcal{L}_K(q, \boldsymbol{\theta}) \geq \mathcal{L}_M(q, \boldsymbol{\theta}), \quad \text{for } K \geq M;$
2. $\log p(\mathbf{x}|\boldsymbol{\theta}) = \lim_{K \to \infty} \mathcal{L}_K(q, \boldsymbol{\theta})$ if $\frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x}, \phi)}$ is bounded.

If $K > 1$ the bound could be tighter.

$$
\mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x}, \phi)};
$$

$$
\mathcal{L}_K(q, \boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_K \sim q(\mathbf{z}|\mathbf{x}, \phi)} \log \left(\frac{1}{K} \sum_{k=1}^{K} \frac{p(\mathbf{x}, \mathbf{z}_k|\boldsymbol{\theta})}{q(\mathbf{z}_k|\mathbf{x}, \phi)} \right).
$$

- $\mathcal{L}_1(q, \boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta})$;
- $\mathcal{L}_\infty(q, \boldsymbol{\theta}) = \log p(\mathbf{x}|\boldsymbol{\theta})$.
- Which $q^*(\mathbf{z}|\mathbf{x}, \phi)$ gives $\mathcal{L}(q^*, \boldsymbol{\theta}) = \log p(\mathbf{x}|\boldsymbol{\theta})$?

Burda Y., Grosse R., Salakhutdinov R. Importance Weighted Autoencoders, 2015

Importance Weighted Autoencoders (IWAE)

Objective

$$\mathcal{L}_K(q, \boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_K \sim q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})} \log \left(\frac{1}{K} \sum_{k=1}^{K} \frac{p(\mathbf{x}, \mathbf{z}_k | \boldsymbol{\theta})}{q(\mathbf{z}_k | \mathbf{x}, \boldsymbol{\phi})} \right) \to \max_{\boldsymbol{\phi}, \boldsymbol{\theta}}.$$

Theorem

Gradient signal of $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ vanishes as K increases:

$$\Delta_K = \nabla_{\boldsymbol{\theta}, \boldsymbol{\phi}} \mathcal{L}_K(q, \boldsymbol{\theta}); \quad \text{SNR}_K = \frac{\mathbb{E}[\Delta_K]}{\sigma(\Delta_K)};$$

$$\text{SNR}_K(\boldsymbol{\theta}) = O(\sqrt{K}); \quad \text{SNR}_K(\boldsymbol{\phi}) = O\left(\sqrt{K^{-1}}\right).$$

- IWAE makes the variational bound tighter and extends the class of variational distributions.
- Gradient signal becomes really small, training is complicated.
- IWAE is a standard quality measure for VAE models.

Rainforth T. et al. Tighter variational bounds are not necessarily better, 2018

Outline

1. VAE limitations
2. Posterior collapse and decoder weakening techniques
3. Tighter variational bound
4. Normalizing flows

Likelihood-based models so far...

Autoregressive models

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{j=1}^{m} p(x_j|\mathbf{x}_{1:j-1}, \boldsymbol{\theta})$$

- tractable likelihood,
- no inferred latent factors.

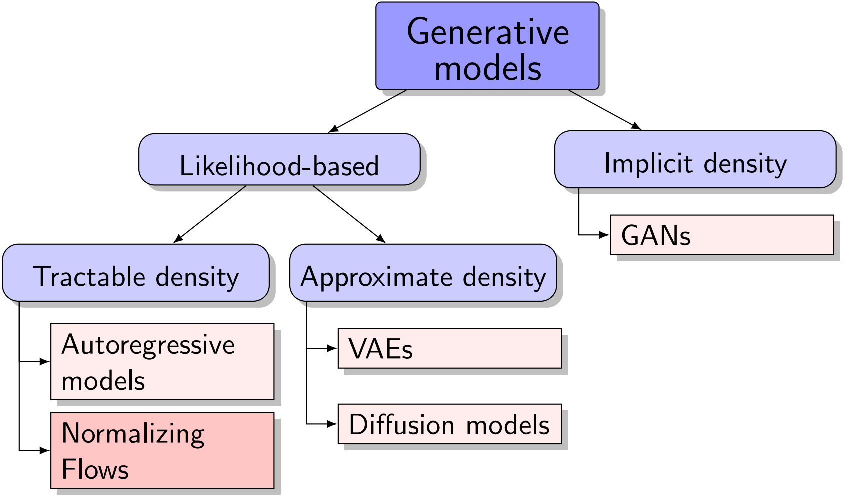
Latent variable models

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

- latent feature representation,
- intractable likelihood.

How to build model with latent variables and tractable likelihood?

Generative models zoo



Normalizing flows prerequisites

Jacobian matrix

Let $f : \mathbb{R}^m \to \mathbb{R}^m$ is a differentiable function.

$$\mathbf{z} = f(\mathbf{x}), \quad \mathbf{J} = \frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \dots & \frac{\partial z_1}{\partial x_m} \\ \dots & \dots & \dots \\ \frac{\partial z_m}{\partial x_1} & \dots & \frac{\partial z_m}{\partial x_m} \end{pmatrix} \in \mathbb{R}^{m \times m}$$

Change of variable theorem (CoV)

Let $\mathbf{x}$ be a random variable with density function $p(\mathbf{x})$ and $f : \mathbb{R}^m \to \mathbb{R}^m$ is a differentiable, **invertible** function (diffeomorphism). If $\mathbf{z} = f(\mathbf{x})$, $\mathbf{x} = f^{-1}(\mathbf{z}) = g(\mathbf{z})$, then

$$p(\mathbf{x}) = p(\mathbf{z})|\det(\mathbf{J}_f)| = p(\mathbf{z}) \left|\det\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right)\right| = p(f(\mathbf{x})) \left|\det\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right)\right|$$

$$p(\mathbf{z}) = p(\mathbf{x})|\det(\mathbf{J}_g)| = p(\mathbf{x}) \left|\det\left(\frac{\partial \mathbf{x}}{\partial \mathbf{z}}\right)\right| = p(g(\mathbf{z})) \left|\det\left(\frac{\partial g(\mathbf{z})}{\partial \mathbf{z}}\right)\right|.$$

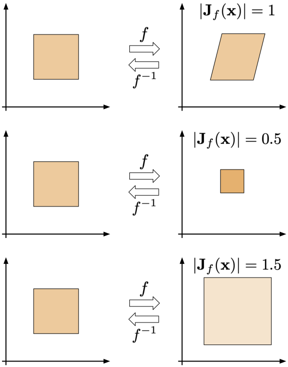
Jacobian determinant

Inverse function theorem

If function f is invertible and Jacobian matrix is continuous and non-singular, then

$$\mathbf{J}_f = \mathbf{J}_{g^{-1}} = \mathbf{J}_g^{-1}, \quad |\det(\mathbf{J}_f)| = \frac{1}{|\det(\mathbf{J}_g)|}$$

- $\mathbf{x}$ and $\mathbf{z}$ have the same dimensionality ($\mathbb{R}^m$).
- $f(\mathbf{x}, \boldsymbol{\theta})$ could be parametric function.
- Determinant of Jacobian matrix $\mathbf{J} = \frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}}$ shows how the volume changes under the transorfmation.



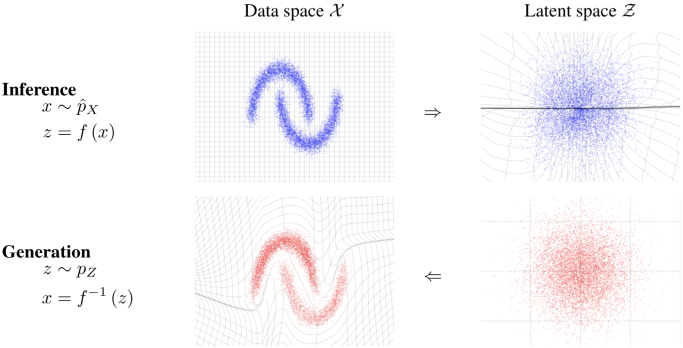
https://jmtomczak.github.io/blog/3/3_flows.html

Fitting flows

MLE problem

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right| = p(f(\mathbf{x}, \boldsymbol{\theta})) \left| \det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} \right) \right|$$

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log |\det(\mathbf{J}_f)| \to \max_{\boldsymbol{\theta}}$$



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

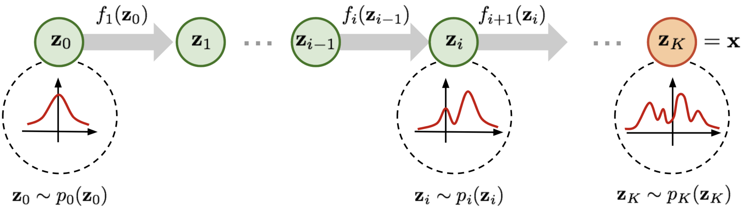
Composition of flows

Theorem

Diffeomorphisms are **composable** (If $\{f_k\}_{k=1}^K$ satisfy conditions of the change of variable theorem, then $\mathbf{z} = f(\mathbf{x}) = f_K \circ \cdots \circ f_1(\mathbf{x})$ also satisfies it).

$$p(\mathbf{x}) = p(f(\mathbf{x})) \left| \det \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = p(f(\mathbf{x})) \left| \det \left(\frac{\partial \mathbf{f}_K}{\partial \mathbf{f}_{K-1}} \cdots \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}} \right) \right| =$$

$$= p(f(\mathbf{x})) \prod_{k=1}^{K} \left| \det \left(\frac{\partial \mathbf{f}_k}{\partial \mathbf{f}_{k-1}} \right) \right| = p(f(\mathbf{x})) \prod_{k=1}^{K} |\det(\mathbf{J}_{f_k})|$$



https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html

Flows

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log |\det(\mathbf{J}_f)|$$

Definition

Normalizing flow is a *differentiable, invertible* mapping from data $\mathbf{x}$ to the noise $\mathbf{z}$.

- **Normalizing** means that the inverse flow takes samples from $\pi(\mathbf{x})$ and normalizes them into samples from the density $p(\mathbf{z})$.
- **Flow** refers to the trajectory followed by samples from $p(\mathbf{z})$ as they are transformed by the sequence of transformations

$$\mathbf{z} = f_K \circ \cdots \circ f_1(\mathbf{x}); \quad \mathbf{x} = f_1^{-1} \circ \cdots \circ f_K^{-1}(\mathbf{z}) = g_1 \circ \cdots \circ g_K(\mathbf{z})$$

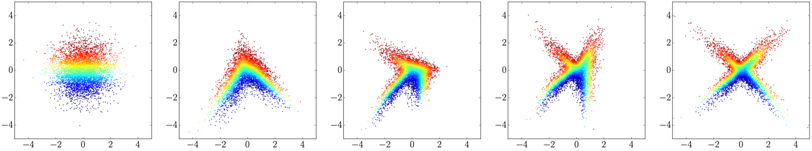
Log likelihood

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f_K \circ \cdots \circ f_1(\mathbf{x})) + \sum_{k=1}^{K} \log |\det(\mathbf{J}_{f_k})|,$$

where $\mathbf{J}_{f_k} = \frac{\partial \mathbf{f}_k}{\partial \mathbf{f}_{k-1}}$.

Flows

Example of a 4-step flow



Flow log likelihood

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log |\det(\mathbf{J}_f)|$$

What is the complexity of the determinant computation?

What we want

- Efficient computation of the Jacobian matrix $\mathbf{J}_f = \frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}}$;
- Efficient sampling from the base distribution $p(\mathbf{z})$;
- Efficient inversion of $f(\mathbf{x}, \boldsymbol{\theta})$.

Papamakarios G. et al. Normalizing flows for probabilistic modeling and inference, 2019

Summary

- Standart VAE has several limitations that we will address later in the course.
- More powerful decoder in VAE leads to more expressive generative model. However, too expressive decoder could lead to the posterior collapse.
- The decoder weakening is a set of techniques to avoid the posterior collapse.
- The IWAE could get the tighter lower bound to the likelihood, but the training of such model becomes more difficult.
- Flow models transform a simple base distribution to a complex one via a sequence of invertible transformations with tractable Jacobian.
- Flow models have a tractable likelihood that is given by the change of variable theorem.