Deep Generative Models
Lecture 4

Roman Isachenko

Ozon Masters

Spring, 2021

Bayesian framework

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.

Meaning

We have unobserved variables $\mathbf{t}$ and some prior knowledge about them $p(\mathbf{t})$. Then, the data $\mathbf{x}$ has been observed. Posterior distribution $p(\mathbf{t}|\mathbf{x})$ summarizes the knowledge after the observations.

Variational Lower Bound

We have set of objects $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$. The goal is to perform Bayesian inference on the unobserved variables $\mathbf{T} = \{\mathbf{t}_i\}_{i=1}^n$.

Evidence Lower Bound (ELBO)

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

We would like to maximize lower bound $\mathcal{L}(q)$.

Mean field approximation

Independence assumption

$$q(\mathbf{T}) = \prod_{i=1}^{k} q_i(\mathbf{T}_i), \quad \mathbf{T} = [\mathbf{T}_1, \dots, \mathbf{T}_k], \, \mathbf{T}_j = \{\mathbf{t}_{ij}\}_{i=1}^{n}, \, \mathbf{t}_i = \{\mathbf{T}_{ij}\}_{j=1}^{k}.$$

Block coordinate optimization of ELBO for $q_j(\mathbf{T}_j)$

$$\mathcal{L}(q) = \int q(\mathbf{T}) \log \frac{p(\mathbf{X}, \mathbf{T})}{q(\mathbf{T})} d\mathbf{T} = \int \left[\prod_{i=1}^{k} q_i(\mathbf{T}_i)\right] \log \frac{p(\mathbf{X}, \mathbf{T})}{\left[\prod_{i=1}^{k} q_i(\mathbf{T}_i)\right]} \prod_{i=1}^{k} d\mathbf{T}_i =$$

$$= \int \left[\prod_{i=1}^{k} q_i\right] \log p(\mathbf{X}, \mathbf{T}) \prod_{i=1}^{k} d\mathbf{T}_i - \sum_{i=1}^{k} \int \left[\prod_{j=1}^{k} q_j\right] \log q_i \prod_{j=1}^{k} d\mathbf{T}_j =$$

$$= \int q_j \left[\int \log p(\mathbf{X}, \mathbf{T}) \prod_{i \neq j} q_i d\mathbf{T}_i\right] d\mathbf{T}_j -$$

$$- \int q_j \log q_j d\mathbf{T}_j + \mathrm{const}(q_j) \to \max_{q_j}$$

Mean field approximation

Block coordinate optimization of ELBO for $q_j(\mathbf{T}_j)$

$$\mathcal{L}(q) = \int q_j \left[\int \log p(\mathbf{X}, \mathbf{T}) \prod_{i \neq j} q_i \, d\mathbf{T}_i \right] d\mathbf{T}_j - \int q_j \log q_j \, d\mathbf{T}_j + \text{const}(q_j) =$$

$$= \int q_j \log \hat{p}(\mathbf{X}, \mathbf{T}_j) d\mathbf{T}_j - \int q_j \log q_j \, d\mathbf{T}_j + \text{const}(q_j) \to \max_{q_j}.$$

Here we introduce

$$\log \hat{p}(\mathbf{X}, \mathbf{T}_j) = \int \log p(\mathbf{X}, \mathbf{T}) \prod_{i \neq j} q_i \, d\mathbf{T}_i = \mathbb{E}_{i \neq j} \log p(\mathbf{X}, \mathbf{T}) + \text{const}(q_j)$$

Final ELBO derivation for $q_j(\mathbf{T}_j)$

$$\mathcal{L}(q) = \int q_j(\mathbf{T}_j) \log \hat{p}(\mathbf{X}, \mathbf{T}_j) d\mathbf{T}_j - \int q_j(\mathbf{T}_j) \log q_j(\mathbf{T}_j) d\mathbf{T}_j + \text{const}(q_j) =$$

$$\int q_j(\mathbf{T}_j) \log \frac{\hat{p}(\mathbf{X}, \mathbf{T}_j)}{q_j(\mathbf{T}_j)} d\mathbf{T}_j + \text{const}(q_j) =$$

$$= -KL(q_j(\mathbf{T}_j) || \hat{p}(\mathbf{X}, \mathbf{T}_j)) + \text{const}(q_j) \to \max_{q_j}.$$

Mean field approximation

Independence assumption

$$q(\mathbf{T}) = \prod_{i=1}^{k} q_i(\mathbf{T}_i), \quad \mathbf{T} = [\mathbf{T}_1, \ldots, \mathbf{T}_k], \quad \mathbf{T}_j = \{\mathbf{t}_{ij}\}_{i=1}^{n}.$$

ELBO

$$\mathcal{L}(q) = -KL(q_j(\mathbf{T}_j) || \hat{p}(\mathbf{X}, \mathbf{T}_j)) + \mathrm{const}(q_j) \to \max_{q_j}.$$

Solution

$$q_j(\mathbf{T}_j) = \mathrm{const} \cdot \hat{p}(\mathbf{X}, \mathbf{T}_j)$$

$$\log \hat{p}(\mathbf{X}, \mathbf{T}_j) = \mathbb{E}_{i \neq j} \log p(\mathbf{X}, \mathbf{T}) + \mathrm{const}$$

$$\log q_j(\mathbf{T}_j) = \mathbb{E}_{i \neq j} \log p(\mathbf{X}, \mathbf{T}) + \mathrm{const}$$

Mean field approximation

ELBO

$$\mathcal{L}(q) = -KL(q_j(\mathbf{T}_j)||\hat{p}(\mathbf{X}, \mathbf{T}_j)) + \text{const}(q_j) \to \max_{q_j}.$$

Solution

$$\log q_j(\mathbf{T}_j) = \mathbb{E}_{i \neq j} \log p(\mathbf{X}, \mathbf{T}) + \text{const}$$

Assumptions:

- $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2] = [\mathbf{Z}, \boldsymbol{\theta}]$, $q(\mathbf{T}) = q(\mathbf{T}_1) \cdot q(\mathbf{T}_2) = q(\mathbf{Z}) \cdot q(\boldsymbol{\theta})$.
- restrict a class of probability distributions for $\boldsymbol{\theta}$ to Dirac delta functions:

  $$q_2 = q(\mathbf{T}_2) = q(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*).$$

  Under the restrictions the exact solution for q_2 is not reached (KL can be greater than 0).

Mean field approximation

General solution

$$\log q_j(\mathbf{T}_j) = \mathbb{E}_{i \neq j} \log p(\mathbf{X}, \mathbf{T}) + \text{const}$$

Solution for $q_1 = q(\mathbf{Z})$

$$\begin{aligned}\log q(\mathbf{Z}) = \int q(\boldsymbol{\theta}) \log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta}) d\boldsymbol{\theta} + \text{const} = \\ = \int \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*) \log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta}) d\boldsymbol{\theta} + \text{const} = \\ = \log p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}^*) + \text{const.}\end{aligned}$$

EM-algorithm (E-step)

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

Mean field approximation

ELBO

$$\mathcal{L}(q) = -KL(q_j(\mathbf{T}_j)||\hat{p}(\mathbf{X}, \mathbf{T}_j)) + \text{const}(q_j) \to \max_{q_j}.$$

ELBO maximization w.r.t. $q_2 = q(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$

$$\begin{aligned}
\mathcal{L}(q_1, q_2) &= -KL(q(\boldsymbol{\theta})||\hat{p}(\mathbf{X}, \boldsymbol{\theta})) + \text{const}(\boldsymbol{\theta}^*) \\
&= \int q(\boldsymbol{\theta}) \log \frac{\hat{p}(\mathbf{X}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} + \text{const}(\boldsymbol{\theta}^*) \\
&= \int q(\boldsymbol{\theta}) \log \hat{p}(\mathbf{X}, \boldsymbol{\theta}) d\boldsymbol{\theta} - \int q(\boldsymbol{\theta}) \log q(\boldsymbol{\theta}) d\boldsymbol{\theta} + \text{const}(\boldsymbol{\theta}^*) \\
&= \int \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*) \log \hat{p}(\mathbf{X}, \boldsymbol{\theta}) d\boldsymbol{\theta} + \text{const}(\boldsymbol{\theta}^*) \to \max_{\boldsymbol{\theta}^*}
\end{aligned}$$

Mean field approximation

ELBO maximization w.r.t. $q_2 = q(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$

$$\mathcal{L}(q_1, q_2) = \int \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*) \log \hat{p}(\mathbf{X}, \boldsymbol{\theta}) d\boldsymbol{\theta} + \text{const} = \log \hat{p}(\mathbf{X}, \boldsymbol{\theta}^*) + \text{const}$$

$$= \mathbb{E}_{i \neq j} \log p(\mathbf{X}, \mathbf{T}) + \text{const} = \mathbb{E}_{q_1} \log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta}^*) + \text{const}$$

$$= \int q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}^*) d\mathbf{Z} + \log p(\boldsymbol{\theta}^*) + \text{const} \to \max_{\boldsymbol{\theta}^*}$$

EM-algorithm (M-step)

$$\mathcal{L}(q, \boldsymbol{\theta}) = \int q(\mathbf{Z}) \log \frac{p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})}{q(\mathbf{Z})} d\mathbf{Z}$$

$$= \int q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) d\mathbf{Z} + \text{const} \to \max_{\boldsymbol{\theta}}$$

Mean field approximation

Solution

$$\log q_j(\mathbf{T}_j) = \mathbb{E}_{i \neq j} \log p(\mathbf{X}, \mathbf{T}) + \text{const}$$

EM algorithm (special case)

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

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

- M-step

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

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

Likelihood-based models so far...

Autoregressive models

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

- 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?

Flows intuition

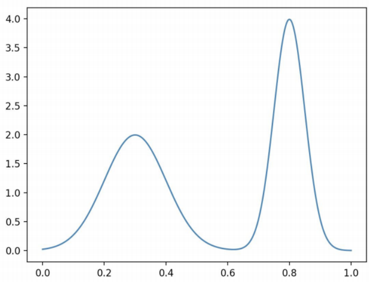
Let ξ be a random variable with density $p(\xi)$. Then

$$\eta = F(\xi) = \int_{-\infty}^{\xi} p(t)dt \sim U[0,1].$$

$$P(\eta < y) = P(F(\xi) < y) = P(\xi < F^{-1}(y)) = F(F^{-1}(y)) = y$$

Hence

$$\eta \sim U[0,1]; \quad \xi = F^{-1}(\eta) \quad \Rightarrow \quad \xi \sim p(\xi).$$



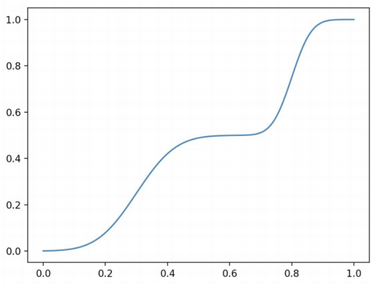


image credit: https://sites.google.com/view/berkeley-cs294-158-sp19/home

Flows intuition

- Let $z \sim p(z)$ is a random variable with base distribution $p(z) = U[0, 1]$.
- Let $x \sim p(x)$ is a random variable with complex distribution $p(x)$ and cdf $F(x)$.
- Then noise variable z can be transformed to x using inverse cdf F^{-1} ($x = F^{-1}(z)$).

How to transform random variable z which has a distribution different from uniform to x?

- Let $z \sim p(z)$ is a random variable with base distribution $p(z)$ and cdf $G(z)$.
- Then $z_0 = G(z)$ has base distribution $p(z_0) = U[0, 1]$.
- Let $x \sim p(x)$ is a random variable with complex distribution $p(x)$ and cdf $F(x)$.
- Then noise variable z can be transformed to x using cdf G and inverse cdf F^{-1} ($x = F^{-1}(z_0) = F^{-1}(G(z))$).

Change of variables

Theorem

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

Then

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

- $\mathbf{x}$ and $\mathbf{z}$ have the same dimensionality (lies in $\mathbb{R}^m$);
- $\left| \det \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = \left| \det \left(\frac{\partial g^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = \left| \det \left(\frac{\partial g(\mathbf{z})}{\partial \mathbf{z}} \right) \right|^{-1}$;
- $f(\mathbf{x}, \boldsymbol{\theta})$ could be parametric function.

Fitting flows

MLE problem

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

Challenge

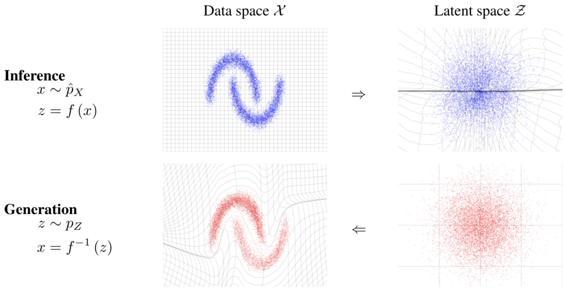
$p(\mathbf{x}|\boldsymbol{\theta})$ can be intractable.

Fitting flow to solve MLE

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

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

Flows



Computational requirement

- Evaluating model density $p(\mathbf{x}|\boldsymbol{\theta})$ requires computing the transformation $\mathbf{z} = f(\mathbf{x}, \boldsymbol{\theta})$ and its Jacobian determinant $\left|\det\left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}}\right)\right|$, and evaluating the density $p(\mathbf{z})$.
- Sampling $\mathbf{x}$ from the model requires the ability to sample from $p(\mathbf{z})$ and to compute the transformation $\mathbf{x} = g(\mathbf{z}, \boldsymbol{\theta}) = f^{-1}(\mathbf{z}, \boldsymbol{\theta})$.

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

Forward KL vs Reverse KL

Fix probabilistic model $p(\mathbf{x}|\boldsymbol{\theta})$ – the set of parameterized distributions .
Instead of searching true $\pi(\mathbf{x})$ over all probability distributions, learn function approximation $p(\mathbf{x}|\boldsymbol{\theta}) \approx \pi(\mathbf{x})$.

Forward KL

$$KL(\pi || p) = \int \pi(\mathbf{x}) \log \frac{\pi(\mathbf{x})}{p(\mathbf{x}|\boldsymbol{\theta})} d\mathbf{x} \rightarrow \min_{\boldsymbol{\theta}}$$

Reverse KL

$$KL(p || \pi) = \int p(\mathbf{x}|\boldsymbol{\theta}) \log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{\pi(\mathbf{x})} d\mathbf{x} \rightarrow \min_{\boldsymbol{\theta}}$$

- What is the difference between these two formulations?
- What do we get in these two cases if $p(\mathbf{x}|\boldsymbol{\theta})$ is a flow model?

Forward KL vs Reverse KL

Forward KL

$$\begin{aligned} KL(\pi || p) &= \int \pi(\mathbf{x}) \log \frac{\pi(\mathbf{x})}{p(\mathbf{x}|\boldsymbol{\theta})} d\mathbf{x} \\ &= \int \pi(\mathbf{x}) \log \pi(\mathbf{x}) d\mathbf{x} - \int \pi(\mathbf{x}) \log p(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x} \\ &= -\mathbb{E}_{\pi(\mathbf{x})} \log p(\mathbf{x}|\boldsymbol{\theta}) + \text{const} \to \min_{\boldsymbol{\theta}} \end{aligned}$$

Monte-Carlo estimation

$$KL(\pi || p) = -\mathbb{E}_{\pi(\mathbf{x})} \log p(\mathbf{x}|\boldsymbol{\theta}) + \text{const} \approx -\frac{1}{n} \sum_{i=1}^{n} \log p(\mathbf{x}_i|\boldsymbol{\theta}) \to \min_{\boldsymbol{\theta}} .$$

MLE problem

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

Forward KL vs Reverse KL

Forward KL

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} \log p(\mathbf{x}_i|\boldsymbol{\theta}) \approx \arg\min_{\boldsymbol{\theta}} KL(\pi||p)$$

Maximum likelihood estimation is equivalent to minimization of the Monte-Carlo estimation of forward KL.

Forward KL for flow model

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

- We need to be able to compute $f(\mathbf{x}, \boldsymbol{\theta})$ and its Jacobian.
- We need to be able to compute the density $p(\mathbf{z})$.
- We don't need to think about computing the function $g(\mathbf{z}, \boldsymbol{\theta}) = f^{-1}(\mathbf{z}, \boldsymbol{\theta})$ until we want to sample from the flow.

Forward KL vs Reverse KL

Reverse KL

$$
\begin{aligned}
KL(p||\pi) &= \int p(\mathbf{x}|\boldsymbol{\theta}) \log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{\pi(\mathbf{x})} d\mathbf{x} \\
&= \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \left[\log p(\mathbf{x}|\boldsymbol{\theta}) - \log \pi(\mathbf{x})\right] \rightarrow \min_{\boldsymbol{\theta}}
\end{aligned}
$$

Reverse KL for flow model

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

$$
KL(p||\pi) = \mathbb{E}_{p(\mathbf{z})} \left[\log p(\mathbf{z}) - \log \left| \det \left(\frac{\partial g(\mathbf{z}, \boldsymbol{\theta})}{\partial \mathbf{z}} \right) \right| - \log \pi(g(\mathbf{z}, \boldsymbol{\theta})) \right]
$$

- We need to be able to compute $g(\mathbf{z}, \boldsymbol{\theta})$ and its Jacobian.
- We need to be able to sample from the density $p(\mathbf{z})$ (do not need to evaluate it).
- We don't need to think about computing the function $f(\mathbf{x}, \boldsymbol{\theta})$.

Composition of flows

Theorem

Diffeomorphisms are **composable** (If f_1, f_2 satisfy conditions of the change of variable theorem (differentiable and invertible), then $\mathbf{z} = f(\mathbf{x}) = f_2 \circ f_1(\mathbf{x})$ also satisfies it).

$$
\begin{aligned}
p(\mathbf{x}) = p(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right| &= p(f(\mathbf{x})) \left| \det \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = \\
= p(f(\mathbf{x})) \left| \det \left(\frac{\partial f_2 \circ f_1(\mathbf{x})}{\partial \mathbf{x}} \right) \right| &= p(f(\mathbf{x})) \left| \det \left(\frac{\partial \mathbf{f}_2}{\partial \mathbf{f}_1} \cdot \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}} \right) \right| = \\
&= p(f(\mathbf{x})) \left| \det \left(\frac{\partial \mathbf{f}_2}{\partial \mathbf{f}_1} \right) \right| \cdot \left| \det \left(\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}} \right) \right|
\end{aligned}
$$

What will we get in the case $\mathbf{z} = f(\mathbf{x}) = f_n \circ \dots \circ f_1(\mathbf{x})$?

Flows

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

Definition

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

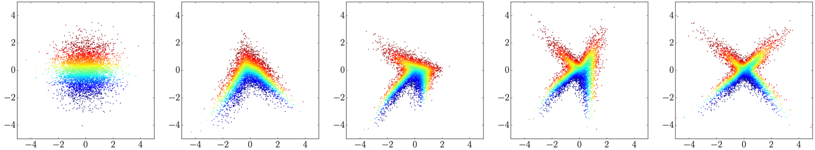
- "Normalizing" means that the inverse flow takes samples from $p(\mathbf{x})$ and normalizes them into samples from density $p(\mathbf{z})$.
- "Flow" refers to the trajectory followed by samples from $p(\mathbf{z})$ as they are transformed by the sequence of transformations

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

$$\begin{aligned} p(\mathbf{x}) &= p(f_K \circ \cdots \circ f_1(\mathbf{x})) \left| \det \left(\frac{\partial f_K \circ \cdots \circ f_1(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = \\ &= p(f_K \circ \cdots \circ f_1(\mathbf{x})) \prod_{k=1}^{K} \left| \det \left(\frac{\partial \mathbf{f}_k}{\partial \mathbf{f}_{k-1}} \right) \right| . \end{aligned}$$

Flows

Example of a 4-step flow



Flow likelihood

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

What is the complexity of the determinant computation?

What we want

- Efficient computation of Jacobian $\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}}$;
- 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

- Mean-field approximation is a general form of approximate variational inference.
- The EM-algorithm and VAE model can be presented as a special case of the mean-field approximation.
- Forward KL minimization is equivalent to MLE. Reverse KL is used in variational inference.
- Flow models transform a simple base distribution to a complex one via a sequence of invertible transformations.
- Flow models have a tractable likelihood that is given by the change of variable theorem.