Bayesian Methods in Machine Learning
Seminar: 14

{Evgenii.Egorov, Anna.Kuzina}@skoltech.ru

Skoltech

Moscow, 2020

Generative Models

- VAE

$$p(x) = \int p(x|z)p(z)dz$$

- GANs

$$x = G(z),\ z \sim p(z)$$

- Autoregressive Models

$$p(x_1, x_2, x_3 \dots) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)\dots$$

- Normalizing Flows

$$z = f^K \circ \cdots \circ f^2 \circ f^1(x), \text{ where } f^i \text{ is invertable } \forall i$$

Generative Models

- VAE

$$p(x) = \int p(x|z)p(z)dz$$

- GANs

$$x = G(z),\ z \sim p(z)$$

- Autoregressive Models

$$p(x_1, x_2, x_3 \dots) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)\dots$$

- **Normalizing Flows**

$$z = f^K \circ \dots \circ f^2 \circ f^1(x), \text{ where } f^i \text{ is invertable } \forall i$$

NF: Change of Variable formula

We'd like to learn invertable transformation from data to noise:

$$z = f(x) \;\Rightarrow x = f^{-1}(z)$$

$$p_x(x) = p_z(z)\left|\det \frac{\partial z}{\partial x}\right| = \text{prior} \;\cdot\; \text{volume change}$$

$$\log p_x(x) = \log p_z(f(x)) + \log\left|\det \frac{\partial f(x)}{\partial x}\right|$$

We can combine several f, to make our transformations more powerful

$$x = x_0 \xrightarrow{f^{\mathbf{1}}} x_1 \ldots \xrightarrow{f^K} x_k = z$$

$$z = f^K \circ \cdots \circ f^2 \circ f^1(x)$$

$$\log p_x(x) = \log p_z(f(x)) + \sum_{k=1}^{K} \log|\det \frac{\partial f^k(x)}{\partial x_{k-1}}|$$

Inference with normalizing flows

- **Training**

$$\max_\theta \log p_x(x) = \max_\theta \log p_z(f_\theta(x)) + \sum_{k=1}^{K} \log |\det \frac{\partial f_\theta^k(x)}{\partial x_{k-1}}|$$

At each step we need to:

- Evaluate $f_\theta(x)$
- Compute determinant of the Jacobian matrix

- **Sampling**

$$\hat{z} \sim p(z)$$
$$\hat{x} = f^{-1}(\hat{z})$$

We only need 1 inverse pass

Inference with normalizing flows

- **Training**

$$\max_\theta \log p_x(x) = \max_\theta \log p_z(f_\theta(x)) + \sum_{k=1}^{K} \log|\det \frac{\partial f_\theta^k(x)}{\partial x_{k-1}}|$$

At each step we need to:

  - Evaluate $f_\theta(x)$
  - **Compute determinant of the Jacobian matrix**

- **Sampling**

$$\hat{z} \sim p(z)$$
$$\hat{x} = f^{-1}(\hat{z})$$

We only need 1 inverse pass

Inference with normalizing flows

- **Training**

$$\max_\theta \log p_x(x) = \max_\theta \log p_z(f_\theta(x)) + \sum_{k=1}^{K} \log |\det \frac{\partial f_\theta^k(x)}{\partial x_{k-1}}|$$

At each step we need to:

- Evaluate $f_\theta(x)$
- **Compute determinant of the Jacobian matrix**

- **Sampling**

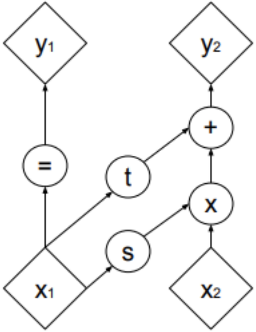
$$\hat{z} \sim p(z)$$
$$\hat{x} = f^{-1}(\hat{z})$$

We only need 1 inverse pass

Let's choose f, for which determinant of the Jacobian is easy to compute

Idea 1: Affine Coupling Layer

Transformation used in Nice and RealNVP models.



Split input vector into two parts:

$$z_{1:d} = x_{1:d},$$
$$z_{d+1:D} = x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d}).$$

Where $s(\cdot)$ and $t(\cdot)$ are arbitrary functions and $\odot$ is point-wise multiplication.

Task

- Derive $\frac{\partial f(x)}{dx}$
- Compute $\log \det \frac{\partial f(x)}{dx}$

Solution: Affine Coupling Layer

$$z_{1:d} = x_{1:d},$$
$$z_{d+1:D} = x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d}).$$

Task

- Derive $\frac{\partial f(x)}{dx}$

$$\frac{\partial f(x)}{dx} = \begin{bmatrix} I & 0 \\ \left[\frac{\partial f(x)}{dx_{\mathbf{1}:d}}\right]_{d+1:D} & \exp(s(x_{1:d})) \end{bmatrix}$$

- Compute $\log\det \frac{\partial f(x)}{dx}$

$$\det \frac{\partial f(x)}{dx} = \prod_{i=1}^{d} \exp(s(x_{1:d}))_i$$
$$\log\det \frac{\partial f(x)}{dx} = \sum_{i=1}^{d} s(x_{1:d})_i$$

Idea 2: Invertable 1x1 convolutions

It was noticed that permuting channels of the images between coupling layers improves flow performance. In Glow authors propose to use 1x1 convolution as a generalization of permutation operation.
Given input $x \in \mathbb{R}^{h \times w \times c}$ and weight matrix $\mathbf{W} \in \mathbb{R}^{c \times c}$. We can define this transformation as:

$$f(x)_{i,j} = x_{i,j}\mathbf{W}$$

Task
Compute log-determinant of such transformation.

Solution: Invertable 1x1 convolutions

Given input $x \in \mathbb{R}^{h \times w \times c}$ and weight matrix $\mathbf{W} \in \mathbb{R}^{c \times c}$. We can define this transformation as:

$$f(x)_{i,j} = x_{i,j}\mathbf{W}$$

Solution

$$\frac{\partial f(x)_{i,j}}{\partial x_{i,j}} = \mathbf{W}$$

$$\log \det \frac{\partial f(x)}{\partial x} = h \cdot w \cdot \log \det \mathbf{W}$$

Since c is usually small, it is pretty cheap to compute $\log \det$ and inverse.

And now, let's practice