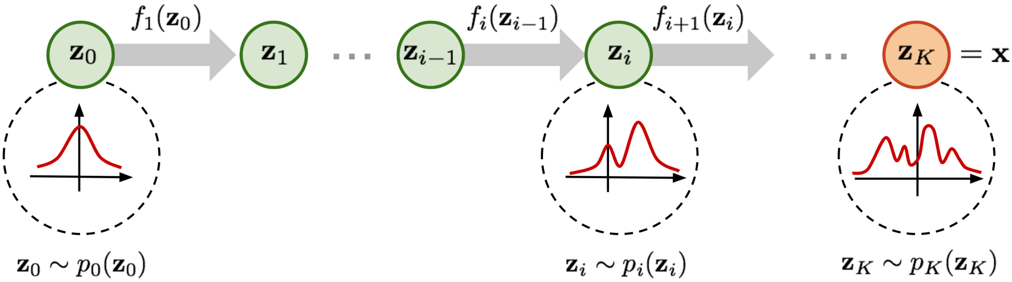
Normalizing Flows

Talgat Daulbaev

June 19, 2019

Model	**Density**
Generative Adversarial Networks (GAN)	No
Variational Autoencoders (VAE)	Yes
Flow-based models	Yes

Normalizing Flows in General



Mappings f_k should be bijective and smooth

Change of Variables Theorem

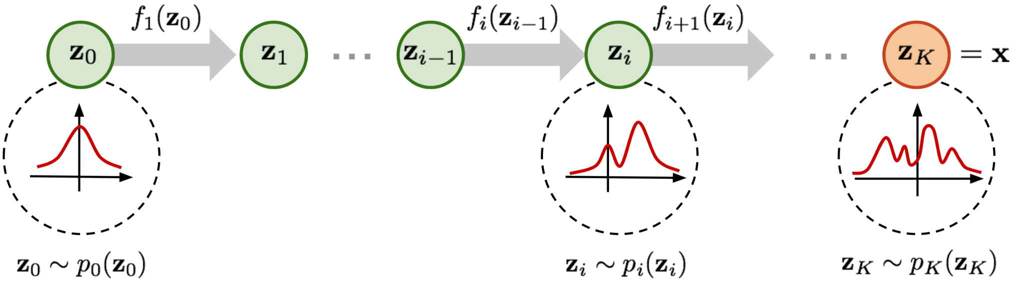
$$p(\boldsymbol{z}_k) = p(\boldsymbol{z}_{k-1}) \left| \det \frac{\partial f_k(\boldsymbol{z}_{k-1})}{\partial \boldsymbol{z}_{k-1}} \right|^{-1}$$

Image Credit: `lilianweng.github.io`

Tasks that should be managed

- Sampling $\boldsymbol{x} \sim p(\boldsymbol{x})$
- Estimating p at an arbitrary point $\boldsymbol{x}$

How to train the model



$$\mathbb{E}_{\mathbf{x}} \left[-\log p_\theta(\mathbf{x}) \right] \to \min_\theta$$

$$\mathbb{E}_{\mathbf{x}} \left[-\log p\left(f_\theta^{-1}(\mathbf{x})\right) + \log \det \left| \frac{\partial f_\theta(\mathbf{x})}{\partial \mathbf{x}} \right| \right] \to \min_\theta$$

Planar Flows

A family of mappings:

$$f(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{u}h(\boldsymbol{w}^\top \boldsymbol{z} + b),$$

where h is a smooth element-wise non-linearity

Density recomputations:

$$p(\boldsymbol{z}_k) = p(\boldsymbol{z}_{k-1}) \left| \det \frac{\partial f_k(\boldsymbol{z}_{k-1})}{\partial \boldsymbol{z}_{k-1}} \right|^{-1} = p(\boldsymbol{z}_{k-1}) \left| \det \left(I + \boldsymbol{u}h'(\boldsymbol{w}^\top \boldsymbol{z} + b)\boldsymbol{w}^\top \right) \right| =$$

$$= \left\{ \det(I + \boldsymbol{u}\boldsymbol{v}^\top) = 1 + \boldsymbol{v}^\top \boldsymbol{u} \right\} = p(\boldsymbol{z}_{k-1}) \left(1 + \boldsymbol{w}^\top \boldsymbol{u}h'(\boldsymbol{w}^\top \boldsymbol{z} + b) \right)$$

Real NVP

Affine coupling layer:

$$\begin{cases} \boldsymbol{y}_{1:d} = \boldsymbol{x}_{1:d} \\ \boldsymbol{y}_{d+1:D} = \boldsymbol{x}_{d+1:D} \odot \exp\left(s\left(\boldsymbol{x}_{1:d}\right)\right) + t\left(\boldsymbol{x}_{1:d}\right) \end{cases}$$

Inverse:

$$\begin{cases} \boldsymbol{x}_{1:d} = \boldsymbol{y}_{1:d} \\ \boldsymbol{x}_{d+1:D} = \left(\boldsymbol{y}_{d+1:D} - t\left(\boldsymbol{x}_{1:d}\right)\right) \odot \exp\left(-s\left(\boldsymbol{x}_{1:d}\right)\right) \end{cases}$$

Jacobian:

$$\begin{bmatrix} \boldsymbol{I}_{d\times d} & \mathbf{0}_{d\times(D-d)} \\ \frac{\partial \mathbf{y}_{d+1:D}}{\partial \mathbf{x}_{1:d}} & \operatorname{diag}\left(\exp\left(s\left(\mathbf{x}_{1:d}\right)\right)\right) \end{bmatrix}$$

"Density estimation using real NVP" by Dinh et al. (ICML 2017)

Glow

Invertible 1×1 convolutions

$$\log \left| \det \left(\frac{d \, \mathrm{conv2D}(\mathbf{h}; \mathbf{W})}{d\mathbf{h}} \right) \right| = h \cdot w \cdot \log |\det(\mathbf{W})|$$

$$\mathbf{W} = \mathbf{PL}(\mathbf{U} + \mathrm{diag}(\mathbf{s})) \quad \Longrightarrow \quad \log |\det(\mathbf{W})| = \mathrm{sum}(\log |\mathbf{s}|)$$

"Glow: Generative Flow with Invertible 1×1 Convolutions" by Kingma et al. (NeurIPS 2018)

Glow: Results

Autoregressive Flows (MAF and IAF)

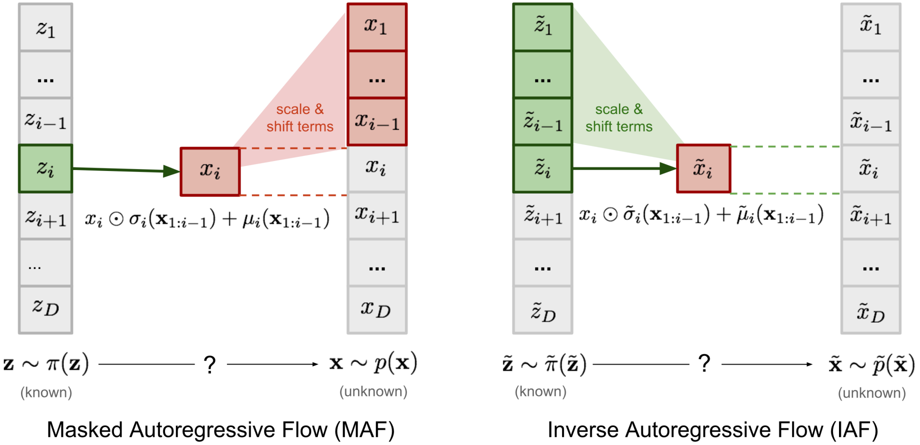


Image Credit: `lilianweng.github.io`

Continuous Normalizing Flows (CNF)

Normalizing Flows:

$$\log p(\boldsymbol{z}_k) = \log p(\boldsymbol{z}_{k-1}) - \log \left| \det \frac{\partial f_k(\boldsymbol{z}_{k-1})}{\partial \boldsymbol{z}_{k-1}} \right|$$

Continuous Normalizing Flows:

$$\begin{cases} \frac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}t} = f(\boldsymbol{z}, t) \\ \boldsymbol{z}(t_0) = \boldsymbol{z}_0 \end{cases}$$

If $f \in \mathrm{Lip}(\boldsymbol{z}) \cap \mathrm{C}(t)$:

$$\frac{\mathrm{d} \log p(\boldsymbol{z}(t))}{\mathrm{d}t} = -\mathrm{Tr}\left(\frac{\partial f_k(\boldsymbol{z}_{k-1})}{\partial \boldsymbol{z}_{k-1}} \right)$$

Neural Ordinary Differential Equations

$$\begin{cases} \dfrac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}t} = f(\boldsymbol{z}, t) \\ \boldsymbol{z}(0) = \boldsymbol{z}_0 \end{cases}$$

$$L\left(\mathbf{z}\left(t_1\right)\right) = L\left(\int_{t_0}^{t_1} f(\mathbf{z}(t), t; \theta)\, dt\right)$$

$$\frac{dL}{d\theta} = -\int_{t_1}^{t_0} \left(\frac{\partial L}{\partial \mathbf{z}(t)}\right)^T \frac{\partial f(\mathbf{z}(t), t; \theta)}{\partial \theta} dt$$

Estimating p at an arbitrary point x

$$\begin{bmatrix} \boldsymbol{z} \\ \log p(\boldsymbol{x}) - \log p(\boldsymbol{z}(0)) \end{bmatrix} = \int_{t_1}^{t_0} \begin{bmatrix} f(\boldsymbol{z}(t), t; \theta) \\ -\mathrm{tr}\left(\frac{\partial f}{\partial \boldsymbol{z}(t)}\right) \end{bmatrix} \mathrm{d}t,$$

$$\begin{bmatrix} \boldsymbol{z}(t_1) \\ \log p(\boldsymbol{x}) - \log p(\boldsymbol{z}(t_1)) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x} \\ 0 \end{bmatrix}$$

FFJORD

$$\begin{aligned}
\log p\left(\mathbf{z}\left(t_1\right)\right) &= \log p\left(\mathbf{z}\left(t_0\right)\right) - \int_{t_0}^{t_1} \operatorname{Tr}\left(\frac{\partial f}{\partial \mathbf{z}(t)}\right) dt \\
&= \log p\left(\mathbf{z}\left(t_0\right)\right) - \int_{t_0}^{t_1} \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\boldsymbol{\epsilon}^T \frac{\partial f}{\partial \mathbf{z}(t)} \boldsymbol{\epsilon}\right] dt \\
&= \log p\left(\mathbf{z}\left(t_0\right)\right) - \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\int_{t_0}^{t_1} \boldsymbol{\epsilon}^T \frac{\partial f}{\partial \mathbf{z}(t)} \boldsymbol{\epsilon}\, dt\right]
\end{aligned}$$