3/27 Normalizing Flows Cont'd

Invertible maps

$$\underset{\text{latent}}{\overset{\in \mathbb{R}^d}{z}} \longrightarrow \underset{\text{data}}{\overset{\in \mathbb{R}^d}{x}} \qquad x = f_\theta^{-1}(z)$$

$$z = f_\theta(x)$$

$z \sim N(0,1)^d$ "$q(z)$" $\quad x \sim p_{data}(x)$

$$\boxed{p_\theta(x) = q(z = f_\theta(x)) \left| \frac{\partial f_\theta}{\partial x} \right|}$$

can train w/ max likelihood loss

$$L = -\sum_{x \sim p_{data}} \log p_\theta(x)$$

need tractable Jacobian

$\left| J = \frac{\partial f_\theta}{\partial x} \right|$ $O(d^3)$ operations in general

But if J is upper/lower triangular only $O(d)$!

$$\begin{cases} z_1 = f_1(x_1) \\ z_2 = f_2(x_2; x_1) \\ \vdots \\ z_d = f_d(x_d; x_1, \dots x_{d-1}) \end{cases}$$

Autoregressive transform

$p(x_1) \longrightarrow q(z_1)$

$p(x_2 | x_1) \longrightarrow q(z_2)$

$\vdots$

$p(x_d | x_1, \dots x_{d-1}) \longrightarrow q(z_d)$

each step is transforming 1d conditional density

$$p(x_1, \dots x_d) = p(x_1) p(x_2|x_1) \cdots p(x_d | x_1, \dots x_{d-1})$$

so autoregressive can still model most general joint density!

need family of invertible ~~fns~~ 1d fns

— affine $z_n = a_n(x_1, \dots x_{n-1}) x_n + b_n(x_1, \dots x_{n-1})$

NNs

can eval all d coords w/ single pass — masking

"Masked Autoregressive Flow"
Papamakarios et al 2017

$x \to z$ fast (one pass NN) OE
$z \to x$ slow (d passes NN) gen.

IAF defined inverse, OE slow
gen fast

can't train w/ max likelihood
need distillation, teacher/student

---

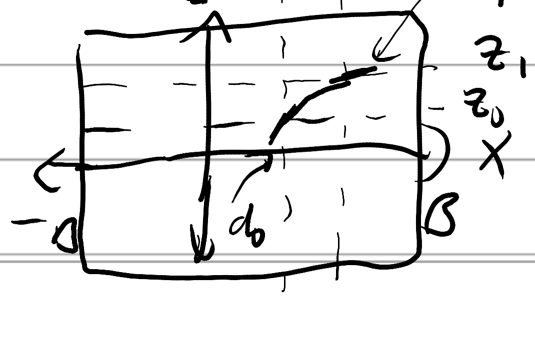
other transforms

– Rational Quadratic Spline

- monotonic
- has analytic inverse

$$z = f(x) = z_0 + (z_1 - z_0)\left(\frac{s\xi^2 + d_0\xi(1-\xi)}{s + (d_1 + d_0 - 2s)\xi(1-\xi)}\right)$$

$$s = \frac{z_1 - z_0}{x_1 - x_0} \qquad \xi = \left(\frac{x - x_0}{x_1 - x_0}\right) \in (0,1)$$



x_0, x_1
z_0, z_1 parameters of spline
d_0, d_1 NNs

— Coupling layers

(RealNVP Dinh + 2017)

some simple fn like affine or RQS

$$z_1 = x_1 \qquad z_{k+1} = C_{k+1}\left(x_{k+1}, \mu_{k+1}(x_1, \ldots x_k)\right)$$

$$\vdots \qquad\qquad \vdots$$

$$z_k = x_k \qquad z_d = C_d\left(x_d, \mu_d(x_1, \ldots x_k)\right)$$

equally fast DE & gen!

can be as expressive as MAF w/ perm layers

— Glow based on affine coupling layer

Kingma & Dhariwal 2018

+ multiscale

→ towardsdatascience explainer

+ MNIST example

CaloFlow (2106.05285) example