Continuous NFs 3/31

NF: $x_0 \xrightarrow{f_0} x_1 \xrightarrow{f_1} \cdots \xrightarrow{f_{N-1}} x_N$

$$p_i(x_i) = p_{i-1}(x_{i-1}) \left| \frac{\partial f_i}{\partial x_{i-1}} \right|^{-1}$$

- Drawbacks: – need tractable Jacobian & invertibility – expressivity
  – memory – NFs have trouble scaling up in d

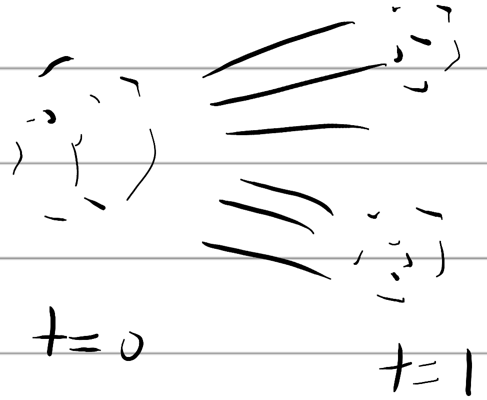
- Promising idea to overcome these limitations:
  CNFs (R. Chen + "Neural ODEs" NeurIPS 2018)

It's natural to consider $N \to \infty$, infinitesimal limit of NFs

Continuous flow: $x_i \to x_t \quad t \in [0,1]$

$p_i(x_i) \longrightarrow p_t(x_t)$ "probability path"

$$\begin{cases} x_0 = z \\ x_1 = x \end{cases} \qquad \begin{aligned} p_0 &= q(z) \\ p_1 &= p_{data}(x) \end{aligned}$$



Characterized by ODE

$$\frac{dx_t}{dt} = u_t(x_t)$$

$$dx_t = \underbrace{u_t(x_t)\,dt}_{\text{this is like infinitesimal transf}}$$

infinitesimal linear so don't need to worry about invertibility!

$$\underbrace{p_{t+dt}(x_{t+dt})} = p_t(x_t)\left|\frac{\partial x_t}{\partial x_{t+dt}}\right|$$

$$= p_t(x_t) + dt\,\frac{dp_t}{dt}$$

$$= p_t(x_t)\det\left(1+\frac{\partial u_t}{\partial x_t}dt\right)^{-1}$$

Log both sides

$$\log p_t + dt\,\frac{d\log p_t}{dt} = \log p_t - \underbrace{\log\det}_{=\mathrm{Tr}\log}\left(1+\frac{\partial u_t}{\partial x_t}dt\right)$$

$$\Rightarrow \boxed{\frac{d\log p_t}{dt} = -\vec{\nabla}_{x_t}\cdot u_t(x_t)}$$

Corollary:

$$\frac{d}{dt} p_t = \frac{\partial p_t}{\partial t} + \frac{\partial p_t}{\partial x} \cdot \frac{dx}{dt} = \frac{\partial p_t}{\partial t} + \frac{\partial p_t}{\partial x} \cdot u_t$$

$$\shortparallel$$

$$-\frac{\partial u_t}{\partial x} \cdot p_t \implies \boxed{\begin{gathered} \frac{\partial p_t}{\partial t} = -\nabla_x \cdot j(x) \\ j(x) = p_t(x) u_t(x) \end{gathered}}$$

Continuity eqn ... prob flux.

conservation of prob.

CNF - generation: $x_1 = x_0 + \int_0^1 dt\, u_t(x_t)$

CNF - density estimation
$$\begin{cases} x_0 = x_1 - \int_0^1 dt\, u_t(x_t) \\ p_1 = p_0(x_0) - \int_0^1 \vec{\nabla}_{x_t} \cdot u_t(x_t)\, dt \end{cases}$$

$\downarrow$

Gen & DE equal compute

— equally slow, must solve neural ODE for both!

Makes max likelihood training of CNFs problematic

- $\log p_\theta(x) = \log p_{base}(z) - \underbrace{\int_0^1 dt \, \mathrm{Tr} \frac{\partial u_t}{\partial x_t}}$

only trace not det

so $\mathcal{O}(d^3) \to \mathcal{O}(d)$

- But need to integrate ODE - trade $\mathcal{O}(d^3)$ for # time steps

$\Rightarrow$ still way too slow in practice!

- on the bright side, u_t can be arbitrary NN
  $\rightarrow$ no memory restrictions as d increases