CONTINUOUS NORMALIZING FLOW

Table of Contents

Discrete Normalizing Flow (DNF)

Goal: Imagine we want to turn a simple distribution into a more complex one

Apply a sequence of invertible tranformations

$\{f_1, \text{———}, f_k\}$, $f_i : \mathbb{R}^d \to \mathbb{R}^d$

$z_{i-1} \to z_i$

Train: $z_k = f_k \circ f_{k-1} \circ \text{———} \circ f_1(x)$

Infer: $x = f_k^{-1} \circ f_{k-1}^{-1} \circ \text{———} \circ f_1^{-1}(z_k)$

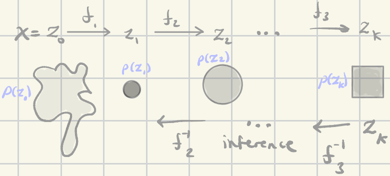
an image $x = z_0$, $z_k \in \mathbb{R}^d$, $z_k \sim N(0, I)$

$p(x)$ $p(z)$

Let $P_x(x)$, $P_z(z)$ be PDFs

$\Rightarrow$ Since f_i invertible,

$f_2 : z_1 \rightarrow z_2$



Discrete change-of-variables (cov)
(and log cov) formula

$$\log p(x) = \log \underbrace{p(z_K)}_{\text{Gaussian}} + \sum_{i=1}^{K} \log \left| \det \overbrace{\frac{\partial f_i(z_{i-1})}{\partial z_{i-1}}}^{J_{f_i} \text{ : jacobian}} \right| \quad (A)$$

what's the problem with discrete?

1) Invertibility: You need specially designed invertible blocks
2) Expressivity: Finite stacks struggle with complex dists unless made deep/structured which gets expensive
3) Optimization: More layers you stack $\rightarrow$ more computationally expensive training gets

From DNF $\rightarrow$ Continuous Normalizing Flow (CNF)

Each discrete step becomes an infinitesimal transformation and the composition converges to an ODE flow

$$\frac{dx}{dt} = f_\theta(x_t, t)$$

Learn a vector field that continuously drags a simple dist into the data dist

no more finite invertible layers

What is an ODE?

It tells you how something changes over time

$$\frac{dx}{dt} = f(x, t)$$

The instantaneous ROC wrt time

tells you the slope at that point

Continuous Normalizing Flow (CNF)

$x_0 \sim p_0$ is the transformed sample wwt match data

$x_T \sim N(0, I)$ is base distribution (say Gaussian)

Instantaneous Change-of-variables in continuous time

$$\frac{d}{dt} \lg p_t(x_t) = -\mathrm{Tr}\Big(\overbrace{\frac{\partial f_\theta(x_t, t)}{\partial x_t}}^{J_{f_\theta}}\Big)$$

$$\lg p(x_0) = \lg p(x_T) - \int_0^T \mathrm{Tr}\left(\frac{\partial f_\theta}{\partial x_t}\right) dt$$

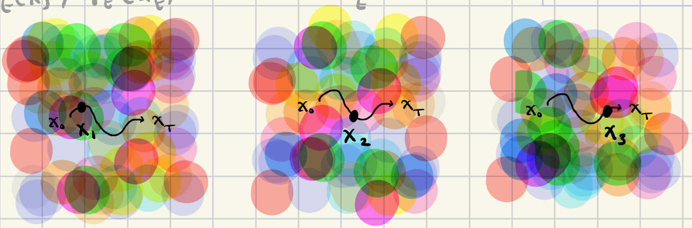
p_t = PDF indexed by time, so $p_t(x_t) = p(t)(x_t)$
$= p(x_t, t)$

whole flow is now a continuous trajectory $x_t : t \to x$

$x_t = x(t)$
$= (1-t)x + a$

Problem: Trace is the computational bottleneck!

$p_t(x) \neq p_t(x_t)$ $\quad$ p_t



$t = 1$ $\quad p_1(x_1) = a \in \mathbb{R}$ $\quad t = 2$ $\quad t = 3$

Think of p_t like temperature

Discrete Flow	Continuous Flows
Product / log-sum of Jacobian determinants	Limit of infinitely many infinitesimal Jacobians / integral of Jacobian trace
Transformation: A warp of space (invertible layer)	Continuous tranformation: a time-dependent velocity field that drags probability mass around

Total Derivative

x a coordinate $\mathbb{R}^d$

x_t a trajectory

and $g(x_t, t)$

when we differentiate we have to account for 2 things

i) How the field g changes with time at a fixed point $\frac{\partial g}{\partial t}$

ii) How the field changes because you're moving along the trajectory through space

$$\nabla_x g \cdot \frac{dx_t}{dt}$$

That's why, for $x \in \mathbb{R}^d$

Lagrangian — dot product — Eulerian

$g(x(t),t)$ $$\frac{d}{dt} g(x_t, t) = \frac{\partial g}{\partial x}(x_t, t) \cdot \frac{dx_t}{dt} + \frac{\partial g}{\partial t}(x_t, t)$$

where $\frac{\partial g}{\partial x}(x_t, t) = \left(\frac{\partial g_1}{\partial x_1}, \frac{\partial g_2}{\partial x_2}, \dots, \frac{\partial g_d}{\partial x_d} \right) = \nabla_x \cdot g(x_t, t)$

and $\frac{dx_t}{dt} = \left(\dot{x}_{t,1}, \dot{x}_{t,2}, \dots, \dot{x}_{t,d} \right)$ a vector

$\Longrightarrow$ $\frac{\partial g}{\partial x}(x_t, t) \cdot \frac{dx_t}{dt}$ a dot product $\longrightarrow$ scalar

Total Derivative — $\rho_t(x_t) = \rho(x_t, t)$

$$\frac{d}{dt} g(x_t, t) = \underbrace{\nabla_x g(x_t, t) \cdot \frac{dx_t}{dt}}_{\text{Lagrangian}} + \underbrace{\frac{\partial g}{\partial t}(x, t)\Big|_{x=x_t}}_{\text{Eulerian}}$$

$$\frac{d}{dt} g_t(x_t) = \nabla_x g_t(x_t) \frac{dx}{dt} + \frac{\partial}{\partial t} g_t(x)\Big|_{x=x_t}$$

So we take the derivative of $\frac{d}{dt} \log p_t(x_t)$

$$\frac{d}{dt} \log p_t(x_t) = \frac{\partial}{\partial t} \log p_t(x) \Big|_{x=x_t} + \nabla_x \log p_t(x_t, t) \frac{dx_t}{dt}$$

$$= \frac{\partial}{\partial t} \log p_t(x) \Big|_{x=x_t} + \nabla_x \log p_t(x_t, t)\, f_\theta(x_t, t)$$

How does log density of x evolve as we push it through a flow?

$$\boxed{\text{Recall } \frac{dx}{dt} = f_\theta(x_t, t)}$$

Why the Total Derivative?

Because 1) The density field p_t changes wrt time $\left(\frac{\partial}{\partial t}\right)$

2) The sample x_t also moves through space as time evolves $\left(\nabla_x \cdot \frac{dx}{dt}\right)$

Now we have the formula

$$\boxed{\frac{d}{dt} \log p_t(x_t) = \frac{\partial}{\partial t} \log p_t(x) \Big|_{x=x_t} + \nabla_x \log p_t(x_t, t)\, f_\theta(x_t, t)}$$

Why the $|_{x=x_t}$ Notation?

when applying the chain rule, we want be explicit that

- $\frac{\partial}{\partial t} \log p_t(x)$ means derivative wrt t while treating x constant
- Then evaluate the expression at the actual moving location x_t

$$\frac{d}{dt} \log p_t(x_t) = \frac{\partial}{\partial t} \log p_t(x) \Big|_{x=x_t} + \nabla_x \log p_t(x_t, t)\, f_\theta(x_t, t)$$

Continuity equation

$$\boxed{\frac{\partial}{\partial t} \lg p_t(x) = -\nabla_x f_\theta(x_t, t) - f_\theta(x_t, t) \nabla_x \lg p_t(x)}$$

$$= \left[-\nabla_x f_\theta(x_t, t) - \cancel{f_\theta(x_t, t) \nabla_x \log p_t(x)} \right]_{x=x_t} + \cancel{\nabla_x \log p_t(x_t, t) f_\theta(x_t, t)}$$

$$\therefore \boxed{\frac{d}{dt} \log p_t(x_t) = -\nabla_x \cdot f_\theta(x_t, t)}$$