Understanding Flow Matching

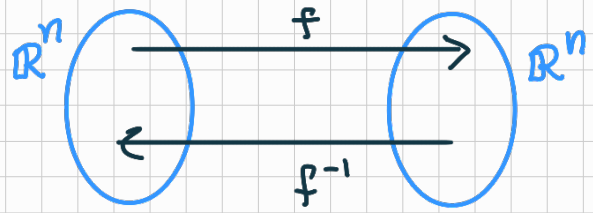
by Cristian Lazo Quispe

Author: Cristian Lazo Quispe

Diffeomorphism

It is a function $f : R^n \to R^n$ that is:

- Bijective (invertible)
- Differentiable (smooth)
- Has a differentiable inverse



⇨ It means the transformation is smooth and reversible, and there are no tears, folds, or discontinuities.

Given Random Variable $x \sim p(x)$ and diffeomorphism function Φ then

$$\tilde{x} = \Phi(x) \quad \text{and} \quad \tilde{P}(x) = \frac{P(\Phi^{-1}(\tilde{x}))}{\left|\det \frac{\partial \Phi}{\partial x}\right|}$$

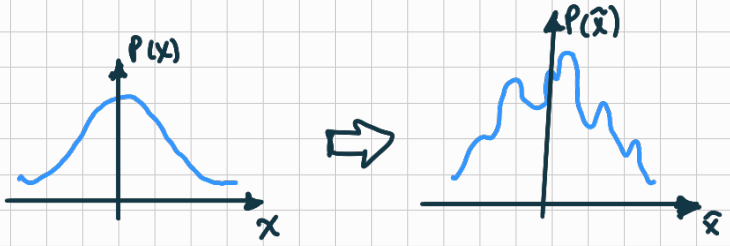
$\tilde{x}$ called twins

$\left|\det \frac{\partial \Phi}{\partial x}\right|$ Jacobian of Φ

★ Note: KL-divergence (P, q) satisfy diffeomorphism variance

$KLD(P, q) = KLD(\tilde{P}, \tilde{q})$ for any Φ

Continuity Equation



- In Physics

$$\frac{\partial \rho(x,t)}{\partial t} + \nabla \cdot (\rho(x,t) \cdot \vec{v}(x,t)) = 0$$

$\rho(x,t)$: mass density at x, t

$v(x,t)$: velocity field (how matter flow)

This means that mass is neither created nor destroyed only redistributed

- In Probability Distributions

$$\frac{\partial P(x,t)}{\partial t} + \nabla \cdot (P(x,t), \vec{v}(x,t)) = 0$$

$P(x,t)$: probability density at x, t

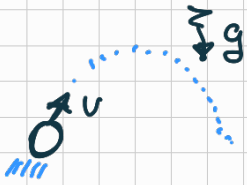
$v(x,t)$: vector field that transports samples

this equation ensures that the flow of samples and the evolution of the density are consistent

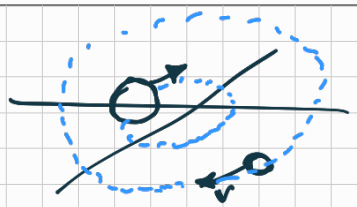
Author: Cristian Lazo Quispe

Differential Equations

$$\ddot{\theta} = -\frac{g}{L}\sin\theta - \mu\dot{\theta}(t)$$



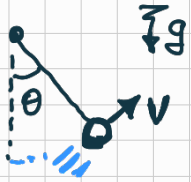
$$\ddot{y}(t) = -g$$



$$\ddot{\vec{x}}_i = G m_2 \left(\frac{\vec{x}_2 - \vec{x}_1}{\|\vec{x}_2 - \vec{x}_1\|}\right)\left(\frac{1}{\|\vec{x}_2 - \vec{x}_1\|^2}\right)$$

Sometimes it is hard to solve it. However, Vector field $\vec{v}$ is often the solution

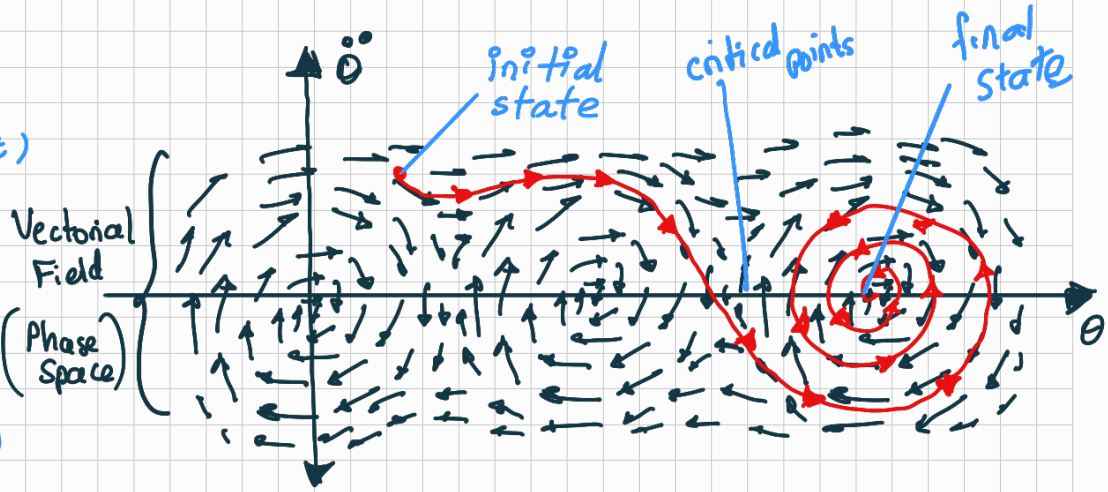
Example:



$$\ddot{\theta} = -\frac{g}{L}\sin\theta - \mu\dot{\theta}(t)$$

Second order ODE

Computational solution using Vector field



• Phase Space (Spaces encoding all kind of states of changing systems)

ODE Solver:

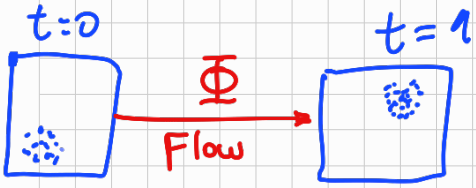
$$\dot{x}(t) = \vec{v}(x,t) = f(x,t)$$

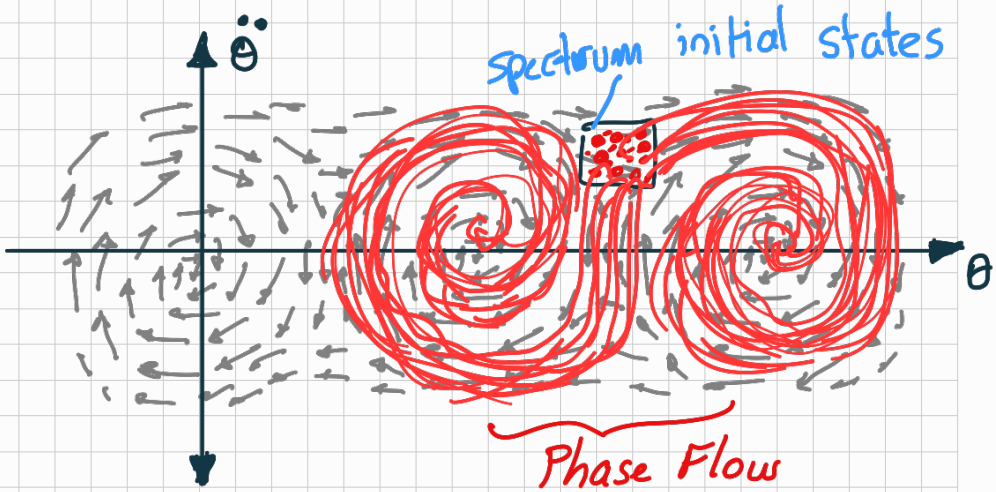
$$x(t) = x_0 + \int_0^t \vec{V}(x(\tau),\tau)\,d\tau$$

$x(t)$: t state; x_0: initial state; $\vec{V}$: vector field

$$X(t) = \Phi_{t_0}^{t}(x_0)$$

Flow map





• Phase Flow (collection of all possible trajectories)

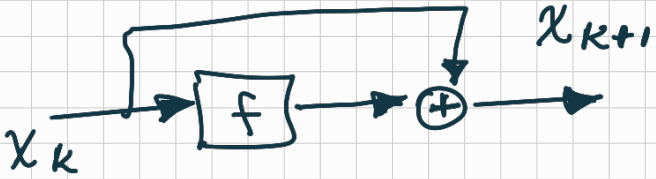
Author: Cristian Lazo Quispe

Neural ODE

Model dynamical system

Residual Network (Resnet)

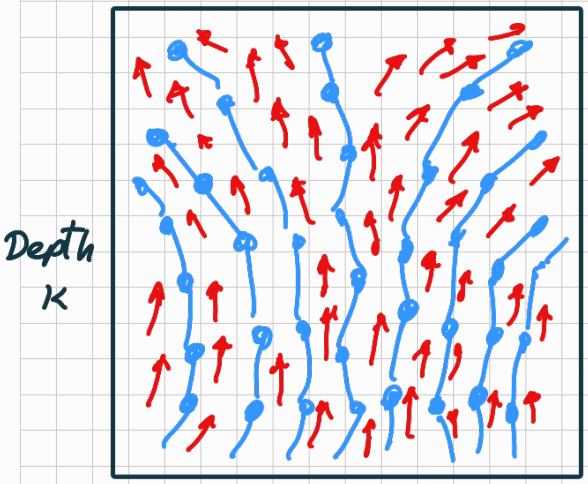
discrete



$$x_{k+1} = x_k + f(x_k)$$

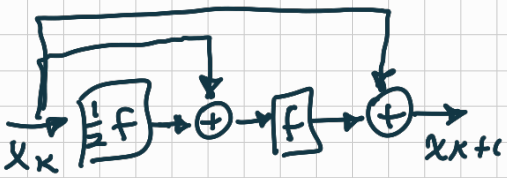
$$x_{k+1} = x_k + \Delta x_k$$

it can be seen as euler integrator!



input / hidden / output

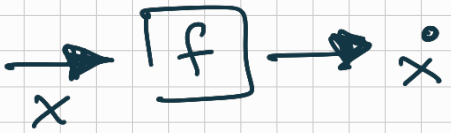
better approximation



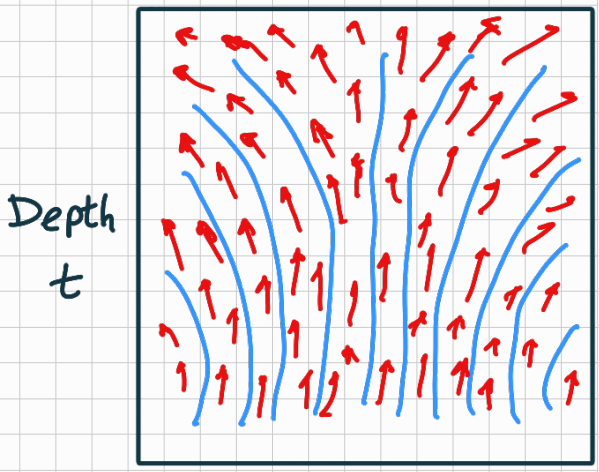
$$x_{k+1} = x_k + f\left(x_k + \tfrac{1}{2} f(x_k)\right)$$

Neural ODE

continuous



$$\dot{x} = \frac{dx}{dt} = f_\theta(x)$$



input / hidden / output

$$x_{t_0+\Delta t} = x_{t_0} + \int_{t_0}^{t_0+\Delta t} f_\theta(x(z))\, dz$$

flow map, ϕ

Free parameters: $f_\theta, t_0, \Delta t$

Hidden state: $x(z)$

$$x(z) : \Phi(x(t_0), t_0, z; \theta)$$

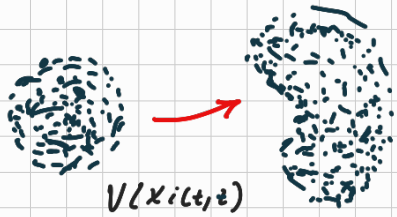
Author: Cristian Lazo Quispe

Optimal Transport

Problem of transforming one distribution into another in the most cost-efficient way

Particles perspective (evolution) location

$$\dot{x}_i(t) = V(x_i(t), t)$$
$$x_i(0) = x_{i,0} \quad \text{initial state}$$



$V(x_i(t), t)$ is globally Lipschitz

by there is a constant L

$$\|f(x) - f(y)\| \leq L \|x - y\| \quad \forall x, y$$

It means that it is smooth at there is not a crazy jump in the function or infinity slope

For V,

$V(x,t)$ Velocity field

if $\rho^N(0) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_{i,0}} \xrightarrow{N \to \infty} \rho_0(x)$, (delta dirac)

then $\rho^N(t) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_i(t)} \xrightarrow{N \to \infty} \rho(x,t)$

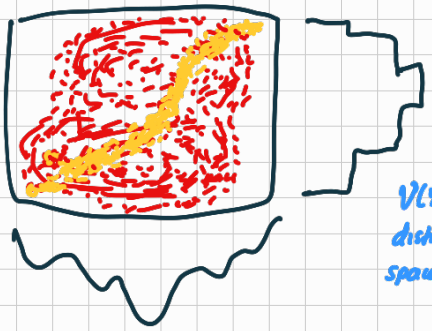
(all probability mass at a single point x)

I have N particles moving in space. Their collective presence approximates a density

set of all joint distributions (couplings)

Π

π_0 optimal coupling



$V(y)$ probability distribution over space Y (target)

$\mu(x)$ probability distribution over space X (source)

$\Pi(\mu, \nu)$ set of all joint distributions of transport μ to ν

$C(x,y)$ cost of moving mass from x to y

the Kantorovich Formulation

$$\min_{\pi \in \Pi(\mu,\nu)} \int_{X \times Y} c(x,y)\, d\pi(x,y)$$

Searching over all possible joint distributions $\pi(x,y)$ that match μ and ν.

Wasserstein Distance

$$W_2(\mu,\nu) = \left(\min_{\pi \in \Pi(\mu,\nu)} \int_{X \times Y} \|x-y\|^2 \, d\pi(x,y) \right)^{1/2}$$

It is the optimal transport cost

- Displacement interpolation:

McCann introduces the idea of displacement interpolation a way to smoothly interpolate between 2 distributions μ and ν using the optimal transport T.

identity map

$$\rho_t = ((1-t) \cdot id + tT) \# \mu$$

$$\pi_t(x,y) = (1-t)x + ty$$

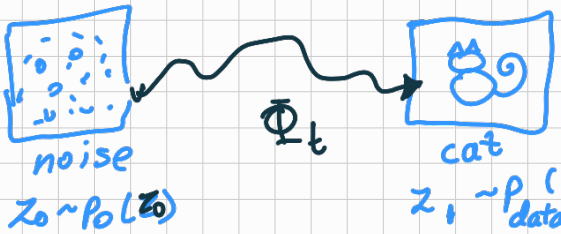
Boundaries:

$$\pi_0(x,y) = x$$
$$\pi_1(x,y) = y$$

Author: Cristian Lazo Quispe

CNF — Continuous Normalizing Flows (CNFs)

Learn a transformation from latent $z_0 \sim p_0(z)$ to data $z(1) \sim p_{data}(z_1)$



assume Φ_t diffeomorphic mapping

$\Phi = f_\theta(x,t)$ model

$\Phi_0(x) = x$

Vector Field: $V: \underbrace{[0,1]}_{t} \times \underbrace{\mathbb{R}^d}_{x} \to \underbrace{\mathbb{R}^d}_{v}$

It defines $\frac{dz(t)}{dt} = V(z(t), t) = \Phi = f_\theta(x,t)$

$z = \Phi_t(x)$

$\frac{d(\Phi_t(x))}{dt} = V_t(\Phi_t(x))$

Probability density path: $P: \underbrace{[0,1]}_{t} \times \underbrace{\mathbb{R}^d}_{x} \to \underbrace{\mathbb{R}_{>0}}_{\text{number real}}$

in instance of time

$p_t = [\phi_t]_* p_0(x) = p_0(\phi_t^{-1}(x)) \det\left[\frac{\partial \phi_t^{-1}(x)}{\partial x}\right]$

push-forward

$p(t,x) \to \int_{\mathbb{R}^d} p(t,x)\,dx = 1 \quad \forall\, t \in [0,1]$

(number)

It means all probability densities function over the time

Objective: $L_{CNF} = -\log p_1(z(1))$ We want to maximize the log-probability of data samples

It means learning a map from z_0 to z_1, along with the exact densities $p(z_0)$ and $p(z_1)$, by defining a Flow from z_0 to z_1, which consequently induces a flow from $p(z_0)$ to $p(z_1)$.

How to solve:

$$\boxed{\frac{\partial p(x,t)}{\partial t} + \nabla \cdot (p(x,t)\cdot \vec{v}(x,t)) = 0}$$

theorem of liouville

$\frac{\partial p}{\partial t} + \nabla \langle p, v \rangle = 0$

$\frac{\partial p}{\partial t} + p\nabla v + \nabla p v = 0$

$\frac{\partial p}{\partial t} + \nabla p \cdot v = -p\nabla v$

$\frac{dp}{dt} = \frac{\partial p}{\partial t} + \frac{\partial p}{\partial z}\cdot\frac{\partial z}{\partial t} = -p\nabla v$

$\frac{dp}{dt} = -p\nabla \cdot v$

$\frac{1}{p}\frac{dp}{dt} = -\nabla \cdot v = \frac{d \log p_t(x)}{dt} = -\nabla \cdot V_t(x)$

$\log p(z(1)) = \log p(z(0)) - \int_0^1 \nabla \cdot v\, dt$

$L_{CNF} = -\log p(z(0)) + \int_0^1 \nabla \cdot v\, dt$

Author: Cristian Lazo Quispe

$$L_{CNF} = -\log(P(z_0)) + \int_0^1 \nabla \cdot V \, dt$$

d : dimension of input

$$\nabla \cdot V = tr\left(\frac{\partial V(z)}{\partial z}\right)$$

$$tr(J) = \sum_{i=1}^{d} \frac{\partial V(z_i)}{\partial z_i}$$

Jacobian trace → expensive to compute

Image: $d = C \times h \cdot w$

RGB → $3 \times h \cdot w$

$$J = \frac{\partial V}{\partial z} = \begin{bmatrix} \partial V/\partial z_1 \\ \vdots \\ \partial V/\partial z_4 \end{bmatrix}$$

how to solve it → Hutchison's trick

given $\epsilon \sim N(0, I)$

$$\nabla \cdot V = tr(J) = E_\epsilon[\epsilon^T J \epsilon]$$

$$= E[\epsilon^T \nabla_z (\epsilon^T V(z))]$$

$\epsilon^T \nabla_z (\epsilon^T V(z))$ is simple compute

It could be computed with autograd over z

$$\epsilon^T V = \sum \epsilon_i V_i$$

$$\nabla_z \epsilon^T V = \sum \epsilon_i \nabla_z V_i = J^T \epsilon$$

$\epsilon^T V$: scalar; $J^T \epsilon$: scalar

$$L_{CNF} = -\log p_1(z(1)) = -\log(P(z_0)) + \int_0^1 tr\left(\frac{\partial V}{\partial z}\right) dt$$

$$L_{CNF} = -\log p_1(z(1)) = -\log(P(z_0)) + \int_0^1 E[\epsilon^T \nabla_z (\epsilon^T V(z))] dt$$

the original paper proposes this : We don't know z_0 that generates $z(1)$ directly → how is it obtained ?

Doing Flow: $z(1) \xrightarrow{\Phi} z(0)$

$$z_1 = z_0 + \int_0^1 \vec{V}(z(t), t) \, dt$$

$$z_0 = z_1 - \int_0^1 \vec{V}(z(t), t) dt = z_1 + \int_1^0 \vec{V}(z(t), t) dt$$

Training:

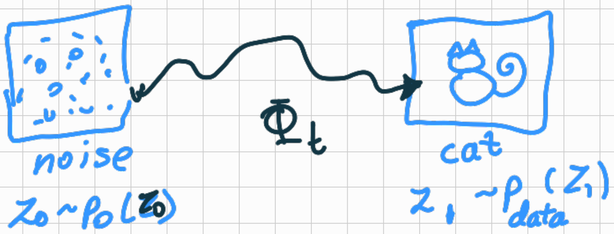
$$L_{CNF} = -\log p_1(z(1)) = -\log\left(P\left(z_1 + \int_1^0 \vec{V}(z(t), t) dt\right)\right) - \int_1^0 E[\epsilon^T \nabla_z (\epsilon^T V(t))] dt$$

Inference: $z_1 = z_0 + \int_0^1 \vec{V}(z(t), t) \, dt$

Author: Cristian Lazo Quispe

Flow Matching

Learn a transformation from latent $z_0 \sim p_0(z)$ to data $z(1) \sim p_{data}(z_1)$



assume Φ_t diffeomorphic mapping

$\Phi = f_\theta(x,t)$ model

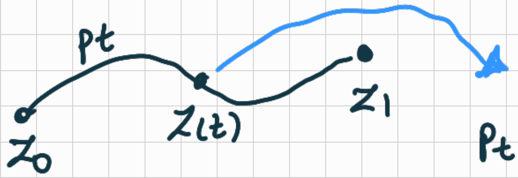
$\Phi_0(x) = x$

Vector Field: It defines $\frac{dz(t)}{dt} = V(z(t), t)$ $\quad \Phi = f_\theta(x,t)$

$V: \underbrace{[0,1]}_{t} \times \underbrace{R^d}_{x} \to \underbrace{R^d}_{v}$

$z = \Phi_t(x)$ $\qquad \frac{d(\Phi_t(x))}{dt} = V_t(\Phi_t(x))$

Probability density path:



in instance of time

$p_t = [\Phi_t]_* p_0(x) = p_0(\Phi_t^{-1}(x)) \det\left[\frac{\partial \Phi_t^{-1}}{\partial x}(x)\right]$

↑ push-forward

$P: \underbrace{[0,1]}_{t} \times \underbrace{R^d}_{x} \to \underbrace{R_{>0}}_{\text{number real}}$

$\underbrace{p(t,x)}_{\text{number}} \to \int_{R^d} p(t,x)\,dx = 1 \quad \forall t \in [0,1]$

it means all probability densities function over the time

Objective: $\mathcal{L}_{FM} = E_{t, p_t(x)} \| \underset{\text{model}}{V_t(x)} - u_t(x) \|^2$

We want to reduce the mean square error of the vector field predicted over the real

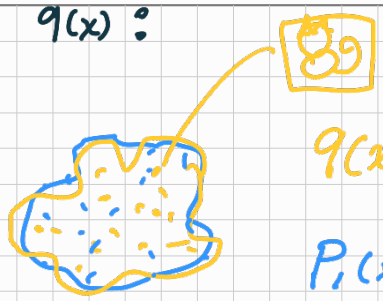
😎 It means learning a map from an intermediate point z_t to z_1, guided by the exact flow of the vector field $V(t,x)$

How to solve it: how to get $u_t(x)$ and $p_t(x)$

We don't have access to a specific path that maps z_0 to z_1, nor a closed-form expression for u_t that generates the desired p_t, we need to compute it somehow. In contrast, CNF don't care about the explicit path or the vector field, CNF only focus on the start and end distributions connected through a continuous flow.

Author: Cristian Lazo Quispe

Define $q(x)$:



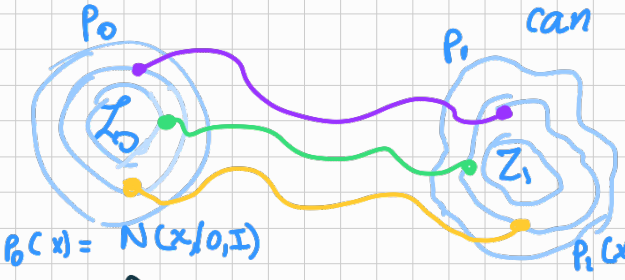
$q(x)$: unknow distribution (True world) $p_1(x) \simeq q(x)$

$p_1(x)$: target distribution (model world) we have samples parametric model

Define $p_t(x)$:

how to define $p_t(x)$? there are many probability paths that can satisfy $p_1(x) \simeq q(x)$ & $p_0(x) = z_0$ noise

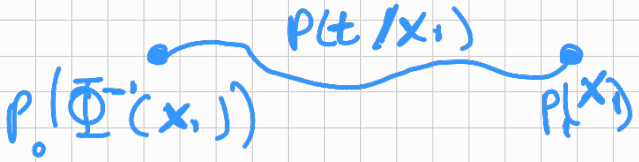
(Phase Flow)



Paper denotes:
$p_0(x) = p$
$p_1(x) = q$

- Define conditional probability path $p_t(x/x_1)$

we have data and x_1 is a data point



given x_1 we denote

$p_t(x/x_1)$ that satisfy

$p_0(x/x_1) = p(x)$ ← noise, $t=0$

$p_1(x/x_1)$ = distribution concentrated around $x = x_1$

- Define marginal probability path

$$p_t(x) = \int p_t(x, x_1)\, dx_1$$

$$p_t(x) = \int p_t(x/x_1)\, q(x_1)\, dx_1$$

$$p_t(x) = \int p_t(x/x_1)\, q(x_1)\, dx_1 \quad \text{where} \quad p_1(x) = \int p_1(x/x_1) q(x_1) dx_1 \simeq q(x)$$

- Define marginal vector field

$$\boxed{\frac{\partial p_t(x)}{\partial t} + \nabla \cdot (p_t(x) \cdot u_t(x)) = 0}$$

$$\rightarrow \frac{\partial p_t(x/x_1)}{\partial t} + \nabla \cdot (p_t(x/x_1) \cdot u(x/x_1)) = 0$$

$$q(x_1) \cdot \left[\frac{\partial p_t(x/x_1)}{\partial t} + \nabla \cdot (p_t(x/x_1) \cdot u(x/x_1)) \right] = 0$$

$$\frac{\partial p_t(x/x_1)\, q(x_1)}{\partial t} + \nabla \cdot (p_t(x/x_1) \cdot u(x/x_1) \cdot q(x_1)) = 0$$

$$\int \frac{\partial p_t(x/x_1) q(x_1)}{\partial t} dx_1 + \nabla \cdot \left(\int p_t(x/x_1) \cdot u(x/x_1) q(x_1)\, dx_1 \right) = 0$$

Author: Cristian Lazo Quispe

$$\int \frac{\partial p_t(x|x_1) q(x_1)}{\partial t} dx_1 + \nabla \cdot \left(\int p_t(x|x_1) \cdot u(x|x_1) q(x_1) dx_1 \right) = 0$$

$$\frac{\partial p_t(x)}{\partial t} + \nabla \cdot \left(\int p_t(x|x_1) \cdot u(x|x_1) q(x_1) dx_1 \right) = 0$$

$$\boxed{\frac{\partial p_t(x)}{\partial t} + \nabla \cdot \left(p_t(x) \cdot u_t(x) \right) = 0}$$

$$p_t(x) \cdot u_t(x) = \int p_t(x|x_1) \cdot u(x|x_1) q(x_1) dx_1$$

$$u_t(x) = \frac{1}{p_t(x)} \int p_t(x|x_1) \, u(x|x_1) \, q(x_1) dx_1$$

$$u_t(x) = \int u_t(x|x_1) \cdot \frac{p_t(x|x_1) \, q(x_1)}{p_t(x)} dx_1$$

where $u_t(x|x_1) : \mathbb{R}^d \to \mathbb{R}^d$ is a conditional vector field that generates $p_t(x|x_1)$

Paper key observation:

The marginal vector field $u_t(x)$ generates the marginal probability path $p_t(x)$

this generates the following theorem 1

Theorem 1:

Given vector fields $u_t(x|x_1)$ that generate conditional probability paths $p_t(x|x_1)$, for any distribution $q(x_1)$, the marginal vector field u_t generates the marginal probability path p_t, i.e., u_t and p_t satisfy the continuity equation

$$L_{FM} = E_{t, p_t(x)} \| v_t(x) - u_t(x) \|^2$$

model

$u_t(x)$ and $p_t(x)$ are still intractable to compute

We could trying to solve it in a simple way

$$u_t(x) = \int u_t(x|x_1) \cdot \frac{p_t(x|x_1) \, q(x_1)}{p_t(x)} dx_1$$

Author: Cristian Lazo Quispe

$$L_{FM} = E_{t, p_t(x)} \| \underset{\text{model}}{v_t(x)} - u_t(x) \|^2$$

$$E_{t, x\sim p_t(x)} \left(\|v_t(x)\|^2 - \langle 2\, v_t(x) \cdot u_t(x) \rangle + \|u_t(x)\|^2 \right)$$

$$E_{t, x\sim p_t(x)} \|v_t(x)\|^2 = \int \|v_t(x)\|^2 p_t(x)\,dx = \int \|v_t(x)\|^2 \int p_t(x|x_1)\, q(x_1)\, dx_1\, dx$$

$$= \iint \|v_t(x)\|^2 p_t(x|x_1)\, q(x_1)\, dx_1\, dx$$

$$E_{t, x\sim p_t(x)} \|v_t(x)\|^2 = E_{t, x\sim p_t(x|x_1),\, x_1 \sim q(x_1)} \|v_t(x)\|^2$$

$$E_{t, x\sim p_t(x)} \| \langle 2\, v_t(x) \cdot u_t(x) \rangle \| = \int \langle 2 v_t(x) \cdot u_t(x) \rangle\, p_t(x)\,dx$$

$$2 \int v_t(x) \cdot u_t(x)\, p_t(x)\,dx$$

$$2 \int v_t(x) \cdot \int u_t(x|x_1) \underbrace{\frac{p_t(x|x_1)\, q(x_1)}{p_t(x)}}\, dx_1 \cancel{p_t(x)}\, dx$$

$$= 2 \int v_t(x) \cdot \int u_t(x|x_1)\, p_t(x|x_1)\, q(x_1)\, dx_1\, dx$$

$$= 2 \iint \underbrace{v_t(x) \cdot u_t(x|x_1)}_{\langle 2 v_t(x) \cdot u_t(x|x_1) \rangle} p_t(x|x_1)\, q(x_1)\, dx_1 dx$$

$$E_{t, x\sim p_t(x)} \| \langle 2\, v_t(x) \cdot u_t(x) \rangle \| = E_{t,\, x\sim p_t(x|x_1),\, x_1 \sim q(x_1)} \| \langle 2 v_t(x) \cdot u_t(x|x_1) \rangle \|$$

$$E_{t, x\sim p_t(x)} \|u_t(x)\|^2 = \int \|u_t(x)\|^2 p_t(x)\,dx = \int \|u_t(x)\|^2 \int p_t(x|x_1)\, q(x_1)\, dx_1\, dx$$

$$= \iint \|u_t(x)\|^2 p_t(x|x_1)\, q(x_1)\, dx_1\, dx$$

$$E_{t, x\sim p_t(x)} \|u_t(x)\|^2 = E_{t,\, x\sim p_t(x|x_1),\, x_1 \sim q(x_1)} \|u_t(x)\|^2$$

$$E_{t, p_t(x)} = \|v_t(x) - u_t(x)\|^2 = E_{t,\, x\sim p_t(x|x_1),\, x_1 \sim q(x_1)} \left(\|v_t(x)\|^2 - \langle 2\, v_t(x) \cdot u_t(x|x_1) \rangle + \|u_t(x)\|^2 \right)$$

Author: Cristian Lazo Quispe

$$E_{t,p_t(x)} = \|v_t(x) - u_t(x)\|^2 = E_{t,\, x\sim p_t(x/x_1),\, x_1\sim q(x_1)} \left(\|v_t(x)\|^2 - \langle 2\, v_t(x) \cdot u_t(x/x_1)\rangle + \|u_t(x)\|^2 \right)$$

$$E_{t,\, x\sim p_t(x/x_1),\, x_1\sim q(x_1)} \left(\|v_t(x)\|^2 - \langle 2\, v_t(x) \cdot u_t(x/x_1)\rangle + \|u_t(x)\|^2 \right) + \|u_t(x/x_1)\|^2 - \|u_t(x/x_1)\|^2$$

$$E_{t,\, x\sim p_t(x/x_1),\, x_1\sim q(x_1)} \left(\|v_t(x)\|^2 - \langle 2\, v_t(x) \cdot u_t(x/x_1)\rangle + \|u_t(x/x_1)\|^2 + \|u_t(x)\|^2 - \|u_t(x/x_1)\|^2 \right)$$

$$E_{t,\, x\sim p_t(x/x_1),\, x_1\sim q(x_1)} \left(\|v_t(x) - u_t(x/x_1)\|^2 + \|u_t(x)\|^2 - \|u_t(x/x_1)\|^2 \right)$$

not depend of the model in the optimization

$$L_{CFM} = E_{t,\, x\sim p_t(x/x_1),\, x_1\sim q(x_1)} \|v_t(x) - u_t(x/x_1)\|^2 \qquad t \sim U[0,1]$$

The 2nd Key observation is therefore:

> The L_{FM} and L_{CFM} objectives have identical gradients w.r.t θ

this generate the following theorem 2

> **Theorem 2.** Assuming that $p_t(x) > 0$ for all $x \in \mathbb{R}^d$ and $t \in [0,1]$, then, up to a constant independent of θ, L_{CFM} and L_{FM} are equal. Hence, $\nabla_\theta L_{FM}(\theta) = \nabla_\theta L_{CFM}(\theta)$

Now we known that we can express the objective with a conditional probability $p_t(x/x_1)$ and conditional vector field $u_t(x/x_1)$.

How to construct $p_t(x/x_1)$ and $u_t(x/x_1)$?

↓

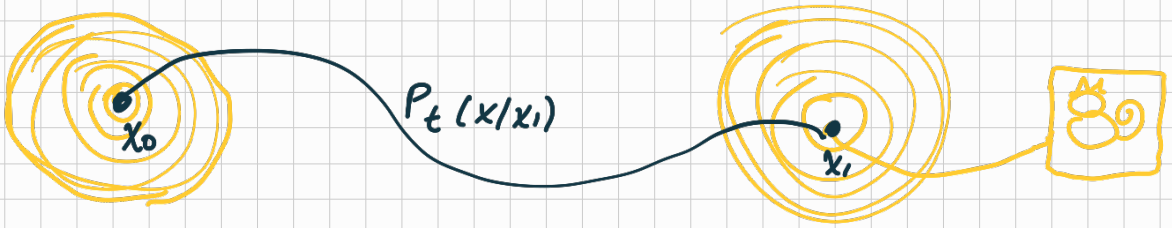
We know that $\mathcal{L}_{CFM}$ works with any choice of $p_t(x/x_1)$ and $u_t(x/x_1)$

Let's consider $p_t(x/x_1) = \mathcal{N}(x \mid \mu_t(x_1), \sigma_t(x_1)^2 I)$

family of Gaussian conditional probability paths.

where $\mu : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ is the time-dependent mean of the Gaussian Distribution

$\sigma : [0,1] \times \mathbb{R} \to \mathbb{R}_{>0}$ describes a time-dependent scalar standard desviation (std)



we set

$\mu_0(x_1) = 0$

$\sigma_0(x_1) = 1 \to p_0(x) = p(x) = \mathcal{N}(x \mid 0, I)$ (noise)

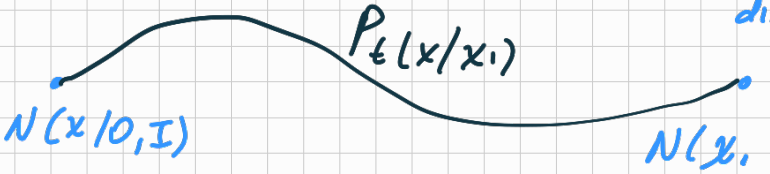
$p_0(x/x_1) = p_0(x)$

$\mu_1(x_1) = x_1$

$\sigma_1(x_1) = \sigma_{min}$ sufficiently small

so that

$p_1(x/x_1)$ is concentrated Gaussian distribution centered at x_1.



there are many vector fields V_t that generate any particular probability path and many Flow Φ_t because $\frac{d(\Phi_t(x))}{dt} = V_t(\Phi_t(x))$

↳ let's consider a flow $\psi_t(x)$ as a canonical transformation for Gaussian distributions (conditioned on x_1)

$$\psi_t(x, x_1) = \psi_t(x) = \sigma_t(x_1) \cdot x + \mu_t(x_1)$$

$$[\psi_t]_* p(x) = p_t(x/x_1)$$

$$\boxed{\frac{d}{dt}\phi_t(x) = u_t(\phi_t(x))}$$

This flow then provides a vector field that generates the conditional probability path: $\frac{d}{dt}\psi_t(x) = u_t(\psi_t(x)/x_1)$

$\psi_0(x_0) = x_0$

$\sigma_0(x_1) \cdot x_0 + \mu_0(x_1) = x_1$

$1 \cdot x_0 + 0 = x_1$

$x_1 = x_1$

$\psi_1(x_1) = x_1$

$\sigma_1(x_1) \cdot x_1 + \mu_1(x_1) = x_1$

$\sigma_{min} \cdot x_1 + x_1 = x_1$

$x_1 \approx x_1$

Satisfies the property of diffeomorphic transformation

$$\psi_t(x) = \sigma_t(x_1)\cdot x + \mu_t(x_1) \qquad x = \frac{\psi_t(x) - \mu_t(x_1)}{\sigma_t(x_1)}$$

$$\frac{d}{dt}\psi_t(x) = u_t(\psi_t(x)/x_1) = \frac{d}{dt}\left[\sigma_t(x_1)\cdot x + \mu_t(x_1)\right]$$

$$\frac{d\sigma_t}{dt}(x_1)\cdot x + \frac{d\mu_t(x_1)}{dt}$$

$$\frac{d}{dt}\psi_t(x) = u_t(\psi_t(x)/x_1) = \frac{d\sigma_t}{dt}(x_1)\cdot\left[\frac{\psi_t(x) - \mu_t(x_1)}{\sigma_t(x_1)}\right] + \frac{d\mu_t}{dt}(x_1)$$

it allows to use any x value in the formulation

$$\frac{d}{dt}\psi_t(x) = w_t(\psi_t(x)) = u_t(\psi_t(x)/x_1)$$

$$w_t(x) = u_t(x/x_1) = \frac{\sigma_t'(x_1)}{\sigma_t(x_1)}(x - \mu_t(x_1)) + \mu_t'(x_1)$$

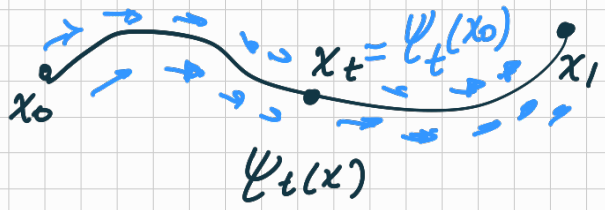
Theorem 3: Let $p_t(x/x_1)$ be a Gaussian probability path, and ψ_t its corresponding flow map. Then, the unique vector field that defines ψ_t has the form:

$$u_t(x/x_1) = \frac{\sigma_t'(x_1)}{\sigma_t(x_1)}(x - \mu_t(x_1)) + \mu_t'(x_1)$$

Consequently $u_t(x/x_1)$ generates the Gaussian path $p_t(x/x_1)$

$$L_{CFM} = E_{t,\, x\sim p_t(x/x_1),\, x_1\sim q(x_1)} \| V_t(x) - u_t(x/x_1)\|^2$$

Writing in terms of the intermediate point x_t



We know that $\psi_t(x_0) = x_t$ flow of x_0

$$L_{CFM} = E_{t,\, p_t(x/x_1),\, q(x_1)} \| V_t(x_t) - u_t(x_t/x_1)\|^2$$

$$E_{t,\, p_t(x_0/x_1),\, q(x_1)} \| V_t(\psi_t(x_0)) - u_t(\psi_t(x_0)/x_1)\|^2$$

$$E_{t,\, p(x_0),\, q(x_1)} \| V_t(\psi_t(x_0)) - \frac{d}{dt}\psi_t(x_0)\|^2$$

$p(x_0)$: noise; $q(x_1)$: data; V_t: model

Remember that $\psi_t(x)$ is conditioned on x_1

Author: Cristian Lazo Quispe

$$p_t(x|x_1) = \mathcal{N}(x \mid \mu_t(x_1), \sigma_t(x_1)^2 I)$$

$$\psi_t(x_0, x_1) = \psi_t(x) = \sigma_t(x_1)\, x + \mu_t(x_1)$$

$$u_t(x) = u_t(x|x_1) = \frac{\sigma_t'(x_1)}{\sigma_t(x_1)}\left(x - \mu_t(x_1)\right) + \mu_t'(x_1)$$

Boundaries:
$\psi_0(x_0) = x_0$
$\psi_1(x_1) = x_1$

$$\mathbb{E}_{t,\ p(x_0),\ q(x_1)} \left\| v_t(\psi_t(x_0)) - \frac{d}{dt}\psi_t(x_0) \right\|^2$$

t: noise; $p(x_0)$: data; v_t: model

Now $\mathcal{L}_{CFM}(\theta)$ is in terms of x_0 and $\psi_t(x_0, x_1)$

this formulation is fully general for arbitrary functions $\mu_t(x_1)$ and $\sigma_t(x_1)$ we can set them to any differentiable function satisfying the desired boundary conditions.

Example 1: Diffusion conditional VFs

Diffusion models start with data points and gradually add noise until it approximates pure noise (stochastic process)

In diffusion: the reversed (noise → data) Variance Exploding (VE) path has the form

$$p_t(x) = \mathcal{N}(x/x_1, \sigma_{1-t}^2 I)$$

σ_t is an increasing function

$\sigma_0 = 0 \quad \sigma_1 >> 1$

$$\mu_t(x_1) = x_1 \qquad \sigma_t(x_1) = \sigma_{1-t}$$

$$u_t(x|x_1) = -\frac{\sigma_{1-t}'}{\sigma_{1-t}}(x - x_1)$$

The reversed (noise → data) Variance Preserving (VP) diffusion path has the form

$p_t(x|x_1) = \mathcal{N}(x/\alpha_{1-t}x_1, (1-\alpha_{1-t}^2)I)$, where

$\alpha_t = e^{-\frac{1}{2}T(t)}$, $T(t) = \int_0^t \beta(s)ds$

and $\beta(t)$ is the noise scale function

$$\mu_t(x_1) = \alpha_{1-t}x_1 \qquad \sigma_t(x_1) = \sqrt{1-\alpha_{1-t}^2}$$

$$u_t(x|x_1) = \frac{\alpha_{1-t}'}{1-\alpha_{1-t}^2}(\alpha_{1-t}x - x_1)$$

$$= -\frac{T'(1-t)}{2}\left[\frac{e^{-T(1-t)}x - e^{-\frac{1}{2}T(1-t)}x_1}{1-e^{-T(1-t)}}\right]$$

Note: These p_t were previously derived as solutions of diffusion process, they do not actually reach a true noise distribution in finite time.

Example 2: Optimal Transport conditional VFs

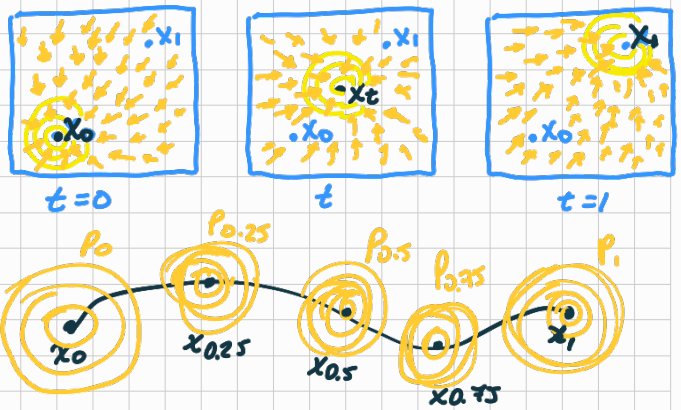
Let's define $\mu_t(x)$ and $\sigma_t(x)$ to simple change linearly in time.

$$\mu_t(x) = t\,x_1 \qquad \sigma_t(x) = 1 - (1-\sigma_{min})t$$

Conditional Flow

$$\psi_t(x_0, x_1) = \psi_t(x) = (1-(1-\sigma_{min})t)x + tx_1$$

$$w_t(x) = u_t(x|x_1) = \frac{x_1 - (1-\sigma_{min})x}{1-(1-\sigma_{min})t}$$



The conditional flow $\psi_t(x)$ is in fact the Optimal Transport (OT) displacement map between Gaussians $p_0(x|x_1)$ and $p_1(x|x_1)$

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t,\ q(x_1),\ p(x_0)} \left\| v_t(\psi_t(x_0)) - (x_1 - (1-\sigma_{min})x_0) \right\|^2$$

If σ_{min} is 0 then

$$\psi_t(x) = (1-t)x + tx_1$$

$$\psi_t(x) = x_t$$

Author: Cristian Lazo Quispe

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t,\, q(x_1),\, p(x_0)} \left\| v_t(\psi_t(x_0)) - (x_1 - (1-\sigma_{min})x_0) \right\|^2$$

Flow matching learns an exact vector field v_t to follow a simple, optimal transport-based path from noise to data, using only a closed-form loss over interpolated samples.
(simple and elegant generative model framework)

Personally, I consider these to be the main key ideas:

- Objective function of learning a vector field

$$v_t = \frac{d\psi_t}{dt} \qquad \mathcal{L}_{FM} \; \mathbb{E}_{t, p_t} \| v_t(x) - u_t(x) \|^2$$

- Conditional probability path as family of Gaussian

$$p_t(x/x_1) = \mathcal{N}(x \mid \mu_t(x_1), \sigma_t(x_1)^2 I)$$

- Optimal transport inspired interpolation

$$\mu_t(x) = t\, x_1 \qquad \sigma_t(x) = 1 - (1-\sigma_{min})t$$

Training ($\sigma_{min} = 0$):

```
t = random(0, 1)
x0 = noise
x_t = (1 - t)·x_0 + t x_1
u_t = (x_t - x_0)/(1 - t)
v_t = model(xt, t)
loss = (v_t - u_t)^2
```

Inference ($\sigma_{min} = 0$): (basic solver ODE)

```
x0 = noise
x = x0          dt = 1/N_steps
for i in N_steps:
    t = i · dt
    v = model(x, t)
    x = x + v dt
```

Author: Cristian Lazo Quispe