Flow & Diffusion Models

Theory

ODE/SDE

Goals:

① Flow and Diffusion models from first principles

② The necessary amount of mathematics.

③ Implement & apply these algos.

Structure:

1.
   a. Formalize generating an image
   b. Construct Flow & Diffusion models.

2. Define Training objective

3. Define training algorithm [score matching & flow matching]

④ Network architectures + Conditioning

⑤ Advanced Topics: Alignment, Complex Data Types, Distillation.

Lecture 1

SECTION-1 Generation to Sampling.

Data Distribution: Distribution of objects that we want to generate. [p_{data}]

$p_{data}: \mathbb{R}^d \to \mathbb{R}_{\geq 0}$,

we don't know this prob. density.

$z \longmapsto p_{data}(z)$.

↓

sample.

Datasets:

A dataset consists finite number of samples from the data dist.

$z_1, z_2, \cdots, z_N \sim p_{data}$.

Conditional Generation.

$z \sim p_{data}(\cdot \mid y)$

↘ First this

↗ Next this

Generative Models convert samples from init dist. into samples from the data dist.

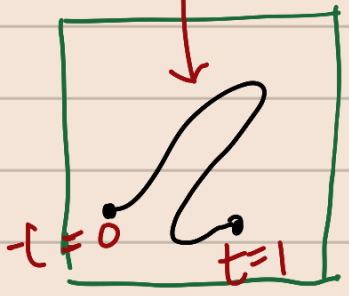
$x \sim p_{init}$ → Generative model → $z \sim p_{data}$.

SECTION-2: Flow & Diffusion Models.

2A Flow Models

Trajectory :- $X : [0, t=1] \to \mathbb{R}^d$

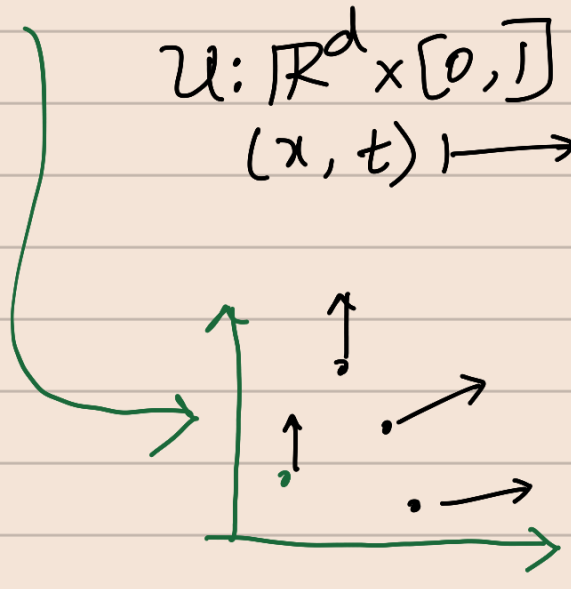
Essentially, $t \mapsto X_t$



Vector Field:

$u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$

$(x, t) \mapsto u_t(x)$



Gives a direction at every point.

Ordinary Differential Equation (ODE):

$$X_0 = x_0 \quad \text{[Initial Condition]}$$

$$\frac{d}{dt} X_t = u_t(X_t) \quad \text{[ODE]}$$ → This describes the direction of the X_t, which is

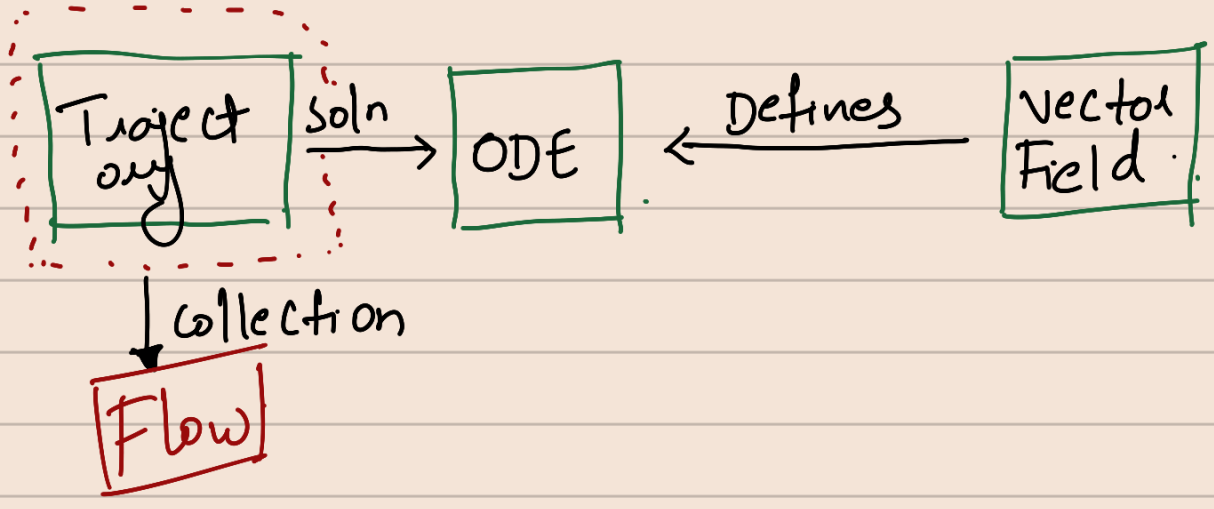
given by location of x_t, and specified by u_t

Flow: $\psi : \mathbb{R}^d \times [0,1] \longrightarrow \mathbb{R}^d$ (spatial, time)

$$(x_0, t) \mapsto \psi_t(x_0)$$

$$\psi_0(x_0) = x_0 \quad \& \quad \frac{d}{dt}\psi_t(x_0) = u_t(\psi_t(x_0))$$

flow is essentially a collection of solutions to an ODE for a lot of initial conditions



Existence & Uniqueness Theorem ODEs

Picard-Lindelöf theorem: If a vector fld $u_t(x)$ is continously differentiable with bounded derivatives, then a unique solution to ODE

$$X_0 = x_0, \quad \frac{d}{dt} X_t = u_t(X_t)$$

exists.

In other words, a flow map exists. More generally this is true if vector field is Lipschitz.

Key Takeaway: In the cases of practical interest for machine learning, unique solutions to ODE / flows exists.

Example Linear ODE

Simple Vector Field:

$$u_t(x) = -\theta x \quad (\theta > 0)$$

Claim: Flow is given by.

$$\psi_t(x_0) = \exp(-\theta t) x_0$$

Proof:

Initial Condition:

$$\psi_t(x_0) = \exp(0) x_0 = x_0$$

ODE:

$$\frac{d}{dt} \psi_t(x_0) = \frac{d}{dt} \exp(-\theta t) x_0 = -\theta \exp(-\theta t) x_0$$
$$= -\theta \psi_t(x_0)$$

$= u_t(\psi_t(x_0))$

Numerical Simulation of an ODE:

Euler Method.

For a given vector field u_t, initial condition x_0,

take a small step in the direction of vector field

and then return trajectory.

Flow Model:

$$p_{init} \xrightarrow[ODE]{} p_{data}$$

Neural Network: $u_t^\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ (Parameters)

Random init: $X_0 \sim p_{init}$

ODE: $\frac{d}{dt} X_t = u_t^\theta(x_t)$ (Neural Network Vector Field)

Goal: $X_1 \sim p_{data}$

Training = Find parameters θ such that

$X_0 \sim p_{init}$, $dX_t = u_t^\theta(x_t)\,dt$ $\xrightarrow{\text{implies}}$ $X_1 \sim p_{data}$

start with init dist. | Follow along the vector field | The dist. of final point = data dist.

Lecture 2.0

Deriving Training Target:

we need to explicitly derive it, unlike in regression & classification.

Margin Vector Field → Flow Matching

Margin score Fn → Score Matching.

Conditional Prob Path → Conditional Vector Field → Conditional score Fn (↳ Per single Data Point)

Marginal Prob path → Marginal vector field → Marginal score Fn (↳ Across dist. of data points.)

Probability Paths: The path from noise to data.

The interpolation between noise dist & data

A prob. path only specifies the marginals dist. It says nothing about the evolution of a single particle.

Dirac Dist: $z \in \mathbb{R}^d$, δ_z $\quad X \sim \delta_z \Rightarrow X = z$

Always returns the same thing

Conditional probability path: $p_t(\cdot|z)$

① $p_t(\cdot|z)$ distribution over $\mathbb{R}^d$

② $p_0(\cdot|z) = p_{init}$, $p_1(\cdot|z) = \delta_z$

Example: Gaussian probability path.

$p_t(\cdot|z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d)$ | Noise schedulers

① True

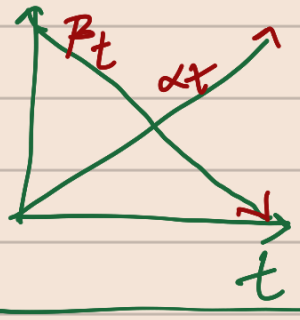
② $p_0(\cdot|z) = \mathcal{N}(0, I_d)$ ✓

$p_1(\cdot|z) = \mathcal{N}(z, 0)$ ✓

α_t, β_t s.t

$\alpha_0 = 0$ $\beta_0 = 1$

$\alpha_1 = 1$ $\beta_1 = 0$



Marginal probability path: p_t

$z \sim p_{data}$, $X \sim p_t(\cdot|z)$

$\Rightarrow$ $X \sim p_t$

forget z.

① $p_t(x) = \int p_t(x|z)\, p_{data}(z) \cdot dz$

② $p_0 = p_{init}$, $p_1 = p_{data}$

Conditional & Marginal Vector Field:

Conditional Vector Field: $u_t^{target}(x|z)$

$\left(\begin{array}{l} 0 \le t \le 1 \\ x, z \in \mathbb{R}^d \end{array}\right)$

z

$p_{init} \xrightarrow[\text{ODE}]{p_t(\cdot|z)} \delta_z$

such that

$$x_0 \sim p_{init}, \quad \frac{d}{dt} x_t = u_t^{target}(x_t|z)$$

$p_0(\cdot|z)$ — If we follow this vec. field, we should be necessarily following the Prob. Path

$$\Rightarrow x_t \sim p_t(\cdot|z) \quad (0 \le t \le 1)$$

Cond. Gaussian VF:

$$u_t^{target}(x|z) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t\right) z + \frac{\dot{\beta}_t}{\beta_t} x$$

$\dot{\alpha}_t = \frac{d\alpha_t}{dt}$, $\dot{\beta}_t$

Theorem (Marginalization Trick):

the marginal vector field by

$$u_t^{target}(x) = \int u_t^{target}(x|z) \frac{p_t(x|z)\, p_{data}(z)}{p_t(x)} dz$$

satisfies

$$X_0 \sim p_{init}, \frac{d}{dt} X_t = u_t^{target}(X_t) \Rightarrow X_t \sim p_t \quad (0 \le t \le 1).$$

Essentially, if we are following this marginal density then we are essentially following this marginal path. This means we'll end up at p_{data} in the end.

Continuity Equation:

We'll use to prove the above statement.

Let us consider a flow model with vector field u_t^{target} with $X_0 \sim p_{init}$. Then $X \sim p_t, \forall 0 \le t \le 1$ i.f.f

$$\partial_t p_t x = -\text{div}(p_t u_t^{target}) x, \forall x, 0 \le t \le 1$$

Flow Matching: u_t^θ

Goal: $u_t^\theta \approx u_t^{target}$.

Flow matching loss: $L_{fm}(\theta) = \mathbb{E}\left[\| u_t^\theta(x) - u_t^{target}(x) \|^2\right]$

$t \sim \text{unif}$ ← Uniform in $[0,1]$

$z \sim p_{data}$ ← draw data point

$x \sim p_t(\cdot|z)$ ← draw from cond. path.

Intractable

↓

Conditional to the rescue.

Conditional Flow Matching loss:

$$L_{CFM}(\theta) = \mathbb{E}\left[\| u_t^\theta(x) - u_t^{target}(x|z) \|^2\right]$$

$t \sim \text{unif}$

$z \sim p_{data}$

$x \sim p_t(\cdot|z)$

↓

Tractable ✓

Minimizer? $u_t^{target}(x|z)$

The beauty of FM will be seen when it will be shown that minimizing L_{CFM} is enough to minimize L_{FM}.

Theorem $L_{fm}(\theta) = L_{CFM}(\theta) + C$.

for $C > 0$ & independent of θ.

⇒ (1) For minimizer θ^* of L_{CFM}

$$u_t^{\theta^*} = u_t^{target}$$

(2) $\nabla_\theta L_{CFM} = \nabla_\theta L_{FM}(\theta)$

⇒ SGD the same.

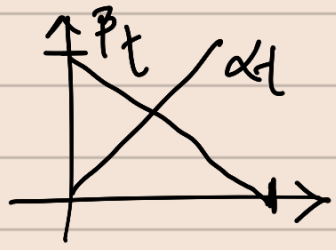
Flow Matching Training Procedure

- for each mini-batch of data do
  - sample a data example z
  - sample a random time $t \sim Unif(0,1)$
  - sample $x \sim p_t(\cdot|z)$
  - compute loss L_{CFM}.
  - update model params θ.

$\mathcal{L}_{CFM}$ for Gaussian Conditional path:

$$p_t(\cdot|z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d)$$

$$u_t^{target}(x|z) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t\right)z + \frac{\dot{\beta}_t}{\beta_t}x$$



Sampling from $p_t(\cdot|z)$ by:

$$\varepsilon \sim \mathcal{N}(0, I_d) \Rightarrow \alpha_t z + \beta_t \varepsilon \overset{\text{def}}{=} x \sim p_t(\cdot|z)$$

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{\substack{t\sim \text{unif} \\ z\sim p_{data} \\ x\sim \mathcal{N}(\alpha_t z, \beta_t^2 I_d)}}\left[\left\| u_t^\theta(x) - \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t\right)z - \frac{\dot{\beta}_t}{\beta_t}x \right\|^2\right]$$

$x = \alpha_t z + \beta_t \varepsilon$

$$= \mathbb{E}_{\substack{t\sim \text{unif} \\ z\sim p_{data} \\ \varepsilon\sim \mathcal{N}(0, I_d)}}\left[\left\| u_t^\theta(\alpha_t z + \beta_t \varepsilon) - (\dot{\alpha}_t z - \dot{\beta}_t \varepsilon) \right\|^2\right]$$

Cond OT Path: with specific α_t & β_t.

$\alpha_t = t \quad \dot{\alpha}_t = 1$

$\beta_t = 1 - t \quad \dot{\beta}_t = -1$

$$L_{CFM} \quad \mathbb{E}\left[\left\| u_t^\theta(\alpha_t z + \beta_t \varepsilon) - (z - \varepsilon)\right\|\right]^2.$$

The OT corresponds to the optimal transport path (straight line path) from noise to data.

Simple loss!

Lecture 4

A Guided CFM Objective:

Observation 1

For fixed y, we obtain the unguided problem, and may adapt an unguided objective to obtain:

$$L_{CFM}^{guided}(\theta; y) = \mathbb{E}_{\square} \left\| u_t^\theta(x|y) - u_t^{target}(x|z)\right\|^2$$

$$\square = z \sim p_{data}(z|y), t \sim Unif(0,1), x \sim p_t(x|z)$$

Observation 2

By varying y, the above yields a guided objective for general y:

$$\mathcal{L}_{CFM}^{guided}(\theta) = \square \left\| u_t^\theta(x|y) - u_t^{target}(x|z) \right\|^2$$

$$\square = (z,y) \sim p_{data}(z,y),\ t \sim Unif(0,1),\ x \sim p_t(x|z)$$

Guided Sampling Procedure:

Obtain a trained guided vector field $u_t^\theta(x|y)$

1. Select a prompt $y \in \mathcal{Y}$
2. Initialize $X_0 \sim p_{init}$
3. Simulate $dX_t = u_t^\theta(X_t|y)dt$ from $t=0$ to $t=1$.

Can we do better? At least empirically, yes.

People have realized you could trade-off

diversity for perceptual quality

CFG.

Classifier Free Guidance.

We'll build this using Gaussian Paths

Recall: A Gaussian Conditional Prob path.

$$P_t(x|z) = N(\alpha_t x, \beta_t^2 I_d)$$

where α_t, β_t are continuously diff'able, monotonic functions satisfying $\alpha_1 = \beta_0 = 1$ & $\alpha_0 = \beta_1 = 0$

For Gaussian Probability Paths it can be shown that

$$u_t^{target}(x|y) = u_t^{target}(x) + b_t \nabla \log p_t(y|x)$$

$$b_t = \frac{\dot{\alpha}_t \beta_t^2 - \dot{\beta}_t \beta_t \alpha_t}{\alpha_t}$$

For fixed w, we may define

$$\tilde{u}_t(x|y) = u_t^{target}(x) + w b_t \nabla \log p_t(y|x)$$

Rearranging yields

$$\tilde{u}_t(x|y) = (1-w)\underbrace{u_t^{\text{target}}(x)}_{\text{Unguided}} + w\underbrace{u_t^{\text{target}}(x|y)}_{\text{Guided}}.$$

CFG Training

$$u_t^{\text{target}}(x) = u_t^{\text{target}}(x|y=\emptyset)$$

we may now train a single model $u_t^\theta(x|y)$, $y \in \{y, \emptyset\}$ by re-using $L_{\text{CFM}}^{\text{guided}}(\theta)$ & occasionally setting $y = \emptyset$.

$$L_{\text{CFM}}^{\text{CFG}}(\theta) = \mathbb{E}_{\square} \| u_t^\theta(x|y) - u_t^{\text{target}}(x|z) \|^2$$

$\square = (z,y) \sim p_{\text{data}}(z,y)$ with prob η, $y \leftarrow \emptyset$,
$t \sim \text{Unif}[0,1]$, $x \sim p_t(x|z)$

CFG Sampling Procedure

Simulate $dx_t = \left[(1-w)u_t^\theta(x_t|\emptyset) + w\,u_t^\theta(x_t|y)\right]dt$

from $t=0$ to $t=1$