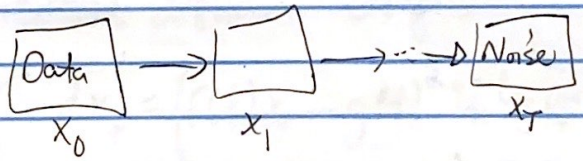
Diffusion Generative Models

Impressive framework that DALLE2, midjourney etc based on

Better image quality than GANs, but slower.

Original Idea: forward diffusion – noising
gradually add noise to input

Key insight! ⟹ reverse diffusion – denoising
can gradually remove noise from input!



$$q(x_t|x_{t-1}) = \mathcal{N}(x_t | \mu = \sqrt{1-\beta_t}\, x_{t-1}, \sigma = \sqrt{\beta_t})$$

β_t "noise schedule" $x_t = \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\, \varepsilon_{t-1}$

$\varepsilon_{t-1} \sim N(0,1)$

{ "Brownian motion"
"Wiener process"

can prove ↓ (this is why we need the $\sqrt{1-\beta}$ & $\sqrt{\beta}$ factors)

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}\, x_0, \sqrt{1-\bar{\alpha}_t})$$

$$\bar{\alpha}_t = \prod_{s=1}^{t} (1-\beta_s)$$

Same Gaussian prob path as in flow matching!

β_t s.t. $\bar{\alpha}_T \to 0$ and $N(x_T; 0, 1)$ as $T \to \infty$
chosen

Original version discrete (eg DDPM Ho et al 2020)

↓

Continuous time version (Song et al 2020, 2021)
"Score-based generative modeling"

$$dx_t = f(t)\,x_t\,dt + g(t)\,dw$$

- dw: Wiener process
- $f(t)$: "drift" coeff
- $g(t)$: "diffusion" coeff
- generates path
- solve/integrate $t=0 \to t=T$

$x_t = \gamma_t x_0 + \sigma_t \varepsilon$

$N(x_t \mid \gamma_t x_0, \sigma_t)$

$f = \frac{\dot{\gamma}_t}{\gamma_t}$

$g = \sqrt{2\sigma_t^2\left(\frac{\dot{\gamma}_t}{\gamma_t}\frac{\dot{\gamma}_t}{\gamma_t}\right)}$

reverse diffusion "Langevin" (step denoising)

$$dx_t = \left(f(t)\,x_t - g(t)^2 \nabla_x \log p_t(x)\right)dt + g(t)\,dw$$

Solve $t=T \to t=0$

$\nabla_x \log p_t(x)$: score function

$p_t(x)$ = uncond prob density of x_t

⇒ To generate, just need to learn (1) score $\nabla_x \log p_t(x)$!

→ should be much easier than learning $p_t(x)$ itself (don't need normalization)

- ~~[illegible]~~

Trick to learn $\nabla_x \log p_t(x_t)$ → conditional score matching

- Claim: NN

$$\min_\theta \mathbb{E}_{\substack{x_t \sim p_t(x_t), \\ t \sim U[0,1]}} \| s_\theta(x_t, t) - \nabla_{x_t} \log p_t(x_t) \|^2$$

$$= \min_\theta \mathbb{E}_{\substack{x_t, t, \\ x_0 \sim p(x_0) \\ \text{by data distn}}} \| s_\theta(x_t, t) - \nabla_{x_t} \log p_t(x_t | x_0) \|^2$$

$$\nabla_{x_t} \log p_t(x_t | x_0) = \frac{x_t - \alpha_t x_0}{\sigma_t^2}$$

- pf

$$\int dt \int dx_t p_t(x_t | x_0) dx_0 p(x_0) \left(s_\theta^2 - 2 s_\theta \nabla_{x_t} p_t(x_t | x_0) + \text{const} \right)$$

(int. by pts)

$$\left(\nabla_{x_t} s_\theta \right) p_t(x_t | x_0) p(x_0)$$

→ So a very simple objective for learning score!

As in FM, s_θ can be any NN, not restricted like in NF