Langevin Dynamics

Monte Carlo:	Computational algorithm with repeated random sampling for solving numerical problems.
Markov chain:	A sequential stochastic model, in which each event statistically depend only on its predecessor.
Stochastic Differential Equation (SDE):	differential equation with random entities included.

Given a well behaved PDF $P(x)$, we aim to draw iid samples from it.

The approach taken is the Langevin SDE equation:

x_t is the temporal derivative of the continuous time varying $x(t)$.

$$\boxed{x_t = \underbrace{\nabla_x \log P(x)}_{\text{score function}} + \sqrt{2}\, w_t}$$

w_t is the temporal deriv. of a brownian motion which is accumulation of canonical gaussian white noise vector.

Claim: (if initialized with $x(0) \sim N(0, I)$) the above process produces at ∞ fair samples from $P(x)$, i.e, the empirical distribution over many such runs is $P(x)$.

Some further discretization and.

Langevin dynamics

$$\text{LD}: \boxed{x_{k+1} = x_k + \tau \cdot \nabla_x \log P(x_k) + \sqrt{2\tau} \cdot z_k}$$

As we have access to the Score-Function via a denoiser approximation, we can implement the following algorithm and get samples from the target probability distribution, just as desired

Initialization: $\underline{x}_0 \sim \mathbb{N}(\underline{0}, \mathbf{I})$, $\sigma_* = 0.01$

for $k = 0:1:K-1$

$\underline{x}_{k+1} = \underline{x}_k + \tau \cdot \nabla_{\underline{x}} \log P(\underline{x}_k) + \sqrt{2\tau} \cdot \underline{z}_k$

By the way, if we apply this iterative formula without the noise perturbation, we end up converging to the

→ because after removing this we are left with score

$$= \underline{x}_k + \frac{\tau}{\sigma_*^2}\left(D(\underline{x}_k, \sigma_*) - \underline{x}_k\right) + \sqrt{2\tau} \cdot \underline{z}_k$$

end

modes (peaks) of $P(\underline{x})$, but this should not surprise us

function. It will find the shortest path to the peak.

For this algo to work,

- τ should be very small $(\to 0)$
- High quality Score
- lot of iterations $(>100{,}000)$

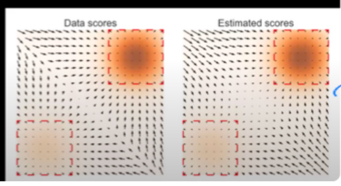
Challenges of LD algo

Problem 1:

- Recall that we are dealing with images: These are vectors in very high dimensions ($n = 1e6$), while their manifold is of low dimension
- Thus, most of the embedding space – the cube $[0,1]^n \in \mathbb{R}^n$ – is empty
- What is the Score value for a random initial point $\underline{x}_0$? The probability at this point is $P(\underline{x}_0) \approx 0$, the log becomes $-\infty$ and the gradient is either undefined or simply zero
- So, what would be the force that pulls us towards the image manifold?

Problem 2:

- Everything we do relies on the ability of our denoiser to approximate well the Score-Function
- In lower-probability regions of the manifold, we are not likely to have sufficient examples to train on, and the approximation becomes poor
- Thus, iterating in these regions is likely to behave inconsistently



→ as seen in the middle no idea of score.