Denoising Diffusion Probabilistic Models

Trevor Norton

October 26, 2023

What are diffusion models?

Generative models are unsupervised learning methods which determine the distribution $p(x)$ from which training samples are drawn.

- Examples: Autoregressive Models, Normalizing Flows, Variational Auto-Encoders, and Generative Adversarial Networks

Diffusion probabilistic models (or, more simply, diffusion models) are a form of generative models that inject noise into a distribution and then try to learn the denoising process.

- state-of-the-art performance in image generation, shape generation, and music generation

Diffusion models learn to denoise

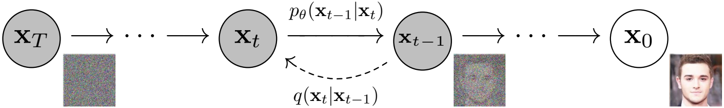

Figure: The forward process $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ continually adds noise. The backward process $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ learns to reverse the noising process [1].

To sample from the learned distribution, take noisy samples (typically from some simple tractable distribution) and reverse the noising process.

Paper 1: Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Diffusion models first proposed in [2].

- Inspired by methods in thermodynamics and statistics (in particular, Annealed Importance Sampling).
- Destroy the distribution using an iterative diffusion process and then learn to reverse the process.
- Use Markov chains with Gaussian transitions as forward and backward processes.

Forward process

Distribution to learn: $q(\mathbf{x}_0)$

Diffusion kernel: $T_\pi(\mathbf{y}|\mathbf{y}'; \beta)$

Forward process: $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = T_\pi(\mathbf{x}_t \mid \mathbf{x}_{t-1}; \beta_t)$

Stationary distribution: $\pi(\mathbf{y}) = \int T_\pi(\mathbf{y} \mid \mathbf{y}'; \beta)\pi(\mathbf{y}')\,\mathrm{d}\mathbf{y}'$

Forward process

Distribution to learn: $$q(\mathbf{x}_0)$$

Diffusion kernel: $$T_\pi(\mathbf{y}|\mathbf{y}';\beta) = \mathcal{N}(\mathbf{y}; \sqrt{1-\beta}\mathbf{y}', \beta\mathbf{I})$$

Forward process: $$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = T_\pi(\mathbf{x}_t \mid \mathbf{x}_{t-1}; \beta_t)$$

Stationary distribution: $$\pi(\mathbf{y}) = \int T_\pi(\mathbf{y} \mid \mathbf{y}'; \beta)\pi(\mathbf{y}')\,\mathrm{d}\mathbf{y}'$$

$$\implies \pi(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{I})$$

What is the forward process doing?

- If $\mathbf{x}_0 \sim \delta_0$ (i.e. $\mathbf{x}_0 = 0$ with probability 1), then $\mathbf{x}_1 \sim \mathcal{N}(0, \beta \mathbf{I})$
$\Rightarrow$ the forward process is blurring the initial data
- Repeated steps in the Markov chain leads the distribution closer to pure noise: $q(\mathbf{x}_T) \approx \pi(\mathbf{x}_T)$ for T large.
- Noise schedule β_t may be treated as fixed hyperparameters or trained as a parameters of the model. (Generally want β_t increasing.)

Reverse process

For small β_t, the reverse process has approximately normal Gaussian transitions.

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$
$$p_\theta(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

Need to choose θ that will reverse the process.

Ancestral sampling: After learning p_θ, sampling can be done by sampling from $\mathcal{N}(\mathbf{0}; \mathbf{I})$ and going through the reverse Markov chain.

Training

We want to maximize the model log-likelihood:

$$\begin{aligned}
&\int \log p_\theta(\mathbf{x}_0) q(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0 \\
&\geq \underbrace{\int \log \left[p(\mathbf{x}_T) \Pi_{t=1}^{T} \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t \mid \mathbf{x}_{t-1})} \right] q(\mathbf{x}_{0:T}) \mathrm{d}\mathbf{x}_{0:T}}_{=:-L} \\
&\geq K
\end{aligned}$$

The term K is computationally tractable, and can be used as a lower-bound for the log-likelihood. Training then chooses μ_θ and Σ_θ (and β_t) to maximize K.

Figure: Random samples for model trained on CIFAR-10 dataset [2].

Paper 2: Denoising Diffusion Probabilistic Models

- In [1], they take the diffusion model and define a new variational training condition.
- The variational loss results in better sample quality.
- The new loss also helps demonstrate connections with score-based models (as we will see more clearly next week).

Negative Log-Likelihood

$$
\begin{aligned}
L = \\
\mathbb{E}_q \Big[& \underbrace{D_{KL}(q(\mathbf{x}_T \mid \mathbf{x}_0) \mid\mid p(\mathbf{x}_T))}_{L_T} \\
& + \sum_{t>1} \underbrace{D_{KL}(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t \mathbf{x}_0) \mid\mid p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t))}_{L_{t-1}} \\
& - \underbrace{\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)}_{L_0} \Big]
\end{aligned}
$$

- L_T is constant when β_t are held fixed.
- The KL divergences in L_{t-1} can be written in closed forms since the distributions are normal.

Variational Bound

Fixing $\Sigma_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I}$ and reparameterizing

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \overline{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right)$$

where $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \Pi_{s=1}^t \alpha_s$.

Minimizing L_{t-1} is equivalent to minimizing

$$\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \overline{\alpha}_t)} \| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\overline{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \overline{\alpha}_t} \boldsymbol{\epsilon}, t) \|^2 \right]$$

- The authors note that this resembles *score matching*: trying to learn the (Stein) score $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$
- Dropping the weighted terms gives a simplified loss

$$L_{simple}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t)\right\|^2\right]$$

- Using L_{simple} improves sampling quality in image generation
    - Dropping the weighting puts for emphasis on learning to denoise at large t, which is more difficult.

Figure: Generated samples on CelebA-HQ [1].

Jonathan Ho, Ajay Jain, and Pieter Abbeel.
Denoising Diffusion Probabilistic Models.
In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli.
Deep Unsupervised Learning using Nonequilibrium Thermodynamics.
In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265. PMLR, June 2015. ISSN: 1938-7228.