Introduction to Score based Generative modelling

Gabriel V. Cardoso

Hi! Paris reading group, 27/02/2024

Overview

Background

Generative Models

- **Task**: generate new samples from a distribution of interest q_{d} defined over $\mathbb{R}^d$.
- **Context**: We rely only on a dataset $\mathcal{D}$ of i.i.d samples from q_{d}.
- **Examples**: Generative Adversarial Networks (GANs)[1], Normalizing Flows[2] and *Score-based generative models*[3].

---

[1] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2672–2680.

[2] Rezende, D., & Mohamed, S. (2015, July). Variational inference with normalizing flows. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd international conference on machine learning* (pp. 1530–1538, Vol. 37). PMLR. https://proceedings.mlr.press/v37/rezende15.html

[3] Song, Y., & Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, *32*.

Definitions

- Let $\mathcal{B}(\mathbb{R}^d)$ be the set of Borelian subsets of $\mathbb{R}^d$.

Definitions

- Let $\mathcal{B}(\mathbb{R}^d)$ be the set of Borelian subsets of $\mathbb{R}^d$.
- Let $d \in \mathbb{N}_*$ and $\mathsf{P}_0(\mathbb{R}^d)$ denote the set of (Borelian) probability measures in $\mathbb{R}^d$.

Definitions

- Let $\mathcal{B}(\mathbb{R}^d)$ be the set of Borelian subsets of $\mathbb{R}^d$.
- Let $d \in \mathbb{N}_*$ and $\mathsf{P}_0(\mathbb{R}^d)$ denote the set of (Borelian) probability measures in $\mathbb{R}^d$.
- $\mathsf{P}_2(\mathbb{R}^d) := \{p \in \mathsf{P}_0(\mathbb{R}^d) | \mathbb{E}_{X \sim p}\left[X^2\right] < \infty\}$.

Definitions

- Let $\mathcal{B}(\mathbb{R}^d)$ be the set of Borelian subsets of $\mathbb{R}^d$.
- Let $d \in \mathbb{N}_*$ and $\mathsf{P}_0(\mathbb{R}^d)$ denote the set of (Borelian) probability measures in $\mathbb{R}^d$.
- $\mathsf{P}_2(\mathbb{R}^d) := \{p \in \mathsf{P}_0(\mathbb{R}^d) | \mathbb{E}_{X \sim p}\left[X^2\right] < \infty\}$.
- For $\mathsf{q}_{\mathrm{d}} \in \mathsf{P}_0(\mathbb{R}^d)$ that admits a density w.r.t the Lebesgue measure, we define the *score* of q_{d} as

$$\mathsf{s}(x) := \nabla \log \mathsf{q}_{\mathrm{d}}(x)\,.$$

Unadjusted Langevin Algorithm (ULA)

ULA

$X_0 \sim \mu_0$, for $t \in \mathbb{N}_*$

$$X_t := X_{t-1} + \gamma \nabla \log \mathsf{q}_{\mathrm{d}}(X_{t-1}) + (2\gamma)^{1/2} \epsilon_t \,, \tag{1}$$

where $\epsilon_t \sim \mathcal{N}(0, \mathrm{I}_d)$ and $\gamma > 0$.

ULA guarantees

Wasserstein 2

For $(p_1, p_2) \in \mathsf{P}_2(\mathbb{R}^d)^{\otimes 2}$ we define

$$\mathcal{C}(p_1, p_2) := \{\pi \in \mathsf{P}_0(\mathbb{R}^{2d}) | \pi(A \times \mathbb{R}^d) = p_1(A); \pi(\mathbb{R}^d \times B) = p_2(B) \\ \text{for} \quad (A, B) \in \mathcal{B}(\mathbb{R}^d)^{\otimes 2}\} \,.$$

We define the *Wasserstein 2* distance between p_1 and p_2 as

$$W_2^2(p_1, p_2) := \min_{\pi \in \mathcal{C}(p_1, p_2)} \int \|x - y\|^2 \pi(x, y) \mathrm{d}x \mathrm{d}y \,.$$

ULA guarantees

ULA guarantees from Durmus et al., 2019, Corollary 10[4]

Assume the score is m-concave, L Lipschitz and $\epsilon > 0$. Let $\mu_t := \mathrm{Law}(X_t)$. If

- $\gamma_\epsilon < \min\left\{m\epsilon/(4Ld), L^{-1}\right\}$ and $t_\epsilon > \log(2W_2^2(\mu_0, \mathsf{q}_\mathrm{d})/\epsilon)\gamma_\epsilon^{-1}m^{-1}$

then

$$W_2^2(\mu_{t_\epsilon}, \mathsf{q}_\mathrm{d}) < \epsilon\,.$$

[4] Durmus, A., Majewski, S., & Miasojedow, B. (2019). Analysis of langevin monte carlo via convex optimization. *Journal of Machine Learning Research*, *20*(73), 1–46. http://jmlr.org/papers/v20/18-173.html Corollary 10.

Implicit Score Matching

Score Matching

Goal: Learn the score of q_{d} with a Neural Network s_θ, where $\theta \in \Theta \subset \mathbb{R}^p$.

Score Matching

Goal: Learn the score of q_{d} with a Neural Network s_θ, where $\theta \in \Theta \subset \mathbb{R}^p$.

Score Matching

$$\operatorname{argmin}_\theta \mathbb{E}_{X\sim \mathsf{q}_{\mathrm{d}}}\left[\|\mathsf{s}_\theta(X) - \mathsf{s}(X)\|^2\right] . \tag{2}$$

Learning the score from data

Hyvärinen, 2005[5] shows if $\lim_{\|x\|\to\infty} \mathsf{s}_\theta(x)\mathsf{q}_\mathrm{d}(x) = 0$, then the score matching objective (2) is equivalent to

Implicit score matching loss

$$\operatorname{argmin}_\theta \mathbb{E}_{X\sim\mathsf{q}_\mathrm{d}}\left[\nabla \cdot \mathsf{s}_\theta(X) + 1/2\|\mathsf{s}_\theta(X)\|^2\right]. \tag{3}$$

[5] Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, *6*(24), 695–709. http://jmlr.org/papers/v6/hyvarinen05a.html

Proof

$$\mathbb{E}_{X\sim \mathsf{q}_{\mathrm{d}}}\left[\|\mathsf{s}(X)-\mathsf{s}_\theta(X)\|^2\right] = \mathbb{E}_{X\sim \mathsf{q}_{\mathrm{d}}}\left[\|\mathsf{s}(X)\|^2\right] \\ - 2\mathbb{E}_{X\sim \mathsf{q}_{\mathrm{d}}}\left[\mathsf{s}(X)^T\mathsf{s}_\theta(X)\right] + \mathbb{E}_{X\sim \mathsf{q}_{\mathrm{d}}}\left[\|\mathsf{s}_\theta(X)\|^2\right] .$$

Proof

$$
\begin{aligned}
\mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\|\mathsf{s}(X)-\mathsf{s}_\theta(X)\|^2\right] &= \mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\|\mathsf{s}(X)\|^2\right] \\
&\quad - 2\mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\mathsf{s}(X)^T\mathsf{s}_\theta(X)\right] + \mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\|\mathsf{s}_\theta(X)\|^2\right] .
\end{aligned}
$$

Note that

$$
\mathsf{s}(x)^T\mathsf{s}_\theta(x) = \sum_{i=1}^d \mathsf{s}_{\theta,i}(x)\partial_{x_i}\log \mathsf{q}_\mathrm{d}(x) .
$$

Proof

$$\mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\|\mathsf{s}(X)-\mathsf{s}_\theta(X)\|^2\right] = \mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\|\mathsf{s}(X)\|^2\right] \\ - 2\mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\mathsf{s}(X)^T\mathsf{s}_\theta(X)\right] + \mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\|\mathsf{s}_\theta(X)\|^2\right] .$$

Note that

$$\mathsf{s}(x)^T\mathsf{s}_\theta(x) = \sum_{i=1}^{d} \mathsf{s}_{\theta,i}(x)\partial_{x_i}\log \mathsf{q}_\mathrm{d}(x) .$$

For $i \in [\![1, d]\!]$,

$$\mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\mathsf{s}_{\theta,i}(X)\partial_{x_i}\log \mathsf{q}_\mathrm{d}(X)\right] = \int \mathsf{s}_{\theta,i}(x)\partial_{x_i}\log \mathsf{q}_\mathrm{d}(x)\mathsf{q}_\mathrm{d}(x)\mathrm{d}x .$$

Proof

$$\mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\|\mathsf{s}(X)-\mathsf{s}_\theta(X)\|^2\right] = \mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\|\mathsf{s}(X)\|^2\right] - 2\mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\mathsf{s}(X)^T\mathsf{s}_\theta(X)\right] + \mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\|\mathsf{s}_\theta(X)\|^2\right] .$$

Note that

$$\mathsf{s}(x)^T\mathsf{s}_\theta(x) = \sum_{i=1}^d \mathsf{s}_{\theta,i}(x)\partial_{x_i}\log \mathsf{q}_\mathrm{d}(x) .$$

For $i \in [\![1,d]\!]$,

$$\mathbb{E}_{X\sim \mathsf{q}_\mathrm{d}}\left[\mathsf{s}_{\theta,i}(X)\partial_{x_i}\log \mathsf{q}_\mathrm{d}(X)\right] = \int \mathsf{s}_{\theta,i}(x)\partial_{x_i}\log \mathsf{q}_\mathrm{d}(x)\mathsf{q}_\mathrm{d}(x)\mathrm{d}x .$$

Since $\partial_{x_i}\log \mathsf{q}_\mathrm{d}(x)\mathsf{q}_\mathrm{d}(x) = \partial_{x_i}\mathsf{q}_\mathrm{d}(x)$, we have

$$\int \mathsf{s}_{\theta,i}(x)\partial_{x_i}\log \mathsf{q}_\mathrm{d}(x)\mathsf{q}_\mathrm{d}(x)\mathrm{d}x = -\int \partial_{x_i}\mathsf{s}_{\theta,i}(x)\mathsf{q}_\mathrm{d}(x)\mathrm{d}x .$$

- $\nabla \cdot s_\theta$ costly in high dimensions.
- Score estimate inaccurate in low data regions, furthermore ULA moves can be stuck in each mode.

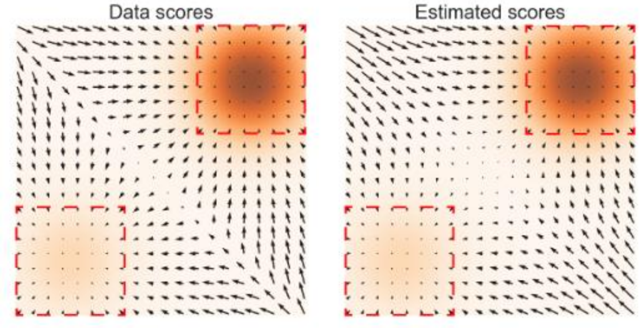


Figure: Illustration of score on low data regions, from Y. Song and Ermon, 2019[6].

---

[6]Song, Y., & Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems, 32*.

Code Break!

https://github.com/gabrielvc/tutorial_ddim

Denoising score matching

Denoising score matching

Vincent, 2011[7] introduces the idea of learning the score of $\mathsf{q}_\sigma(\mathrm{d}x_\sigma) = \int q_\sigma(x_\sigma|x)\mathsf{q}_\mathrm{d}(\mathrm{d}x)$ where $q_\sigma(\cdot|x) = \mathcal{N}(x, \sigma^2\,\mathrm{I})$.

[7] Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, *23*(7), 1661–1674. https://doi.org/10.1162/NECO_a_00142

Denoising score matching

Vincent, 2011[7] introduces the idea of learning the score of $\mathsf{q}_\sigma(\mathrm{d}x_\sigma) = \int q_\sigma(x_\sigma|x)\mathsf{q}_\mathrm{d}(\mathrm{d}x)$ where $q_\sigma(\cdot|x) = \mathcal{N}(x, \sigma^2\,\mathrm{I})$.

Denoising score matching loss

$$\mathrm{argmin}_\theta\, \mathbb{E}_{X_\sigma \sim q_\sigma(\cdot|X), X\sim \mathsf{q}_\mathrm{d}} \left[\|\mathsf{s}_{\theta,\sigma}(X_\sigma) - \nabla \log q_\sigma(X_\sigma|X)\|^2 \right] .$$

[7] Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, *23*(7), 1661–1674. https://doi.org/10.1162/NECO_a_00142

Proof

Let $q_{0|\sigma}(\mathrm{d}x_0|x_\sigma) := q_{\sigma|0}(x_\sigma|x_0)\mathsf{q}_{\mathrm{d}}(\mathrm{d}x_0)/\mathsf{q}_\sigma(x_\sigma)$.

Proof

Let $q_{0|\sigma}(\mathrm{d}x_0|x_\sigma) := q_{\sigma|0}(x_\sigma|x_0)\mathsf{q}_{\mathrm{d}}(\mathrm{d}x_0)/\mathsf{q}_\sigma(x_\sigma)$. By Fisher's identity

$$
\begin{aligned}
\nabla \log \mathsf{q}_\sigma(x_\sigma) &= \frac{\nabla \mathsf{q}_\sigma(x_\sigma)}{\mathsf{q}_\sigma(x_\sigma)} = \mathbb{E}_{X_0 \sim \mathsf{q}_{\mathrm{d}}}\left[\frac{\nabla q_{\sigma|0}(x_\sigma|X_0)}{\mathsf{q}_\sigma(x_\sigma)}\right] \\
&= \mathbb{E}_{X_0 \sim \mathsf{q}_{\mathrm{d}}}\left[\nabla \log q_{\sigma|0}(x_\sigma|X_0)\frac{q_{\sigma|0}(x_\sigma|X_0)}{\mathsf{q}_\sigma(x_\sigma)}\right] \\
&= \mathbb{E}_{X_0 \sim q_{0|\sigma}(\cdot|x_\sigma)}\left[\nabla \log q_{\sigma|0}(x_\sigma|X_0)\right] .
\end{aligned}
$$

Proof

Let $q_{0|\sigma}(\mathrm{d}x_0|x_\sigma) := q_{\sigma|0}(x_\sigma|x_0)\mathsf{q}_\mathrm{d}(\mathrm{d}x_0)/\mathsf{q}_\sigma(x_\sigma)$. By Fisher's identity

$$\begin{aligned}
\nabla \log \mathsf{q}_\sigma(x_\sigma) = \frac{\nabla \mathsf{q}_\sigma(x_\sigma)}{\mathsf{q}_\sigma(x_\sigma)} &= \mathbb{E}_{X_0 \sim \mathsf{q}_\mathrm{d}}\left[\frac{\nabla q_{\sigma|0}(x_\sigma|X_0)}{\mathsf{q}_\sigma(x_\sigma)}\right] \\
&= \mathbb{E}_{X_0 \sim \mathsf{q}_\mathrm{d}}\left[\nabla \log q_{\sigma|0}(x_\sigma|X_0)\frac{q_{\sigma|0}(x_\sigma|X_0)}{\mathsf{q}_\sigma(x_\sigma)}\right] \\
&= \mathbb{E}_{X_0 \sim q_{0|\sigma}(\cdot|x_\sigma)}\left[\nabla \log q_{\sigma|0}(x_\sigma|X_0)\right] .
\end{aligned}$$

Thus

$$\begin{aligned}
&\mathbb{E}_{X_\sigma \sim \mathsf{q}_\sigma}\left[\|\mathsf{s}_\theta(X_\sigma) - \nabla \log \mathsf{q}_\sigma(X_\sigma)\|^2\right] \\
&\quad = \mathbb{E}_{X_\sigma \sim \mathsf{q}_\sigma}\left[\left\|\mathsf{s}_\theta(X_\sigma) - \mathbb{E}_{X_0 \sim q_{0|\sigma}(\cdot|X_\sigma)}\left[\nabla \log q_{\sigma|0}(X_\sigma|X_0)\right]\right\|^2\right] .
\end{aligned}$$

By defining
$\mathsf{q}_{\sigma,0}(\mathrm{d}x_\sigma, \mathrm{d}x_0) = q_{\sigma|0}(x_0|\mathrm{d}x_\sigma)\mathsf{q}_{\mathrm{d}}(\mathrm{d}x_0) = q_{0|\sigma}(\mathrm{d}x_0|x_\sigma)\mathsf{q}_\sigma(\mathrm{d}x_\sigma)$, we have

$$
\begin{aligned}
&\mathbb{E}_{X_\sigma\sim\mathsf{q}_\sigma}\left[\|\mathsf{s}_{\theta,\sigma}(X_\sigma) - \nabla\log\mathsf{q}_\sigma(X_\sigma)\|^2\right] \\
&= \mathbb{E}_{X_\sigma\sim\mathsf{q}_\sigma}\left[\|\mathsf{s}_\theta(X_\sigma)\|^2 - 2\mathsf{s}_\theta(X_\sigma)^T\mathbb{E}_{X_0\sim q_{0|\sigma}(\cdot|X_\sigma)}\left[\nabla\log q_{\sigma|0}(X_\sigma|X_0)\right]\right] + \\
&= \mathbb{E}_{X_\sigma\sim\mathsf{q}_\sigma}\left[\mathbb{E}_{X_0\sim q_{0|\sigma}(\cdot|X_\sigma)}\left[\|\mathsf{s}_\theta(X_\sigma) - \nabla\log q_{\sigma|0}(X_\sigma|X_0)\|^2\right]\right] + \tilde{C} \\
&= \mathbb{E}_{(X_\sigma,X_0)\sim\mathsf{q}_{\sigma,0}}\left[\|\mathsf{s}_\theta(X_\sigma) - \nabla\log q_{\sigma|0}(X_\sigma|X_0)\|^2\right] + \tilde{C}\,,
\end{aligned}
$$

where C and $\tilde{C}$ are constants that do not depend on θ.

Denoising score matching

- No need of calculating derivatives of the score network.
- Mixes better and exploit regions of low data density.
- Approaches $\nabla \log \mathsf{q}_{\mathrm{d}}$ only in the limit $\sigma \to 0$.

Code Break!

https://github.com/gabrielvc/tutorial_ddim

Noise Conditional Score Networks

Noise Conditional Score Networks

Y. Song and Ermon, 2019[8] introduces several noised versions of q_{d}.

Diffused marginals

For $t \in [\![1, T]\!]$ and $v_t > 0$, define $q_{t|0}(x_t|x_0) = \mathcal{N}(x_t; x_0, v_t^2\,\mathrm{I})$ and

$$\mathsf{q}_t(\mathrm{d}x_t) := \int q_{t|0}(\mathrm{d}x_t|x_0)\mathsf{q}_{\mathrm{d}}(\mathrm{d}x_0)\,.$$

[8] Song, Y., & Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems, 32*.

Noise Conditional Score Networks

Train a neural network s_θ to jointly learn the score of $\{\mathsf{q}_t\}_{t=1}^T$:

Diffusion score matching

$$\sum_{t=1}^{T} \gamma_t^2 \mathbb{E}_{X_t \sim q_{t|0}(\cdot|X_0), X_0 \sim \mathsf{q}_\mathrm{d}} \left[\| \mathsf{s}_\theta(X_t, \upsilon_t) - \nabla \log q_{t|0}(X_t|X_0) \|^2 \right] .$$

where $\{\upsilon_t\}_{t=1}^T$ is an increasing sequence of positive values.

Noise Conditional Score Networks

Generate samples by sequential ULA on $\{\mathsf{q}_t\}_{t=1}^{T}$:

Data : X_T^0, k, r, θ

Result : X_0^0

for $t \leftarrow T$ **to** 1 **do**

  for $\ell \leftarrow 1$ **to** k **do**

    set $\gamma = r\upsilon_t^2/\upsilon_T^2$

    draw $\epsilon_{t,\ell} \sim \mathcal{N}(0, \mathrm{I}_d)$

    set $X_t^\ell = X_t^{\ell-1} + (\gamma/2)\mathsf{s}_\theta(X_t^{\ell-1}, \upsilon_t) + \gamma^{1/2}\epsilon_{t,\ell}$

  set $X_{t-1}^0 = X_t^\ell$.

Code Break!

https://github.com/gabrielvc/tutorial_ddim

Denoising diffusion implicit models (DDIM)

Score based generative models

Other then sequential ULA, several samplers are available to sample backwards from the sequence of distributions $\{\mathsf{q}_t\}_{t=1}^{T}$, based on

- Stochastic differential equations[9],
- Ordinary differential equations[10],
- Markov chains[11].

---

[9] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*. https://openreview.net/forum?id=PxTIG12RRHS

[10] Karras, T., Aittala, M., Aila, T., & Laine, S. (2022). Elucidating the design space of diffusion-based generative models. *Proc. NeurIPS*.

[11] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840–6851; Song, J., Meng, C., & Ermon, S. (2021). Denoising diffusion implicit models. *International Conference on Learning Representations*. https://openreview.net/forum?id=St1giarCHLP

DDPM[12]

Define $X_0 \sim \mathsf{q}_{\mathrm{d}}$, $X_t = X_{t-1} + (\upsilon_t^2 - \upsilon_{t-1}^2)^{1/2} \varepsilon_t$ for $t \in [\![1, T]\!]$ with $\varepsilon_t \sim \mathcal{N}(0, \mathrm{I})$.

[12] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840–6851.

DDPM[12]

Define $X_0 \sim \mathsf{q}_{\mathrm{d}}$, $X_t = X_{t-1} + (\upsilon_t^2 - \upsilon_{t-1}^2)^{1/2}\varepsilon_t$ for $t \in [\![1, T]\!]$ with $\varepsilon_t \sim \mathcal{N}(0, \mathrm{I})$.

Then, $\mathrm{Law}(X_t) = \mathsf{q}_t$ and $\mathrm{Law}(X_t | X_0 = x_0) = q_{t|0}(\cdot | x_0)$.

[12] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840–6851.

DDPM[12]

Define $X_0 \sim \mathsf{q}_\mathrm{d}$, $X_t = X_{t-1} + (\upsilon_t^2 - \upsilon_{t-1}^2)^{1/2} \varepsilon_t$ for $t \in [\![1, T]\!]$ with $\varepsilon_t \sim \mathcal{N}(0, \mathrm{I})$.

Then, $\mathrm{Law}(X_t) = \mathsf{q}_t$ and $\mathrm{Law}(X_t | X_0 = x_0) = q_{t|0}(\cdot | x_0)$.

Furthermore, we can write the law of X_{t-1} conditionally on X_t, X_0,

[12] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840–6851.

DDPM[12]

Define $X_0 \sim \mathsf{q}_{\mathrm{d}}$, $X_t = X_{t-1} + (\upsilon_t^2 - \upsilon_{t-1}^2)^{1/2}\varepsilon_t$ for $t \in [\![1, T]\!]$ with $\varepsilon_t \sim \mathcal{N}(0, \mathrm{I})$.

Then, $\mathrm{Law}(X_t) = \mathsf{q}_t$ and $\mathrm{Law}(X_t | X_0 = x_0) = q_{t|0}(\cdot|x_0)$.

Furthermore, we can write the law of X_{t-1} conditionally on X_t, X_0,

$$q_{t-1|t,0}(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_0 + \frac{\upsilon_{t-1}^2}{\upsilon_t^2}(x_t - x_0), (\upsilon_t^2 - \upsilon_{t-1}^2)\frac{\upsilon_{t-1}^2}{\upsilon_t^2}\right) .$$

[12] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840–6851.

DDIM[13]

Inference distribution

For $t \in [\![2, T]\!]$ and $\eta \in (0, \upsilon_{t-1})$, set

$$q^{\eta}_{t-1|t,0}(x_{t-1}|x_t, x_0) := \mathcal{N}\left(x_{t-1}; \mu_{t-1}(x_0, x_t), \eta^2 \, \mathrm{I}_d\right)$$
$$\mu_{t-1}(x_0, x_t) := x_0 + (\upsilon^2_{t-1}/\upsilon^2_t - \eta^2/\upsilon^2_t)^{1/2}(x_t - x_0)\,.$$

The mean μ_{t-1} is chosen to satisfy:

$$q_{t-1|0}(\mathrm{d}x_{t-1}|x_0) = \int q^{\eta}_{t-1|t,0}(\mathrm{d}x_{t-1}|x_t, x_0) q_{t|0}(\mathrm{d}x_t|x_0)\,.$$

[13] Song, J., Meng, C., & Ermon, S. (2021). Denoising diffusion implicit models. *International Conference on Learning Representations.* https://openreview.net/forum?id=St1giarCHLP

DDIM

Full inference process

For $\eta = \{\eta_t \in (0, \upsilon_t)\}_{t=1}^T$, define

$$q^{\eta}_{1:T|0}(\mathrm{d}x_{1:T}|x_0) = q_{T|0}(\mathrm{d}x_T|x_0) \prod_{t=2}^{T} q^{\eta_{t-1}}_{t-1|t,0}(\mathrm{d}x_{t-1}|x_t, x_0)\,,$$

and $q^{\eta}_{0:T}(x_{0:T}) = q^{\eta}_{1:T|0}(\mathrm{d}x_{1:T}|x_0)\mathsf{q}_{\mathrm{d}}(\mathrm{d}x_0)$

DDIM

Full inference process

For $\eta = \{\eta_t \in (0, \upsilon_t)\}_{t=1}^T$, define

$$q^{\eta}_{1:T|0}(\mathrm{d}x_{1:T}|x_0) = q_{T|0}(\mathrm{d}x_T|x_0) \prod_{t=2}^{T} q^{\eta_{t-1}}_{t-1|t,0}(\mathrm{d}x_{t-1}|x_t, x_0)\,,$$

and $q^{\eta}_{0:T}(x_{0:T}) = q^{\eta}_{1:T|0}(\mathrm{d}x_{1:T}|x_0)\mathsf{q}_{\mathrm{d}}(\mathrm{d}x_0)$

The inference process admits the ”right” marginals:

$$\mathsf{q}_t(\mathrm{d}x_t) = \int q^{\eta}_{0:T}(\mathrm{d}x_{0:T})\,. \tag{4}$$

DDIM backward chain

DDIM Recursion

$$\mathsf{q}_{t-1}(\mathrm{d}x_{t-1}) = \int q^{\eta}_{t-1|t,0}(\mathrm{d}x_{t-1}|x_t, x_0)\mathsf{q}_t(\mathrm{d}x_t)q_{0|t}(\mathrm{d}x_0|x_t)\,.$$

where $q_{0|t}(\mathrm{d}x_0|x_t) = q_{t|0}(x_t|x_0)\mathsf{q}_\mathrm{d}(\mathrm{d}x_0)/\mathsf{q}_t(x_t)$.

DDIM backward chain

DDIM Recursion

$$\mathsf{q}_{t-1}(\mathrm{d}x_{t-1}) = \int q^{\eta}_{t-1|t,0}(\mathrm{d}x_{t-1}|x_t, x_0)\mathsf{q}_t(\mathrm{d}x_t)q_{0|t}(\mathrm{d}x_0|x_t)\,.$$

where $q_{0|t}(\mathrm{d}x_0|x_t) = q_{t|0}(x_t|x_0)\mathsf{q}_{\mathrm{d}}(\mathrm{d}x_0)/\mathsf{q}_t(x_t)$.

DDIM Approximation

$$\widehat{\mathsf{q}}_{t-1}(\mathrm{d}x_{t-1}) = \int q^{\eta}_{t-1|t,0}(\mathrm{d}x_{t-1}|x_t, \mu_t(x_t))\mathsf{q}_t(\mathrm{d}x_t)\,,$$

where $\mu_t(x_t) := \mathbb{E}_{X_0 \sim q_{0|t}(\cdot|x_t)}\left[X_0\right]\,.$

DDIM Mean approximation

Note that

$$\begin{aligned} \upsilon_t^2 \mathsf{s}_\theta(x_t, \upsilon_t) &\approx \upsilon_t^2 \mathbb{E}_{x_0 \sim q_{0|t}(\cdot|x_t)} \left[\nabla \log q_{t|0}(x_t|X_0) \right] \\ &= \mathbb{E}_{X_0 \sim q_{0|t}(\cdot|x_t)} \left[X_0 - x_t \right] = \mu_t(x_t) - x_t \,. \end{aligned}$$

DDIM Mean approximation

Note that

$$
\begin{aligned}
\upsilon_t^2 \mathsf{s}_\theta(x_t, \upsilon_t) &\approx \upsilon_t^2 \mathbb{E}_{x_0 \sim q_{0|t}(\cdot|x_t)} \left[\nabla \log q_{t|0}(x_t|X_0) \right] \\
&= \mathbb{E}_{X_0 \sim q_{0|t}(\cdot|x_t)} \left[X_0 - x_t \right] = \mu_t(x_t) - x_t \,.
\end{aligned}
$$

DDIM Backward Markov chain

Let $\mu_{t,\theta}(x_t) = x_t + \upsilon_t^2 \mathsf{s}_\theta(x_t, \upsilon_t)$, $\lambda_T = \mathcal{N}(0, \upsilon_T^2\, \mathrm{I})$ and $\eta = \{\eta_t \in (0, \upsilon_t)\}_{t=0}^T$. Define

$$
\mathsf{p}_{1:T}^\theta(\mathrm{d}x_{1:T}) := \lambda_T(\mathrm{d}x_T) \prod_{t=1}^T p_{t-1|t}^\theta(\mathrm{d}x_{t-1}|x_t) \,.
$$

where $p_{t-1|t}^\theta(\mathrm{d}x_{t-1}|x_t) = q_{t-1|t,0}^{\eta_{t-1}}(\mathrm{d}x_{t-1}|x_t, \mu_{\theta,t}(x_t))$ for $t > 1$ and $p_{0|1}^\theta(x_0|x_1) = \mathcal{N}(x_0; \mu_{1,\theta}(x_1), \eta_0^2\, \mathrm{I})$.

DDIM as variational inference

Kullback-Leibner

$$
\begin{aligned}
&\mathsf{KL}(q^{\eta}_{0:T} \parallel \mathsf{p}_{0:T}) \\
&= \frac{1}{2}\sum_{t=0}^{T-1} \gamma_t^2 \mathbb{E}_{X_t \sim q_{t|0}(\cdot|X_0), X_0 \sim \mathsf{q}_\mathrm{d}} \left[\|\mu_{\theta,t}(X_t) - X_0\|^2 \right] \\
&+ \frac{1}{2}\upsilon_T^{-2} \mathbb{E}_{\mathsf{q}_\mathrm{d}} \left[\|X_0\|^2 \right] + C \,,
\end{aligned}
$$

where $\gamma_t := \left[\upsilon_t - (\upsilon_{t-1}^2 - \eta_{t-1}^2)^{1/2}\right] (\eta_{t-1}\upsilon_t)^{-1}$ for $t > 0$, $\gamma_0 = \eta_0$ and C is a constant that does not depend on θ.

KL calculation

$$\mathsf{KL}(q^\eta_{0:T} \parallel \mathsf{p}^\theta_{0:T})$$

$$= \int \log\left(\frac{\mathsf{q}_\mathrm{d}(x_0) q_{T|0}(x_T|x_0) \prod_{t=2}^T q^\eta_{t-1|t,0}(x_{t-1}|x_t, x_0)}{\lambda_T(x_T) \prod_{t=1}^T p^\theta_{t-1|t}(x_{t-1}|x_t)}\right) q^\eta_{0:T}(\mathrm{d}x_{0:T})$$

$$= \sum_{t=2}^T \int \mathsf{KL}(q^\eta_{t-1|t,0}(\cdot|x_t, x_0) \parallel p^\theta_{t-1|t}(\cdot|x_t))\mathsf{q}_{t,0}(\mathrm{d}x_t, \mathrm{d}x_0)$$

$$+ \int \mathsf{KL}(\mathsf{q}_\mathrm{d} \parallel p^\theta_{0|1}(\cdot|x_1))\mathsf{q}_1(\mathrm{d}x_1) + \int \mathsf{KL}(q_{T|0}(\cdot|x_0) \parallel \lambda_T)\mathsf{q}_\mathrm{d}(\mathrm{d}x_0)\,.$$

Intermediate KL

$$
\begin{aligned}
&\mathsf{KL}(q^{\eta}_{t-1|t,0}(\cdot|x_t, x_0) \parallel p^{\theta}_{t-1|t}(\cdot|x_t)) \\
&\quad = (2\eta^2_{t-1})^{-1} \|\mu_{t-1}(x_0, x_t) - \mu_{t-1}(\mu_{t,\theta}(x_t), x_t)\|^2 \\
&\quad = (2\eta^2_{t-1})^{-1} \left[1 + (\upsilon^2_{t-1}/\upsilon^2_t - \eta^2_{t-1}/\upsilon^2_t)^{1/2}\right]^2 \|x_0 - \mu_{t,\theta}(x_t)\|^2 \\
&\quad = (2\eta^2_{t-1}\upsilon^2_t)^{-1} \left[\upsilon_t + (\upsilon^2_{t-1} - \eta^2_{t-1})^{1/2}\right]^2 \|x_0 - \mu_{t,\theta}(x_t)\|^2 \\
&\quad = (1/2)\gamma^2_t \|x_0 - \mu_{t,\theta}(x_t)\|^2 \,.
\end{aligned}
$$

Other terms

$$
\begin{aligned}
\mathsf{KL}(q_{T|0}(\cdot|x_0) \parallel \lambda_T) &= (2\upsilon_T^2)^{-1}\|x_0\|^2 \\
\mathsf{KL}(\mathsf{q}_{\mathrm{d}} \parallel p^{\theta}_{0|1}(\cdot|x_1)) &= -\int \log p^{\theta}_{0|1}(x_0|x_1)\mathsf{q}_{\mathrm{d}}(\mathrm{d}x_0) - \mathcal{H}(\mathsf{q}_{\mathrm{d}}) \\
&= (2\eta_0^2)^{-1}\|x_0 - \mu_{1,\theta}(x_1)\|^2 + (d/2)\log(2\pi\eta_0) \\
&- \mathcal{H}(\mathsf{q}_{\mathrm{d}})\,.
\end{aligned}
$$

Code Break! https://github.com/gabrielvc/tutorial_ddim

Deep image prior

Deep image prior

Let $\tilde{x}$ be a corrupted version of an image (inpaiting, denoising) and $\mathsf{m} \in \{0, 1\}^d$ the associated mask. Ulyanov et al., 2018[14] proposes solving the reconstruction task by

$$\mathrm{argmin}_\theta \left\| \mu_\theta(z) \odot \mathsf{m} - \tilde{x} \odot \mathsf{m} \right\|^2 ,$$

where z is a fixed seed.

---

[14] Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2018). Deep image prior. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deep image prior

Figure: Inpainting example from Ulyanov et al., 2018[15]

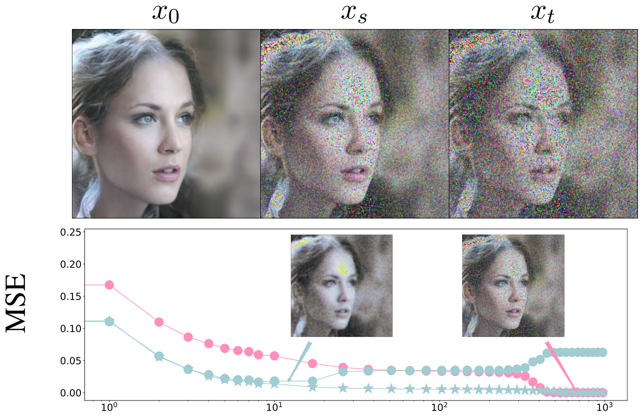
[15] Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2018). Deep image prior. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Denoising tasks

Suppose $\mu_{t,\theta}(x_t)$ is a UNet. Consider the following losses:

$$L_{t|s}(\theta) := \|\mu_{t,\theta}(x_t) - x_s\|^2 \quad \text{and} \quad L_{t|0}(\theta) := \|\mu_{t,\theta}(x_t) - x_0\|^2 .$$

We train $\mu_{t,\theta}(x_t)$ to minimize $L_{t|s}(\theta)$.



Conclusion

Interesting papers

- Going further on Diffusion models: Karras et al., 2022[16], Y. Song, Sohl-Dickstein, et al., 2021[17], Y. Song, Durkan, et al., 2021.[18].

[16] Karras, T., Aittala, M., Aila, T., & Laine, S. (2022). Elucidating the design space of diffusion-based generative models. *Proc. NeurIPS*.

[17] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*. https://openreview.net/forum?id=PxTIG12RRHS

[18] Song, Y., Durkan, C., Murray, I., & Ermon, S. (2021). Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, *34*, 1415–1428.

Interesting papers

- Diffusion models as priors for inverse problems: Chung et al., 2023[19], Cardoso et al., 2023[20], Wu et al., 2023[21].

---

[19] Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., & Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=OnD9zGAGT0k

[20] Cardoso, G., Idrissi, Y. J. E., Corff, S. L., & Moulines, E. (2023). Monte carlo guided diffusion for bayesian linear inverse problems.

[21] Wu, L., Trippe, B. L., Naesseth, C. A., Blei, D. M., & Cunningham, J. P. (2023). Practical and asymptotically exact conditional sampling in diffusion models.

Interesting papers

- Developpements on diffusion models: Rombach et al., 2022[22], Y. Song et al., 2023[23].

[22] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

[23] Song, Y., Dhariwal, P., Chen, M., & Sutskever, I. (2023). Consistency models.

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As
there was some unprocessed data that should have been added to th
final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page wi
go away, because LaTeX now knows how many pages to expect for t
document.