CS F425: Deep Learning

36 Diffusion Models

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

May 02, 2023 ON-CAMPUS Campus @ BITS-Pilani [Jan-May 2023]

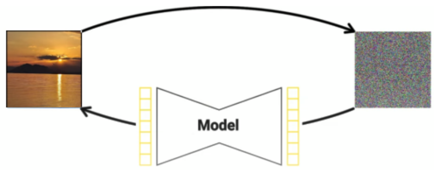
http://ktiwari.in/dl

Diffusion Models [1]

Probability distributions of data-sets is complex

Makes learning, sampling, inference, and evaluation to be analytically/computationally non tractable (in general)

1. Systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process.
2. Then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model.



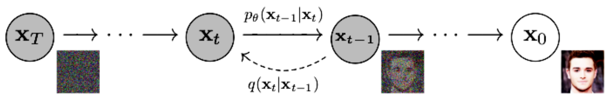
[1] **Cite: 1135** Sohl-Dickstein, Jascha, et al. **Deep unsupervised learning using nonequilibrium thermodynamics.** International Conference on Machine Learning. PMLR, 2015.

Diffusion Models [2]

High quality image synthesis

By training a weighted variational bound designed according to connection between diffusion probabilistic models and denoising score matching with Langevin dynamics

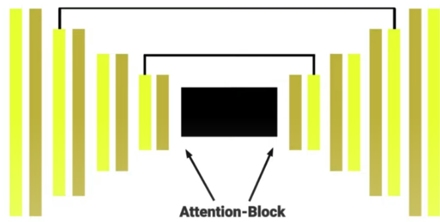
1. Predict image noise, to subtract for getting original one.
2. Noise injection is regulated by a schedule: linear/cosine



[2] **Cite: 1842** Ho, Jonathan, Ajay Jain, and Pieter Abbeel. **Denoising diffusion probabilistic models.** Advances in Neural Information Processing Systems 33 (2020): 6840-6851.

Architecture

U-Net like architecture



1. Down/up sample with resNet block.
2. Skip connection with attention block at certain resolution.

Diffusion Models

Algorithm 3 Sending $\mathbf{x}_0$

1: Send $\mathbf{x}_T \sim q(\mathbf{x}_T|\mathbf{x}_0)$ using $p(\mathbf{x}_T)$
2: **for** $t = T-1, \ldots, 2, 1$ **do**
3: Send $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0)$ using $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$
4: **end for**
5: Send $\mathbf{x}_0$ using $p_\theta(\mathbf{x}_0|\mathbf{x}_1)$

Algorithm 4 Receiving

1: Receive $\mathbf{x}_T$ using $p(\mathbf{x}_T)$
2: **for** $t = T-1, \ldots, 1, 0$ **do**
3: Receive $\mathbf{x}_t$ using $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$
4: **end for**
5: **return** $\mathbf{x}_0$

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Improved Diffusion Models [3]

Variance should also be learned → it improves log likelyhood.

1. Predict image noise, to subtract for getting original one.
2. Cosine noise injection schedule

[3] **Cite: 586** Nichol, Alexander Quinn, and Prafulla Dhariwal. **Improved denoising diffusion probabilistic models.** International Conference on Machine Learning. PMLR, 2021.

Thank You!

Thank you very much for your attention!