Denoising Diffusion models

Sébastien Pierre, Oscar Roux, Youssoufa Talba

7 Mars 2023

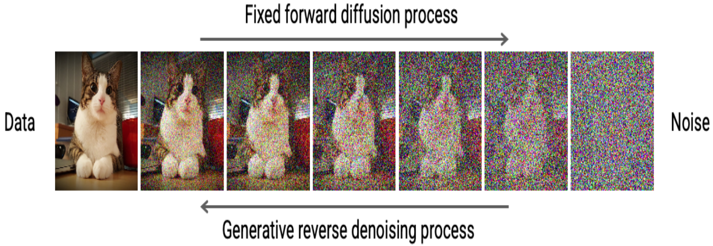
Introduction

[1] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. arXiv preprint arXiv:1503.03585.
[2] Ho, J., Jain, A., Abbeel, P. (2020). Denoising diffusion probabilistic models. arXiv preprint arXiv:2006.11239.

Sommaire

Principle of denoising diffusion models

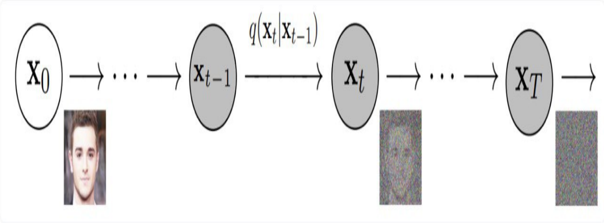
Generative models : generate a new datapoint based on a dataset



Goal : Learn the reverse diffusion process in order to sample new datapoints with random gaussian noise

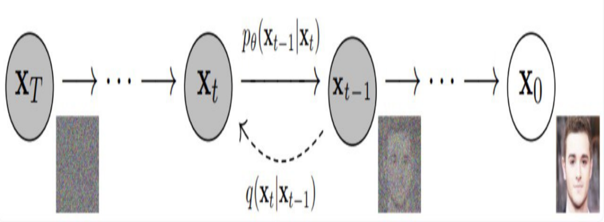
Forward diffusion process

Forward process: Adding at each step a gaussian noise to the data



$q(x_t|x_{t-1}) \sim \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t I)$ with the process β_t being learned or set in advance.

Backward process: From pure noise to a sample of the target distribution



$p_\theta(x_{t-1}|x_t) \sim \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$ where a neural network learns the parameters $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t))$

The β_ts are chosen in order to ensure that the reverse process is also gaussian (For our purpose, we only need to take small β_ts)

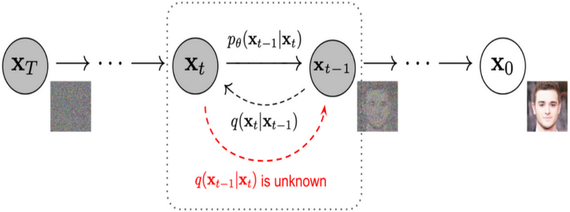
Fasten the computations

$$
\begin{aligned}
x_t &= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1} \\
&= \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \text{ with } \alpha_t = 1-\beta_t
\end{aligned}
$$

Recursively (since we are summing independant gaussian rv), we obtain $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_0$
Where we denote $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$

Work with a tractable loss

training and loss simplification



$$\mathbb{E}[-log p_\theta(x_0)] \leq \mathbb{E}_q[-log \frac{p_\theta(x_0:T)}{q(x_1:T|x_0)}] = L_{vub}$$

training and loss simplification

Let's rewrite this loss with Kullback-Leibler divergence

$$L_{vlb} = L_0 + L_1 + ... + L_{T-1} + L_T$$

$$L_0 = -log p_\theta(x_0|x_1)$$

$$L_{t-1} = D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))$$

$$L_T = D_{KL}(q(x_T|x_0)||p(x_T))$$

training and loss simplification

$$
\begin{aligned}
L_{\mathrm{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\Big] \\
&= \mathbb{E}_q\Big[\log \frac{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T)\prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=1}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \Big(\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\cdot\frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}\Big) + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\Big] \\
&= \mathbb{E}_q[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]
\end{aligned}
$$

training and loss simplification

Choices made for the implementation of our diffusion process :

- forward diffusion :

  β can be learnt or fixed before training

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

- Reverse diffusion :

$$q(\mathrm{x}_{t-1}|\mathrm{x}_t, \mathrm{x}_0) = \mathcal{N}(\mathrm{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathrm{x}_t, \mathrm{x}_0), \tilde{\beta}_t \mathrm{I})$$

$$p_\theta(\mathrm{x}_{t-1}|\mathrm{x}_t) = \mathcal{N}(\mathrm{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathrm{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathrm{x}_t, t))$$

$$\text{Variance : } \textstyle\sum_\theta(x_t, t) = \sigma_t^2 \mathbf{1}$$
$$\sigma_t^2 = \beta_t$$

$$\text{We obtain } p_\theta(\mathrm{x}_{t-1}|\mathrm{x}_t) = \mathcal{N}(\mathrm{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathrm{x}_t, \mathrm{x}_0), \beta_t \mathrm{I})$$

- Simplification of the Kullback-Leibler divergence

$$L_{t-1} = D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))$$

becomes

$$L_{t-1} = \mathbb{E}_{\mathrm{x}_0,\epsilon}\left[\frac{1}{2\|\mathbf{\Sigma}_\theta(\mathrm{x}_t,t)\|_2^2}\|\tilde{\boldsymbol{\mu}}_t(\mathrm{x}_t,\mathrm{x}_0) - \boldsymbol{\mu}_\theta(\mathrm{x}_t,t)\|^2\right]$$

- From a mean predictor to a noise predictor :

$$\tilde{\boldsymbol{\mu}}_t = \frac{1}{\sqrt{\alpha_t}}\left(\mathrm{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right)$$

$$\boldsymbol{\mu}_\theta = \frac{1}{\sqrt{\alpha_t}}\left(\mathrm{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\right)$$

$$\begin{aligned}
L_t &= \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t)\|_2^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0)-\boldsymbol{\mu}_\theta(\mathbf{x}_t,t)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta\|_2^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right)-\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\right)\right\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t,t)\|^2\right]
\end{aligned}$$

Principle of the implemented algortihm

Algorithm 1 Training

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

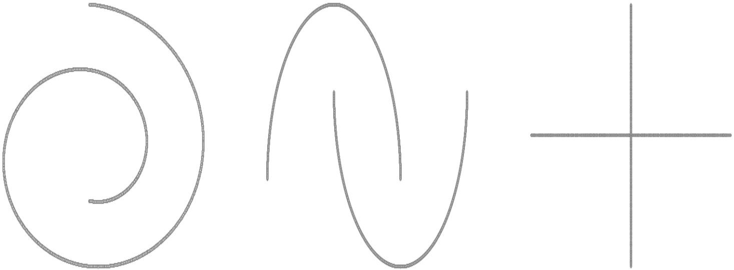
Algorithm 2 Sampling

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

DDPM algorithm

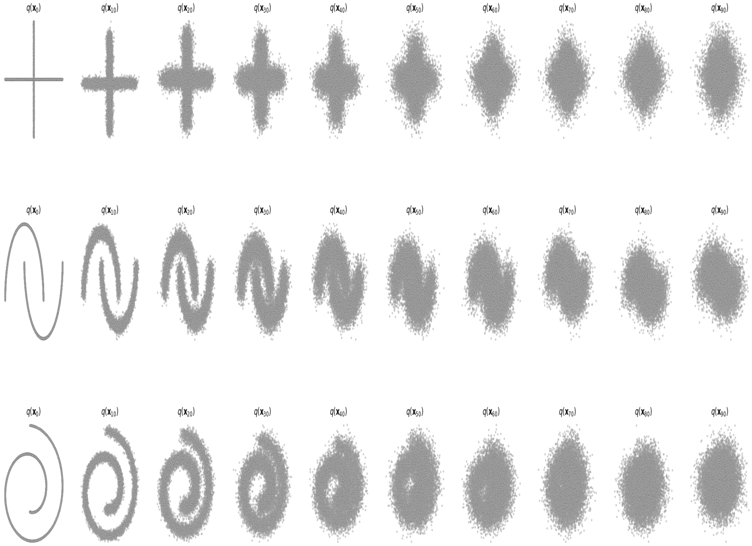
Experiment with 3 toy datasets

Experiments with 3 toy 2D datasets : swiss roll, moons, cross

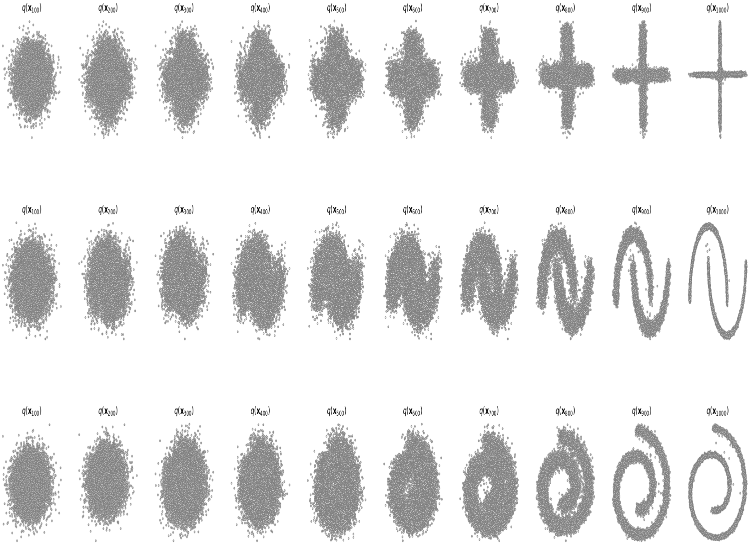


Different characteristics

Experiment with 3 toy datasets : forward process



Experiment with 3 toy datasets : reverse process



Trajectories of data during forward and reverse process

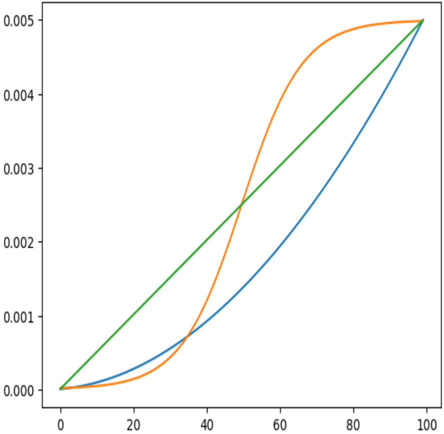
What is the evolution of position of the data during forward and reverse process ?
Does the starting position influence the ending position ?

How does the choice of β_t influence the learning process ?

What is a good choice for β_t schedule ?

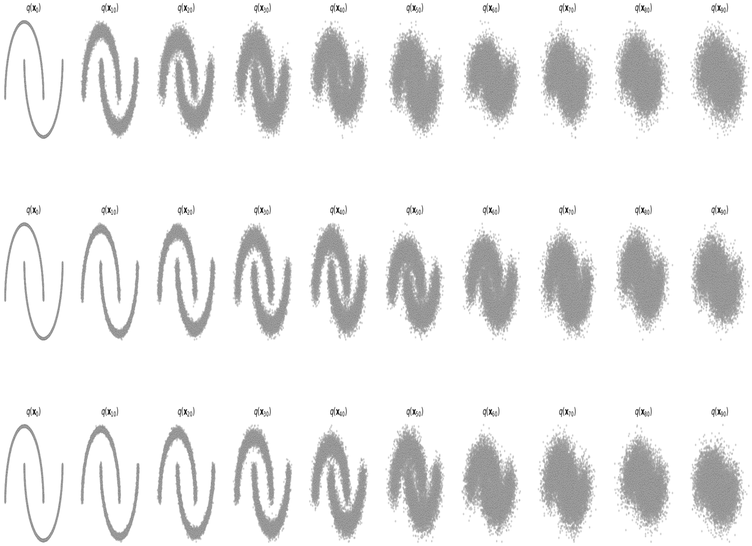
- Must be small compared to the data
- Must be not that small so that we have isotropic gaussian noise "quickly" at the end of the forward process
- Must be inscreasing

How does the choice of β_t influence the learning process ?



betas

How does the choice of β_t influence the learning process ?



How does the choice of β_t influence the learning process ?