Denoising Diffusion Probabilistic Models (DDPM)

Nguyễn Văn Minh 23520945
Nguyễn Huy Phước 23521234
Nguyễn Văn Hồng Thái 23521418

Trường Đại học Công nghệ Thông tin

Ngày 12 tháng 1 năm 2025

Nội Dung

Nội dung

Mục tiêu của nhóm

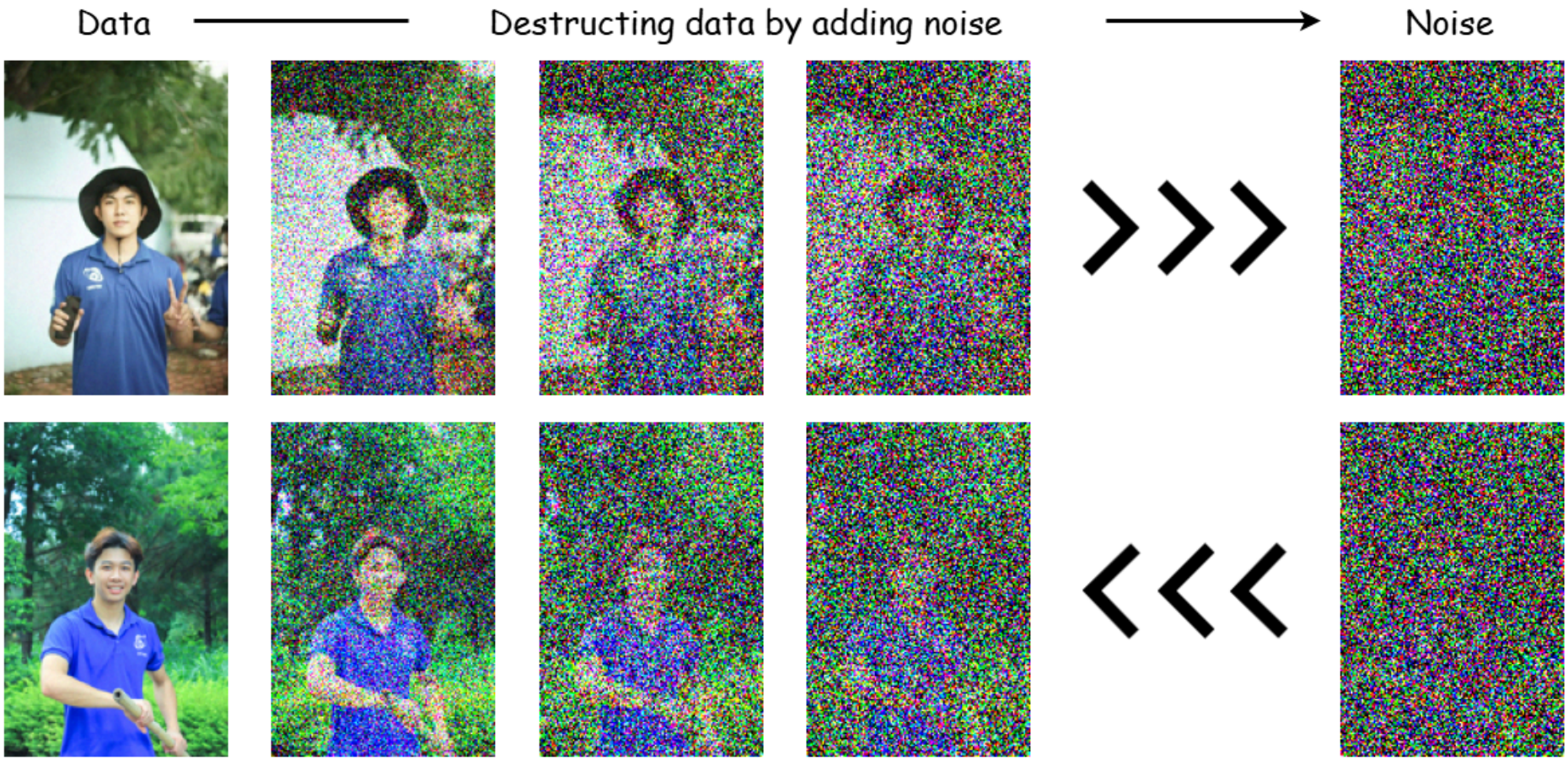
- Nhóm hiểu ý nghĩa của toán học của DDPM.
- Hiểu rõ các nguyên lý cơ bản của DDPM, bao gồm cơ chế khuếch tán (diffusion process) và khử nhiễu (denoising process).
- Nắm bắt cách mô hình sử dụng phương pháp học xác suất để tạo ra dữ liệu mới từ dữ liệu nhiễu.
- Đánh giá DDPM so với các mô hình sinh khác như GAN (Generative Adversarial Networks) hay VAEs (Variational Autoencoders).
- Phân tích ưu và nhược điểm của DDPM khi áp dụng vào các bài toán thực tế.

Ý tưởng tổng quan

Ý tưởng của phương pháp này là biến đổi phân bố dữ liệu thành một phân bố có thể lấy mẫu được. Việc sinh dữ liệu sẽ bắt đầu từ phân bố này, sau đó biến đổi ngược về phân bố ban đầu là 2 quá trình chính để khởi tạo và huấn luyện mô hình. Mô hình cần học được phép biến đổi ngược. Quá trình biến đổi này được mô tả bằng một chuỗi các phân bố, chúng ta sẽ sử dụng quá trình ngẫu nhiên để mô tả chuỗi này.

Ý tưởng tổng quan

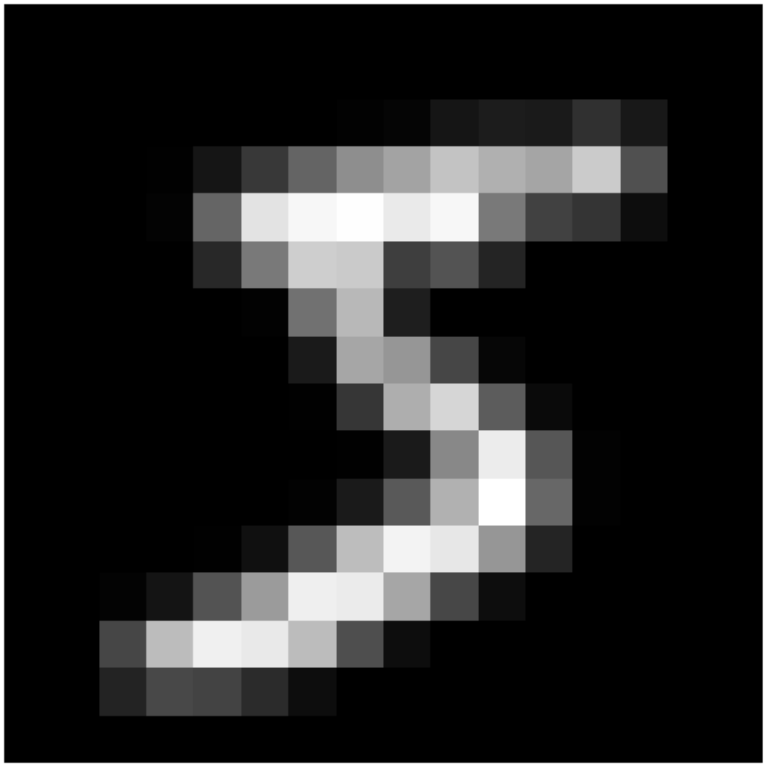
- Quá trình khuếch tán thuận là quá trình chuyển phân phối dữ liệu phức tạp sang một phân phối đơn giản và có thể dễ dàng làm việc.
- Quá trình đảo ngược làm ngược lại quá trình trên, chuyển hóa mẫu phân phối đơn giản về lại với ảnh thực được sử dụng để sinh dữ liệu. Chúng ta sử dụng mạng neural để học cách thực hiện quá trình đảo ngược.


Data
Destructing data by adding noise
Noise
Data
Generating samples by denoising
Noise

Denoising Diffusion Probabilistic Models - Pixel Space

- Là không gian lưu trữ dữ liệu, ta chuyển đổi hình ảnh thành ma trận với mỗi phần tử biểu thị giá trị đen trắng hoặc giá trị màu tại một pixel cụ thể. Đây cũng sẽ là không gian dữ liệu mà mô hình sẽ làm việc.
- Tuy nhiên, các giá trị pixel cần được chuẩn hóa [0,1] trước khi đưa vào mô hình, nhằm tránh các vấn đề Gradient Exploding/Vanishing khi làm việc với khoảng [0,255] và giúp giảm phương sai dữ liệu, cải thiện tốc độ hội tụ của thuật toán tối ưu.

Denoising Diffusion Probabilistic Models - Pixel Space



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	5	21	27	25	47	23	0	0
0	0	0	1	21	55	96	136	156	188	169	158	195	77	0	0
0	0	0	3	97	218	237	244	224	237	117	64	51	13	0	0
0	0	0	0	40	117	198	194	60	80	36	0	0	0	0	0
0	0	0	0	0	2	109	177	29	0	0	0	0	0	0	0
0	0	0	0	0	0	25	159	144	67	6	0	0	0	0	0
0	0	0	0	0	0	1	52	167	205	89	10	0	0	0	0
0	0	0	0	0	0	0	1	25	131	226	83	2	0	0	0
0	0	0	0	0	0	2	25	86	170	245	99	2	0	0	0
0	0	0	0	1	16	84	181	233	222	144	35	0	0	0	0
0	0	3	20	80	149	229	225	159	68	13	0	0	0	0	0
0	0	67	180	230	223	180	75	13	0	0	0	0	0	0	0
0	0	34	69	65	42	13	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Các định nghĩa toán liên quan

- Chuỗi Markov
  Đây là một chuỗi biến đổi toán học mô tả một quá trình ngẫu nhiên mà trạng thái sau chỉ phụ thuộc trạng thái hiện tại và không phụ thuộc vào các trạng thái trước đó:

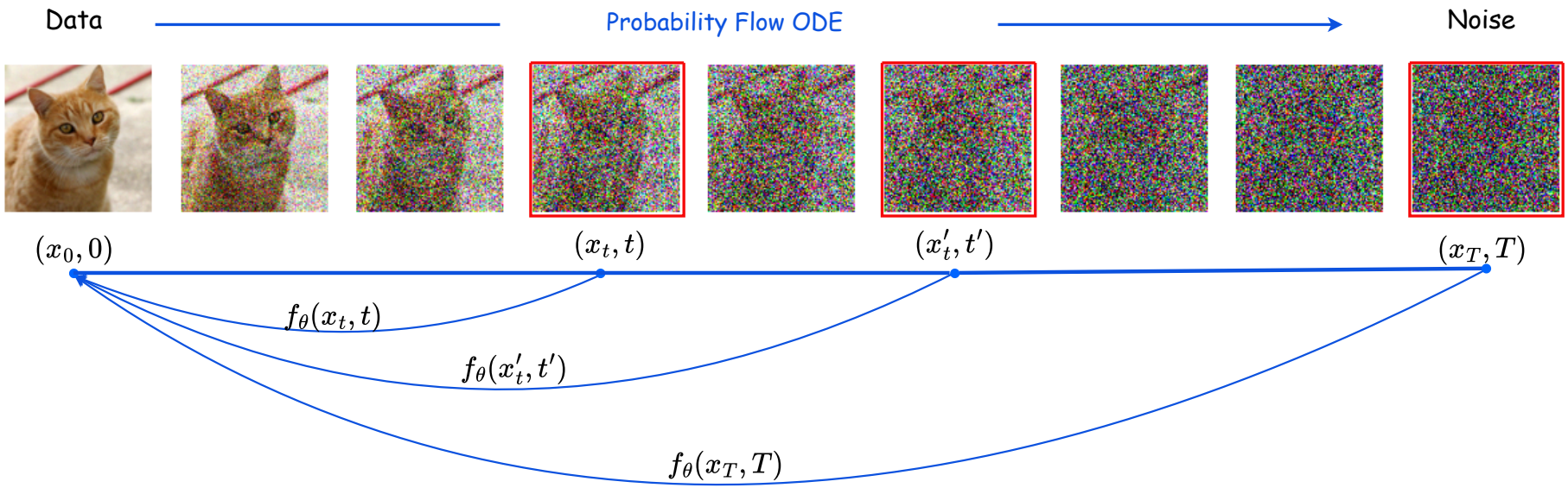
$$P(X_{n+m} = i \mid X_1, \ldots, X_n) = P(X_{n+m} = i \mid X_n).$$

- Nhân biến đổi
  Đây là một hàm mô tả xác suất chuyển từ trạng thái này sang trạng thái khác trong một quá trình ngẫu nhiên. Trong chuỗi Markov, nhân biến đổi chính là ma trận chuyển tiếp hoặc phân phối có điều kiện.

$$K(x_{t-1}, x_t) = P(X_t = x_t \mid X_{t-1} = x_{t-1})$$

Nội dung

Minh họa


Data
Probability Flow ODE
Noise
$(x_0, 0)$
(x_t, t)
(x'_t, t')
(x_T, T)
$f_\theta(x_t, t)$
$f_\theta(x'_t, t')$
$f_\theta(x_T, T)$

Công thức

Ở mỗi quá trình biến đổi điểm dữ liệu, ta sẽ thêm dần dần một lượng nhỏ nhiễu ϵ, nhiễu này sẽ được lấy mẫu dựa vào phân phối gauss (tức là $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$).
Ta có thể hiểu: $x_t = x_{t-1} + \epsilon_{t-1}$,

- ϵ là lượng nhiễu được thêm vào.
- x_0 là dữ liệu; x_1, x_2, ..., x_T là các biến ẩn có cùng số chiều với x_0. T là số bước biến đổi.

Ta cần kiểm soát độ lớn nhiễu thông qua giá trị β_t (tốc độ khếch tán) và đồng thời giảm bớt sự ảnh hưởng của dữ liệu gốc. Ta có công thức:

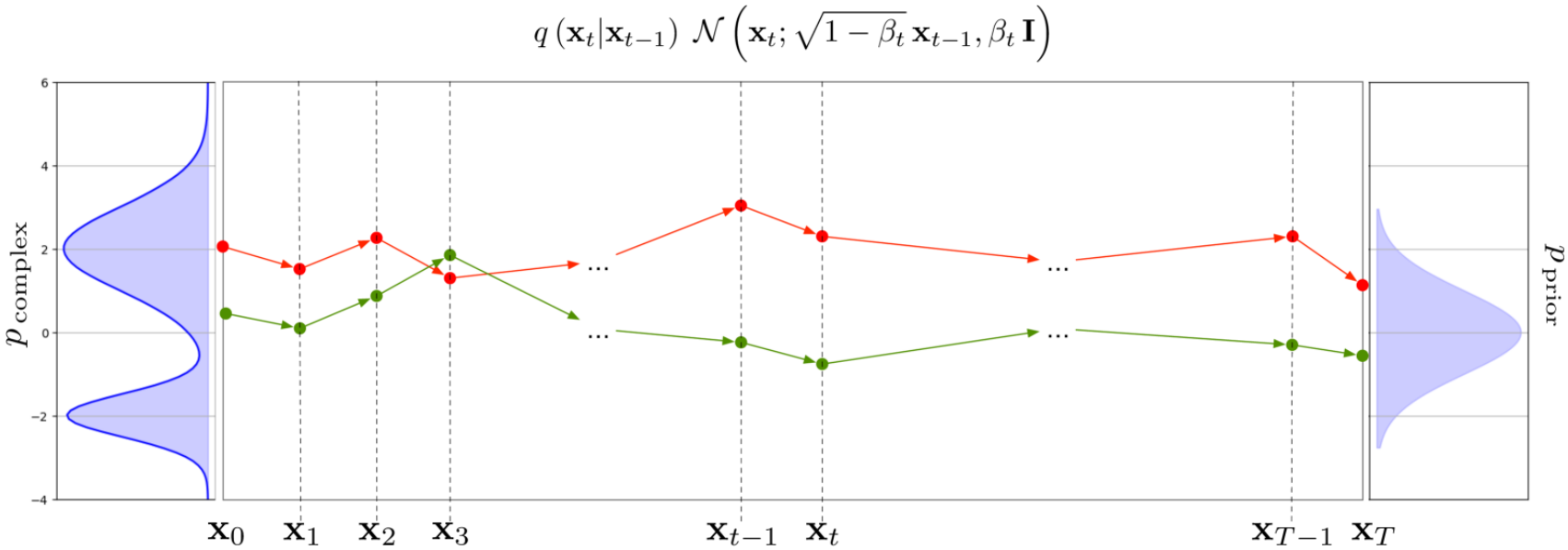
$$x_t = \sqrt{1-\beta_t} x_{t-1} + \sqrt{\beta_t}\epsilon_t \text{ với } \epsilon_t \sim \mathcal{N}(0, \mathbf{I})$$

Công thức

Ta cần kiểm soát độ lớn nhiễu thông qua giá trị β_t (tốc độ khếch tán) và đồng thời giảm bớt sự ảnh hưởng của dữ liệu gốc. Ta có công thức:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_t \text{ với } \epsilon_t \sim \mathcal{N}(0, \mathbf{I})$$

Quá trình khuếch tán trên là mô tả bằng một chuỗi Markov, nghĩa là trạng thái x_t chỉ phụ thuộc vào x_{t-1}. Và từ đó, ta ký hiệu q là hàm mật độ của quá trình khuếch tán, ta có:



Lưu ý - Quá trình nhiễu hóa

Ta có một điều cần đảm bảo , giá trị T cần phải rất lớn và kéo theo tốc độ khếch tán phải đủ nhỏ. T phải đủ lớn là vì: Quá trình thuận thêm nhiễu Gaussian tại mỗi bước:

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

với β_t là lượng nhiễu rất nhỏ. Khi T lớn:

- Nhiễu được thêm dần dần, đảm bảo thông tin dữ liệu x_0 bị phá hủy từ từ.
- Nếu T nhỏ, nhiễu β_t phải lớn hơn, dẫn đến mất thông tin nhanh chóng và gây khó khăn trong quá trình khử nhiễu.

Đảm bảo phân phối x_T gần $\mathcal{N}(0, \mathbf{I})$

Mục tiêu của quá trình thuận là biến x_0 thành nhiễu trắng $x_T \sim \mathcal{N}(0, \mathbf{I})$. Với T lớn:

- Nhiễu được tích lũy dần dần, đảm bảo x_T tiệm cận phân phối Gaussian chuẩn.
- Nếu T nhỏ, x_T có thể chưa đủ nhiễu hóa, làm giảm hiệu quả quá trình ngược.

Cải thiện chất lượng tái tạo dữ liệu

Cải thiện:Cải thiện chất lượng tái tạo dữ liệu Số bước T lớn cho phép khử nhiễu dần dần, cải thiện chất lượng dữ liệu tái tạo (như hình ảnh hoặc âm thanh). Với T nhỏ, dữ liệu tái tạo dễ bị mờ hoặc thiếu chi tiết.

Tổng quát công thức

Ta có một vấn đề, nhìn vào công thức trình bày ở trên ta cần phải tính một cách tuần tự $x_1, x_2, ..., x_t$ mà vì t lớn nên việc tính toán rất mất thời gian, vì vậy ta có một đề xuất tổng quát hóa công thức trên tức là ta sẽ tính x_t từ x_0.

Phân tích

Biểu diễn tại từng bước Ở mỗi bước, x_t được sinh ra từ x_{t-1}:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \mathbf{I}).$$

Biểu diễn cho t

$$x_t = \sqrt{\beta_t}x_{t-1} + \sqrt{1-\beta_t}\epsilon_{t-1} \tag{1}$$

$$\implies x_t = \sqrt{\beta_t}\left(\sqrt{\beta_{t-1}x_{t-2}} + \sqrt{1-\beta_{t-1}}\epsilon_{t-2}\right) \tag{2}$$

$$+ \sqrt{1-\beta_t}\epsilon_{t-1} \quad \text{(mở rộng với } x_{t-1}\text{)} \tag{3}$$

Khai triển: $x_t = \sqrt{\beta_{t-1}\beta_t}x_{t-2} + \underbrace{\sqrt{\beta_t(1-\beta_{t-1})}\epsilon_{t-2}}_{\text{RV1}} + \underbrace{\sqrt{1-\beta_t}\epsilon_{t-1}}_{\text{RV2}}$

Tổng quát hóa và rút gọn công thức

- Tổng quát hóa cho t Hai số hạng cụ thể là RV1 và RV2 ở trên là hai biến ngẫu nhiên phân phối chuẩn với trung bình bằng không và phương sai $\beta_t(1-\beta_t)$, và $(1-\beta)$ tương ứng.
  Mà ta có 2 tính chất, với hai biến ngẫu nhiên độc lập X và Y:
  - $E[X+Y] = E[X] + E[Y]$
  - $VAR[X+Y] = VAR[X] + VAR[Y]$

Tổng quát hóa và rút gọn công thức

Áp dụng tính chất lên $RV1$ và $RV2$, ta có:

$$\Rightarrow x_t = \sqrt{\beta_t\beta_{t-1}}x_{t-2} + \underbrace{\sqrt{\beta_t(1-\beta_{t-1})}\epsilon_{t-2}}_{\text{RV1}} + \underbrace{\sqrt{1-\beta_t}\epsilon_{t-1}}_{\text{RV2}}$$

$$= \sqrt{\beta_t\beta_{t-1}}x_{t-2} + \sqrt{1-\beta_t\beta_{t-1}}\bar{z}_{t-2} \quad (\bar{z}_{t-2}\text{là biến sau khi hợp nhất})$$

$$= \dots$$

$$\Rightarrow x_t = \sqrt{\bar{\beta}_t}x_0 + \sqrt{1-\bar{\beta}_t}\epsilon \quad (\text{vì } \bar{\beta}_t = \prod_{i=1}^{T}\beta_i)$$

Từ đây, ta có rút ra công thức tính nhanh x_t từ x_0 như sau:
$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\beta}_t}x_0, (1-\bar{\beta}_t)\mathbf{I})$

Nội dung

Phân tích tổng quan quá trình đảo ngược

Quá trình đảo ngược cũng là một chuỗi Markov có những trạng thái như quá trình thuận nhưng theo chiều ngược lại. Phân phối của quá trình sinh gồm T bước áp dụng nhân biến đổi (tương tự quá trình thuận):

$$p(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

Với $p(x_T) = \mathcal{N}(x_T; 0, I)$, x_t là điểm cuối cùng của quá trình thuận hoặc điểm được sinh ngẫu nhiên ban đầu.
Để đảo ngược, ta cần xác định nhân biến đổi ngược $p(x_{t-1} \mid x_t)$ và biến đổi T bước để thu được x_0. Điểm dữ liệu tại bước t tuân theo phân phối Gaussian tuy nhiên, dữ liệu gốc hoặc dữ liệu sinh thì không tuân theo phân phối gaussian.

Phân tích tổng quan

Ý tưởng là ta sẽ cố gắng sấp xỉ $p(x_{t-1} \mid x_t)$, tức là chúng ta sẽ định nghĩa một phân phối mới $q(x_{t-1} \mid x_t)$. Với β_t nhỏ thì quá trình thuận và quá trình đảo ngược sẽ có cùng functional form.
Ta có thể viết nhân biến đổi ngược dưới dạng tổng quát nhất như sau:

$$q(x-1 \mid x) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

Phân tích chi tiết

Ở bước đầu tiên, trạng thái khởi tạo là từ một phân phối Gaussian chuẩn:

$$x^{(T)} \sim \mathcal{N}(0, I)$$

Đây là trạng thái hoàn toàn ngẫu nhiên, tương ứng với dữ liệu đã được thêm nhiễu hoàn toàn qua quá trình thuận (forward process).

Biến đổi tại mỗi bước đảo ngược

Quá trình từ $x^{(T)}$ về $x^{(0)}$ được thực hiện qua T bước ngược lại, mỗi bước tuân theo quy tắc:

$$p(x^{(t-1)} \mid x^{(t)}) = \mathcal{N}(x^{(t-1)}; \mu_\theta(x^{(t)}, t), \Sigma_\theta(x^{(t)}, t))$$

Ước lượng mean (μ_θ) Hàm mean $\mu_\theta(x^{(t)}, t)$ được định nghĩa như sau:

$$\mu_\theta(x^{(t)}, t) = \frac{1}{\sqrt{\alpha_t}} \left(x^{(t)} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x^{(t)}, t) \right)$$

Trong đó:

- $\epsilon_\theta(x^{(t)}, t)$: Hàm mạng neural ước lượng nhiễu được thêm vào tại bước t.
- β_t: Lượng nhiễu tại bước t (noise schedule).
- $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$.

Xử lý nhiễu ngẫu nhiên

Tại mỗi bước, mẫu ngẫu nhiên được rút ra từ phân phối Gaussian:

$$x^{(t-1)} \sim \mathcal{N}(\mu_\theta(x^{(t)}, t), \Sigma_\theta(x^{(t)}, t))$$

Trong đó $\Sigma_\theta(x^{(t)}, t)$ thường được giả định là một hàm đơn giản của β_t, ví dụ: $\Sigma_\theta(x^{(t)}, t) = \beta_t I$.

Nội dung

Hàm mục tiêu

Hàm mất mát được tính theo khoảng cách trung bình trong không gian Euclide đối với phân phối được dự đoán ra và phân phối gốc trong quá trình nhiễu hóa và khử nhiễu.

Toàn bộ quá trình đảo ngược tối ưu hóa theo hàm log-likelihood của dữ liệu gốc $x^{(0)}$:

$$L = \mathbb{E}_{x^{(0)},\epsilon\sim\mathcal{N}(0,I),t}\left[\|\epsilon - \epsilon_\theta(x^{(t)}, t)\|^2\right]$$

Hàm mất mát này tối ưu mạng ϵ_θ, giúp mô hình học cách dự đoán nhiễu được thêm vào dữ liệu.

Sự biến đổi trong quá trình nhiễu hóa và khử nhiễu của model DDPM cũng tuần tự theo các bước như model VAEs nên chúng ta có ý tưởng hàm loss của DDPM dựa trên VAEs.

Hàm mục tiêu

Bình thường theo hàm loss của VAEs ta có là

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] \; - \; D_{KL}(q_\phi(z|x) \parallel p_\theta(z))(*)$$

- Ta có $p_\theta(x_0)$ phân phối ngược mà chúng ta đang mô hình hóa. θ biểu diễn các tham số mà mô hình phải tối ưu hóa để thực hiện phân phối ngược
- $q_\phi(z|x)$ là phân phối xấp xỉ, được mô hình hóa bởi mạng suy diễn của VAE. ϕ là các tham số của mạng suy diễn, thường được tối ưu hóa trong quá trình huấn luyện.
- D_{KL} là phân kỳ Kullback-Leibler để đo lường sự khác biệt giữa hai phân phối này.

Biến đổi công thức

Trong DDPM, chúng ta có thể coi x_0 là **biến quan sát được**, trong khi $x_{1:T}$ được xem như **các biến tiềm ẩn**, q là quá trình khuếch tán thuận. Thay thế cách biểu diễn này vào phương trình (*), ta thu được:

$$
\begin{aligned}
\log p_\theta(x_0) &\geq \mathbb{E}_{q(x_{1:T}|x_0)}\big[\log p_\theta(x_0|x_{1:T})\big] - D_{\mathsf{KL}}\big(q(x_{1:T}|x_0)\|p_\theta(x_{1:T})\big) \\
&= \mathbb{E}_{q(x_{1:T}|x_0)}\big[\log p_\theta(x_0|x_{1:T})\big] - \mathbb{E}_{q(x_{1:T}|x_0)}\big[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{1:T})}\big] \\
&= \mathbb{E}_q\big[\log p_\theta(x_0|x_{1:T})\big] - \mathbb{E}_q\big[\log q(x_{1:T}|x_0) - \log p_\theta(x_{1:T})\big] \\
&= \mathbb{E}_q\big[\log p_\theta(x_0|x_{1:T}) - \log q(x_{1:T}|x_0) + \log p_\theta(x_{1:T})\big] \\
&= \mathbb{E}_q\big[\log p_\theta(x_0|x_{1:T}) + \log p_\theta(x_{1:T}) - \log q(x_{1:T}|x_0)\big] \\
&= \mathbb{E}_q\big[\log \frac{p_\theta(x_0|x_{1:T})p_\theta(x_{1:T})}{q(x_{1:T}|x_0)}\big] \\
&= \mathbb{E}_q\big[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\big].
\end{aligned}
$$

Tối ưu hàm mục tiêu ?

Việc huấn luyện được thực hiện bằng cách tối ưu chặn trên của Negative Log Likelihood:

$$
\begin{aligned}
L_{\text{VLB}} &= -\mathbb{E}_q \left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right] \\
&= \mathbb{E}_q \left[\log \frac{q(x_{1:T} \mid x_0)}{p_\theta(x_{0:T})} \right] \geq -\mathbb{E}_q \left[\log p_\theta(x_0) \right]
\end{aligned}
$$

Biến đổi hàm L_{VLB}

Ở đây chúng ta áp dụng định nghĩa chuẩn về kỳ vọng, KL-Divergence và logarit vào phương trình ban đầu ta được như sau:

$$
\begin{aligned}
L_{\text{VLB}} &= \mathbb{E}_q \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} \right] \\
&= \mathbb{E}_q \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)} \right] \\
&= \mathbb{E}_q \left[-\log p_\theta(x_T) + \sum_{t=1}^{T} \log \frac{q(x_t|x_{t-1})}{p_\theta(x_{t-1}|x_t)} \right]
\end{aligned}
$$

Biến đổi hàm L_{VLB}

Để biến đổi tiếp thì chúng ta nhắc lại quy tắc Bayes:

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1}, x_0) \cdot q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

$$L_{\text{VLB}} = \mathbb{E}_{q(x_{1:T}|x_0)} \left[-\log p_\theta(x_T) + \sum_{t=2}^{T} \log \frac{q(x_{t-1}|x_t)}{p_\theta(x_{t-1}|x_t)} + \log \frac{q(x_0|x_1)}{p_\theta(x_0|x_1)} \right]$$

$$= \mathbb{E}_q \left[-\log p(x_T) + \log \frac{q(x_1|x_0)}{p_\theta(x_0|x_1)} + \sum_{t=2}^{T} \log \frac{q(x_t|x_0)}{q(x_{t-1}|x_0)} \right.$$

$$\left. + \sum_{t=2}^{T} \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} \right.$$

Biến đổi hàm L_{VLB}

$$
\begin{aligned}
L_{\text{VLB}} &= \mathbb{E}_q \left[-\log p(x_T) + \log \frac{q(x_1|x_0)}{p_\theta(x_0|x_1)} + \log \frac{q(x_T|x_0)}{q(x_1|x_0)} \right. \\
&\quad \left. + \sum_{t=2}^{T} \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} \right] \\
&= \mathbb{E}_q \left[-\log p(x_T) - \log p_\theta(x_0|x_1) + \log q(x_1|x_0) - \log q(x_1|x_0) \right. \\
&\quad \left. + \log q(x_T|x_0) + \sum_{t=2}^{T} \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} \right] \\
&= \mathbb{E}_{q(x_{1:T}|x_0)} \left[-\log \frac{p(x_T)}{q(x_T|x_0)} - \log p_\theta(x_0|x_1) \right. \\
&\quad \left. + \sum_{t=2}^{T} \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} \right]
\end{aligned}
$$

Biến đổi hàm L_{VLB}

$$
\begin{aligned}
L_{\text{VLB}} &= \mathbb{E}_{q(x_{1:T}|x_0)}\left[-\log p_\theta(x_0|x_1)\right] - \mathbb{E}_{q(x_T|x_0)}\left[\log \frac{p(x_T)}{q(x_T|x_0)}\right] \\
&\quad - \sum_{t=2}^{T} \mathbb{E}_{q(x_t,x_{t-1}|x_0)}\left[\log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}\right] \\
&= \mathbb{E}_{q(x_{1:T}|x_0)}\left[-\log p_\theta(x_0|x_1)\right] + D_{\text{KL}}\left(q(x_T|x_0) \,\|\, p(x_T)\right) \\
&\quad + \sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}\left[D_{\text{KL}}\left(q(x_{t-1}|x_t,x_0) \,\|\, p_\theta(x_{t-1}|x_t)\right)\right] \\
&= -L_0 + L_T + \sum_{t=2}^{T} L_{t-1}
\end{aligned}
$$

Thành phần L_T

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

Xét 2 thành phần của KL divergence bên trên. Thành phần đầu tiên, $q(x_T|x_0)$ chỉ phụ thuộc vào quá trình thuận và không chứa tham số tối ưu được. Thành phần thứ hai, $p_\theta(x_T)$ được chọn trước là $\mathcal{N}(0, I)$. Do đó, L_T là hằng số trong quá trình huấn luyện và có thể bỏ qua thành phần này.

Thành phần L_0

Nếu như các transition kernel $p(x_{t-1}|x_t)$ với $t > 1$ chiếu một không gian liên tục này sang không gian liên tục khác thì với $t = 1$ nó cần chiếu một không gian liên tục về một không gian rời rạc (không gian của input).

Do sự khác biệt đó, thành phần cuối cùng của quá trình đào ngược được tính theo cách riêng:

$$p_\theta(x_0|x_1) = \prod_{i=1}^{D} \int_{-\infty}^{\delta_+^i(x_0^i)} \mathcal{N}(x; \mu_\theta^i(x_1, 1), \sigma_1^2)\, dx$$

$$\delta_+^i(x) = \begin{cases} \infty & \text{if } x = 1 \\ x + \frac{1}{255} & \text{if } x < 1 \end{cases}, \quad \delta_-^i(x) = \begin{cases} -\infty & \text{if } x = -1 \\ x - \frac{1}{255} & \text{if } x > -1 \end{cases}$$

Một số vấn đề mà DDPM gặp phải

- Quá trình suy diễn (khử nhiễu) diễn ra chậm, do rằng không như quá trình thuận có công thức tổng quát. Quá trình nghịch bắt buộc ta đi từng giai đoạn 1, mỗi giai đoạn rất tốn thời gian vì mỗi bước yêu cầu chạy qua toàn bộ mạng. Và như ta đã phân tích số bước T cần đủ lớn để mô hình hoạt động tốt.
- Chất lượng mẫu, ảnh cấu thành có thể không đảm bảo các chi tiết cao, nhỏ có thể bị mờ. Do quá trình khử nhiễu ngược trong DDPM dựa trên việc mô hình ước lượng nhiễu ở mỗi bước. Nhiễu trong mỗi bước được tạo ra từ một phân phối Gaussian, việc khử nhiễu từ phân phối này không thể tái tạo các chi tiết đặc thù.
- Khả năng mở rộng của DDPM, rất kém hiệu quả nếu ta áp dụng trên các loại dữ liệu khác 2D, dữ liệu 3D có cấu trúc không gian phức tạp hơn.YYêu cầu các mô hình học sâu phải hiểu rõ hơn về mối quan hệ không gian ba chiều giữa các điểm. Dữ liệu video có yếu tố thời gian và mối quan hệ giữa các khung hình nó phức tạp hơn rất nhiều so với việc tạo mẫu từ ảnh tĩnh.

Nội dung

FID score

FID đo khoảng cách giữa phân phối của các đặc trưng từ hình ảnh sinh ra (p_g) và hình ảnh thật (p_r). FID sử dụng đặc trưng trích xuất từ một mạng tiền huấn luyện để biểu diễn hình ảnh trong không gian tiềm ẩn.
FID dựa trên sự so sánh hai phân phối Gaussian (phân phối của đặc trưng từ hình ảnh thật và hình ảnh sinh ra).

Công thức

$$FID = \|\mu_r - \mu_g\|_2^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}})$$

Trong đó:

- μ_r, Σ_r: Trung bình và ma trận hiệp phương sai của đặc trưng hình ảnh thật.
- μ_g, Σ_g: Trung bình và ma trận hiệp phương sai của đặc trưng hình ảnh sinh ra.
- $\|\mu_r - \mu_g\|_2^2$: Khoảng cách Euclidean giữa hai vector trung bình.
- $\text{Tr}(\cdot)$: Dấu vết (trace) của ma trận.

Inception score

Inception Score đánh giá chất lượng và sự đa dạng của hình ảnh sinh ra dựa trên dự đoán phân loại từ mạng Inception v3.

- Chất lượng: Hình ảnh sắc nét sẽ có phân phối dự đoán $p(y|x)$ tập trung vào một lớp cụ thể.
- Đa dạng: Hình ảnh sinh ra khác nhau sẽ tạo ra phân phối $p(y)$ (trung bình của các $p(y|x)$) đồng đều trên các lớp.

Công thức

$$IS = \exp\left(\mathbb{E}_x\left[D_{KL}(p(y|x)\|p(y))\right]\right)$$

Trong đó:

- $p(y|x)$: Phân phối dự đoán xác suất nhãn y của một hình ảnh x.
- $p(y) = \frac{1}{N}\sum_{i=1}^{N} p(y|x_i)$: Phân phối tổng hợp từ tất cả hình ảnh sinh ra.
- $D_{KL}(p(y|x)\|p(y))$: Khoảng cách Kullback-Leibler (KL) giữa $p(y|x)$ và $p(y)$.

Nội dung

VAEs model

VAEs được thiết kế để học cách biểu diễn ẩn (latent representation) của dữ liệu, kết hợp bộ tự mã hóa và phương pháp biến nhân. **Hàm mục tiêu** Hàm mục tiêu của VAEs là Evidence Lower Bound (ELBO). Đây là một giới hạn dưới của log-likelihood của dữ liệu, chúng ta sẽ tối lưu hàm ELBO này theo hướng cực tiểu hóa hàm cận dưới.

Công thức được biểu diễn như sau

$$\text{ELBO} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(z, N(0, I_d))$$

Phân tích VAEs

Hàm này có 2 thành phần chính

- Phần tái tạo dùng để reconstruct lại input ban đầu. Các hàm loss thông dụng là Mean Square Error hay Mean Absolute Error. Trong trường hợp ảnh nhị phân, ta có thể sử dụng binary cross entropy.

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$$

- Phần KL divergence sử dụng KL divergence (khoảng cách giữa 2 phân phối xác suất) giữa phân phối chuẩn với trung bình $\mathbb{E}(z)$ và phương sai $\mathbb{V}(z)$ với phân phối chuẩn chuẩn tắc d chiều $N(0, I_d)$.

$$KL(z, N(0, I_d)) = \frac{1}{2}\sum_{i=1}^{d}\left(\mathbb{V}(z_i) - \log \mathbb{V}(z_i) - 1 + \mathbb{E}(z_i)^2\right)$$

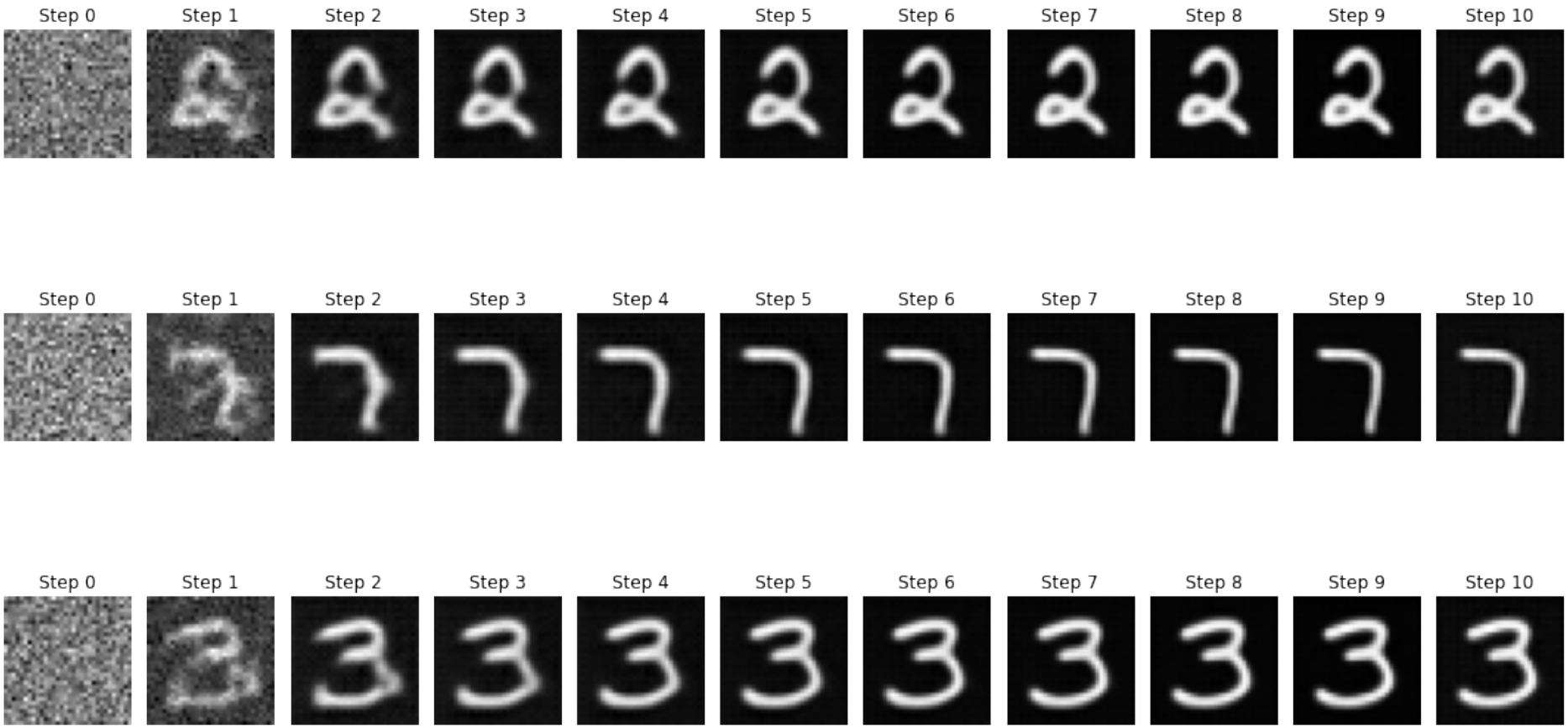
Điểm yếu của VAEs

Vấn đề mà VAEs gặp phải trong quá trình huấn luyện để tối ưu hóa hàm mục tiêu

- Hiện tượng "Posterior Collapse": KL divergence $KL(z, N(0, I_d))$ có thể trở nên quá lớn, khiến mô hình ưu tiên tối ưu hóa phần này thay vì phần tái tạo.
- Nếu KL divergence quá lớn, mô hình sẽ ưu tiên làm cho phân phối ẩn gần với phân phối tiên nghiệm, bỏ qua việc tái tạo dữ liệu. Ngược lại, nếu reconstruction term quá lớn, mô hình có thể bỏ qua việc học một phân phối ẩn có cấu trúc tốt.
- Khoảng cách giữa ELBO và log-likelihood có thể lớn, mô hình sẽ không thể tối ưu hóa đúng với phân phối đúng.
- Khi $\mathbb{V}(z_i)$ tiến gần đến 0, $\log \mathbb{V}(z_i)$ tiến đến $-\infty$, gây ra các vấn đề số học.

DDPM

Chạy thực nghiệm DDPM trên tập dữ liệu MNIST và tiến hành khử nhiễu từ nhiễu random và nhận được mẫu ảnh như dưới đây.



Hình: 10 epochs, 1000 steps per epochs, 300ms/epochs

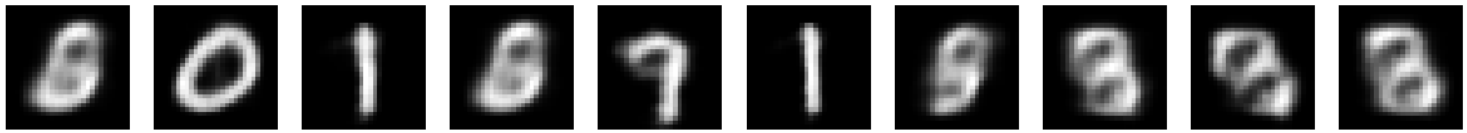
GAN

Chạy thực nghiệm GAN trên tập dữ liệu MNIST và tiến hành khử nhiễu từ nhiều random và nhận được mẫu ảnh như dưới đây.

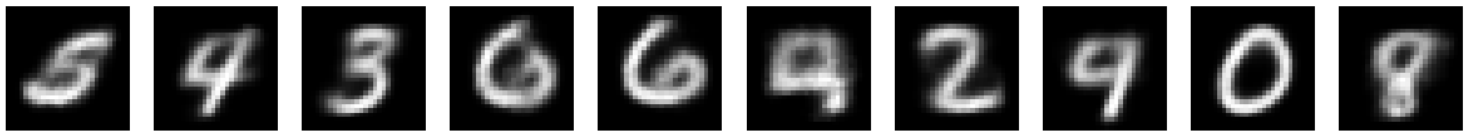
Hình: 30 epochs, 550 steps per epoch,60ms per steps

VAEs

Chạy trên tập dữ liệu MNIST sau đó tiến hành khử nhiều và thu được kết quả như dưới đây.



Hình: 10 Epochs, 547 steps per epoch, 30ms per steps



Hình: 30 Epochs, 547 steps per epoch, 30ms per steps

VAEs

Hình: 70 Epochs, 547 steps per epoch, 50ms per steps

Trích dẫn

Nội dung của bài thuyết trình được trích dẫn từ nguồn:

- Diffusion Models cơ bản[1];
- Sinh dữ liệu với mô hình diffusion và mô hình dạng SDE tổng quát[2]
- Alex Nichol, Prafulla Dhariwal, Improved Denoising Diffusion Probabilistic Models[3]

[1] https://viblo.asia/p/diffusion-models-co-ban-phan-1-E1XVOx884Mz#_ii-diffusion-trong-deep-learning-6

[2] https://viblo.asia/p/sinh-du-lieu-voi-mo-hinh-diffusion-va-mo-hinh-dang-sde-tong-quat-Ljy5V3jMk_cong-thuc-cac-phan-bo-trong-qua-trinh-thuan-cua-mo-hinh-diffusion-16

[3] https://arxiv.org/pdf/2102.09672