Машинное обучение (Machine Learning)

Диффузионные модели

Уткин Л.В.

ПОЛИТЕХ
Санкт-Петербургский
политехнический университет
Петра Великого

Общее

- https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
- Идея диффузионных моделей основана на идеях неравновесной термодинамики.
- Они определяют марковскую цепь шагов диффузии, чтобы медленно добавлять случайный шум к данным, а затем учатся обращать процесс диффузии вспять, чтобы создавать желаемые выборки данных из шума. В отличие от VAE, модели диффузии обучаются с помощью фиксированной процедуры, а скрытая переменная имеет высокую размерность (такую же, как исходные данные).

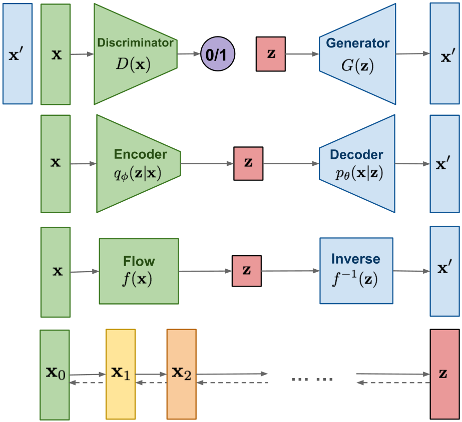
Отличие от других моделей

GAN: Adversarial training

VAE: maximize variational lower bound

Flow-based models: Invertible transform of distributions

Diffusion models: Gradually add Gaussian noise and then reverse



Прямой диффузионный процесс (1)

- Дана точка сгенерированная из реального распределения данных $\mathbf{x}_0 \sim q(\mathbf{x})$
- Определим прямой процесс диффузии, в котором поэтапно за T шагов добавляем небольшое количество гауссова шума к примеру, создавая последовательность зашумленных примеров $\mathbf{x}_1, \ldots, \mathbf{x}_T$

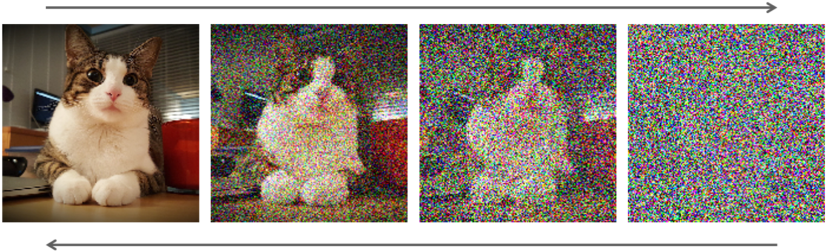
Прямой диффузионный процесс (2)

- Размеры шага контролируются дисперсией $\{\beta_t \in (0,1)\}_{t=1}^T$ которая указывает, сколько шума мы хотим добавить на одном шаге

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}),$$
$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

- Точка $\mathbf{x}_0$ постепенно теряет свои отличительные черты по мере увеличения шага t.

Прямой диффузионный процесс (3)



Прямой диффузионный процесс (4)

- Замечательное свойство процесса - мы можем генерировать $\mathbf{x}_t$ на произвольном временном шаге t в явном виде, используя репараметризацию.
- Пусть $\alpha_t = 1 - \beta_t$ и $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Тогда

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1};\ \boldsymbol{\epsilon}_{t-1}, \boldsymbol{\epsilon}_{t-2}, .. \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2};\ \bar{\boldsymbol{\epsilon}}_{t-2} \text{ объед-ет 2 н.р.}$$

$$= \dots$$

$$= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$$

Итого

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

Прямой диффузионный процесс (5)

- Напомним, что когда мы объединяем два норм. распред. с разной дисперсией $\mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$ и $\mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I})$, новое распределение $\mathcal{N}(\mathbf{0}, (\sigma_1^2 + \sigma_2^2)\mathbf{I})$.
- Здесь объединенное СКО равно $\sqrt{(1-\alpha_t) + \alpha_t(1-\alpha_{t-1})} = \sqrt{1-\alpha_t\alpha_{t-1}}$.
- Обычно можно делать больший шаг обновления, когда пример становится более шумным, поэтому $\beta_1 < \beta_2 < \cdots < \beta_T$ и $\bar{\alpha}_1 > \cdots > \bar{\alpha}_T$

Связь со стохастической градиентной динамикой Ланжевена (1)

- Динамика Ланжевена — это понятие из физики (стат. модел-е молекулярных систем)
- В сочетании со SGD стохастическая градиентная динамика Ланжевена создает примеры из плотности $p(\mathbf{x})$, используя только градиенты $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ в марковской цепи обновлений:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \frac{\delta}{2}\nabla_{\mathbf{x}} \log q(\mathbf{x}_{t-1}) + \sqrt{\delta}\boldsymbol{\epsilon}_t, \text{ где } \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

где δ - размер шага.

Связь со стохастической градиентной динамикой Ланжевена (2)

- Когда $T \to \infty$, $\epsilon \to 0$, $\mathbf{x}_T$ определяется истинной плотностью $p(\mathbf{x})$.
- По сравнению со стандартным SGD стохастическая градиентная динамика Ланжевена вводит гауссов шум в обновления параметров, чтобы избежать коллапса в локальные минимумы.

Обратный диффузионный процесс (1)

- Если мы сможем обратить описанный выше процесс и произвести выборку из $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, мы сможем воссоздать истинную выборку из входного гауссовского шума $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- Если даже β_t достаточно мал, он также будет гауссовым.
- К сожалению, мы не можем просто оценить $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, т.к. для этого нужно использовать весь датасет, и поэтому нужно научить модель p_θ для аппроксимации этих условных вероятностей, чтобы запустить процесс обратной диффузии.

Обратный диффузионный процесс (2)

- К сожалению, мы не можем просто оценить $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, т.к. для этого нужно использовать весь датасет, и поэтому нужно научить модель p_θ для аппроксимации этих условных вероятностей, чтобы запустить процесс обратной диффузии.

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t),$$

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

- "Обратная" условная вероятность поддается обработке, если она условна по $\mathbf{x}_0$

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

Обратный диффузионный процесс (3)

- Используем правило Байеса

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$
$$\propto \exp\left(-\frac{1}{2}\left(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\right)\right)$$

Обратный диффузионный процесс (4)

$$= \exp\left(-\frac{1}{2}\left(\frac{\mathbf{x}_t^2 - 2\sqrt{\alpha_t}\mathbf{x}_t\mathbf{x}_{t-1} + \alpha_t\mathbf{x}_{t-1}^2}{\beta_t}\right.\right.$$

$$\left.\left. + \frac{\mathbf{x}_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\mathbf{x}_{t-1} + \bar{\alpha}_{t-1}\mathbf{x}_0^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\right)\right)$$

$$= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)\mathbf{x}_{t-1}^2\right.\right.$$

$$\left.\left. - \left(\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right)\mathbf{x}_{t-1} + C(\mathbf{x}_t, \mathbf{x}_0)\right) + C(\mathbf{x}_t, \mathbf{x}_0)\right)$$

$C(\mathbf{x}_t, \mathbf{x}_0)$ - некоторая функция, не включающая $\mathbf{x}_{t-1}$.

Репараметризация среднего и дисперсии

Следуя функции плотности Гаусса и репараметризации среднего значения и дисперсии ($\alpha_t = 1 - \beta_t$ и $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$):

$$\tilde{\beta}_t = 1/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) = 1/(\frac{\alpha_t - \bar{\alpha}_t + \beta_t}{\beta_t(1-\bar{\alpha}_{t-1})})$$

$$= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$$

Зачем репараметризация?

- Мы не можем распространять градиент обратно, когда осуществляется генерация выборки.
- Чтобы сделать процесс обучаемым, вводится прием репараметризации:
    - можно выразить случайную величину $\mathbf{x}$ как детерминированную переменную $\mathbf{z} = F_{\phi}(\mathbf{x}, \varepsilon)$, где ε вспомогательная независимая случайная величина, а функция F_{ϕ}, параметризованная ϕ, преобразует ε в $\mathbf{z}$.
    - если $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \alpha, \beta\mathbf{I})$, то $\mathbf{z} = \alpha + \beta \odot \varepsilon$, где $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Обратный диффузионный процесс (6)

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) = \left(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right) / \left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)$$

$$= \left(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right)$$

$$\times \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$$

$$= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0$$

Обратный диффузионный процесс (7)

Так как $\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t)$, то

$$\tilde{\boldsymbol{\mu}}_t = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t)$$

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

ELBO - по аналогии с VAE

$$
\begin{aligned}
-\log p_\theta(\mathbf{x}_0) &\le -\log p_\theta(\mathbf{x}_0) + D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0) \| p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)) \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{\mathbf{x}_{1:T}\sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})/p_\theta(\mathbf{x}_0)}\right] \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_q\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} + \log p_\theta(\mathbf{x}_0)\right] \\
&= \mathbb{E}_q\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right]
\end{aligned}
$$

ELBO - по аналогии с VAE

- Устанавливаем границу L_{VLB}. Пусть

$$L_{\mathrm{VLB}} = \mathbb{E}_{q(\mathbf{x}_{0:T})}\left(\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right) \geq -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0)$$

- Также легко получить тот же результат, используя неравенство Йенсена. Пусть мы хотим минимизировать кросс-энтропию в качестве цели обучения:

$$\begin{aligned} L_{\mathrm{CE}} &= -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0) \\ &= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}\right) \\ &= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} d\mathbf{x}_{1:T}\right) \end{aligned}$$

Неравенство Йенсена

$$\begin{aligned} L_{\text{CE}} &= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right) \\ &\leq -\mathbb{E}_{q(\mathbf{x}_{0:T})} \log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \\ &= \mathbb{E}_{q(\mathbf{x}_{0:T})} \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] = L_{\text{VLB}} \end{aligned}$$

Неравенство Йенсена: $f\left(\sum_{i=1}^n q_i x_i\right) \leq \sum_{i=1}^n q_i f(x_i)$, если $f(x_i)$ - выпуклые и $q_1 + \ldots + q_n = 1,\ q_i \geq 0$

Дальнейшее преобразование

Представим L_{VLB} как комбинацию нескольких членов KL-дивергенции и энтропии:

$$
\begin{aligned}
L_{\text{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \\
&= \mathbb{E}_q\left[\log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=1}^{T} \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right] \\
&=
\end{aligned}
$$

Дальнейшее преобразование

$$= \mathbb{E}_q \Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \Big]$$

$$= \mathbb{E}_q \Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T} \log \left(\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \cdot \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} \right)$$

$$+ \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \Big] =$$

Дальнейшее преобразование

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

Дальнейшее преобразование

$$L_{\text{VLB}} = L_T + L_{T-1} + \cdots + L_0$$

где

$$L_T = D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))$$

$$L_t = D_{\text{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})),\ 1 \leq t \leq T-1$$

$$L_0 = -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$$

В итоге

- Каждый член KL в L_{VLB} (кроме L_0) сравнивает два распределения Гаусса, и поэтому они могут быть вычислены в явном виде.
- L_T является постоянной и может быть проигнорировано во время обучения, поскольку q не имеет обучаемых параметров и $\mathbf{x}_T$ является гауссовским шумом.

Параметризация потерь для обучения (1)

- Напомним, что нам нужно обучить нейронную сеть для аппроксимации условных распределений вероятностей в процессе обратной диффузии

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

- Хотим обучить $\boldsymbol{\mu}_\theta$, чтобы предсказывать

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

Параметризация потерь для обучения (2)

- Поскольку $\mathbf{x}_t$ доступен в качестве входных данных во время обучения, мы можем репараметрировать член гауссовского шума, чтобы он предсказывал ϵ_t на основе входных данных $\mathbf{x}_t$ на шаге t:

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

- Отсюда

$$\mathbf{x}_{t-1} = \mathcal{N}\left(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\right)$$

Параметризация потерь для обучения (3)

- L_t параметризуется, чтобы минимизировать разницу с $\tilde{\boldsymbol{\mu}}$.

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

Упрощение

- Обучение диффузионной модели лучше работает с упрощенной целевой функцией, которая игнорирует весовой коэффициент:

$$
\begin{aligned}
L_t^{\text{simple}} &= \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\epsilon_t}\left[\left\|\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t)\right\|^2\right] \\
&= \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\epsilon_t}\left[\left\|\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)\right\|^2\right]
\end{aligned}
$$

- Окончательно $L_{\text{simple}} = L_t^{\text{simple}} + C$, где C - константа, независящая от θ.

Алгоритм обучения

Algorithm 1 Training

1: **repeat**
2: $\quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $\quad t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: $\quad$ 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

Алгоритм генерации

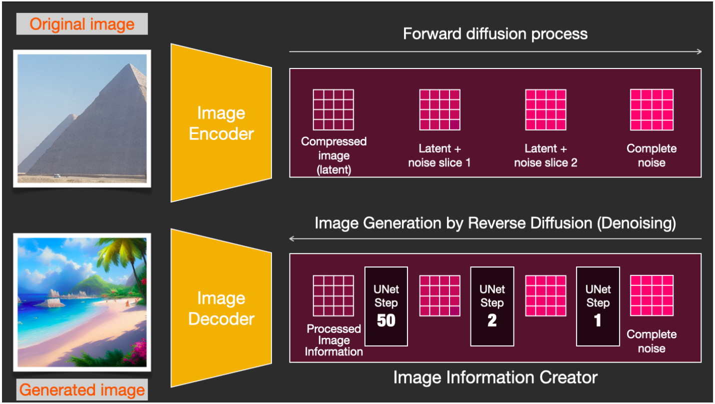
Algorithm 2 Sampling

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

Диффузия сжатых данных, а не пиксельного изображения

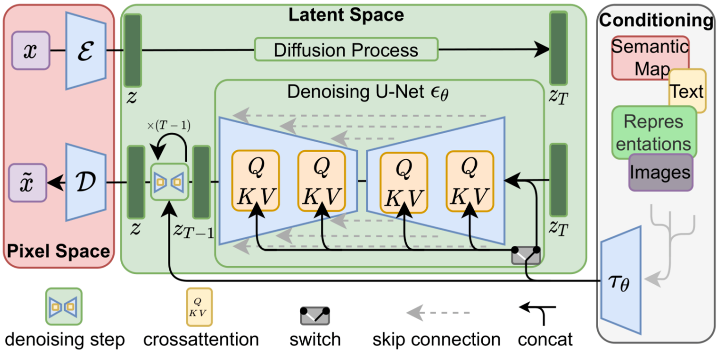
- Для ускорения процесса генерации изображений Stable Diffusion выполняет процесс диффузии не с самими пиксельными изображениями, а со сжатой версией изображения переходом в скрытое пространство (Latent diffusion model - **LDM**).
- Сжатие выполняется при помощи автокодера, который сжимает изображение в скрытое пространство при помощи своего кодера, а затем восстанавливает его при помощи декодера.
- Со сжатыми латентным представлением выполняется прямой процесс диффузии.

Диффузия сжатых данных



https://habr.com/ru/post/693298/

Схема генерация изображения из текста



Еще описание

- Процессы диффузии и denoising происходят на скрытом векторе $\mathbf{z}$.
- Модель denoising представляет собой time-conditioned U-Net с блоками ResNet, дополненную механизмом cross-attention для обработки информации для создания изображений (например, метки классов, семантические карты, размытые варианты изображения) и для представления $\epsilon_\theta(\mathbf{x}_t, t)$.

Еще описание

- Если функция потерь для обычной диффузионной модели имеет вид:

$$L_t^{\text{simple}} = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\epsilon_t}\left[\left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right\|^2\right]$$

- то для латентной модели:

$$L_t^{\text{simple}} = \mathbb{E}_{t\sim[1,T],\mathcal{E}(\mathbf{x}),\epsilon_t}\left[\left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)\right\|^2\right]$$

где $\mathcal{E}$ - кодер

Еще описание

- Это эквивалентно объединению представлений различных модальностей в модель с cross-attention. Каждый тип информации связан с доменным кодером τ_θ для проецирования входа y в промежуточное представление, которое может быть отображено в cross-attention элемент, $\tau_\theta(y)$:

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right) \cdot \mathbf{V}$$

где $\mathbf{Q} = \mathbf{W}_Q^{(i)} \cdot \varphi_i(\mathbf{z}_i)$, $\mathbf{K} = \mathbf{W}_K^{(i)} \cdot \tau_\theta(y)$, $\mathbf{V} = \mathbf{W}_V^{(i)} \cdot \tau_\theta(y)$ и $\mathbf{W}_Q^{(i)} \in \mathbb{R}^{d \times d_\epsilon^i}$, $\mathbf{W}_K^{(i)}, \mathbf{W}_V^{(i)} \in \mathbb{R}^{d \times d_\tau}$, $\varphi_i(\mathbf{z}_i) \in \mathbb{R}^{N \times d_\epsilon^i}$, $\tau_\theta(y) \in \mathbb{R}^{M \times d_\tau}$

Почему U-Net?

- Модели диффузии тесно связаны с идеей автокодеров с шумоподавлением.
- Кроме того, U-Net-подобные архитектуры являются очень распространенной архитектурой для автокодиров изображений.
- Диффузионная U-Net может обусловливать свой вывод текстовыми эмбедингами через слои cross-attention, которые добавляются как к кодеру, так и к декодеру U-Net, обычно между блоками ResNet.

Вопросы

?