

# Stable-Diffusion models

Голов В. А.

Конспекты по пройденному материалу

16 января 2023 г.

## СОДЕРЖАНИЕ

<b>I Диффузионная модель</b>	<b>1</b>
I.1 Что такое диффузионная модель?	1
I.2 Определение процесса обучения . .	1
I.3 Наблюдения и некоторые выводы .	2

## I. ДИФФУЗИОННАЯ МОДЕЛЬ

### I.1. ЧТО ТАКОЕ ДИФФУЗИОННАЯ МОДЕЛЬ?

Положим существует  $q(x_0)$  – распределение исходных данных. То есть распределение в котором выборка  $x_0 \sim q(x_0)$ . **Прямой диффузионный процесс**  $q(x_t|x_{t-1})$  зашумляет данные Гауссовым шумом на каждом шаге  $t$ .

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

где  $\forall t, 0 < \beta_t < 1$  и  $\beta_t > \beta_{t-1}$ . В классической нотации нормальное распределение  $\mathcal{N}(\mu, \sigma^2)$  или в общем виде  $\mathcal{N}(\vec{\mu}, \Sigma)$  зависит от параметров смещения  $\mu$  и разброса  $\sigma$  (среднее и стандартное отклонение). В данном случае  $\mu_t = \sqrt{1 - \beta_t}x_{t-1}$  и  $\sigma_t^2 = \beta_t$ . Преобразование зашумления можно определить при помощи добавления аддитивного шума  $\varepsilon \sim \mathcal{N}(0, I)$  как

$$x_t = \mu_t + \sigma_t \cdot \varepsilon = \sqrt{1 - \beta_t}x_{t-1} + \sqrt{\beta_t}\varepsilon. \quad (2)$$

Это следует из того факта, что если  $\xi \sim \mathcal{N}(0, 1)$ , то  $\eta = \sigma\xi + \mu \sim \mathcal{N}(\mu, \sigma^2)$ . Заметим, что каждое  $\beta_t$  не является постоянной от времени и называется **запланированным разбросом** и может задаваться по-разному (линейно, квадратически, синусом и тд.).

Таким образом, если бы мы знали условное распределение  $p(x_{t-1}|x_t)$ , мы бы могли запустить процесс в обратном порядке и получить  $x_0$  выборку из зашумленной  $x_T$ , где  $t = 0, \dots, T$ .

Так как  $p(x_{t-1}|x_t)$  мы не знаем, приблизим его при помощи параметризованной функции распределения  $p_\theta(x_{t-1}|x_t)$ , где  $\theta$  – веса, обновляемые в

процессе обучения. Так как нормальное распределение зависит от двух параметров, введем параметризованные среднее и разброс ( $\mu_\theta$  и  $\Sigma_\theta$ ). Тогда наше параметризованное распределение имеет вид

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \quad (3)$$

Заметим, что авторы статьи [1] обучают модель только на среднем, а разброс фиксируют как  $\Sigma_\theta = \sigma^2 I = \beta_t I$ , что было улучшено в статье [2].

### I.2. ОПРЕДЕЛЕНИЕ ПРОЦЕССА ОБУЧЕНИЯ

Если рассматривать  $q$  и  $p_\theta$  как VAE, то можно воспользоваться *variational lower bound* (ELBO) для максимизации правдоподобия. В данном случае ELBO преобразуется в сумму  $L = L_0 + L_1 + \dots + L_T$ , где все  $L_t$  кроме  $L_0$  имеют вид MSE ( $L_2$  нормы).

Заметим, что для получения  $x_t$  из  $x_0$  не нужно проделывать все шаги между ними. При известных  $\beta_t$  достаточно выполнить преобразование

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1 - \alpha_t)I), \quad (4)$$

где  $\alpha_t = 1 - \beta_t$  и  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ . Это хорошо, так как позволяет нам оптимизировать случайные члены функции потерь  $L$  (случайным образом семплировать выборку по  $t$ ). Помимо этого, данное свойство позволяет нам использовать **аппроксимацию аддитивного шума** вместо аппроксимации среднего. То есть наше среднее принимает вид

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon_\theta(x_t, t) \right), \quad (5)$$

что позволяет ввести функцию потерь  $L_t$  вида

$$\begin{aligned} L_t &= \|\varepsilon - \varepsilon_\theta(x_t, t)\|_2^2 = \\ &= \|\varepsilon - \varepsilon_\theta(\sqrt{\alpha_t}x_0 + \varepsilon\sqrt{1 - \bar{\alpha}_t}, t)\|_2^2. \end{aligned} \quad (6)$$

Таким образом алгоритм обучения можно свести к виду Алгоритма 1.

Другими словами, речь идет о следующем:

---

**Алгоритм 1:** Обучение модели

---

**повторять**

$x_0 \sim q(x_0);$   
 $t \sim \mathcal{U}[1, T];$   
 $\varepsilon \sim \mathcal{N}(0, I);$   
 $\theta \leftarrow \theta -$   
 $\tau \nabla_{\theta} \|\varepsilon - \varepsilon_{\theta}(\sqrt{\alpha_t}x_0 + \varepsilon\sqrt{1 - \alpha_t}, t)\|_2^2$   
/\* Градиентный спуск \*/

**до тех пор, пока не покрыто;**

---

1. Сэмплируем выборку  $x_0$  из реального распределения  $q(x_0)$ ;
2. Сэмплируем уровень шума из дискретного равномерного распределения  $\mathcal{U}[1, T]$ ;
3. Генерируем шум из нормального распределения и зашумляем данные (как показано выше);
4. На основе зашумленных изображений обучаем сеть определять уровень аддитивного шума.

Далее рассмотрим другие алгоритмы, необходимые при обучении. Алгоритм 2. используется

---

**Алгоритм 2:** Сэмплирование

---

$x_T \sim \mathcal{N}(0, I);$   
**цикл**  $t = T, \dots, 1$  **выполнять**  
  **если**  $t > 1$  **тогда**  
     $z \sim \mathcal{N}(0, I);$   
  **иначе**  
     $z = \vec{0};$   
   $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \varepsilon_{\theta}(x_t, t) \right) + z\sigma_t;$   
**вернуть**  $x_0$

---

авторами статьи [1] для отслеживания прогресса. По-факту речь о том, чтобы сгенерировать шум  $x_T$  самостоятельно, а затем с использованием модели привести его к  $x_0$ . То есть в идеале должно получиться изображение из исходного распределения  $q(x_0)$ .

### 1.3. Наблюдения и некоторые выводы

На выходе имеются смешанные ощущения от модели. С одной стороны прямой процесс зашумления данных кажется очень простым за счет сэмплирования по временному шагу и возможности зашумления  $x_0 \rightarrow x_t$  без промежуточных шагов.

Обратный проход, который и представляет из себя результирующую нейронную сеть требует использовать пошаговое удаление шума, что приводит к циклу при генерации изображения (см алгоритм 2).

Напрашивается вопрос: "можно ли использовать другой инструментарий для получения такого же результата?".

Рассмотрим  $q(x_t|x_{t-1})$  как некоторый дифференциальный закон. Тогда можно сформулировать задачу

$$\begin{cases} \frac{dx(t)}{dt} = f_{\theta}(x(t), t, \theta), \\ x(0) = x_0 \sim q(x_0), \\ x(T) = x_T \sim \mathcal{N}(0, I), \end{cases} \quad (7)$$

где  $f_{\theta}$  - некоторая параметризованная функция, которую необходимо обучить зашумлять исходные данные в рамках ОДУ. Тогда обратных проход дает нам  $x_0$  из  $x_T$

$$x_0 = \int_T^0 f_{\theta}(x(t), t, \theta) dt. \quad (8)$$

Данный подход лишает нас возможности обучать сеть на случайных членах  $L_t$ . Однако можно обучать на выборочных отрезках равной длины с фиксированным шагом  $h$ . Таким образом мы получаем контролируемую непрерывную производную динамики диффузионного процесса.

Этот процесс больше похож на решение задачи **нормализации потока** и может решаться как задача максимизации правдоподобия. Использование ОДУ гарантирует непрерывность и дифференцируемость результирующей функции, что делает трансформацию более устойчивой.

## СПИСОК ЛИТЕРАТУРЫ

- [1] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. June 2020. [arXiv:2006.11239](https://arxiv.org/abs/2006.11239).
- [2] A. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. *CoRR*, abs/2102.09672, 2021. URL: <https://arxiv.org/abs/2102.09672>, [arXiv:2102.09672](https://arxiv.org/abs/2102.09672).