Diffusion Model

Arpan Maity

CS461
National Institute of Science Education and Research Bhubaneswar

30-10-2023

What are diffusion models?

Diffusion models are a class of probabilistic generative models used in machine learning and deep learning.

The core concept involves a methodical and gradual breakdown of the patterns within a data distribution by using a step-by-step forward diffusion process. Subsequently, a complementary reverse diffusion process is learned to reconstruct those patterns, resulting in a versatile and manageable generative model for the data.

They can be used to generate images, audio etc.

The basic Model

- As mentioned earlier it has two processes
  - The forward process
  - The reverse process
- The forward process does not include Machine learning.
- The reverse process is based on the Machine learning.
- The ML part learns how to remove noise for a noisy data and make it less noisy.
- The architecture can be UNet based where the data is projected with ResNet-Block and downsampling to a bottle neck(small resolution) and then again upsampling and using Res-Net block it is projected to the original size.(At some resolutions there can be attention blocks as well)

Notations

- x_t= The image(data) at a particular timestep t. x_0 is thus the original image.
- $q(x_t|x_{t-1})$ corresponds to the forward process.
- $p(x_{t-1}|x_t)$ corresponds to the reverse process.
- $\mathcal{N}$ represents Normal distribution.
- $\mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$ In this x_t is the output, and the rest are inputs,the second term is the mean and the third term is the variance.
- The β lie in between 0 and 1 and these are called schedules and are chnaged in every timestep.

The Forward process

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

A little trick

$$\alpha_t = 1 - \beta_t$$
$$\bar{\alpha_t} = \Pi_{s=1}^{t} a_s$$

Reparameterization technique:

$$\mathcal{N}(\mu, \sigma^2) = \mu + \sigma.\epsilon, \epsilon \sim \mathcal{N}(0, 1)$$

Rewrite the process

$$\begin{aligned} q(x_t|x_{t-1}) &= \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \\ &= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon \\ &= \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon \\ &= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon \\ &\ldots \\ &= \sqrt{\alpha_t\alpha_{t-1}\ldots\alpha_1\alpha_0}x_0 + \sqrt{1-\alpha_t\alpha_{t-1}\ldots\alpha_1\alpha_0}\epsilon \end{aligned}$$

The Forward process

$$= \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$

$$\boxed{\therefore q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I)}$$

The Reverse process

$$p(x_{t-1}|x_t)$$

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

We fixed the varinace and thus we will not predict that.We will start by looking at the loss function.

$$-log(p_\theta(x_0))$$

To find this we have keep track of $t-1$ variables which is not possible. So, we will calculate variational lower bound.

$$-log(p_\theta(x_0)) \leq -log(p_\theta(x_0)) + D_{KL}(q(x_{1:T}|x_0)||p_\theta(x_{1:T}|x_0))$$

The plus sign is there because we want to minimise the the loss. This can further be simplfied to,

$$\Sigma_{t=2}^{T} D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}||x_t)) - log(p_\theta(x_0|x_1))$$

The Reverse process

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \beta I)$$
$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \bar{\mu_t}(x_t, x_0), \bar{\beta}_t I)$$

Now we try to calculate the mean squared error between the actual $\bar{\mu_t}$ and the predicted μ_θ

$$L_t = \frac{1}{2\sigma_t^2}||\bar{\mu_t}(x_t, x_0) - \mu_\theta(x_t, t)||^2$$

This can further be simplified to

$$\boxed{||\epsilon - \epsilon_\theta(x_t, t)||^2}$$

So, we optimise

$$L_{simple} = \mathrm{E}_{\mathrm{t,x_0,\epsilon}}\left(||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2\right)$$

Training algorithm

- 1: repeat
- 2: $x_0 \sim q(x_0)$
- 3: $t\sim$ Uniform($\{1, ..., T\}$)
- 4:$\epsilon \sim \mathcal{N}(0, 1)$
- Take gradient descent step on $\nabla_\theta ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)||^2$
- until converged

Sampling algorithm

- 1: repeat
- 2: $x_T \sim \mathcal{N}(0,1)$
- 3: for $t = T, ..., 1$ do
- 4:$z \sim \mathcal{N}(0,1)$ if $t > 1$ else $z = 0$
- $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha_t}}}\epsilon_\theta(x_t, t)\right) + \sigma_t z$
- end for
- return x_0

Recent development

- Increase the depth and decrease width
- More attention layers
- Increase attention heads
- Adaptive Group Normalization

References

- Deep Unsupervised Learning using Nonequilibrium Thermodynamics, Jascha Sohl-Dickstein and Eric A. Weiss and Niru Maheswaranathan and Surya Ganguli, 2015,arXiv:1503.03585.
- Denoising Diffusion Probabilistic Models, Jonathan Ho and Ajay Jain and Pieter Abbeel, 2020, arXiv:2006.11239.
- Improved Denoising Diffusion Probabilistic Models, Alex Nichol and Prafulla Dhariwal, 2021, arXiv:2102.09672.
- Diffusion Models Beat GANs on Image Synthesis, Prafulla Dhariwal and Alex Nichol, 2021, arXiv:2105.05233.

Thank you!