DDPM

Denoising Diffusion Probabilistic Models (2020) 1k

Diffusion process

Forward diffusion process: input image x_0, add Gaussian noise through T steps

Reverse diffusion process, or in general, sampling process of a generative model: train a neural network to recover the image

Forward diffusion

given x_0 sampled from real distribution $q(x)$:

$x_0 \sim q(x)$

add Gaussian noise with variance β_t to x_{t-1}, producing a new latent variable x_t:

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

from x_0 to x_T:

$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$

reparameterization trick: define $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$, $\epsilon_t \sim \mathcal{N}(0, I)$

$$
\begin{aligned}
x_t &= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1} \\
&= \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\
&= \ldots \\
&= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_0
\end{aligned}
$$

Thus we can sample x_t at any arbitrary timestep using

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

This will be our target later on to calculate our tractable objective loss L_t

The variance parameter β_t can be chosen as a constant or a schedule, e.g., in DDPM paper, linear schedule from $\beta_1 = 0.0001$ to $\beta_T = 0.02$

Reverse diffusion

approximate $q(x_{t-1}|x_t)$ with a parameterized model p_θ (e.g. a neural network); since $q(x_{t-1}|x_t)$ will also be Gaussian for small enough β_t, we choose p_θ to be Gaussian and paramterize the mean and variance:

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

from x_T to x_0:

$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$

Training a diffusion model

needs a reference image x_0 to draw an image, we can sample x_t at noise level t conditioned on x_0:

$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \widetilde{\mu}(x_t, x_0), \widetilde{\beta}_t I)$

where $\widetilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$

and $\widetilde{\mu}(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t$

we already saw in the reparameterization trick that

$x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon)$ where $\epsilon \sim \mathcal{N}(0, I)$

combining last two equations:

$\widetilde{\mu}_t(x_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon)$

thus we can use a neural network $\epsilon_\theta(x_t, t)$ to approximate ϵ and consequently the mean:

$\widetilde{\mu}_\theta(x_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t))$

thus the loss function (the denoising term in the ELBO) can be expressed as:

$$
\begin{aligned}
L_t &= \mathbb{E}_{x_0,t,\epsilon}[\frac{1}{2||\Sigma_\theta(x_t,t)||_2^2}||\widetilde{\mu}_t - \mu_\theta(x_t,t)||_2^2] \\
&= \mathbb{E}_{x_0,t,\epsilon}[\frac{1}{2\alpha_t(1-\bar{\alpha}_t)||\Sigma_\theta||_2^2}||\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||_2^2]
\end{aligned}
$$

in DDPM paper, simplified by ignoring the weighting term:

$L_t = \mathbb{E}_{x_0,t,\epsilon}[||\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||_2^2]$

training and sampling algorithms of DDPM paper:

Algorithm 1 Training

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

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **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 $\mathbf{x}_0$

Background Part of Our Course Reserach Project

Our researches belong to the large family of Diffusion models. The models are inspired by *diffusion process*, which is a kind of random process $(X_t)_{t\geq 0}$ that pictures the location of a particle moving at random but also governed by a drift and a random noise. Unlike other generative models which use discriminators (GAN) or encoders (VAE), etc., diffusion models use a Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$ as the approximate posterior $q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)$, called the *forward diffusion process*:

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

This way, the sample $\mathbf{x}_0$ gradually loses its features and as $T \to \infty$, $\mathbf{x}_T$ becomes an isotropic Gaussian distribution. The variances β_t can be learned by reparameterization or held constant as hyperparameters, and expressiveness of the reverse process is ensured in part by the choice of Gaussian conditionals in $p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$, because both processes have the same functional form when β_t are small.

Now we want to reverse the above process to recreate the sample $\mathbf{x}_0$ from the heavily noised $\mathbf{x}_T$. The model then predicts the distriubtion $p_\theta\left(\mathbf{x}_0\right) := \int p_\theta\left(\mathbf{x}_{0:T}\right) d\mathbf{x}_{1:T}$, where $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are latents of the same dimensionality as the data $\mathbf{x}_0 \sim q\left(\mathbf{x}_0\right)$. The joint distribution $p_\theta\left(\mathbf{x}_{0:T}\right)$ is the *reverse process* as a Markov chain with learned Gaussian transitions starting at $p\left(\mathbf{x}_T\right) = \mathcal{N}\left(\mathbf{x}_T; \mathbf{0}, \mathbf{I}\right)$:

$$p_\theta\left(\mathbf{x}_{0:T}\right) := p\left(\mathbf{x}_T\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$$
$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) := \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right)$$