DDPM

A Preprint

1 Main

diffusion probabilistic model(diffusion model) is a parametrized Markov chain which uses variational inference to produce samples matching the data after finite time. Transitions of this chain are learned to reverse a diffusion process, which is a Markov chain that gradually adds noise to the data in the opposite direction of sampling until signal is destroyed. When the diffusion consists of small amounts of Gaussian noise, it is sufficient to set the sampling chain transitions to conditional Gaussians too, allowing for a particularly simple neural network parameterization.

Langevin dynamics is a physics-inspired mathematical concept describing the motion of a particle in a potential field. In statistics and machine learning, it's adapted as an algorithm for drawing samples from a probability distribution. It works by iteratively updating a sample using two main components:

A "drift"term, which pushes the sample towards regions of higher probability (related to the gradient of the log-probability of the target distribution, also known as the score). A "diffusion"or random noise term, which helps the process explore the sample space. The authors establish a connection between their diffusion models and denoising score matching with annealed Langevin dynamics during sampling

More details on Langevin dynamics: Langevin dynamics models particles movement in a potential field with friction and random thermal noise. This equation can be interpreted as a stochastic analogue of Newton:

$$dx_t = -\nabla U(x_t)dt + \sqrt{2\beta^{-1}}dW_t$$

x_t - state(model parameters), $U(x)$ potential energy (basically logp(x)), β - "noise intensity W_t - Brownian motion

Discrete version $\sim$: $x_{k+1} = x_k - \frac{\epsilon}{2}\nabla U(x_k) + \sqrt{\epsilon}\eta_k$

Brownian motion W_t — it is a process that has the following properties: It starts from zero $W_0 = 0$

continuous trajectory

For every t increments $W_{t+1} - W_t$ independent

Normal distribution of increments $W_{t+s} - W_t \sim N(0; s)$

At each moment of time the particle moves completely randomly, and over time its position is described by a normal distribution with variance growing as t.

Comming to diffusion. As a task we want to model data distribution $x \sim q(x_0)$(for example images using latent variable same dimensions as x_0)

$p_\theta(x_0) = \int p_\theta(x_{0:T})dx_{1:T}$ where $x_{0:T}$ Markov chain of states and θ - model parameters. Distribution $p_\theta(x_{0:T})$ describes reverse generation process(like how to get x_0 starting with gaussian noise x_T). We do not deterministically support x_0, but sample it through probabilistic transitions.

Every point in this distribution is probability density (PDF)!

In more formal way we describe process as: $p_\theta(x_{0:T}) = p(x_T) \cdot \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$ $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}, \mu_\theta(x_t; t), \Sigma_\theta(x_t, t))$

$$q(x_{1:T} \mid x_0) := \prod_{t=1}^{T} q(x_t \mid x_{t-1}), \quad q(x_t \mid x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1},\, \beta_t \mathbf{I})$$

Training is performed by optimizing the usual variational bound on negative log likelihood:

$$\mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}] = \mathbb{E}_q[-\log p(x_T) - \sum_{t\geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}]$$

The forward process 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 $p_\theta(x_{t-1} \mid x_t)$, because both processes have the same functional form when β_t are small [?].

A notable property of the forward process is that it admits sampling x_t at an arbitrary timestep t in closed form: using the notation $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, we have

$$q(x_t \mid x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

We define the reverse process as $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 \mathbf{I})$ for $1 < t \leq T$. The variance σ_t^2 is fixed (not learned); we test two choices: $\sigma_t^2 = \beta_t$ and $\sigma_t^2 = \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. The first is optimal when $x_0 \sim \mathcal{N}(0, \mathbf{I})$, the second — when x_0 is fixed. These are upper and lower bounds on reverse process entropy for unit-variance data [?].

To model the mean $\mu_\theta(x_t, t)$, we minimize:

$$\mathcal{L}_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \left\| \tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t) \right\|^2 \right] + C,$$

where C is constant w.r.t. θ, and $\tilde{\mu}_t$ is the posterior mean from the forward process.

Using the reparameterization $x_t(x_0, \epsilon) = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon$, with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, we rewrite:

$$\mathcal{L}_{t-1} - C = \mathbb{E}_{x_0, \epsilon} \left[\frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(x_t(x_0, \epsilon) - \beta_t \frac{\epsilon}{\sqrt{1-\bar{\alpha}_t}} \right) - \mu_\theta(x_t(x_0, \epsilon), t) \right\|^2 \right].$$

1.1 Data Scaling, Reverse Decoder, and Training Objective

We model discrete image data $x_0 \in \{0, \ldots, 255\}^D$ scaled to $[-1, 1]$. To compute discrete log-likelihoods, we set $p_\theta(x_0 \mid x_1)$ to be a discretized Gaussian:

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

with bounds:

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

2 Training and Sampling

To train, we sample $x_0 \sim q(x_0)$, timestep $t \sim \mathcal{U}(\{1, \ldots, T\})$, and noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, and minimize:

$$\nabla_\theta \left\| \epsilon - \epsilon_\theta \left(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t} \epsilon, t \right) \right\|^2$$

The reverse process mean is parameterized as:

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

which leads to the sampling equation:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t z, \quad z \sim \mathcal{N}(0, \mathbf{I})$$

This resembles Langevin dynamics with ϵ_θ acting as a learned gradient. Using this parameterization, the loss becomes:

$$\mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2\right]$$

Algorithm 1 Training

1: repeat
2: $\quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $\quad t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$
5: $\quad$ Take gradient descent step on $\nabla_\theta \left\|\boldsymbol{\epsilon} - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2$
6: until converged

Algorithm 2 Sampling

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