Denoising Diffusion Probabilistic Models

Oswin Krause, PML, 2022

UNIVERSITY OF COPENHAGEN

Diffusion Generative Models:Idea

- Ornstein-Uhlenbeck (OU) processes transform the data distribution to pure noise
- Idea: Learn to invert this process
- This is called Denoising Diffusion
- Current State-of-the-Art Generative models use this.
- Examples: Dall-E uses "stable"/"guided" Diffusion

OU forward process

- Dataset $\mathcal{D} = \{x^{(0)}, \ldots, x^{(\ell)}\}$, $x^{(i)} \in \mathbb{R}^d$
- Pick number of steps T
- Pick $0 < \beta_t < 1$ for $t = 1, \ldots, T$
- Let $q(x_0) = q_{\mathcal{D}}(x_0)$ be the unknown pdf of the data distribution
- The forward conditionals $q(x_t|x_{t-1})$ follow OU process

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

- This models each element of x_t as variable that follows independent OU process.

(Note: variable names follow the names in the literature for diffusion processes)

Backward/Denoising process I

- To generate, we need $q(x_{t-1}|x_t)$.
- With this, we could generate samples via
  - Sample $x_T \sim q(x_T)$
  - Sample $x_{T-1} \sim q(x_{T-1}|x_T)$
  - ...
  - Sample $x_0 \sim q(x_0|x_1)$
  - x_0 is then sample from $q_{\mathcal{D}}$
- However, $q(x_{t-1}|x_t)$ depends on unknown $q_{\mathcal{D}}$.
- We will learn a model $p_\theta(x_{t-1}|x_t)$ instead

Backward/Denoising process II

The modeled distribution is

$$p_\theta(x_0, \ldots, x_T) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

- Assumption: β_t and T chosen large enough such that

$$p_\theta(x_T) \approxeq \mathcal{N}(x_T; 0, I_d)$$

$\rightarrow$ No need to learn $p_\theta(x_T)$.

- For conditionals, we pick

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

- $\mu_\theta(x_t, t)$ will be a deep neural network.

Learning the model I

- Naive: Learn using maximum Likelihood.

$$\min_{\theta} -\frac{1}{\ell}\sum_{i=1}^{\ell} \log p_\theta(x^{(i)})$$

- Problems:
  - Cannot compute marginal

$$p_\theta(x_0) = \int \cdots \int p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)\, dx_1, \ldots, dx_T$$

  - Does not depend on forward process
  - Common difficulty: $p_\theta(x_0|x_1)$ tries to learn everything and gets stuck.

Learning the model II

- Less Naive: Maximum Likelihood on the forward process

$$\min_{\theta} -\frac{1}{\ell}\sum_{i=1}^{\ell}\int q(x^{(i)}, x_1, \ldots, x_T)\log p_\theta(x^{(i)}, x_1, \ldots, x_T)dx_1, \ldots, dx_T$$

- No need for marginals
- Forward process easy to sample from
- Each $p_\theta(x_{t-1}|x_t)$ gets proper learning signal.

Simplifying the objective I

We can decompose the integral:

$$
\begin{aligned}
& -\frac{1}{\ell}\sum_{i=1}^{\ell}\int q(x^{(i)}, x_1, \ldots, x_T)\log p_\theta(x^{(i)}, x_1, \ldots, x_T)dx_1, \ldots, dx_T \\
&= -E_{X_0,\ldots,X_T}\left[\log p_\theta(X_0, \ldots, X_T)\right] \\
&= -E_{X_0,\ldots,X_T}\left[\log p_\theta(X_T) + \sum_{t=1}^{T}\log p_\theta(X_{t-1}|X_t)\right] \\
&= -E_{X_0,\ldots,X_T}\left[\log p_\theta(X_T)\right] - \sum_{t=1}^{T}E_{X_0,\ldots,X_T}\left[\log p_\theta(X_{t-1}|X_t)\right] \\
&= -E_{X_0,X_T}\left[\log p_\theta(X_T)\right] - \sum_{t=1}^{T}E_{X_0,X_{t-1},X_t}\left[\log p_\theta(X_{t-1}|X_t)\right]
\end{aligned}
$$

Simplifying the objective II

Our objective reads

$$-E_{X_0,X_T}\left[\log p_\theta(X_T)\right] - \sum_{t=1}^{T} E_{X_0,X_{t-1},X_t}\left[\log p_\theta(X_{t-1}|X_t)\right]$$

- The first term can be discarded as we pick $p_\theta(x_T)$ constant.
- We continue simplifying the second term.

Simplifying the objective III

We have

$$-E_{X_0,X_{t-1},X_t}\left[\log p_\theta(X_{t-1}|X_t)\right] = -\int q(x_0, x_{t-1}, x_t) \log p_\theta(x_{t-1}|x_t) dx_0, dx_{t-1}, dx_t$$

- Idea: Try to marginalize out variables.
- The more we marginalize the less we have to sample.
- Problem: x_t cannot be marginalized as

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

- We cannot integrate through $\mu_\theta(x_t, t)$.
- But we can integrate x_{t-1}.

Simplifying the objective IV

Reorder:

$$
\begin{aligned}
& - E_{X_0, X_{t-1}, X_t}\left[\log p_\theta(X_{t-1}|X_t)\right] \\
= & - \int q(x_0, x_{t-1}, x_t) \log p_\theta(x_{t-1}|x_t) dx_0, dx_{t-1}, dx_t \\
= & - \int q_{\mathcal{D}}(x_0) q(x_t|x_0) q(x_{t-1}|x_0, x_t) \log p_\theta(x_{t-1}|x_t) dx_0, dx_{t-1}, dx_t
\end{aligned}
$$

- $q(x_t|x_0)$: marginal distribution of OU process
- Marginalized over $x_1, \ldots, x_{t-1}$
- $q(x_{t-1}|x_0, x_t)$: Conditional distribution of x_{t-1} given x_0, x_t
- Since the OU process is a GP, these are both normal distributions

Simplifying the objective V

The Integral can be solved in closed form (tedious):

$$
\begin{aligned}
&-\int q(x_{t-1}|X_0, X_t) \log p_\theta(x_{t-1}|X_t) dx_{t-1} \\
=&\frac{1}{2\beta_t}\|\tilde{\mu}_{t-1}(X_0, X_t) - \mu_\theta(X_t, t)\|^2 + \text{const}
\end{aligned}
$$

Where the mean of $X_{t-1}|X_0, X_t$ is given by

$$
\tilde{\mu}_{t-1}(X_0, X_t) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}X_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}X_t
$$

$$
\bar{\alpha}_j = \prod_{i=1}^{t}(1-\beta_t)
$$

Simplifying the objective VI

Inserting back into our objective, it now reads:

$$
\begin{aligned}
&- \sum_{t=1}^{T} E_{X_0, X_{t-1}, X_t} \left[\log p_\theta(X_{t-1} | X_t) \right] \\
&= \sum_{t=1}^{T} \frac{1}{2\beta_t} E_{X_0, X_t} \left[\| \tilde{\mu}_{t-1}(X_0, X_t) - \mu_\theta(X_t, t) \|^2 \right]
\end{aligned}
$$

Intuition:

- Squared distance between the mean of $X_{t-1}|X_t, X_0$ and the model prediction $\mu_\theta(X_t, t)$
- The model does not know about X_0, thus the best it can do is the mean of $X_{t-1}|X_t$

Simplifying the objective VII

Our objective now reads:

$$\sum_{t=1}^{T} \frac{1}{2\beta_t} E_{X_0, X_t} \left[\|\tilde{\mu}_{t-1}(X_0, X_t) - \mu_\theta(X_t, t)\|^2 \right]$$

- One could train a model with this, but that often works poorly.
- Problem: the model has to learn small perturbations to a (potential) large input
- Approach: find a form of μ_θ that looks close to $\tilde{\mu}_{t-1}$ and has explicit perturbations
- Next: Since μ_θ cannot depend on X_0, eliminate it from $\tilde{\mu}_{t-1}$

Eliminate X_0

Idea: write X_0 in terms of X_t and perturbation ϵ

- Since the forward process is an OU process, we have

$$X_t|X_0 \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}X_0, 1-\bar{\alpha}_t\right)$$

$$\bar{\alpha}_j = \prod_{i=1}^{t}(1-\beta_t) \ .$$

Eliminate X_0

Idea: write X_0 in terms of X_t and perturbation ϵ

- Since the forward process is an OU process, we have

$$X_t|X_0 \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}X_0, 1-\bar{\alpha}_t\right)$$

$$\bar{\alpha}_j = \prod_{i=1}^{t}(1-\beta_t) \ .$$

- Thus,

$$X_t = \sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, \epsilon_t \sim \mathcal{N}(0, I_d)$$

Eliminate X_0

Idea: write X_0 in terms of X_t and perturbation ϵ

- Since the forward process is an OU process, we have

$$X_t|X_0 \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}X_0, 1-\bar{\alpha}_t\right)$$

$$\bar{\alpha}_j = \prod_{i=1}^{t}(1-\beta_t) \ .$$

- Thus,

$$X_t = \sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, \epsilon_t \sim \mathcal{N}(0, I_d)$$

- Solve for X_0:

$$X_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(X_T - \sqrt{1-\bar{\alpha}_t}\epsilon_t\right)$$

Eliminate X_0

We have:

$$\tilde{\mu}_{t-1}(X_0, X_t) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} X_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} X_t$$

and insert

$$X_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(X_T - \sqrt{1-\bar{\alpha}_t}\epsilon_t \right)$$

Then, after LONG and TEDIOUS calculations, we obtain:

$$\tilde{\mu}_{t-1}(X_0, X_t) = \frac{1}{\sqrt{1-\beta_t}} \left(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t \right) \quad .$$

Choosing the model

We computed:

$$\tilde{\mu}_{t-1}(X_0, X_t) = \frac{1}{\sqrt{1-\beta_t}}\left(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right) \quad .$$

We now pick a similar shape of model:

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

Insertion into objective

We have:

$$\|\tilde{\mu}_{t-1}(X_0, X_t) - \mu_\theta(X_t, t)\|^2$$

We computed:

$$\tilde{\mu}_{t-1}(X_0, X_t) = \frac{1}{\sqrt{1-\beta_t}}\left(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right) \quad .$$

And picked:

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

Insertion into objective

We have:

$$\left\| \frac{1}{\sqrt{1-\beta_t}} \left(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_t \right) - \frac{1}{\sqrt{1-\beta_t}} \left(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(X_t, t) \right) \right\|^2$$

We computed:

$$\tilde{\mu}_{t-1}(X_0, X_t) = \frac{1}{\sqrt{1-\beta_t}} \left(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_t \right) \quad .$$

And picked:

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

Insertion into objective

We arrive at:

$$\|\tilde{\mu}_{t-1}(X_0, X_t) - \mu_\theta(X_t, t)\|^2 = \frac{\beta_t^2}{(1-\beta_t)(1-\bar{\alpha}_t)} \|\epsilon_t - \epsilon_\theta(X_t, t)\|^2$$

with

$$X_t = \sqrt{\bar{\alpha}_t} X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, \epsilon_t \sim \mathcal{N}(0, I_d)$$

Bringing everything together

Let us recapitulate what we have done so far:

Training objective

$$-\frac{1}{\ell}\sum_{i=1}^{\ell}\int q(x^{(i)}, x_1, \ldots, x_T)\log p_\theta(x^{(i)}, x_1, \ldots, x_T)dx_1, \ldots, dx_T$$

Step 1: decompose integrals

$$-\sum_{t=1}^{T} E_{X_0, X_{t-1}, X_t}\left[\log p_\theta(X_{t-1}|X_t)\right]$$

Step 2: Integrate over X_{t-1}

$$\sum_{t=1}^{T}\frac{1}{2\beta_t}E_{X_0, X_t}\left[\|\tilde{\mu}_{t-1}(X_0, X_t) - \mu_\theta(X_t, t)\|^2\right]$$

Bringing everything together

Step 3: Simplify the squared norm

$$\sum_{t=1}^{T} \frac{1}{2\beta_t} E_{X_0, X_t} \left[\frac{\beta_t^2}{(1-\beta_t)(1-\bar{\alpha}_t)} \left\| \epsilon_t - \epsilon_\theta(X_t, t) \right\|^2 \right]$$

with

$$X_t = \sqrt{\bar{\alpha}_t} X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, \epsilon_t \sim \mathcal{N}(0, I_d)$$

and

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

Bringing everything together

Step 3: Simplify the squared norm

$$\sum_{t=1}^{T} \frac{\beta_t}{2(1-\beta_t)(1-\bar{\alpha}_t)} E_{X_0, X_t}\left[\|\epsilon_t - \epsilon_\theta(X_t, t)\|^2\right]$$

with

$$X_t = \sqrt{\bar{\alpha}_t} X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, \epsilon_t \sim \mathcal{N}(0, I_d)$$

and

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

Bringing everything together

Step 4: Reparameterization Trick

$$\sum_{t=1}^{T} \frac{\beta_t}{2(1-\beta_t)(1-\bar{\alpha}_t)} E_{X_0,\epsilon_t}\left[\left\|\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)\right\|^2\right]$$

with

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

Final objective

Step 5: Removal of normalization term

$$E_{t,X_0,\epsilon_t}\left[\left\|\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)\right\|^2\right]$$

- This is the final training objective.
- $\epsilon_\theta(X_t, t)$ is our model.
- Practical implementation:
  1. sample t, X_0 uniformly
  2. Compute $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$
  3. Compute squared norm.
  4. Perform gradient descent.

Sampling from the model

We picked

$$X_{t-1}|X_t \sim \mathcal{N}\left(\mu_\theta(X_t, t), \beta_T I_d\right)$$

with

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

Start with $X_T \sim \mathcal{N}(0, I_d)$ and then sample backwards.

Practical Models

How to pick: $\epsilon_\theta(X_t, t)$

- Often: U-Net architecture or similar
- Add an embedding of t to the scaled U-Net blocks
  - Compute $\phi = (\sin(2\pi t/T), \sin(4\pi t/T), \sin(6\pi t/T), \dots)^T$
  - Append ϕ to feature vectors going into the layers
  - Neural Networks perform much better with these embeddings, than using t directly.