Denoising Diffusion Probabilistic Models

Key terms identified:

1. Weighted Variational Bound
2. Langevin Dynamics
3. Progressive Lossy Decompression - Generalization of Autoregressive Decoding

Diffusion Probabilistic Models (DPM)

1. DPM is a parameterized Markov Chain trained using variational inference to produce samples matching the data after finite time.
2. 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.
3. 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.

A certain parameterization of diffusion models reveals an equivalence with denoising score matching over multiple noise levels during training and with annealed langevin dynamics during sampling.

Background - Diffusion Models (DM)

DM are latent variable models of the form $p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}$ where $x_1, ..., x_T$ are latents of the same dimensionality as the data $x_0 \sim q(x_0)$.

Reverse Process

The joint distribution $p_\theta(x_{0:T})$ is called the **reverse process**, it is defined as a markov chain with learned Gaussian transitions starting at $p(X_T) = \mathcal{N}(X_t; 0, \mathbf{I})$:

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

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

From whatever I read, p is always the reverse process.

Forward Process

The thing that distinguishes diffusion models from other types of latent variable models is that the approximate posterior $q(x_{1:T}|x_0)$, is call the **forward process** or **diffusion process** is fixed to a markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, ..., \beta_T$:

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

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

The forward process is always denoted with the alphabet q. In the Diffusion process the forward process is fixed that means the forward process has no role in training it is initialized at the very begining itself, so only the reverse process is trainable.

Training

Training is performed by optimizing the usual variational bound on negative log likelihood. The Eq. 3 (2) is about the Variational bound mainly the ELBO which is derived in the Variational Inference paper by Kingma (1).

Derivation

We start from fixed datapoint x_0:

$$p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}$$

Insert $q(x_{1:T}|x_0)$ (Multiply and Divide):

$$p_\theta(x_0) = \int q(x_{1:T}|x_0)\ \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\ dx_{1:T}$$

The main part: This integral is exactly an expectation under q:

$$p_\theta(x_0) = \mathbb{E}_{q(x_{1:T}|x_0)}\ \left[\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right]$$

Apply logs on both sides:

$$\log p_\theta(x_0) = \log \mathbb{E}_{q(x_{1:T}|x_0)}\ \left[\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right]$$

So, here is where the Jensen Inequality comes in:

$$\log p_\theta(x_0) = \log \mathbb{E}_{q(x_{1:T}|x_0)} \ [\log Z] \ \ where \ \ Z = \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \ > \ 0$$

Since $\log(.)$ is concave, Jensen says:

$$\log \mathbb{E}_q[Z] \geq \mathbb{E}_q[\log Z]$$

Pushing the log inside, it becomes an inequality - so Jensen inequality.

We see inequality flip:

$$-\log \mathbb{E}_q[Z] \leq \mathbb{E}_q[-\log Z]$$

Why do we want to use Jensen here? - The core problem is $\log p_\theta(x_0)$ is a log of an integral (intractable). Even with $\log \mathbb{E}[.]$ it is still had to opitmize. So, Jensen gives a tractable objective where we take that the log is concave. With now, $\mathbb{E}[\log Z]$ - sampling from q is easy.

Then, we expand the joint distributions to get to the following:

$$\mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right]$$

$$\mathbb{E}_q\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right] = \mathbb{E}_q\left[-\log p(x_T) - \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})}\right] =: L$$

The forward process variances β_t can be learned by reparameterization (the reparameterization trick: $z = \mu + \sigma \odot \epsilon$ where the odot is the element wise product and $\epsilon \sim \mathcal{N}(0,1)$ or held constant as hyperparameters, and expressiveness of the reverse process is ensured in part by the choice of the Gaussian conditionals in $p_\theta(x_{t-1}|x_t)$, because both processes have the same functional form when β_t are small from the Nonequilibrium Thermodynamics (3). 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 $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, so we have:

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

Efficient training is therefore possible by optimizing random terms of L with stochastic gradient descent. Further improvements come from variance reduction by rewriting L Eq(3) (2) as:

$$\mathbb{E}_q\left[D_{KL}(q(x_T|x_0) \parallel p(x_T)) + \sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1)\right]$$

The first part is L_T, second is L_{t-1} and the third is L_0.

This equation uses KL divergence to directly compare $p_\theta(x_{t-1}|x_t)$ against forward process posteriors, which are tractable when conditioned on x_0:

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}\Big(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \, \mathbf{I}\Big)$$

$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1-\bar{\alpha}_t}\, x_0 \;+\; \frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\, x_t$$

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\,\beta_t$$

Diffusion models and denoising autoencoders

Explicity connection between diffusion models and denoising score matching (IDK).

Forward process and L_T

Here the β_t is fixed. So no learnable parameters here. So, the L_T part of the above equation is constant and can be ignored.

Reverse process and $L_{1:T-1}$

T is the number of Timesteps.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sum\nolimits_\theta(x_t, t)) \;\; for \;\; 1 < t \leq T$$

$\sum_\theta(x_t, t)) = \sigma_t^2 \mathbf{I}$ to untrained time dependent constants.

The main thing that we have is that in L_{t-1}, we have this term $\sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t))$ so we have q and p.

The main idea is that it would be better have both q and p follow gaussian so as to get to a closed form solution.

They show the *true* forward posterior is Gaussian:

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

Equation 6-7 from (2).

And they choose the model reverse step to also be Gaussian:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$$

Equation in 3.2 section of (2).

So each KL term $D_{KL}(\text{Gaussian} \parallel \text{Gaussian})$ becomes something you can compute in closed form.

The important simplification: fixed variances ⇒ KL becomes (weighted) MSE on means.

If you fix σ_t^2, then the KL between Gaussians is basically:

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

Equation. (8) of (2)

So at this point: **make μ_θ match $\tilde{\mu}_t$.**

Then, we move from μ_θ to ϵ.

So, first forward process can be written as this:

$$x_t = \sqrt{\tilde{\alpha}_t}\, x_0 + \sqrt{1-\tilde{\alpha}_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

We use this reparameterization to rewrite the loss. Now, we will still use the μ_θ but with now ϵ_θ.

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

where ϵ_θ is a function approximator intended to predict ϵ from x_t. To sample $x_{t-1} \sim p_\theta(x_{t-1}|x_t)$ is to compute:

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

where $z \sim \mathcal{N}(0, \mathbf{I})$

References

1. https://arxiv.org/abs/1312.6114 - Auto-Encoding Variational Bayes
2. https://arxiv.org/abs/2006.11239 - Denoising Diffusion Probabilistic Models
3. https://arxiv.org/abs/1503.03585 - Deep Unsupervised Learning using Nonequilibrium Thermodynamics