AMATH 840: Advanced Numerical Methods for Computational and Data Science

Giang Tran
Department of Applied Mathematics, University of Waterloo
Winter 2024

Part 2: Neural Networks

2.3: A Detailed Mathematical Explanation of Denoising Diffusion Probabilistic Models (DDPM)

Winter 2024

Outline

Forward Process

Reverse Process

Denoising Diffusion Probabilistic Models (DDPMs)

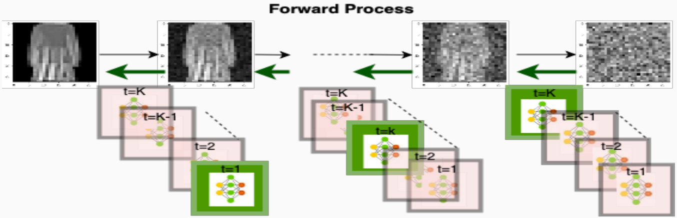

Figure 1: Example of Forward and Reverse Processes.

- All integer Equation numbers (Eq. 1,...) are the same numbers as in DDPM[1].
- The content is based on the previous notes of my PhD student Esha Saha and on discussion with my collaborators Hai Ha Pham (Vietnam National University - Ho Chi Minh City, Vietnam) and Sang Ngoc Pham (EM Normandie Business School, France)

[1]Reference: "Denoising Diffusion Probabilistic Models", by Ho et al, NeurIPS 2020, https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

Forward Process

Definition 1: Let $\mathbf{x}_0 \in \mathbb{R}^d$ from an unknown distribution with p.d.f. $q(\mathbf{x}_0)$. Given a variance schedule $0 < \beta_1, ..., \beta_K < 1$, the forward process is fixed to a Markov chain that gradually adds Gaussian noise to the data:

$$q_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) := \mathcal{N}(\mathbf{x}_k; \sqrt{1-\beta_k}\mathbf{x}_{k-1}, \beta_k \mathbf{I}). \qquad \text{(Eq. 2)}$$

That is,

$$\mathbf{x}_k := \sqrt{1-\beta_k}\mathbf{x}_{k-1} + \sqrt{\beta_k}\,\mathbf{e}, \quad \text{where } \mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \text{ and } k = 1, \ldots, K. \qquad \text{(Eq. 2.1)}$$

Forward Process (cont'd)

Lemma 1: With the assumptions in Definition 1, we have

$$q_{k|0}(\mathbf{x}_k|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_k; \sqrt{\overline{\alpha}_k}\,\mathbf{x}_0, (1-\overline{\alpha}_k)\mathbf{I}), \qquad \text{(Eq. 4)}$$

where $\alpha_k = 1 - \beta_k$ and $\overline{\alpha}_k = \prod_{i=1}^{k} \alpha_i$ for $k = 1, \ldots, K$. That is,

$$\mathbf{x}_k = \sqrt{\overline{\alpha}_k}\,\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_k}\,\widetilde{\mathbf{e}}_k, \qquad \text{(Eq. 4.1)}$$

where $\widetilde{\mathbf{e}}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Note that for any $\tau \geq 1$, $\widetilde{\mathbf{e}}_k$ and $\widetilde{\mathbf{e}}_{k+\tau}$ are not independent.

In particular, if $0 < \beta_1 < \ldots < \beta_K < 1$ or $0 < \gamma \leq \beta_1, \ldots, \beta_K < 1$, we have

$$\mathbf{x}_K \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathbf{I}). \qquad \text{(Eq. 4.2)}$$

Comment: Based on (Eq. 4.2), in the reverse process, we start with $\mathbf{x}_K \sim \mathcal{N}(0, 1)$.

Forward Process (cont'd)

Proof of Lemma 1. Using the reparameterization trick and the fact that the summation of two Gaussian random variables is Gaussian, we can obtain $\mathbf{x}_k$ from $\mathbf{x}_0$:

$$\begin{aligned}
\mathbf{x}_k &= \sqrt{\alpha_k}\,\mathbf{x}_{k-1} + \sqrt{1-\alpha_k}\,\mathbf{e}_{k-1} \\
&= \sqrt{\alpha_k}\left(\sqrt{\alpha_{k-1}}\mathbf{x}_{k-2} + \sqrt{1-\alpha_{k-1}}\mathbf{e}_{k-2}\right) + \sqrt{1-\alpha_k}\mathbf{e}_{k-1} \\
&= \sqrt{\alpha_k\alpha_{k-1}}\,\mathbf{x}_{k-2} + \sqrt{1-\alpha_k\alpha_{k-1}}\,\widetilde{\mathbf{e}}_2 \\
&\;\vdots \\
&= \sqrt{\overline{\alpha}_k}\,\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_k}\,\widetilde{\mathbf{e}}_k,
\end{aligned}$$

where $\widetilde{\mathbf{e}}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $i = 2, \ldots, k$. Therefore, the conditional distribution $q_{k|0}(\mathbf{x}_k|\mathbf{x}_0)$ is

$$q_{k|0}(\mathbf{x}_k|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_k; \sqrt{\overline{\alpha}_k}\,\mathbf{x}_0, (1-\overline{\alpha}_k)\mathbf{I}),$$

Note that $\{\mathbf{e}_k\}_k$ are i.i.d. standard normal and independent of $\mathbf{x}_k$ while $\{\widetilde{\mathbf{e}}_k\}$ depend on each other.

Forward Process (cont'd)

Proof of Lemma 1 (cont'd).

- At $k = K$, we have

$$\mathbf{x}_K = \sqrt{\overline{\alpha}_K}\,\mathbf{x}_0 + \sqrt{1 - \overline{\alpha}_K}\,\mathbf{e},$$

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

- If $0 < \beta_1 < ... < \beta_K < 1$ or $0 < \gamma \leq \beta_1, \ldots, \beta_K < 1$, $\lim_{K \to \infty} \overline{\alpha}_K = 0$. Therefore, $q(\mathbf{x}_K) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathbf{I})$ (converge in distribution), i.e., as the number of timesteps becomes very large, the distribution $q(\mathbf{x}_K)$ will approach the Gaussian distribution with mean $\mathbf{0}$ and covariance $\mathbf{I}$.

Forward Process (cont'd)

Lemma 2: Let $\mathbf{x}_1, \cdots, \mathbf{x}_K$ be the vectors obtained from $\mathbf{x}_0$ by applying the forward process given in Definition 1. Then,

$$q(\mathbf{x}_{1:K} \mid \mathbf{x}_0) = \prod_{k=1}^{K} q(\mathbf{x}_k \mid \mathbf{x}_{k-1}). \qquad \text{(Eq. 2.2)}$$

where $q(\mathbf{x}_{1:K}|\mathbf{x}_0) := q(\mathbf{x}_1, \ldots, \mathbf{x}_K \mid \mathbf{x}_0)$ is the conditional joint distribution of $(\mathbf{x}_1, \ldots, \mathbf{x}_K)$ given $\mathbf{x}_0$.

Forward Process (cont'd)

Proof of Lemma 2. Since the sequence $\{\mathbf{x}_k\}_k$ is a Markov chain, $\mathbf{x}_2$ is independent of $\mathbf{x}_0$ when $\mathbf{x}_1$ is given. Thus, $q(\mathbf{x}_2|\mathbf{x}_1) = q(\mathbf{x}_2|\mathbf{x}_1, \mathbf{x}_0)$.

For $K = 2$, on the right-hand side, we have

$$\begin{aligned} q(\mathbf{x}_1|\mathbf{x}_0)q(\mathbf{x}_2|\mathbf{x}_1) &= q(\mathbf{x}_1|\mathbf{x}_0)q(\mathbf{x}_2|\mathbf{x}_1, \mathbf{x}_0) \\ &= \frac{q(\mathbf{x}_1, \mathbf{x}_0)}{q(\mathbf{x}_0)} q(\mathbf{x}_2|\mathbf{x}_1, \mathbf{x}_0) \\ &= \frac{q(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2)}{q(\mathbf{x}_0)} \\ &= q(\mathbf{x}_1, \mathbf{x}_2|\mathbf{x}_0). \end{aligned}$$

Forward Process (cont'd)

Proof of Lemma 2 (cont'd). For $K = n + 1$, we have

$$\begin{aligned}\prod_{t=1}^{n+1} q(\mathbf{x}_t|\mathbf{x}_{t-1}) &= \prod_{t=1}^{n} q(\mathbf{x}_t|\mathbf{x}_{t-1}) q(\mathbf{x}_{n+1}|\mathbf{x}_n) \\ &= q(\mathbf{x}_1, ..., \mathbf{x}_n|\mathbf{x}_0)\, q(\mathbf{x}_{n+1}|\mathbf{x}_n, ..., \mathbf{x}_0) \\ &= \frac{q(\mathbf{x}_0, ..., \mathbf{x}_n)}{q(\mathbf{x}_0)}\, q(\mathbf{x}_{n+1}|\mathbf{x}_n, ..., \mathbf{x}_0) \\ &= \frac{q(\mathbf{x}_0, ..., \mathbf{x}_n, \mathbf{x}_{n+1})}{q(\mathbf{x}_0)} \\ &= q(\mathbf{x}_1, ...\mathbf{x}_{n+1}|\mathbf{x}_0),\end{aligned}$$

where the second equality is obtained by using the induction hypothesis and the fact that $\mathbf{x}_{n+1}$ is independent of $\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{n-1}$ when $\mathbf{x}_n$ is given. The remaining equalities are obtained by using Bayes' rule.

Outline

Forward Process

Reverse Process

Reverse Process

- The goal of the reverse process is to generate data from the input distribution by sampling from $q(\mathbf{x}_K) = \mathcal{N}(\mathbf{x}_K; 0, \mathbf{I})$ and gradually denoising for which one needs to know the reverse distribution $q(\mathbf{x}_{k-1}|\mathbf{x}_k)$.
- In general, computation of $q(\mathbf{x}_{k-1}|\mathbf{x}_k)$ is intractable without the knowledge of $\mathbf{x}_0$.
- However, we can compute $q(\mathbf{x}_{k-1} \mid \mathbf{x}_k, \mathbf{x}_0)$.

Reverse Process (cont'd)

Lemma 3: With the assumptions of the forward process, the reverse Markov chain conditioned on $\mathbf{x}_0$, $q(\mathbf{x}_{k-1} \mid \mathbf{x}_k, \mathbf{x}_0)$ (for $k \geq 2$), follows a Gaussian distribution:

$$q(\mathbf{x}_{k-1} \mid \mathbf{x}_k, \mathbf{x}_0) = \frac{q(\mathbf{x}_k|\mathbf{x}_{k-1})q(\mathbf{x}_{k-1}|\mathbf{x}_0)}{q(\mathbf{x}_k|\mathbf{x}_0)} \qquad \text{(Eq. 6.1)}$$

$$= \mathcal{N}(\mathbf{x}_{k-1}; \widetilde{\mu}_k(\mathbf{x}_k, \mathbf{x}_0), \widetilde{\beta}_k \mathbf{I}), \qquad \text{(Eq. 6)}$$

where

$$\widetilde{\mu}_k(\mathbf{x}_k, \mathbf{x}_0) := \frac{\sqrt{\alpha_k}(1-\overline{\alpha}_{k-1})}{1-\overline{\alpha}_k}\mathbf{x}_k + \frac{\sqrt{\overline{\alpha}_{k-1}}\beta_k}{1-\overline{\alpha}_k}\mathbf{x}_0 \quad \text{and} \quad \widetilde{\beta}_k = \frac{1-\overline{\alpha}_{k-1}}{1-\overline{\alpha}_k}\beta_k. \qquad \text{(Eq. 7)}$$

The detailed proof is given in the next few slides.

Question: Can we explain intuitively why $q(\mathbf{x}_{k-1} \mid \mathbf{x}_k, \mathbf{x}_0)$ is Gaussian?

Reverse Process (cont'd)

Proof of Lemma 3. We can write the p.d.f. of the reverse Markov chain conditioned on $\mathbf{x}_0$ in terms of the p.d.fs of the forward process:

$$
\begin{aligned}
q(\mathbf{x}_{k-1}|\mathbf{x}_k, \mathbf{x}_0) &= \frac{q_{X_{k-1},X_k,X_0}(\mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{x}_0)}{q_{X_k,X_0}(\mathbf{x}_k, \mathbf{x}_0)} \text{ (Conditional p.d.f)} \\
&= \frac{q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{x}_0) q_{X_{k-1},X_0}(\mathbf{x}_{k-1}, \mathbf{x}_0)}{q(\mathbf{x}_k|\mathbf{x}_0) q_{X_0}(\mathbf{x}_0)} \text{ (Conditional p.d.f)} \\
&= \frac{q(\mathbf{x}_k|\mathbf{x}_{k-1}) q(\mathbf{x}_{k-1}|\mathbf{x}_0) q_{X_0}(\mathbf{x}_0)}{q(\mathbf{x}_k|\mathbf{x}_0) q_{X_0}(\mathbf{x}_0)} \text{(Markov property and Conditional p.d.f.)} \\
&= \frac{q(\mathbf{x}_k|\mathbf{x}_{k-1}) q(\mathbf{x}_{k-1}|\mathbf{x}_0)}{q(\mathbf{x}_k|\mathbf{x}_0)}
\end{aligned}
$$

(Eq. 7.1.)

(cont'd). Substituting (Eq. 2.1) and (Eq. 4.1) to (Eq. 7.1.) yields

$$q(\mathbf{x}_{k-1}|\mathbf{x}_k,\mathbf{x}_0) = \frac{1}{\sqrt{(2\pi\beta_k)^d}} \exp\left(-\frac{1}{2}\frac{(\mathbf{x}_k-\sqrt{\alpha_k}\mathbf{x}_{k-1})^T(\mathbf{x}_k-\sqrt{\alpha_k}\mathbf{x}_{k-1})}{\beta_k}\right)\cdot$$

$$\frac{1}{\sqrt{(2\pi(1-\overline{\alpha}_{k-1}))^d}} \exp\left(-\frac{1}{2}\frac{(\mathbf{x}_{k-1}-\sqrt{\overline{\alpha}_{k-1}}\mathbf{x}_0)^T(\mathbf{x}_{k-1}-\sqrt{\overline{\alpha}_{k-1}}\mathbf{x}_0)}{1-\overline{\alpha}_{k-1}}\right)\cdot$$

$$\left(\sqrt{(2\pi(1-\overline{\alpha}_k))^d}\right) \exp\left(\frac{1}{2}\frac{(\mathbf{x}_k-\sqrt{\overline{\alpha}_k}\mathbf{x}_0)^T(\mathbf{x}_k-\sqrt{\overline{\alpha}_k}\mathbf{x}_0)}{1-\overline{\alpha}_k}\right)$$

$$= \frac{\sqrt{(1-\overline{\alpha}_k)^d}}{\sqrt{(2\pi\beta_k(1-\overline{\alpha}_{k-1}))^d}} \exp\Bigg\{-\frac{1}{2}\frac{1-\overline{\alpha}_k}{\beta_k(1-\overline{\alpha}_{k-1})}\mathbf{x}_{k-1}^T\mathbf{x}_{k-1}+$$

$$\left(\frac{\sqrt{\alpha_k}}{\beta_k}\mathbf{x}_k^T+\frac{\sqrt{\overline{\alpha}_{k-1}}}{1-\overline{\alpha}_{k-1}}\mathbf{x}_0^T\right)\mathbf{x}_{k-1}+\text{terms}(\mathbf{x}_k,\mathbf{x}_0)\Bigg\}$$

Simplifying the calculations, we have

$$q(\mathbf{x}_{k-1}|\mathbf{x}_k,\mathbf{x}_0) = \frac{\sqrt{(1-\overline{\alpha}_k)^d}}{\sqrt{(2\pi\beta_k(1-\overline{\alpha}_{k-1}))^d}} \exp\left(-\frac{1}{2}\frac{(\mathbf{x}_{k-1}-\widetilde{\mu}_k)^T(\mathbf{x}_{k-1}-\widetilde{\mu}_k)}{\widetilde{\beta}_k}\right), \tag{Eq. 7.2}$$

where $\widetilde{\mu}_k$ and $\widetilde{\beta}_k$ are given in (Eq. 7).

Reverse Process (cont'd)

- Our goal is to learn the reverse distribution from the obtained conditional reverse distribution.

- Let $p_\theta(\mathbf{x}_{k-1}|\mathbf{x}_k)$ be the learned reverse distribution. From Markovian theory, we know that $p_\theta(\mathbf{x}_{k-1}|\mathbf{x}_k)$ is also Gaussian (prove this!). The proof is based on two facts:

    - The reverse chain of a Markov chain is also a Markov chain.

    - Under the settings of the forward chain $\{X_k\}_{k=0}^K$ in DDPM, the reverse chain $\{\overline{X}_k := X_{K-k}\}_{k=0}^K$ is also a Markov chain. Moreover, the transition probability density of the reverse chain

    $$\overline{q}_{k,k-1}(\mathbf{y}_k \mid \mathbf{y}_{k-1}) = \frac{\pi_G(\mathbf{y}_{k-1}; 0, \mathbf{I})\pi_G(\mathbf{y}_{k-1}; \sqrt{1-\beta_{K-k+1}}\mathbf{y}_k, \beta_{K-k+1}\mathbf{I})}{\pi_G(\mathbf{y}_k; 0, \mathbf{I})}$$

    is also Gaussian. Here we denote $\pi_G(\mathbf{y}_{k-1}; 0, \mathbf{I})$ the p.d.f of the Gaussian distribution $\mathcal{N}(\mathbf{y}_{k-1}; 0, \mathbf{I})$.

Reverse Process (cont'd)

Definition 2: Under the settings of the forward process, the reverse process $p_\theta(\mathbf{x}_{0:K})$ is defined as a Markov chain with learned Gaussian transitions starting at

$$p(\mathbf{x}_K) = \mathcal{N}(\mathbf{x}_K; \mathbf{0}, \mathbf{I})$$

and

$$p_\theta(\mathbf{x}_{0:K}) := p(\mathbf{x}_K) \prod_{k=1}^{K} p_\theta(\mathbf{x}_{k-1}|\mathbf{x}_k), \qquad \text{(Eq. 1)}$$

where

$$p_\theta(\mathbf{x}_{k-1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k-1}; \mu_\theta(\mathbf{x}_k, k), \Sigma_\theta(\mathbf{x}_k, k)). \qquad \text{(Eq. 1')}$$

The probability the generative model assigns to the data is:

$$p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:K}) d\mathbf{x}_{1:K}, \qquad \text{(Eq. 1.1)}$$

where we denote $d\mathbf{x}_1 d\mathbf{x}_2 \ldots d\mathbf{x}_K$ as $d\mathbf{x}_{1:K}$.

[2]

[2] Deep unsupervised learning using nonequilibrium thermodynamics. PMLR 2015, https://proceedings.mlr.press/v37/sohl-dickstein15.html

Reverse Process (cont'd)

Note that the integral for $p_\theta(\mathbf{x}_0)$ is intractable. Nevertheless, we can evaluate $p_\theta(\mathbf{x}_0)$ via the relative probability of the forward and reverse trajectories as follows:

$$
\begin{aligned}
p_\theta(\mathbf{x}_0) &= \int d\mathbf{x}_{1:K}\; p_\theta(\mathbf{x}_{0:K}) \frac{q(\mathbf{x}_{1:K} \mid \mathbf{x}_0)}{q(\mathbf{x}_{1:K} \mid \mathbf{x}_0)} \\
&= \int d\mathbf{x}_{1:K}\; q(\mathbf{x}_{1:K} \mid \mathbf{x}_0) \frac{p_\theta(\mathbf{x}_{0:K})}{q(\mathbf{x}_{1:K} \mid \mathbf{x}_0)} \\
&\overset{\text{(Eq. 1)\&(Eq. 2.2)}}{=} \int d\mathbf{x}_{1:K}\; q(\mathbf{x}_{1:K} \mid \mathbf{x}_0)\, \frac{p(\mathbf{x}_K) \prod_{k=1}^{K} p_\theta(\mathbf{x}_{k-1} \mid \mathbf{x}_k)}{\prod_{k=1}^{K} q(\mathbf{x}_k \mid \mathbf{x}_{k-1})} \\
&\overset{\text{(Eq. 6.1)}}{=} \int d\mathbf{x}_{1:K}\; q(\mathbf{x}_{1:K} \mid \mathbf{x}_0)\, p(\mathbf{x}_K) \frac{p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)}{q(\mathbf{x}_1 \mid \mathbf{x}_0)} \prod_{k=2}^{K} \frac{p_\theta(\mathbf{x}_{k-1} \mid \mathbf{x}_k) q(\mathbf{x}_{k-1}|\mathbf{x}_0)}{q(\mathbf{x}_{k-1} \mid \mathbf{x}_k, \mathbf{x}_0) q(\mathbf{x}_k|\mathbf{x}_0)} \\
&= \int d\mathbf{x}_{1:K}\; q(\mathbf{x}_{1:K} \mid \mathbf{x}_0)\, \frac{p(\mathbf{x}_K) p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)}{q(\mathbf{x}_K|\mathbf{x}_0)} \prod_{k=2}^{K} \frac{p_\theta(\mathbf{x}_{k-1} \mid \mathbf{x}_k)}{q(\mathbf{x}_{k-1} \mid \mathbf{x}_k, \mathbf{x}_0)} \quad \text{(Eq. 1.2)}
\end{aligned}
$$

DDPM - Recap

- **Forward Process:** Let $\mathbf{x}_0 \in \mathbb{R}^d$ and a variance schedule $\beta_i \in (0, 1)$, for $i = 1, \dots K$. Construct:

$$\mathbf{x}_k = \sqrt{1 - \beta_k}\mathbf{x}_{k-1} + \sqrt{\beta_k}\,\mathbf{e}, \quad k = 1, \dots, K,$$

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

- **Reverse Process:** Generally intractable and learned using a parameterized model,

$$p_\theta(\mathbf{x}_0) = \int d\mathbf{x}_{1:K}\;\; q(\mathbf{x}_{1:K} \mid \mathbf{x}_0)\, p(\mathbf{x}_K) \prod_{k=1}^{K} \frac{p_\theta(\mathbf{x}_{k-1} \mid \mathbf{x}_k)}{q(\mathbf{x}_k \mid \mathbf{x}_{k-1})}.$$

  Here

$$p_\theta(\mathbf{x}_{k-1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k-1}; \mu_\theta(\mathbf{x}_k, k), \Sigma_\theta(\mathbf{x}_k, k)),$$

  where μ_θ and Σ_θ are the learnt mean vector and covariance matrix, respectively.

- **Goal:** Compare $q(\mathbf{x}_0)$ and $p(\mathbf{x}_0) = p_\theta(\mathbf{x}_0)$.

Comparison of Two Distributions

We recall some useful notions to compare two distributions.

Definitions:

3. The cross-entropy of a distribution p relative to another distribution q over a given set is

$$H(q, p) = \mathbb{E}_q[-\log p],$$

where $\mathbb{E}_q[\cdot]$ denotes the expectation with respect to the distribution q.

4. Let p and q be two probability distributions. Then the KL divergence denoted by $D_{KL}(q||p)$ is defined as

$$D_{KL}(q||p) = \mathbb{E}_q\left[\log\left(\frac{q}{p}\right)\right].$$

Roughly speaking, KL divergence $D_{KL}(q||p)$ is a measure of the information lost when q is approximated by p.

Comparison of Two Distributions

Remark: Note that for two probability distributions p and q, we have

$$H(q, p) = H(q, q) + D_{KL}(q||p).$$

So if q is the true distribution and p is an approximated one, then $H(q, q)$ is a constant (not learned) and the cross entropy $H(q, p)$ differs from the KL divergence $D_{KL}(q||p)$ by a constant.

KL Divergence of Two Gaussians

Lemma 4: Let $p \sim \mathcal{N}(\mu_p, \Sigma_p)$ and $q \sim \mathcal{N}(\mu_q, \Sigma_q)$ be two Gaussian distributions on $\mathbb{R}^d$. Then

$$D_{KL}(q \| p) = \frac{1}{2}\left[\log \frac{|\Sigma_p|}{|\Sigma_q|} - d + (\mu_q - \mu_p)^T \Sigma_p^{-1} (\mu_q - \mu_p) + \mathrm{tr}(\Sigma_p^{-1}\Sigma_q)\right].$$

KL Divergence of Two Gaussians (cont'd)

Proof of Lemma 4. Re call that

$$p(\mathbf{x}) = \frac{1}{|\Sigma_p|^{1/2}(2\pi)^{d/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_p)^T\mathbf{\Sigma}_p^{-1}(\mathbf{x}-\mu_p)\right)$$

We have

$$2D_{KL}(q\|p) = 2\mathbb{E}_q\left[\log\left(\frac{q}{p}\right)\right]$$

$$= \log\frac{|\Sigma_p|}{|\Sigma_q|} + \mathbb{E}_q\left(-(\mathbf{x}-\mu_q)^T\mathbf{\Sigma}_q^{-1}(\mathbf{x}-\mu_q)\right) + \mathbb{E}_q\left((\mathbf{x}-\mu_p)^T\mathbf{\Sigma}_p^{-1}(\mathbf{x}-\mu_p)\right)$$

To simplify the second and the third terms, we use the following equality:

> **Lemma 5:** Let X be a random vector in $\mathbb{R}^d$ with mean μ and covariance matrix Σ. Let $A \in \mathbb{R}^{d\times d}$ be a symmetric matrix. Then
>
> $$\mathbb{E}(X^TAX) = \text{tr}(A\Sigma) + \mu^TA\mu.$$

Proof. We have

$$\mathbb{E}(X^TAX) = \mathbb{E}\,\text{tr}\left(X^TAX\right) = \mathbb{E}\,\text{tr}\left(AXX^T\right) = \text{tr}\left(A\mathbb{E}(XX^T)\right)$$

$$= \text{tr}\left(A\left(\texttt{Cov}(X,X) + \mathbb{E}X\,\mathbb{E}X^T\right)\right) = \text{tr}(A\Sigma) + \text{tr}(A\mathbb{E}X\,\mathbb{E}X^T)$$

$$= \text{tr}(A\Sigma) + \text{tr}(A\mu\mu^T) = \text{tr}(A\Sigma) + \text{tr}(\mu^TA\mu) = \text{tr}(A\Sigma) + \mu^TA\mu.$$

KL Divergence of Two Gaussians (cont'd)

Proof of Lemma 4 (cont'd). The second term can be simplified as

$$\mathbb{E}_q(\mathbf{x} - \mu_q)^T \Sigma_q^{-1} (\mathbf{x} - \mu_q) = \operatorname{tr}\left(\Sigma_q^{-1}\Sigma_q\right) + 0^T \Sigma_q^{-1} 0 = \operatorname{tr} I_d = d$$

Similarly, the third term can be simplified as

$$\mathbb{E}_q\left((\mathbf{x} - \mu_p)^T \Sigma_p^{-1} (\mathbf{x} - \mu_p)\right) = \operatorname{tr}(\Sigma_p^{-1}\Sigma_q) + (\mu_q - \mu_p)^T \Sigma_p^{-1} (\mu_q - \mu_p)$$

Cross Entropy Loss Function in Diffusion Models

Theorem. Let $\mathbf{x}_0$ be data drawn from an unknown distribution $q(\mathbf{x}_0)$. Suppose $\mathbf{x}_1, \cdots, \mathbf{x}_K$ be the degraded data obtained by applying the forward process given in Definition 1 and p denotes the (reverse) distribution such that $p(\mathbf{x}_0)$ approximates $q(\mathbf{x}_0)$. Then the cross entropy loss $H(q, p)$ satisfies the following inequality:

$$\begin{aligned} H(q(\mathbf{x}_0), p(\mathbf{x}_0)) \leq & \mathbb{E}_{q(\mathbf{x}_{0:K})}\left[\log \frac{q(\mathbf{x}_K|\mathbf{x}_0)}{p(\mathbf{x}_K)} + \sum_{k=2}^{K} \log \frac{q(\mathbf{x}_{k-1}|\mathbf{x}_k, \mathbf{x}_0)}{p(\mathbf{x}_{k-1}|\mathbf{x}_k)} - \log p(\mathbf{x}_0|\mathbf{x}_1)\right] \\ \leq & D_{KL}(q(\mathbf{x}_K|\mathbf{x}_0) \| p(\mathbf{x}_K)) + \sum_{k=2}^{K} D_{KL}(q(\mathbf{x}_{k-1}|\mathbf{x}_k, \mathbf{x}_0) \| p(\mathbf{x}_{k-1}|\mathbf{x}_k)) \\ & + \mathbb{E}_{q(x_{0:K})}(-\log p(\mathbf{x}_0|\mathbf{x}_1)). \end{aligned} \qquad \text{(Eq. 5)}$$

Cross Entropy Loss Function in Diffusion Models (cont'd)

Proof. The proof is original from [Sohl-Dickstein et al., 15] and recalled in [Ho et al., 20]. We have

$$
\begin{aligned}
H(q(\mathbf{x}_0), p(\mathbf{x}_0)) &\overset{\text{by def.}}{=} -\mathbb{E}_{q(\mathbf{x}_0)}[\log p(\mathbf{x}_0)] \\
&\overset{\text{(Eq. 1.2)}}{=} -\int d\mathbf{x}_0\, q(\mathbf{x}_0) \log \left(\int d\mathbf{x}_{1:K}\;\; q(\mathbf{x}_{1:K} \mid \mathbf{x}_0)\, \frac{p(\mathbf{x}_K)p(\mathbf{x}_0 \mid \mathbf{x}_1)}{q(\mathbf{x}_K|\mathbf{x}_0)} \prod_{k=2}^{K} \frac{p(\mathbf{x}_{k-1} \mid \mathbf{x}_k)}{q(\mathbf{x}_{k-1} \mid \mathbf{x}_k, \mathbf{x}_0)} \right) \\
&\overset{\text{Jensen's ineq.}}{\leq} -\int dx_{0:K}\, q(\mathbf{x}_{0:K}) \log \left(\frac{p(\mathbf{x}_K)p(\mathbf{x}_0 \mid \mathbf{x}_1)}{q(\mathbf{x}_K|\mathbf{x}_0)} \prod_{k=2}^{K} \frac{p(\mathbf{x}_{k-1} \mid \mathbf{x}_k)}{q(\mathbf{x}_{k-1} \mid \mathbf{x}_k, \mathbf{x}_0)} \right) \\
&\leq \int dx_{0:K}\, q(\mathbf{x}_{0:K}) \left[\log \frac{q(\mathbf{x}_K|\mathbf{x}_0)}{p(\mathbf{x}_K)} + \sum_{k=2}^{K} \log \frac{q(\mathbf{x}_{k-1}|\mathbf{x}_k, \mathbf{x}_0)}{p(\mathbf{x}_{k-1}|\mathbf{x}_k)} - \log p(\mathbf{x}_0|\mathbf{x}_1) \right] \\
&\leq D_{KL}(q(\mathbf{x}_K|\mathbf{x}_0) \| p(\mathbf{x}_K)) + \sum_{k=2}^{K} D_{KL}(q(\mathbf{x}_{k-1}|\mathbf{x}_k, \mathbf{x}_0) \| p(\mathbf{x}_{k-1}|\mathbf{x}_k)) + \\
&\qquad + \mathbb{E}_{q(x_{0:K})}(-\log p(\mathbf{x}_0|\mathbf{x}_1)).
\end{aligned}
$$

Cross Entropy Loss Function in Diffusion Models (cont'd)

- From the settings of the diffusion model, the first term on the upper bound

$$D_{KL}(q(\mathbf{x}_K|\mathbf{x}_0)\|p(\mathbf{x}_K))$$

  is constant and hence often ignored when training a diffusion model.

- For the third term on the upper bound,

$$\mathbb{E}_{q(\mathbf{x}_{0:K})}\left[-\log p(\mathbf{x}_0|\mathbf{x}_1)\right],$$

  there are numerous ways to handle this term in practice. For example, the authors in [Ho et al, 20] choose to model this term using a separate discrete decoder.

- For the second term, we first simplify to difference in means, then rewrite in terms of the difference between noises, where the noises are defined based on $\mathbf{x}_k$.

Cross Entropy Loss Function in Diffusion Models (cont'd)

For each $k = 2, \ldots, K$, since $q(\mathbf{x}_{k-1}|\mathbf{x}_k, \mathbf{x}_0)$ and $p(\mathbf{x}_{k-1}|\mathbf{x}_k)$ are Gaussian with the same variance (see the assumptions), using Lemma 4, we have:

$$D_{KL}(q(\mathbf{x}_{k-1}|\mathbf{x}_k, \mathbf{x}_0)\|p_\theta(\mathbf{x}_{k-1}|\mathbf{x}_k)) \overset{\text{Lem. 4}}{=} \mathbb{E}_{q(\mathbf{x}_0,\mathbf{x}_k)} \frac{1}{2\sigma_k^2} \|\tilde{\mu}_k(\mathbf{x}_k, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_k, k)\|_2^2 + C \tag{Eq. 8}$$

$$\overset{\text{(Eq. 7)}}{=} \mathbb{E}_{q(\mathbf{x}_0,\mathbf{x}_k)} \frac{1}{2\sigma_k^2} \left\| \frac{\sqrt{\alpha_k}(1-\overline{\alpha}_{k-1})}{1-\overline{\alpha}_k}\mathbf{x}_k + \frac{\sqrt{\overline{\alpha}_{k-1}}\beta_k}{1-\overline{\alpha}_k}\mathbf{x}_0 - \mu_\theta(\mathbf{x}_k, k) \right\|_2^2 + C \tag{Eq. 8.1}$$

$$\overset{\text{(Eq. 4.1)}}{=} \mathbb{E}_{q(\mathbf{x}_0,\mathbf{x}_k)} \frac{1}{2\sigma_k^2} \left\| \frac{\sqrt{\alpha_k}(1-\overline{\alpha}_{k-1})}{1-\overline{\alpha}_k}\mathbf{x}_k + \frac{\sqrt{\overline{\alpha}_{k-1}}\beta_k}{1-\overline{\alpha}_k} \frac{1}{\sqrt{\overline{\alpha}_k}}(\mathbf{x}_k - \sqrt{1-\overline{\alpha}_k}\widetilde{\varepsilon}_k) - \mu_\theta(\mathbf{x}_k, k) \right\|_2^2 + C \tag{Eq. 8.2}$$

$$= \mathbb{E}_{\mathbf{x}_0,\widetilde{\varepsilon}_k} \frac{1}{2\sigma_k^2} \left\| \frac{1}{\sqrt{\alpha_k}}\mathbf{x}_k(\mathbf{x}_0, \widetilde{\varepsilon}_k) - \frac{\beta_k}{\sqrt{\alpha_k}\sqrt{1-\overline{\alpha}_k}}\widetilde{\varepsilon}_k - \mu_\theta(\mathbf{x}_k(\mathbf{x}_0, \widetilde{\varepsilon}_k), k) \right\|_2^2 + C \tag{Eq. 10}$$

The term C is constant and does not depend on θ.

Cross Entropy Loss Function in Diffusion Models (cont'd)

- Since $\mathbf{x}_k$ is available as input to the model, we may choose the parametrization

$$\boldsymbol{\mu}_\theta(\mathbf{x}_k, k) = \frac{1}{\sqrt{\alpha_k}} \left(\mathbf{x}_k - \frac{\beta_k}{\sqrt{1-\overline{\alpha}_k}} \mathbf{e}_\theta(\mathbf{x}_k, k) \right). \qquad \text{(Eq. 11)}$$

- We can simplify (Eq. 10) as:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}_0, \widetilde{\mathbf{e}}_k} \left[\frac{\beta_k^2}{2\sigma_k^2 \alpha_k (1-\overline{\alpha}_k)} \|\widetilde{\varepsilon}_k - \mathbf{e}_\theta(\mathbf{x}_k, k)\|^2 \right] \\
&\quad = \frac{\beta_k^2}{2\sigma_k^2 \alpha_k (1-\overline{\alpha}_k)} \iint \|\mathbf{e} - \epsilon_\theta(\mathbf{x}_k(\mathbf{x}_0, \mathbf{e}), k)\|^2 q_{X_0}(\mathbf{x}_0) q_\varepsilon(\mathbf{e}) \, d\mathbf{e} \, d\mathbf{x}_0 \\
&\quad = \mathbb{E}_{\mathbf{x}_0, \varepsilon} \left[\frac{\beta_k^2}{2\sigma_k^2 \alpha_k (1-\overline{\alpha}_k)} \|\varepsilon - \mathbf{e}_\theta(\sqrt{\overline{\alpha}_k}\, \mathbf{x}_0 + \sqrt{1-\overline{\alpha}_k}\, \varepsilon, k)\|^2 \right].
\end{aligned} \qquad \text{(Eq. 12)}$$

where $\mathbf{e}_\theta$ now denotes a function approximator intended to predict the noise from $\mathbf{x}_k$.

DDPM Training

Algorithm 1 Training

1: **repeat**
2: $\quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $\quad t \sim \mathrm{Uniform}(\{1, \dots, T\})$
4: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: $\quad$ Take gradient descent step on

$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$

6: **until** converged

DDPM Sampling

Algorithm 2 Sampling

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

Scored-Based Generative Models

To be continued...