The Maths Behind Denoising Diffusion Probabilistic Models

Giovani Tavares
giovanitavares@outlook.com

University of Sao Paulo — January 5, 2025

Definition 0.1 (Variational Lower Bound - Jensen's Inequality). *Let's use the Rule Of Total Probability to find a lower bound for the log-likelihood function.*

$$log[p(\mathbf{x})] = log \int_{\mathbf{Z}} p(\mathbf{x_0}, \mathbf{Z}) d\mathbf{Z} \tag{1}$$

$$= log \int_{\mathbf{Z}} p(\mathbf{x_0}, \mathbf{Z}) \frac{q(\mathbf{Z})}{q(\mathbf{Z})} d\mathbf{Z} \tag{2}$$

$$= log(\mathbb{E}_q \left[\frac{p(\mathbf{x}, \mathbf{Z})}{q(\mathbf{Z})} \right]) \tag{3}$$

$$\textit{the Jensen's inequality tells us} \tag{4}$$

$$f(\mathbf{E}(\mathbf{X})) \geq \mathbf{E}(f(\mathbf{X}))) \tag{5}$$

$$\textit{for any concave function f.} \tag{6}$$

$$\textit{log is concave, hence:} \tag{7}$$

$$log[p(\mathbf{x})] \geq \mathbb{E}_q \left[log \frac{p(\mathbf{x}, \mathbf{Z})}{q(\mathbf{Z})} \right] \tag{8}$$

$$\textit{The Evidence Lower Bound, or simply ELBO, is :} \tag{9}$$

$$L := \mathbb{E}_q \left[log \frac{p(\mathbf{x}, \mathbf{Z})}{q(\mathbf{Z})} \right] \tag{10}$$

$$\tag{11}$$

Definition 0.2 (Variational Lower Bound - KL Divergence). *Let's use the Kullback-Leibler Divergence function to find a lower bound for the log-likelihood function.*

$$\mathbf{D_{KL}} \left[q(\mathbf{Z}))||p(\mathbf{Z}|\mathbf{x}) \right] := \mathbb{E}_q \left[log(q(\mathbf{Z}) - log(p(\mathbf{Z}|\mathbf{x})) \right] \tag{12}$$

$$\textit{using the Bayes' Rule we can write} \tag{13}$$

$$p(\mathbf{Z}|\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{Z})}{p(\mathbf{x})} \tag{14}$$

$$\implies \mathbf{D_{KL}} \left[q(\mathbf{Z}))||p(\mathbf{Z}|\mathbf{x}) \right] = \mathbb{E}_q \left[log(q(\mathbf{Z})) - log(p(\mathbf{x}, \mathbf{Z})) + log(p(\mathbf{x})) \right] \tag{15}$$

$$\textit{the prior of the observed variables X does not depend on q} \tag{16}$$

$$\mathbf{D_{KL}} \left[q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{x}) \right] = \mathbb{E}_q \left[log(q(\mathbf{Z})) - log(p(\mathbf{x}, \mathbf{Z})) \right] + log(p(\mathbf{x})) \tag{17}$$

$$\implies log(p(\mathbf{x})) = \mathbf{D_{KL}} \left[q(\mathbf{Z}))||p(\mathbf{Z}|\mathbf{x}) \right] - \mathbb{E}_q \left[log(q(\mathbf{Z})) - log(p(\mathbf{x}, \mathbf{Z})) \right] \tag{18}$$

$$= \mathbf{D_{KL}} \left[q(\mathbf{Z}))||p(\mathbf{Z}|\mathbf{x}) \right] + \mathbb{E}_q \left[log(p(\mathbf{x}, \mathbf{Z}) - log(q(\mathbf{Z}))) \right] \tag{19}$$

$$\textit{but } \mathbf{D_{KL}} \left[q(\mathbf{Z}))||p(\mathbf{Z}|\mathbf{x}) \right] \geq 0 \tag{20}$$

$$\implies log(p(\mathbf{x})) \geq \mathbb{E}_q \left[log \frac{p(\mathbf{x}, \mathbf{Z})}{q(\mathbf{Z})} \right] \tag{21}$$

$$\textit{The Evidence Lower Bound, or simply ELBO, is :} \tag{22}$$

$$L := \mathbb{E}_q \left[log \frac{p(\mathbf{x}, \mathbf{Z})}{q(\mathbf{Z})} \right] \tag{23}$$

We have found lower bound L for the log-likelihood function that is tractable if rewritten properly. Our goal is to have a process to sample from p such that $logp(\mathbf{x_0})$ is as large as possible. This means that our process will output very likely outputs $\mathbf{x_0}$. In order to do so, DDPMs work in two steps: **noise prediction** and **sampling**. The former is the one responsible for predicting the a noise ϵ_θ of an input $\mathbf{x_t}$ which has has been sampled from the forward process' distribution q. The latter uses such prediction to sample $\mathbf{x_0}$ from p.

0.0.1 Noise Predictor Training Derivation

Having ELBO (L) as a lower bound for the log likelihood function means we can train our noise predictor model to maximize L, i.e., L is a candidate for our model's loss function. In order to do so, further algebraic manipulation must be performed with it in order to make it tractable in a way that the noise that has been added to the input appears somewhere. We demonstrate such manipulations here and will begin by showing that maximing L basically means approximating the distributions $p(\mathbf{x_{t-1}}|\mathbf{x_t})$ and $q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0})$.

If we pay attention to equation 41's terms above, we see that the first term is parameter free, because $p((x_T))$ is fixed and defined as a Gaussian, while $q((x_T|x_0))$ is also Gaussian from the definition of the forward process. Hence, we are left with the second and third terms.

As previously mentioned, we are interested in maximizing L. Using the equation 41, we see that doing so is equivalent to minimizing the KL Divergence between $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$ and $q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0})$. We know that both distributions are Gaussians, which makes computing the KL Divergence between them easier if we know their mean and variance. We will begin by calculating such moments for $q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0})$.

$$q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0}) = \frac{q(\mathbf{x_t}|\mathbf{x_{t-1}}, \mathbf{x_0})q(\mathbf{x_{t-1}}|\mathbf{x_0})}{q(\mathbf{x_t}|\mathbf{x_0})} \tag{24}$$

$$\text{we know the q distribution from Definition \textbf{??}, hence} \tag{25}$$

$$q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0}) \text{ is a product of known Gaussians over another known Gaussian that lets us define} \tag{26}$$

$$\mu_q(\mathbf{x_t}, \mathbf{x_0}) = \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x_t} + (1-\alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x_0}}{(1-\bar{\alpha}_t)} \tag{27}$$

$$\Sigma_q(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I} \tag{28}$$

$$\implies q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0}) = \mathcal{N}(\mathbf{x_{t-1}}; \mu_q(\mathbf{x_t}, \mathbf{x_0}); \Sigma_q(t)) \tag{29}$$

We have just defined the $q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0})$ distribution. Now let's move on to the $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$ distribution.

$$p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}) = \mathcal{N}(\mathbf{x_{t-1}}; \mu_\theta(\mathbf{x_t}); \Sigma_\theta(t)) \tag{30}$$

$$\text{the reverse process variance is defined as the ground truth variance of the forward process:} \tag{31}$$

$$\Sigma_\theta(t) = \Sigma_q(t) \tag{32}$$

$$\text{we are only left with the distribution's mean } \mu_\theta \tag{33}$$

$$p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}) = \mathcal{N}(\mathbf{x_{t-1}}; \mu_\theta(\mathbf{x_t}); \Sigma_q(t)) \tag{34}$$

Equation 50 makes it much easier to calculate $\mathbf{D_{KL}}\left[p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})||q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0})\right]$: now we know we are trying to compute the KL Divergence between two Gaussians with the exact same variance. For that, there is the following result that arives from the definition of such divergence:

$$d_1(x) = \mathcal{N}(\mu_1, \sigma^2) \tag{35}$$

$$d_2(x) = \mathcal{N}(\mu_2, \sigma^2) \tag{36}$$

$$\text{The KL divergence } D_{KL}(d_1|d_2) \text{ is given by:} \tag{37}$$

$$D_{KL}(d_1|d_2) = \frac{(\mu_1 - \mu_2)^2}{2\sigma^2} \tag{38}$$

$$\text{Hence,} \tag{39}$$

$$\mathbf{D_{KL}}\left[p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})||q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0})\right] = \mathbf{D_{KL}}\left(\mathcal{N}(\mathbf{x_{t-1}}; \mu_\theta(\mathbf{x_t}); \Sigma_q(t)), \mathcal{N}(\mathbf{x_{t-1}}; \mu_q(\mathbf{x_t}, \mathbf{x_0}); \Sigma_q(t))\right) \tag{40}$$

$$= \frac{1 - \bar{\alpha}_t}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \left|\left|(\mu_\theta - \mu_q)_2^2\right|\right| \tag{41}$$

As our goal is to minimize the KL Divergence, from equation 59 we see that such goal comes down to basically minimizing the difference $\mu_\theta - \mu_q$, i.e., **we just need to minimize the difference between the means of the reverse and forward processes' distributions**. We need to define the reverse process' distribution mean prediction by taking a look at the forward process' one.

$$\mu_q(\mathbf{x_t}, \mathbf{x_0}) = \frac{(1 - \bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x_t} + (1 - \alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x_0}}{(1 - \bar{\alpha}_t)} \tag{42}$$

$$\text{we can define the prediction} \tag{43}$$

$$\hat{\mu}_q(\mathbf{x_t}, \mathbf{x_0}) = \mu_\theta(\mathbf{x_t}) \tag{44}$$

$$= \frac{(1 - \bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x_t} + (1 - \alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_\theta}{(1 - \bar{\alpha}_t)} \tag{45}$$

In equation 63 we see that we are using our reverse process model's prediction $\mathbf{x}_\theta$ in the prediction of its distribution's mean, which let's us rewrite $\left|\left|(\mu_\theta - \mu_q)_2^2\right|\right|$ in terms of $\mathbf{x_0}$ and $\mathbf{x}_\theta$ which leves us with the following for the KL Divergence:

$$\mathbf{D_{KL}}\left(\mathcal{N}(\mathbf{x_{t-1}}; \mu_\theta(\mathbf{x_t}); \Sigma_q(t)), \mathcal{N}(\mathbf{x_{t-1}}; \mu_q(\mathbf{x_t}, \mathbf{x_0}); \Sigma_q(t))\right) = \frac{(1 - \bar{\alpha}_t)(\bar{\alpha}_{t-1})}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \left|\left|(\mathbf{x}_\theta - \mathbf{x_0})_2^2\right|\right| \tag{46}$$

The author's of the DDPM paper mention that equation 64 can be used as the loss function to train the reverse process model. On the other hand, we now that the forward process actually predict the noise that was added to an input $\mathbf{x_t}$ intead of predicting $\mathbf{x}_\theta$ directly. This means that the loss function must account for the error prediction somehow. This is achieved by further analysing $\mathbf{x_0}$ and $\mathbf{x}_\theta$ and remembering how $\mathbf{x}_\theta$ was defined in the forward process.

$$q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} \times x_0; (1 - \bar{\alpha}_t)\mathbb{I}) \tag{47}$$

$$\text{which let's us write} \tag{48}$$

$$\mathbf{x_t} = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon \tag{49}$$

$$\implies \mathbf{x_0} = \frac{\mathbf{x_t} - \sqrt{1 - \bar{\alpha}_t}\epsilon}{\sqrt{\bar{\alpha}_t}} \tag{50}$$

$$\text{for a Standard Gaussian Noise } \epsilon. \tag{51}$$

$$\text{We can now define our prediction } \hat{\epsilon} = \epsilon_\theta \tag{52}$$

$$\mathbf{x}_\theta = \frac{\mathbf{x_t} - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta}{\sqrt{\bar{\alpha}_t}} \tag{53}$$

$$\implies \frac{(1 - \bar{\alpha}_t)(\bar{\alpha}_{t-1})}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \left|\left|(\mathbf{x}_\theta - \mathbf{x_0})_2^2\right|\right| = \frac{(1 - \bar{\alpha}_t)(\bar{\alpha}_{t-1})}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \frac{(1 - \alpha_t)^2}{(1 - \bar{\alpha}_t)\alpha_t} \left|\left|(\epsilon_\theta - \epsilon)_2^2\right|\right| \tag{54}$$

Even though equation 72 could be used directly as the loss function for the noise predictor, the DDPM paper authors mention that optimizing $\left|\left|(\epsilon_\theta - \epsilon)_2^2\right|\right|$ without the scaling factor with the cumulative noise α_t is enough. Hence, we have finally defined a function to be minimized for the noise predictor training and hence write its algorithm.

Algorithm 1: Noise Predictor Training

```
repeat
    $\mathbf{x_0} \sim \mathbf{q(x_0)}$    ▷ Sample image from training set
    $\mathbf{x_0} \sim \mathbf{Uniform(\{1, \ldots, T\})}$    ▷ Sample the step of the Forward Process Markov Chain
    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$    ▷ Sample standard gaussian noise to be added to the input
    $\mathbf{x_t} = \sqrt{\bar{\alpha}_\mathbf{t}}\mathbf{x_0} + \sqrt{\mathbf{1} - \bar{\alpha}_\mathbf{t}}\epsilon$    ▷ Forward Process/ Generating Noisy Image
    Take Gradient Descent Step on $\nabla_\theta(||\epsilon - \epsilon_\theta(\mathbf{x_t}, \mathbf{t})||)$
until converged
```

0.0.2 Sampling Algorithm Derivation

Now that we have defined a way to predict an input image $\mathbf{x_t}$'s noise, we need a way to use such prediction to reconstruct the original de-noised image $\mathbf{x_0}$, i.e., we need a way to sample from $p(\mathbf{x_0})$. To do so, let's recall how we have defined the $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$ distribution.

$$p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}) = \mathcal{N}(\mathbf{x_{t-1}}; \mu_\theta(\mathbf{x_t}); \Sigma_q(t)) \tag{55}$$

$$\mu_\theta(\mathbf{x_t}) = \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x_t} + (1-\alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_\theta}{(1-\bar{\alpha}_t)} \tag{56}$$

$$\mathbf{x}_\theta = \frac{\mathbf{x_t} - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta}{\sqrt{\bar{\alpha}_t}} \tag{57}$$

$$\implies \mu_\theta(\mathbf{x_t}) = \frac{\mathbf{x_t}}{\sqrt{\alpha_t}} - \frac{(1-\alpha_t)(\sqrt{1-\bar{\alpha}_t})}{(1-\bar{\alpha}_t)(\sqrt{\alpha_t})}\epsilon_\theta = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x_t} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\right) \tag{58}$$

$$\Sigma_\theta(t) = \Sigma_q(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I} \tag{59}$$

$$\text{We now have defined } \mathbf{x_{t-1}}\text{'s mean and variance given } \mathbf{x_t} \text{ which let's us write it as} \tag{60}$$

$$\mathbf{x_{t-1}} = \mu_\theta(\mathbf{x_t}) + \sqrt{\Sigma_\theta(t)}\mathbf{z} \tag{61}$$

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{62}$$

We have now a way to generate $\mathbf{x_{t-1}}$ from $\mathbf{x_t}$. This means that if we have $\mathbf{x_1}$ we can generate $\mathbf{x_0}$. This let's us finally define our sampling algorithm:

Algorithm 2: Sampling

```
repeat
    $\mathbf{x_T} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$    ▷ Sample random noisy image
    for $t = T, \ldots, 1$ do
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$ else $\mathbf{z} = \mathbf{0}$
    $\mathbf{x_{t-1}} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x_t} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x_t}, \mathbf{t})\right) + \sqrt{\Sigma_\theta(t)}\mathbf{z}$    ▷ Sampling $\mathbf{x_{t-1}}$ from $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$
    end for
    return $\mathbf{x_0}$
```

0.1 Recap

We have studied both the forward and reverse process that make up DDPMs. We have seen that good samples (or images) are generated by maximing the reverse process' likelihood $logp_\theta(\mathbf{x_0})$ lower bound, the Evidence Lower Bound, ELBO, or simply L.

We have rewritten ELBO in a way that its maximization turns out to be dual with minimizing the Kullback-Leibler (KL) divergence between $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$ and $q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0})$. Such KL-Divergence minimization was then translated to a noise predictor training. After the predictor training algorithm was defined, we have shown how to use such prediction to sample from $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$ and finally reconstructing $\mathbf{x_0}$, the original denoised image.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[2] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *CoRR*, abs/1503.03585, 2015.