

Diffusion Notes

Nicholas Leland

October 2024

Contents

1 Step by Step Diffusion: An Elementary Tutorial	1
1.1 Fundamentals of Diffusion	1

1 Step by Step Diffusion: An Elementary Tutorial

The first source that we will be taking a look at is **Step by Step Diffusion: An Elementary Tutorial**. This is meant to pose as an introductory course to the concept of Diffusion, and should help us gain a general understanding of the topic.

1.1 Fundamentals of Diffusion

Gaussian diffusion is the act of taking images that have had noise applied to them, *denoising* the image in order to gain a deeper understanding of the framework that the image is built on.

We will begin by exploring how we inject our images with noise.

Definition 1. *the Forward Process can be described as follows:*

$$x_{t+1} := x_1 + n_t, n_t \sim N(0, \sigma^2)$$

Essentially, we are creating a target distribution p^* that is generated using a data set we have identified. Then, using this formula, we generate multiple random variables $(x_0, x_1, x_2, \dots, x_t)$. With these random variables (in our case, images), we can identify that we slowly begin to approach the **Gaussian Distribution**, therefore, we can sample the Gaussian rather than our originally defined p^* distribution.

We can evaluate the two distributions using a formula known as KL Divergence.

Definition 2. *The **Kullback-Leibler (KL) divergence** between two distributions P and Q is defined as:*

$$D_{KL}(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}.$$

or for a continuous distribution:

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} P(X) \log \frac{P(x)}{Q(x)}.$$

- $P(x)$ is the true probability distribution
- $Q(x)$ is the approximating (or target) probability distribution
- $D_{KL}(P||Q)$ is the KL Divergence which is non-negative and measures how much information is lost when Q is used to approximate P .

So would we be utilizing Continuous or Discrete variables? You might think initially that due to the format of image files given as:

$$P_{ij} \in [0, 255]^3 \text{ for } i = 1, \dots, W \text{ and } j = 1, \dots, H.$$

Realistically though, our model will think within the space of 0-1 instances which we will then convert into our discrete space. We will find later on that this quantization is actually quite a problematic idea with images.

It might be important to have these definition's on hand as well, mainly just as a refresher.

Definition 3. *Probability Density Function*

$$P(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right).$$

where:

- x is a d -dimensional random vector.
- μ is the mean vector
- Σ is the covariance matrix.

Definition 4. *Covariance Matrix*

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])].$$

where:

- $\mathbb{E}[X]$. is the expected value (mean) of X .
- $\mathbb{E}[Y]$. is the expected value (mean) of Y .

Now that we have initial definitions out of the way, let's reevaluate our problem. We have an image, we want to define some way to reconstruct an image of that nature. Our sub problem is essentially:

"Given a sample marginally distributed as p_t , introduce a sample marginally distributed as p_{t-1} "

Now that we have established our noisy image, the goal is to create a model which can identify features and effectively denoise our image. This method is known as a **reverse sampler**.

We would effectively move through each position until indefinitely reaching our original, target distribution ($p_0 = p^*$). This leads us to the primary idea behind diffusion, rather than learning how to go from a noised image back in one step (more in line to how a VAE functions), we instead go through multiple steps and learn the process behind each.

We can construct reverse samplers in many different ways, the **DDPM Sampler** is known as the standard diffusion sampler. This uses a very simple strategy, at a time t , given the input z , where z is a sample from p_t , output a sample from the conditional distribution.

$$p(x_{t-1} | x_t = z).$$

This seems very simple but realistically there is much more to the entire process. If we were to have a model for every single step of x_t , this would be quite a large process. The key is identifying the *per step noise* or σ .

If this amount of noise that we are adding per step is small enough, the distribution becomes simple.

Definition 5. Diffusion Reverse Process: For small σ , the Gaussian diffusion process defined in (1), the conditional distribution $p(x_{t-1}|x_t)$ is itself close to Gaussian. That is, for all the times t and conditioning $z \in \mathbb{R}^d$, there exists some mean parameter $\mu \in \mathbb{R}^d$ such that:

$$p(x_{t-1}|x_t = z) \approx \mathcal{N}(x_{t-1} : \mu, \sigma^2).$$

There is a lot going on here, let's talk about it in a bit more detail. Essentially speaking, if we have the distribution that we are trying to understand, by targeting a certain point and minimizing our σ value, the distribution approaches that of the Gaussian at a much more aggressive rate. This allows us to have one missing value to focus on, the mean (μ).

This is a really important statement, so let's go through this one more time.

- For a given time t and a conditioning value x_t , learning the mean of $p(x_{t-1}|x_t)$ is sufficient to learn the full conditional distribution $p(x_{t-1}|x_t)$.
- Now that we know we must just determine the mean (μ), we can just utilize as simple tool like **simple linear regression** to begin.
- Our goal is to estimate $\mathbb{E}[x_{t-1}, x_t]$

Definition 6. Standard Regression Loss : Remember, for any distribution over (x, y) , we have:

$$\begin{aligned} \arg \min_f \mathbb{E} [||f(x) - y||^2] &= \mathbb{E}[y|x] \\ \mu_{t-1}(z) &:= \mathbb{E}[x_{t-1}|x_t = z] \\ \mu_{t-1} &= \arg \min_{f: \mathbb{R}^d \rightarrow \mathbb{R}^d} \mathbb{E}_{x_t, x_{t-1}} ||f(x_t) - x_{t-1}||_2^2 \\ \arg \min_{f: \mathbb{R}^d \rightarrow \mathbb{R}^{dx_d}} \mathbb{E}_{x_t, x_{t-1}, t} ||f(x_{t-1} + t) - x_{t-1}||_2^2 \end{aligned}$$

Essentially, we are optimizing the μ for the minidifference between our given distribution and the noise that is generated on the image. x_0 is from our target distribution p^* . When target p^* is a distribution on images, then the corresponding regression problem is exactly an **image denoising objective**. This objective can then be approached with deep learning techniques such as traditional CNN's.

Through this process we have taken our goal of **learning to sample from an arbitrary distribution** to the standard **problem of regression**.