Diffusion Notes

Nicholas Leland

October 2024

Contents

Chapter 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

1.1.1 Gaussian 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:*

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

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 $(x_0, x_1, x_2, \ldots, x_t$. 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 then 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*
- $\mathcal{Q}(x)$ *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:

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

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*

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

where:

- x *is a d-dimensional random vector.*
- μ *is the mean vector*
- $\sum$ *is the covariance matrix.*

Definition 4. *Covariance Matrix*

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

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 then 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 $z \subset \mathbb{R}^d$, there exists some mean parameter $\mu \subset \mathbb{R}^d$ 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), $we can just utilize a simple tool like$**simple linear regression**$! to begi$
- 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:*

$$\arg\min_f \mathbb{E}\left[||f(x) - y||^2\right] = \mathbb{E}[y|x]$$
$$\mu_{t-1}(z) := \mathbb{E}[x_{t-1}|x_t = z]$$
$$\mu_{t-1} = \arg\min_{f:\mathbb{R}^d \to \mathbb{R}^d} \mathbb{E}_{x_t, x_{t-1}} ||f(x_t) - x_{t-1}||_2^2$$
$$\arg\min_{f:\mathbb{R}^d \to \mathbb{R}^{dxd'}} \mathbb{E}_{x_t, x_{t-1}, t} ||f(x_{t-1} + t) - x_{t-1}||_2^2$$

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**.

1.1.2 Diffusions in the Abstract

Forget about the Gaussian now. Let's break down our *diffusion-like generative model.*

- Start with a target distribution (p^*)
- Pick a base distribution ($q(x)$) that is easy to sample from (Standard Gaussian or i.i.d bits)
- Construct a sequence of distributions which interpolate between the target p^* and the base distribution q. We now have a set of distributions $p_0, p_1, p_2, \ldots, p_t$.
- With these distributions, $p_0 = p^*$ (Our Target), $p_t = q$ (base distribution) and the rest of the distributions, $p_t - 1, pt$, are marginally close to one another. The smaller steps the more accurately we can assume the base distribution.
- Then we learn a *reverse sampler* to take our p_t distributions and transform them into p_{t-1}. This is how we train the model, or our *learning step.*

Definition 7. ***Reverse Sampler*** *Given a sequence of marginal distributions p_t, a reverse sampler for step t is a potentially stochastic function F_t such that if x_t p_t, then the marginal distribution of $F_t(x_t)$ is exactly p_{t-1}:*

$$F_t(z) : z\ p_t \equiv p_{t-1}$$

Remember, stochastic functions are just functions that have an unpredictable outcome, which is entirely valid based on the assumption of how noise is added.

There are many possible reverse samplers that are available, some are even *deterministic*, or include no randomness. In some cases, it is possible to instantiate in a discrete setting where the distribution p^* is over a finite set and there are corresponding "interpolating distributions" and reverse samplers.

We will be evaluating three very popular reverse samplers here:

- The DDPM sampler
- The DDIM Sampler (More deterministic
- The Family of *flow-matching models*, a generalization of DDIM

1.1.3 Discretization

Very quickly, we run into the problem regarding dealing with discrete values (float vs int). The process of taking a function that is normally a continuous function and transferring that into one that is discrete is known as **discretization**. The goal is to reduce the amount of error down to a set *negligible amount* by the end of the process.

Currently we have explained the process of de-noising an image as finding a step value (p_{t-1} where the value is *extremely small*, therefore we can draw the same assumptions about the baseline distribution that we are using. We run into a problem when evaluating how small this step (p_t, p_{t-1} actually ends up being.

As a refresher, we have a time-evolving function, $p(x,t)$, which starts from the target distribution (p_0 at time $t=0$) and ends at the noisy distribution (p_t at time $t=1$). It is important to note, this is not a traditional function but rather the current status of our distribution at the set points (x and t). To reiterate, $p(t_0,x)$ is our original distribution (our clean image) and $p(t_1,x)$ is our distribution where we represent our target distribution.

$$p(x, k\Delta t) = p_k(x), where \Delta t = \frac{1}{T}.$$

Here, T is the *fineness* of the steps that we are taking. The more steps (and presumably the smaller the σ value), the closer the image is from step to step.

We run into a very interesting problem now. We need to ensure that the variance of the final distribution p_t is *independent of the number discretization steps.* To ensure that this is a reality, we need to be more specific about our incremental variances.

Let's look at the following given fact. If $x_k = x_{k-1} + \mathcal{N}(0, \sigma^2)$ then $x_t \sim \mathcal{N}(x_0, T\sigma^2)$

Essentially, because we are pulling from our target distribution so many times, normally, this would cause an overlap to occur, which in turn would aggressively scale our variance. This would result in our variance scaling at a factor of $T \cdot \sigma^2$. This would be quite problematic as the stacking variance would skew our distribution. To accommodate for this, we will instead by scaling the variance by the number of steps.

$$Total\ Variance = T \cdot (\sigma^2 \cdot \Delta t) = T \cdot (\sigma^2 \cdot \frac{1}{T}) = \sigma_2^2.$$

Remember, $\Delta t = \frac{1}{T}$ is the amount that we will be scaling by. With this fact, we will be choosing.

$$\sigma = sigma_q \sqrt{\Delta t}.$$

where σ_q^2 is the *desired terminal variance.* This concept of $\sqrt{\Delta t} scaling will be very important, and it's the baseline that SDE (Stochast$ with the desired terminal variance, we ensure that the variance of P_t is always σ_q^2 regardless of our T.

At this point, we will be taking a temporary pause with this section and instead be jumping to the next chapter of these notes. This will build up the fundamentals regarding Stochastic Differential Equations, which will be very relative for the upcoming diffusion notes.

Chapter 2

Stochastic Differential Equations

2.1 Differential Equations

This section will be primarily reviewing general differential equations and their fundamentals. We will be going over the baselines with some basic videos, and then diving into more traditional mathematical approaches after that.

2.1.1 Differential Equations, a tourist's guide

The first resource that we will be evaluating is the 3Blue1Brown video series regarding differential equations. This is avalible here: https://www.youtube.com/watch?v=p$_d$i4Zn4wz4list = $PLZHQObOWTQDNPOjrT6KVlfJuKtYTftqH6$

Differential Equations are essentially a function in which it is easier to document the change in numbers rather then the direct values.

Why they grow or shrink over why it is one value over another.

Differential equations are one of two: Ordinary Differential Equations (one input) or Partial Differential Equations (multiple inputs)

- Ordinary Differential Equations or **ODE**'s, are typically involving time. This is very commonly associated with physics calculations.
- Partial Differential Equations or **PDE**'s, deal with multiple inputs like a temperature at points in a solid body or the velocity of a fluid at different points in space.

Say we have an item falling. Based on earths gravity, our object is falling at a rate of $-9.8\frac{\frac{m}{s}}{s}$. This itself can be expressed as a differential equation, where it's derivative, gives the velocity, and the second derivative as the acceleration. We turn this into a differential equation problem when we begin to ask the reverse question. If we look at a generalized problem, we find that as we work through this problem backwards, we begin to introduce additional constants that are free to change. If we want to find the function which has an acceleration of 9.8, we are looking for a function with $-g$ as its derivative. This ends up introducing a t constant, which we know as time. If we do the same with our new function, looking for the original function given our acceleration and time constant, we now introduce an additional constant, one that changes the starting position. There can be many functions with each derivative, which is why we now need to clarify given the new constants. Again, if my acceleration is a second or d er derivative, -9.8. When I am trying to find the inverse of the derivative, there are many functions that might end up with this derivative, we need to isolate with the additional time position.

What if the forces depend on the location of the body? At this rate, we have an interchangeable exchange between a potential location and the resulting differential variable.

Often in differential equations, the puzzles you face involve finding a derivative or higher order derivative which is defined interms of the function itself.

2.2 18.S096 Lecture 21: Stochastic Differential Equations

This next section of notes will be directly from an MiT lecture avalible on youtube at https://www.youtube.com/watch?v=qdbkvD us. This is primarily from the background of a FinTech domain, but should also go through some aspects that will be applicable for diffusion before we jump into the next section.

Say we are given a Differential Equation.

$$dX = \mu(t, X(t))dt + \sigma(t, X(t)dB(t).$$

Our goal is to find a stochastic process $X(t)$ that satisfies a given function.