Diffusion

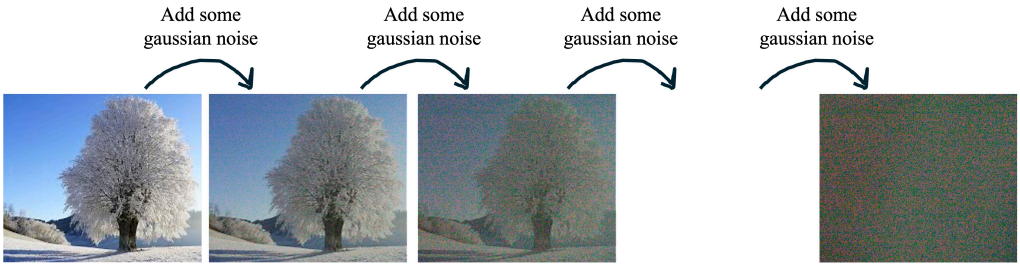
Diffusion Task

Goal: Create a model that can generate pictures of trees from the "tree photo distribution"

Data: Many pictures of trees:

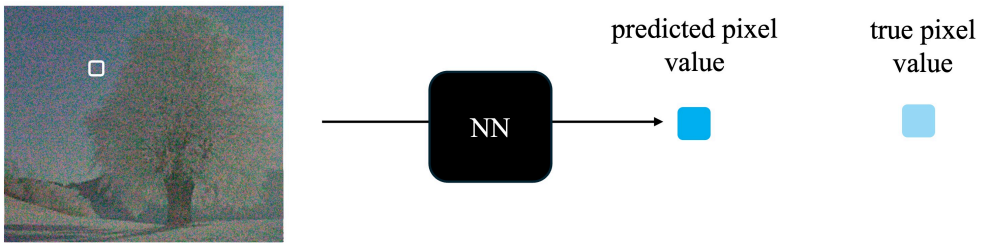
Big Picture Idea

Supplement your dataset by iteratively adding gaussian noise to pixels, then train a deep learning model to remove noise.



A reasonable number of steps would be to add 10% noise each time step, so that after 10 timesteps each pixel is fully noise.

The key task is to train a deep neural network to predict the "denoised" value of pixels:



Loss: Mean squared error between the predicted pixels and the true color. Set the parameters of your neural network to minimize loss. Then you have a model which can remove 10% noise one step at a time. Start with random noise, then run it through your denoising neural network 10 times.

Theory Behind Diffusion Models

The magic of diffusion models lies in the Gaussian noise process. Let's break it down:

1. Adding Noise: The Forward Process

At each step t, we add Gaussian noise to the pixel values:

$$x_{t+1} = x_t + n_t \quad \text{where } n_t \sim N(0, \sigma^2).$$

This gradually turns the original image into pure noise.

2. Removing Noise: The Reverse Process

To reverse this process, we need the conditional distribution $x_{t-1}|x_t$. Here's the surprising part:

Critical Fact #1: $x_{t-1}|x_t$ is Gaussian with known variance
If the noise variance σ^2 is small enough, the distribution of $x_{t-1}|x_t$ can be approximated as:

$$x_{t-1}|x_t \sim N(\mu_{t-1}(x_t), \sigma^2),$$

where:

- $\mu_{t-1}(x_t)$: The mean of the Gaussian, which depends on x_t.
- σ^2: The known variance of the noise.

This is great news! It means we only need to estimate the mean $\mu_{t-1}(x_t)$ to fully describe $x_{t-1}|x_t$.

3. Training the Neural Network

Critical Fact #2: Standard regression is all you need!
To train the neural network, we need it to predict $\mu_{t-1}(x_t)$, the mean of the Gaussian. How do we measure the quality of the predictions?

The difference between the predicted Gaussian $q_\theta(x_{t-1}|x_t)$ (from the neural network) and the true Gaussian $p(x_{t-1}|x_t)$ can be measured by the **KL divergence**. Happily in this case:

$$\text{Minimizing KL divergence } \Leftrightarrow \text{Minimizing mean squared error (MSE)}.$$

This is only true because the distributions are Gaussian. Thus, we can simply train the neural network to minimize the MSE between its prediction of pixel values ($\mu_{t-1}(x_t)$) and the true pixel values.

Once trained, the neural network can iteratively denoise an image, starting with random noise, until it generates a clear, realistic image.

4. The Complete Diffusion Algorithm

Here's the full workflow for a diffusion model:

1. **Forward process:** Add Gaussian noise to turn images into pure noise.
2. **Reverse process:** Train a neural network to predict the mean $\mu_{t-1}(x_t)$ and remove noise step-by-step.
3. **Image generation:** Start with random noise and run the neural network in reverse T times to generate a realistic image.

This elegant approach combines simple Gaussian noise with the power of deep learning to generate stunning results!

Proof Sketch of Critical Idea #1

Claim: $x_{t-1}|x_t$ is approximately Gaussian when the noise variance σ^2 is small enough.

We start with Bayes' theorem to express the conditional probability:

$$\mathrm{P}(x_{t-1}|x_t) = \frac{\mathrm{P}(x_t|x_{t-1})\,\mathrm{P}(x_{t-1})}{\mathrm{P}(x_t)}$$

We are going to think about the log of this expression. This is because the log of a Gaussian is a quadratic function, which will make our math easier. We can write:

$$\log \mathrm{P}(x_{t-1}|x_t) = \log \mathrm{P}(x_t|x_{t-1}) + \log \mathrm{P}(x_{t-1}) - \log \mathrm{P}(x_t)$$

Let's break down the terms in this expression.

Forward process likelihood:
From the forward process, x_t given x_{t-1} is Gaussian:

$$\mathrm{P}(x_t|x_{t-1}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_t - x_{t-1})^2}{2\sigma^2}\right).$$

Taking the log, we get:

$$\log \mathrm{P}(x_t|x_{t-1}) = -\frac{(x_t - x_{t-1})^2}{2\sigma^2} + \text{constant}$$

Prior on x_{t-1}:
The prior $p(x_{t-1})$ is the probability of x_{t-1} at the previous step. This one is hard to know! What is the prior distribution of a pixel of a tree? However we employ a really neat trick. Its log-density can be Taylor expanded around x_t, assuming x_{t-1} is close to x_t:

$$\log \mathrm{P}(x_{t-1}) \approx \log \mathrm{P}(x_t) + \Big[\frac{\partial}{\partial x} \log \mathrm{P}(x_t)\Big](x_{t-1} - x_t)$$

Completing the square:
There are two terms in the above expressions involving the difference between x_{t-1} and x_t. As a helpful step, we will complete the square for these terms.

Complete the square for this sum of terms:

$$-\frac{(x_t - x_{t-1})^2}{2\sigma^2} + \Big[\frac{\partial}{\partial x} \log \mathrm{P}(x_t)\Big](x_{t-1} - x_t)$$

First, rewrite $(x_t - x_{t-1})^2$:

$$(x_t - x_{t-1})^2 = (x_{t-1} - x_t)^2$$

Allowing us to rewrite our sum as:

$$-\frac{1}{2\sigma^2}(x_{t-1} - x_t)^2 + \Big[\frac{\partial}{\partial x} \log \mathrm{P}(x_t)\Big](x_{t-1} - x_t)$$

Factor out $-\frac{1}{2\sigma^2}$ to make the quadratic term more explicit:

$$-\frac{1}{2\sigma^2}\left[(x_{t-1} - x_t)^2 - 2\sigma^2\Big[\frac{\partial}{\partial x} \log \mathrm{P}(x_t)\Big](x_{t-1} - x_t)\right]$$

The expression inside the brackets is a quadratic expression in $(x_{t-1} - x_t)$. Let's complete the square for:

$$(x_{t-1} - x_t)^2 - 2\sigma^2\Big[\frac{\partial}{\partial x} \log \mathrm{P}(x_t)\Big](x_{t-1} - x_t)$$

This allows us to rewrite the quadratic expression as:

$$\left[(x_{t-1} - x_t) - \sigma^2 \Big[\frac{\partial}{\partial x} \log \mathrm{P}(x_t)\Big]\right]^2 - \left(\sigma^2 \Big[\frac{\partial}{\partial x} \log \mathrm{P}(x_t)\Big]\right)^2$$

Substitute this back. Let K stand in for constant:

$$\begin{aligned}
\log \mathrm{P}(x_{t-1}|x_t) &= \log \mathrm{P}(x_t|x_{t-1}) + \log \mathrm{P}(x_{t-1}) - \log \mathrm{P}(x_t) \\
&= -\frac{(x_t - x_{t-1})^2}{2\sigma^2} + \Big(\log \mathrm{P}(x_t) + \Big[\frac{\partial}{\partial x} \log \mathrm{P}(x_t)\Big](x_{t-1} - x_t)\Big) - \log \mathrm{P}(x_t) + K \\
&= -\frac{(x_t - x_{t-1})^2}{2\sigma^2} + \Big[\frac{\partial}{\partial x} \log \mathrm{P}(x_t)\Big](x_{t-1} - x_t) + K \\
&= -\frac{1}{2\sigma^2}\left[\left(x_{t-1} - x_t - \sigma^2 \Big[\frac{\partial}{\partial x} \log \mathrm{P}(x_t)\Big]\right)^2\right] + K
\end{aligned}$$

Final Result: Recall that the log of the gaussian PMF looks like this:

Let $X \sim N(\mu, \sigma^2)$. What is the log of the PMF of X?

$$\log \mathrm{P}(X = x) = -\frac{1}{2\sigma^2}(x - \mu)^2 + K$$

From the above, we see that $x_{t-1}|x_t$ is Gaussian. How do we know this? The distribution is identical, up to additive factors, to the log-density of a Normal distribution.

$$x_{t-1}|x_t \sim N\left(\mu_{t-1}, \sigma^2\right),$$

where:

$$\mu_{t-1} = x_t + \sigma^2 \Big[\frac{\partial}{\partial x} \log \mathrm{P}(x_t)\Big]$$

The variance remains σ^2, which is fixed from the forward process.