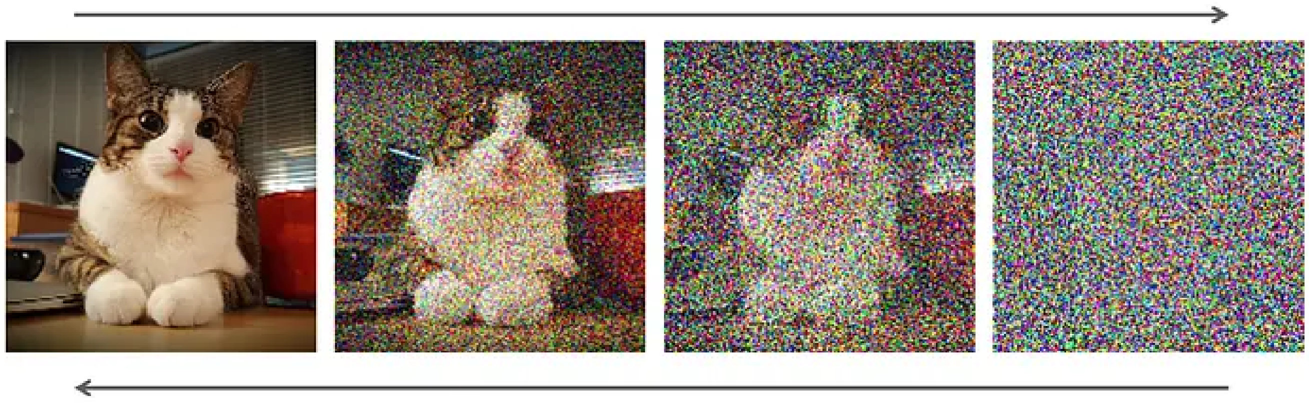
Diving into Diffusion Models

Saurabh Raj · Follow

5 min read · Jun 3, 2023

82 1

The blog will focus on the working principles of Diffusion models.

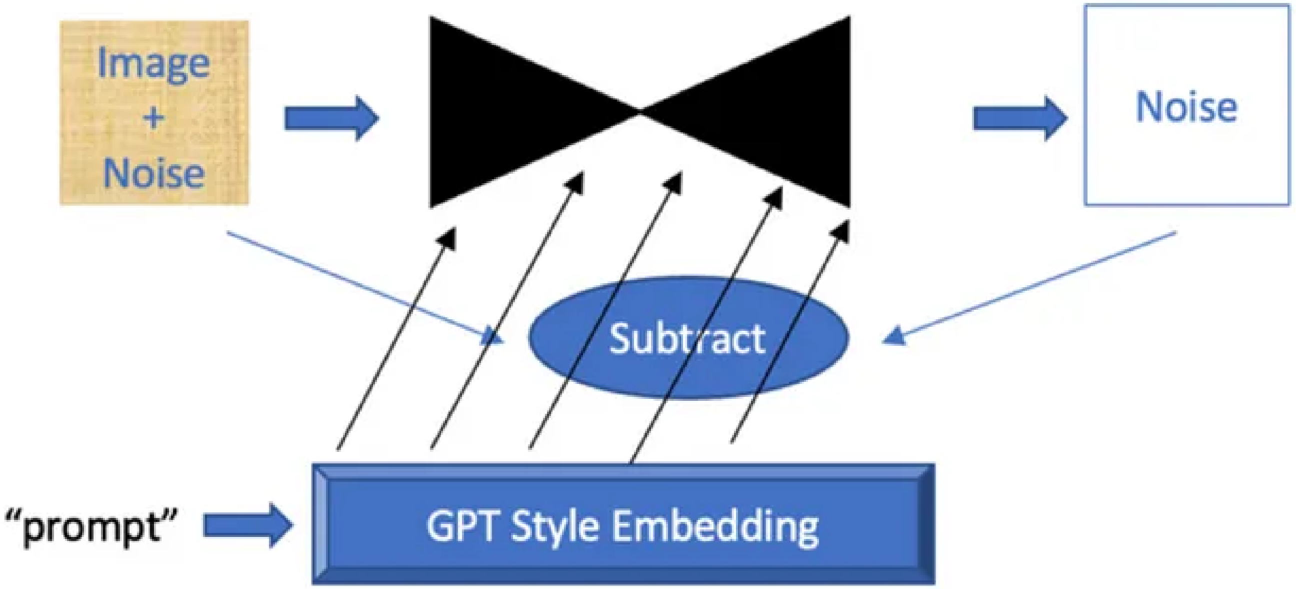


Source: https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/

Introduction

Nowadays, as Generative AI is making grade everywhere, most of us have become familiar with Midjourney, DALL-E, etc. These APIs are able to generate attractive images, given a simple to complex prompt by the end user.

A very basic intuition behind the algorithm used is that they try to get a sensible image (given the prompt) out of a noise by iteratively removing noise. Suppose we have an image of a dog, and we progressively keep on adding noise to it, so that the image becomes an absolute noise, in which we cannot recognise any object as such. Then we must also be able to estimate noise in the noisy image and remove the noise progressively to get a sensible image of the dog. The noise removal is assisted by the GPT style embedding obtained out of the prompt. So that while estimating noise, the network keep the semantic meaning of prompt into context. The pixels in the noisy image belong to Normal Distribution and we try to sample noise from normal distribution.



The above intuition is good for starter but there are multiple aspects of the logic which we will see in this blog.

Sampling

We have a noise image (NIM), we pass it through a neural network, and our network predicts some noise which we essentially subtract from our noise image (NIM). We repeat this iteratively multiple times.

Step 1: Sample a random noise sample using random method of the framework in use. Here we need to mention the dimension of the image which we are trying to generate. Let us call this sample as **org_samples.**

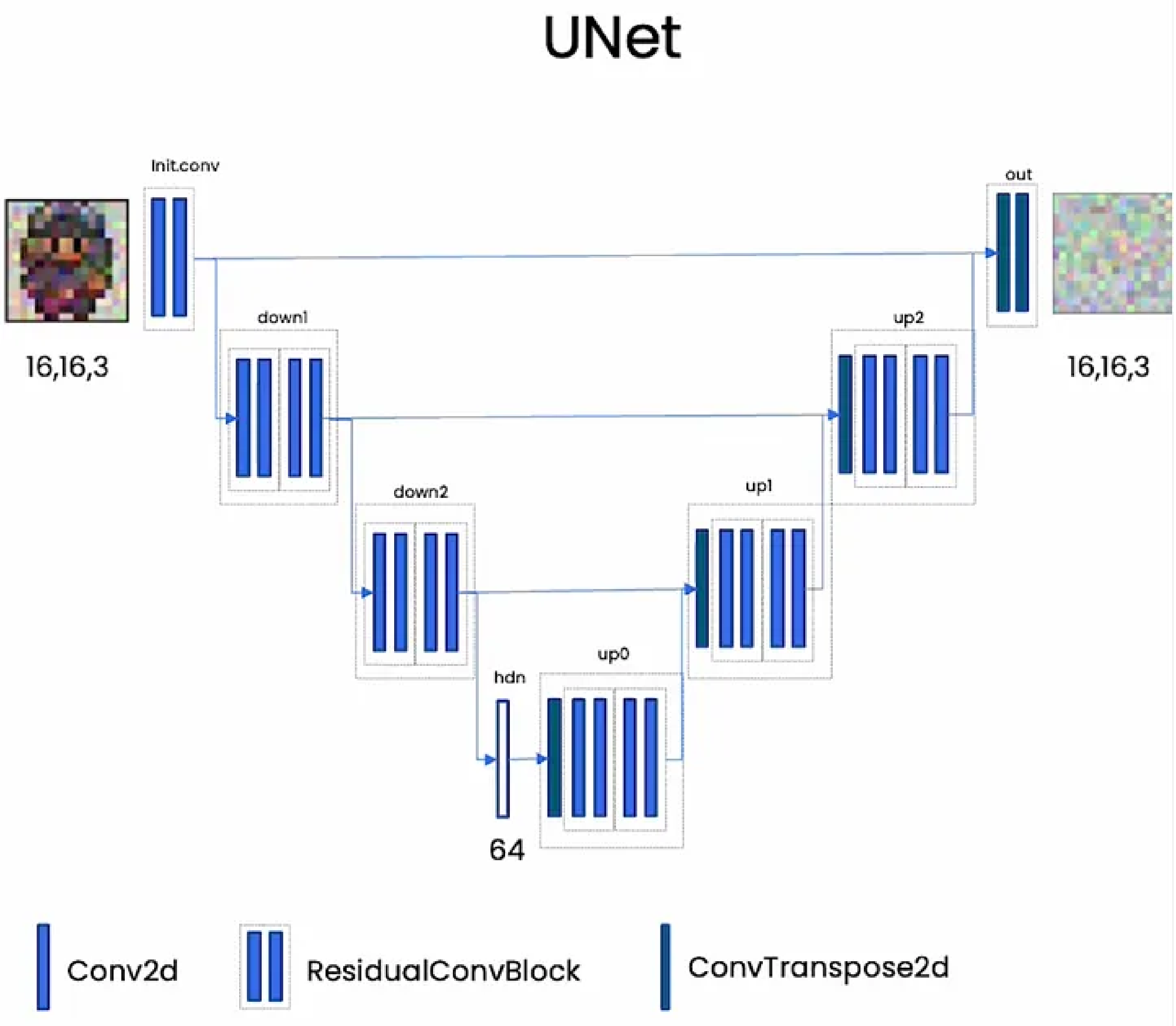
Step 2: Initiate a for loop to iterate from T timestep to 0 (reverse iteration).

Step 3: Then we create some **extra noise**.

Step 4: Then we pass the org_samples into our neural network to predict the **estimated noise.** This estimated noise will be subtracted from our original noise samples.

Step 5: We subtract the estimated noise from original noise using a DDPM algorithm (Denoising Diffusion Probabilistic Model) and then add the extra noise created in step 2. This extra noise is added because adding some extra noise after subtraction gives more accurate results (experimentally speaking).

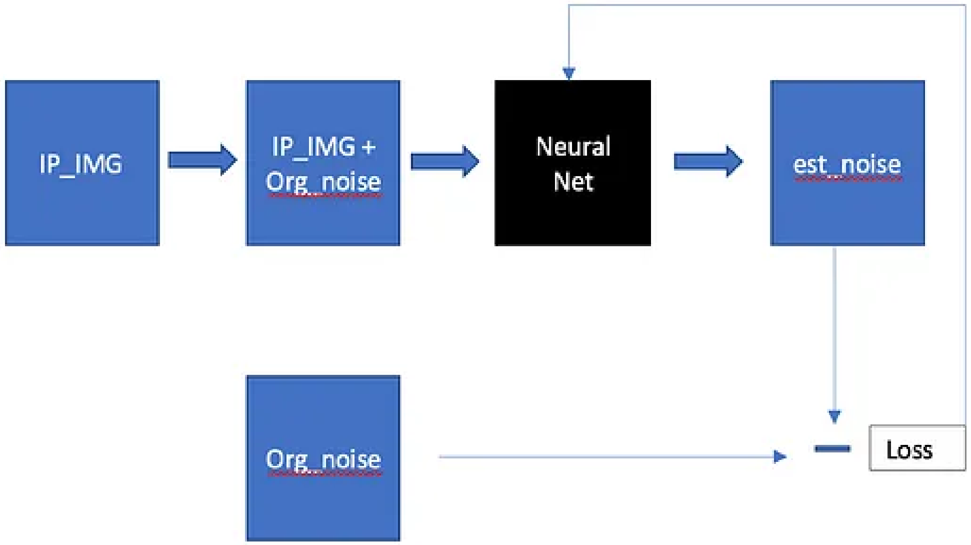
Neural Network Architecture



Source:https://www.deeplearning.ai/

The network is a typical U-Net architecture which consists of downssampling followed by upsampling and assisted by Residual connections. We feed an input image and the output is a noise of the same size as that of input. The key thing about U-Net is that the input and output is of same size. Left part of the Unet is responsible for compressing all the information of input image into an embedding (hdn-64). We do this using multiple convolution layers. Then the right part is responsible for upsampling the embedded information into an output of the same size. We use same number of convolution layers for upsampling. While performing the upsampling, we can use other embeddings such as **context vector** of input promt (text) and **time embeddings** (tells model what time stamp it is) to assist our compressed embedding of input image (hdn). We add the time embedding and multiple the context embedding to our hdn.

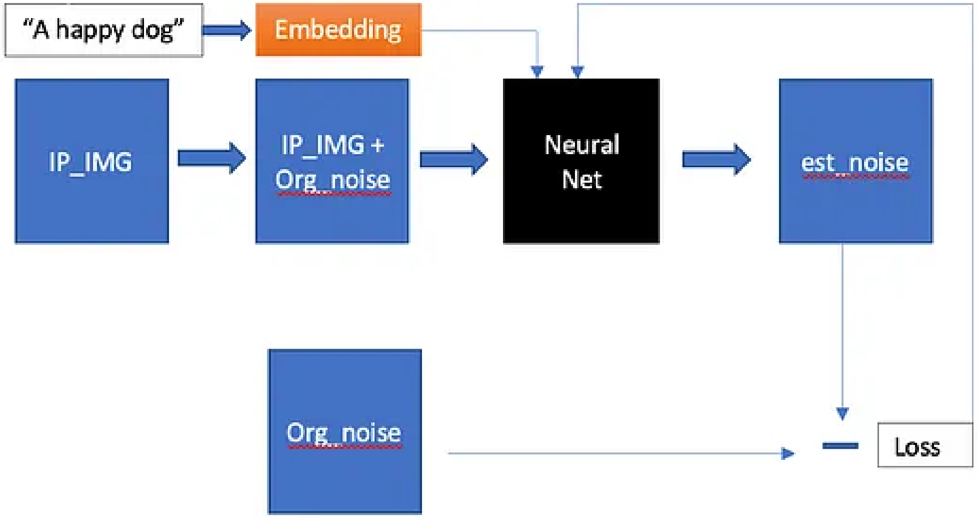
Training



We randomly sample the org_noise and then add it to the input image (IP_IMG). Then we train our model in a way that it should predict a noise (est_noise). Then we calculate the loss using estimated noise (est_noise) and original noise (Org_noise). We can use Mean Squared Error for loss calculation and then backpropagate. In this way our neural network learns to predict the noise by learning the distribution input image pixels (noise vs not noise). For more stable performance, the level of noise are randomly sampled for different images.

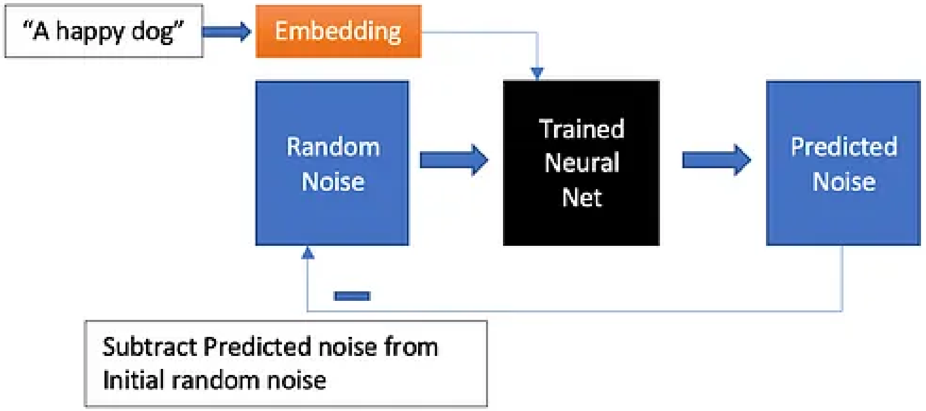
A bit more about Context Embedding

Context Embeddings are vectors that captures the meaning/information of



As explained in above section, we pass the embedding vector of text description of input image. For example, if we have an image of a happy dog, then we can pass the text "A happy dog" through GPT like embedding layer and pass the context embedding into out network.

While evaluating, we start with a random noise and a text description (prompt imagined by the user). Then we iteratively predict the noise based on context embedding and eventually we get an image close to our imagination.



Summary

We trained a Neural Network (UNet like) to learn the distribution of noise and not noise from input image based on a textual description embedding. Sampling is a key step where we created extra noise as well. We added extra noise after subtracting the estimated noise from original noise. The subtraction of estimated noise was done using DDPM algorithm. While evaluating, we start with a random noise and keep on removing noise based on text description embedding untill we get a sensible image.

Reference: https://www.deeplearning.ai/

Artificial Intelligence

Deep Learning

Generative Ai

Diffusion Models

Neural Networks

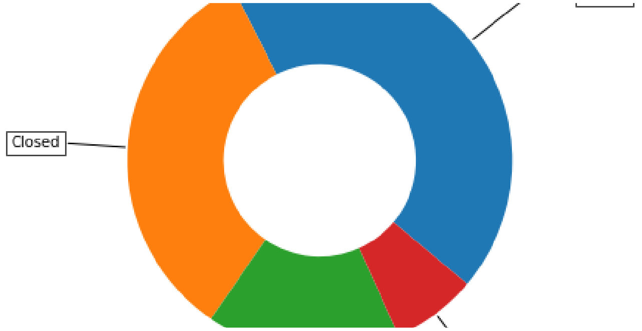
Written by Saurabh Raj

Follow

53 Followers

City, University of London' 23 , IIT Jammu’ 20

More from Saurabh Raj



Saurabh Raj

Case Study: Customer Complaints Analysis & ML model training

A detailed case study of customer complaints in a firm and training multiple machine...

10 min read · Sep 25, 2020

9 1

Saurabh Raj

Dealing with Imbalanced Dataset (Company Bankruptcy Prediction)

A case study on famous company bankruptcy dataset which attempts to solve the highly...

11 min read · Jan 29, 2023

53 1

Saurabh Raj

An example of Proportional Sampling

A walk through the concept of proportional sampling by an example explanation with...

4 min read · Sep 19, 2020

6 1

Saurabh Raj

Road Damage Detection using AI

A comprehensive case study for detecting the type of damages on roads using state of...

10 min read · Jun 29, 2021

24 2

See all from Saurabh Raj

Recommended from Medium

Naman Rastogi

Navigating DDPMs—A Closer Look At Denoising Diffusion…

In the realm of nature's uncertainty, probability becomes the beacon that…

28 min read · Aug 17, 2023

114 1

Jonathan Kernes in Towards Data Science

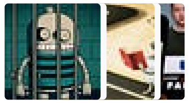
Diffusion Models

What are they, how do they work, and why now?

23 min read · Dec 13, 2022

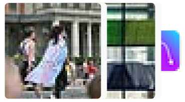
290 4

Lists



AI Regulation

6 stories · 283 saves

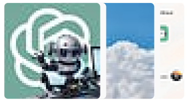


Generative AI Recommended Reading

52 stories · 639 saves

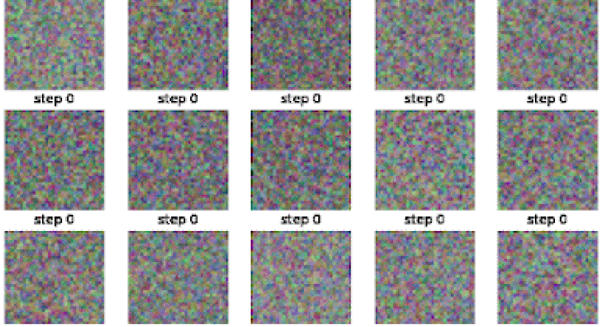
Natural Language Processing

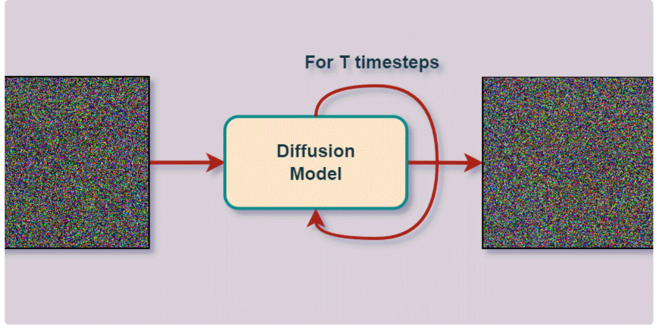
1117 stories · 588 saves



ChatGPT

23 stories · 406 saves





Abdulkader Helwan

Diffusion models from scratch

Diffusion models are a type of generative deep learning model that can generate new...

✦ · 4 min read · Oct 4, 2023

49 2

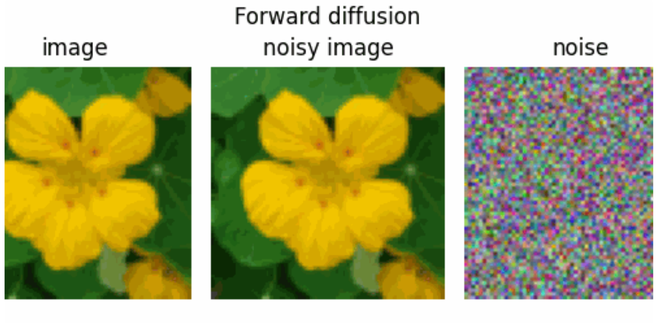
Gabriel Mongaras in Better Programming

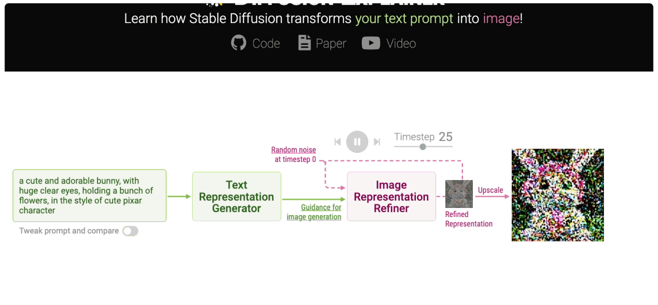
Diffusion Models — DDPMs, DDIMs, and Classifier Free Guidance

A guide to the evolution of diffusion models from DDPMs to Classifier Free guidance

28 min read · Mar 13, 2023

616 8




Learn how Stable Diffusion transforms your text prompt into image!

a cute and adorable bunny, with huge clear eyes, holding a bunch of flowers, in the style of cute pixar character

Tweak prompt and compare

Timestep 25

Code
Paper
Video
Text Representation Generator
Random noise at timestep 0
Guidance for image generation
Image Representation Refiner
Refined Representation
Upscale

kerolos yacoub

Diffusion Models

Introduction to diffusion models:

8 min read · Nov 14, 2023

Seongmin ... in Polo Club of Data Science | Georgi...

Stable Diffusion Explained and Visualized for Everyone

4 min read · Nov 11, 2023

4

172

See more recommendations