Diffusion Models

Mollaev D.E.

May 9, 2023

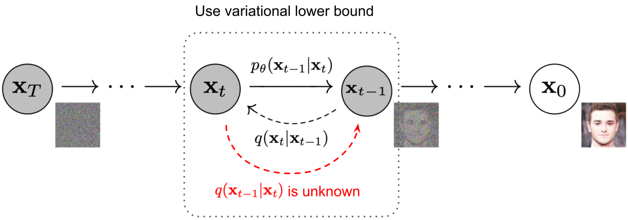
Project description

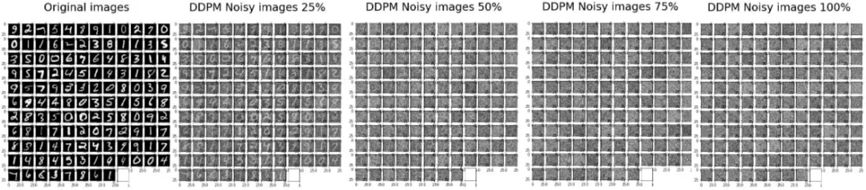
- Datasets - MNIST, CIFAR10
- Model - DDPM (Denoising Diffusion Probabilistics Models) with Unet
- Metrics - FID, INCEPTION SCORE

Let $x_0 \sim q(x_0)$, then we add small amount Gaussian noise to the sample in T steps. And we get sequence of noisy samples $x_1, ..., x_T$
Then conditional probability

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} * x_{t-1}, b_t\mathcal{I})$$

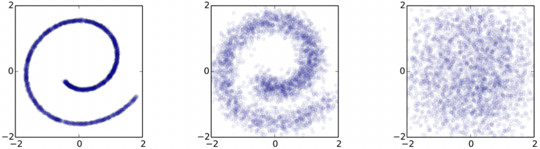
Eventually when $T \to \infty$, x_T is equivalent to an isotropic Gaussian distribution





$t = 0$ $t = \frac{T}{2}$ $t = T$

The forward trajectory
$q(\mathbf{x}_{0:T})$

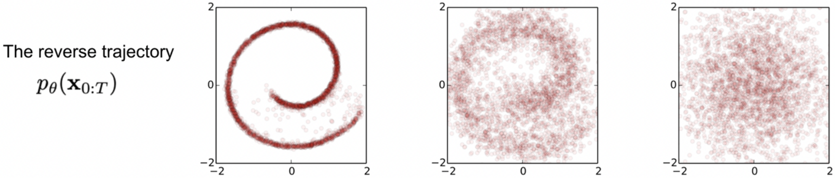


DDPM

Reverse diffusion process

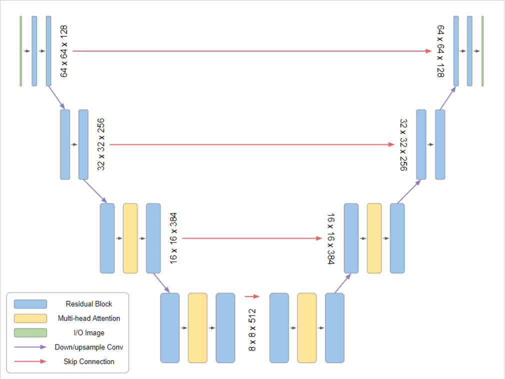
If we can reverse the forward process and sample from $q(x_{t-1}|x_t)$, we will be able to recreate the true sample from a Gaussian noise input, $x_T \sim \mathcal{N}(0, \mathcal{I})$.
But we cannot easily estimate $q(x_{t-1}|x_t)$ because it needs to use the entire dataset and therefore we need to learn a model(Unet) p_θ to approximate these conditional probabilities in order to run the reverse diffusion process.

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$



Link for visualization

DDPM
Unet



64 x 64 x 128
64 x 64 x 128
32 x 32 x 256
32 x 32 x 256
16 x 16 x 384
16 x 16 x 384
8 x 8 x 512
Residual Block
Multi-head Attention
I/O Image
Down/upsample Conv
Skip Connection

Algorithms

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

FID(Frechlet Inception Distance) is a performance metric that calculates the distance between the feature vectors of real images and the feature vectors of generate images

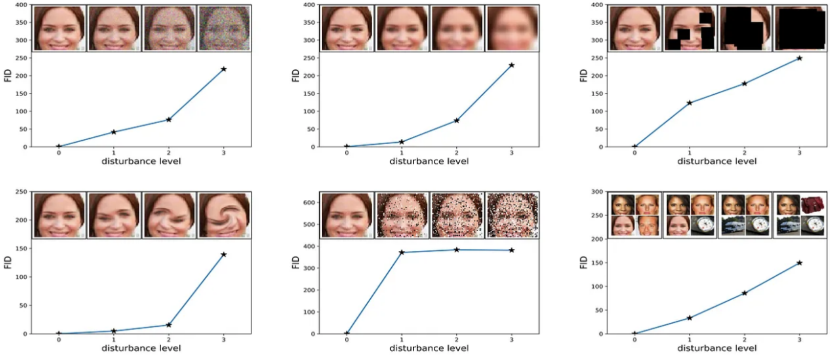
$$d^2((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\big(\boldsymbol{C} + \boldsymbol{C}_w - 2\big(\boldsymbol{C}\boldsymbol{C}_w\big)^{1/2}\big)$$

How to calculate FID?

1. Use the Inception V2 pre-trained model to extract the feature vectors of real images and generated images by the generator
2. Calculate the feature-wise mean of the feature vectors generated in step 1
3. Generate the covariance matrices of the feature vectors — C, C_w
4. Calculate trace
5. Calculate the squared difference of the mean vectors calculated in step 2
6. Finally, add the output of step 4 and step 5

Metrics

FID



The Inception Score(IS) is an objective performance metric, used to evaluate the quality of generated images or synthetic images. It measures how realistic and diverse the output images are.
It measures two things:

- **Diversity** (Variety) — How diverse the generated images are —The entropy of the overall distribution should be high.
- **Quality** (Goodness)— How good the generated images are — Low entropy with high predictability is required.

$$\mathrm{IS}(G) = \exp\left(\mathbb{E}_{\mathbf{x}\sim p_g} D_{KL}(\, p(y|\mathbf{x}) \parallel p(y) \,) \right)$$

- **Conditional Probability Distribution** — $p(y|x)$. It should be highly predictable and with low entropy. Here y is the set of labels and x is the image.
- **Marginal Probability Distribution** — $p(y)$

$$\int_z p(y|x = G(z))dz$$

Here, $G(z)$ is the generated image by the generator model when provided with a latent vector. If the data distribution for y is uniform with high entropy, then the synthetic images will be diverse.

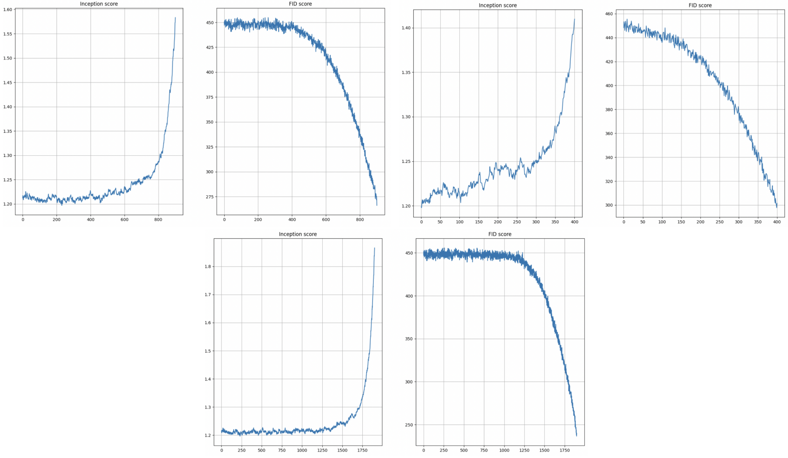
How to calculate IS?

1. Pass the generated images through the Inception model to get the conditional label distribution $p(y|x)$
2. Calculate the marginal probability distribution $p(y)$
3. Calculate the KL Divergence between $p(y)$ and $p(y|x)$
4. Calculate the sum over classes and take the average of outputs over images
5. Finally, take the exponential of the averaged value.

Results

I trained three generators with 500, 1000, 2000 steps on dataset CIFAR10.

Plot mterics for 1000, 500, 2000 steps:



Visualization

- Link for visualization(1000 steps on CIFAR10)
- Link for visualization(500 steps on CIFAR10)
- Link for visualization(2000 steps on CIFAR10)

Sources

Links:

- Denoising Diffusion Probabilistic Models
- Lil'Log What are Diffusion Models?
- Medium: Generating images with DDPMs: A PyTorch Implementation
- Medium: A Very Short Introduction to Inception Score(IS)
- Medium: A Very Short Introduction to Frechlet Inception Distance(FID)
- Special course on Mechmath MSU: Introduction to Machine and Deep Learning Theory