Introduction to Diffusion Models

Andreas Makris

30/10/2024

Data  Noise

Outline

- Quick history lesson
- DDPM
- Coding with PyTorch

The "history" of Diffusion Models

March 2015

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

July 2019

Generative Modeling by Estimating Gradients of the Data Distribution

(NCSN)

June 2020

Denoising Diffusion Probabilistic Models

(DDPM)

November 2020

Score-Based Generative Modeling through Stochastic Differential Equations

May 2021

Diffusion Models Beat GANs on Image Synthesis

(Guided Diffusion)

May 2022

Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding

(Imagen)

March 2024

Scaling Rectified Flow Transformers for High-Resolution Image Synthesis

(Stable Diffusion 3)

The “history” of Diffusion Models

March 2015

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

July 2019

Generative Modeling by Estimating Gradients of the Data Distribution

(NCSN)

June 2020

Denoising Diffusion Probabilistic Models

(DDPM)

13600 citations

November 2020

Score-Based Generative Modeling through Stochastic Differential Equations

May 2021

Diffusion Models Beat GANs on Image Synthesis

(Guided Diffusion)

May 2022

Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding

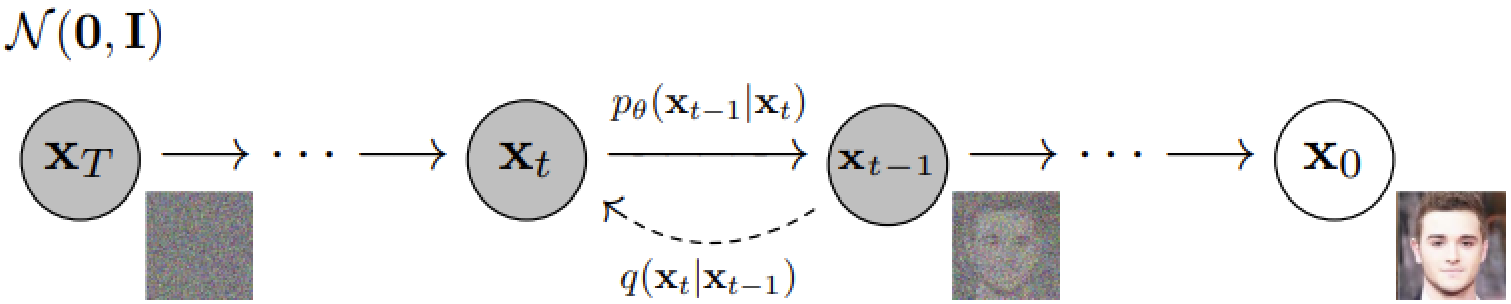
(Imagen)

March 2024

Scaling Rectified Flow Transformers for High-Resolution Image Synthesis

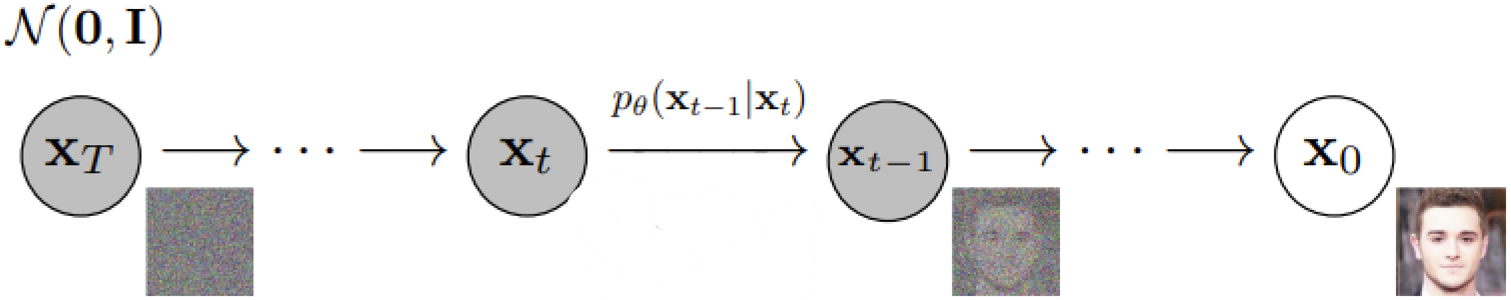
(Stable Diffusion 3)

DDPM


$\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$

DDPM

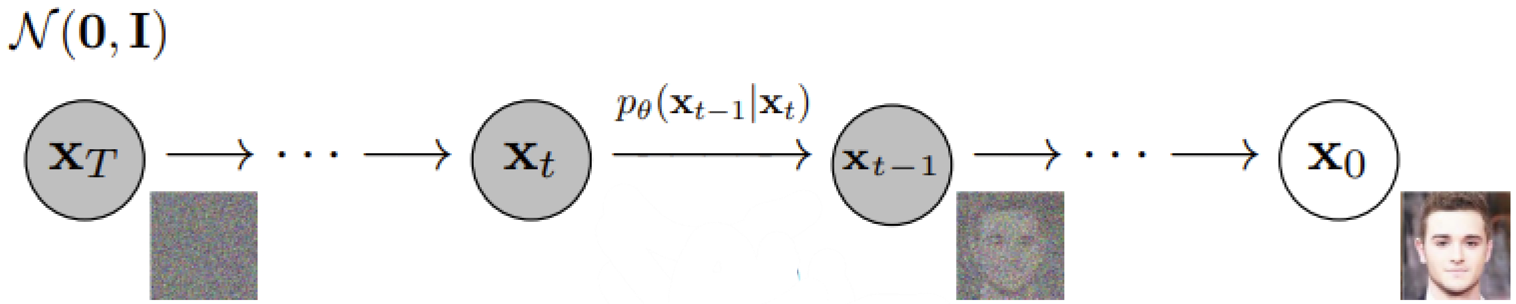
We have some data $x_0^{(1)}, \ldots, x_0^{(N)} \sim q_{data}(x_0)$
We want to sample from the data distribution.



How should we train the neural network?

DDPM

We have some data $x_0^{(1)}, \ldots, x_0^{(N)} \sim q_{data}(x_0)$
We want to sample from the data distribution.



$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right)$$

How should we train the neural network?

DDPM

We have some data $x_0^{(1)}, \ldots, x_0^{(N)} \sim q_{data}(x_0)$
We want to sample from the data distribution.

$$\beta_1, \ldots, \beta_T$$

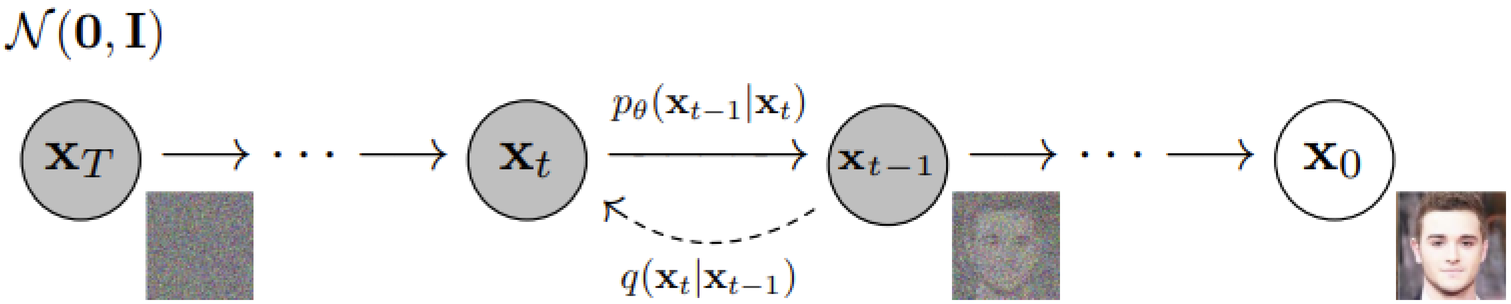
$$\alpha_t := 1 - \beta_t$$

$$\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$$

$$\sigma_t^2 = \beta_t$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$



$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right)$$

How should we train the neural network?

DDPM

We have some data $x_0^{(1)}, \dots, x_0^{(N)} \sim q_{data}(x_0)$
We want to sample from the data distribution.

$$\beta_1, \dots, \beta_T$$

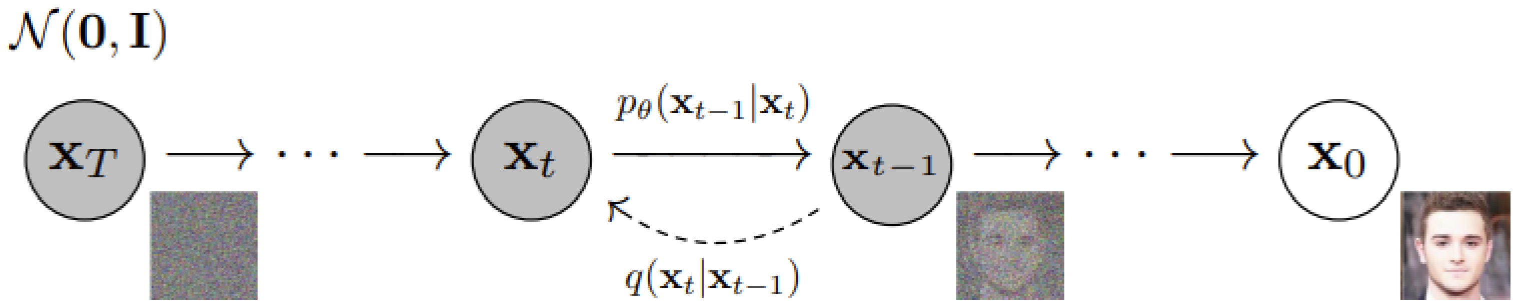
$$\alpha_t := 1 - \beta_t$$

$$\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$$

$$\sigma_t^2 = \beta_t$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$



$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right)$$

How should we train the neural network?

Maximise a lower bound to the log likelihood of the data

$$L_{\text{simple}} = \mathbb{E}_{\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \text{Unif}(1,T)} \left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\underbrace{\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}_{\boldsymbol{x}_t}, t\right)||^2 \right]$$

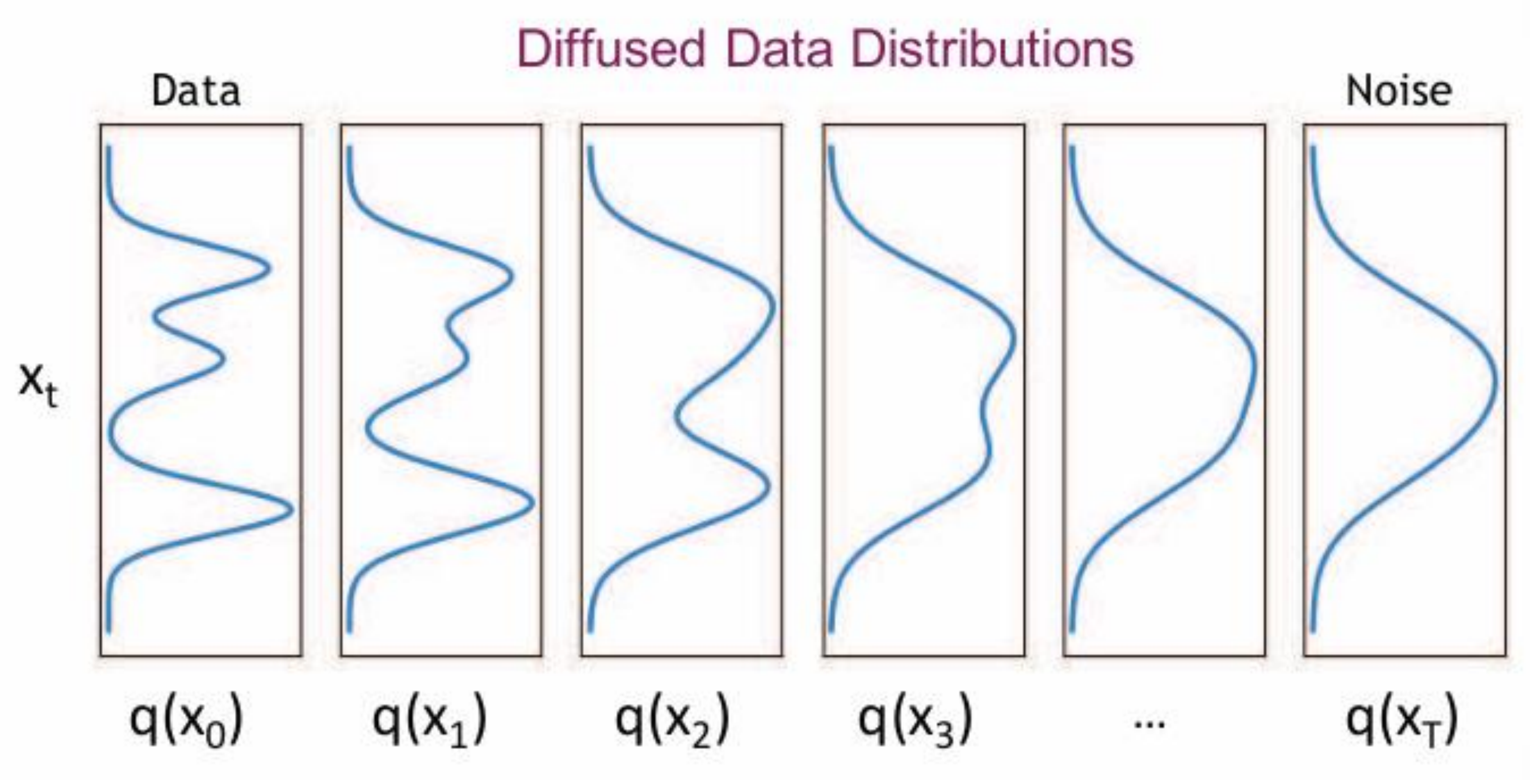


Figure from Probabilistic Machine Learning: Advanced Topics

Contributions of the DDPM paper

- New simple and effective weighted variational lower bound to train diffusion models.
- Hyperparameter choices that lead to good results:
    - Constant forward process variances
    - Scale data to [-1, 1]
    - Train the model to predict noise
    - U-Net with self-attention and group normalisation
    - T = 1000

Time to Code!

What I cannot create, I do not understand.

~Richard Feynman, 1988

Why PyTorch? https://paperswithcode.com/trends

https://colab.research.google.com/drive/1S3NY8Uj5GYwUxE3kAQvaftsvfM1j7Kw1#scrollTo=Qw3n7Fpn4cbc