diffusion

denoising diffusion probabilistic models”。DDPM: https://arxiv.org/abs/2006.11239

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** $\mathbf{x}_0$

Input: noised image + iteration
model output: noise predicted -> denoising

stable diffusion

1. text encoder to vector
2. generation model (diffusion) Denoising U-Net
3. decoder to final version in pixel space parallel training
   mid journey: during training process, illustrates the results encoded from the denoising images

text encoder:

1. gpt coding/ BIRT
2. criteria: CLIP score/ FID-10k

   - **FID**: standard -> pre-trained CNN classification model -> representation ； the distance between the representation of the generated images and the representation of the real images (assumption of Gaussians distribution)
     两组distribution的距离 limitation: need a large scale of generated images

   - **CLIP**: An additional Image Encoder model CLIP score: the vectors similarity between the encoded text and encoded generated image representations

decoder:

feature: Training without knowing the correspondence between images and text intermediate:

- compressed image: sample and downsample -> train
- latent representation: auto-encoder ??
  - input: H*W*3 latent: h*w*c (exceeding vision dimension)

generation model:

input: noised image + text
output: intermediate
text(additional): condition (can be ignored during inferation) 加噪过程，改为加在中间杂序上，使用auto-encoder的encoder部分 train a noise predictor denoising: initialized by sampling normal distribution noise

algorithm

part 1 training

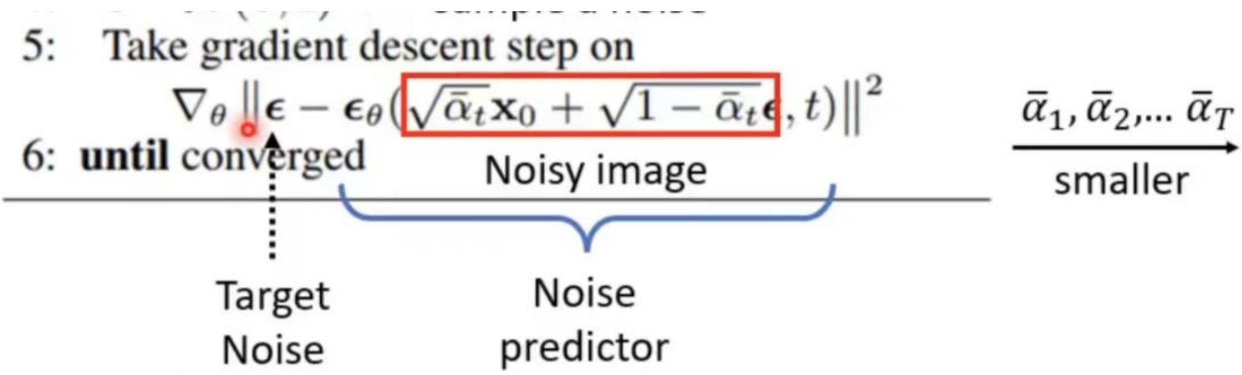
Algorithm 1 Training

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** $\mathbf{x}_0$

loss function during training: 2. xo -> clean images 4. \$\epsilon\$ samples from normal distribution (\$\mu\$ = 0,v = 1) 5.



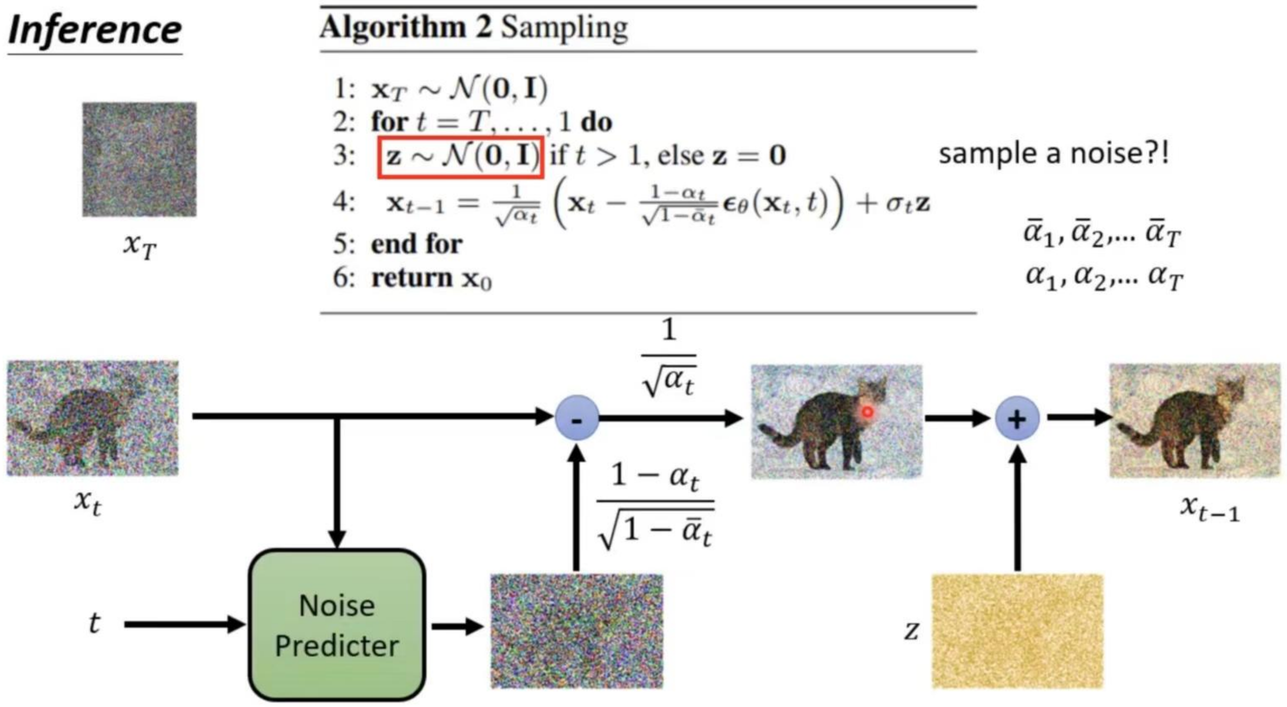
inside: weighted sum, noising
the larger t is, the more proportion the noise added
\$\epsilon_\theta\$: noise predictor input: noiy image + t(step/iteration) output: predicted noise image

compared with the target noise you have sampled in **step 4**

difference with origin steps noise and denoise step by step < DDPM training > predicting the noise by once why?

sample

generate image



strangeness: elinimate the predicted noise and add a new one afterward (plus signal)

theory

map the generated **distribution** to the actual world distribution Q: to measure the similarity of the two- A: maximum likelihood Estimation:(MLE)
$P_\theta(x)<->P_{data}(x)$
sample
all objective for image generation model

KL divergence

KL diverges: 衡量两种分布差异程度 definition：$D_{KL}(P | Q) = \int p(x) \log \left(\frac{p(x)}{q(x)}\right) dx$ 非对称性

VAE encoder

q(z|x) z: distribution (major Gaussians) given the data x (x -> image to imitate)

course:C4 11:30