

生成式模型的发展

VAE-GAN-Diffusion

姓名 李磊

学号 241260275

Huffman compression
哈夫曼编码

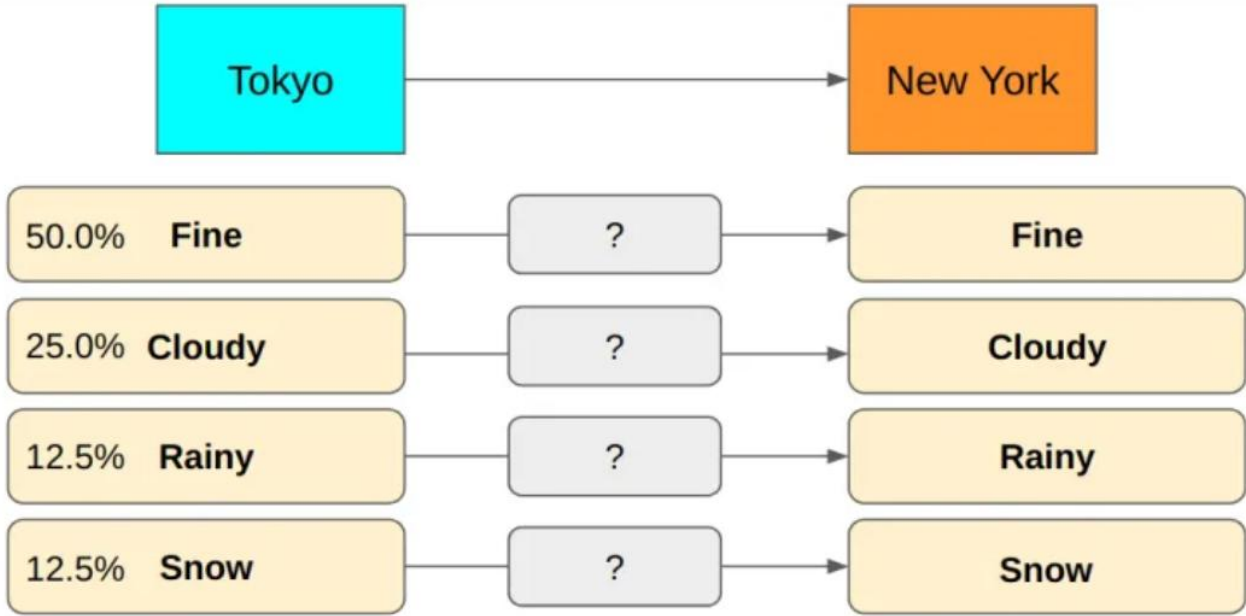
0	1	1	0
---	---	---	---	-------

字符编码原则：

- 前缀不重复（prefix-free）,从任一位置解码

提示：

- 频率相关:高频短，低频长

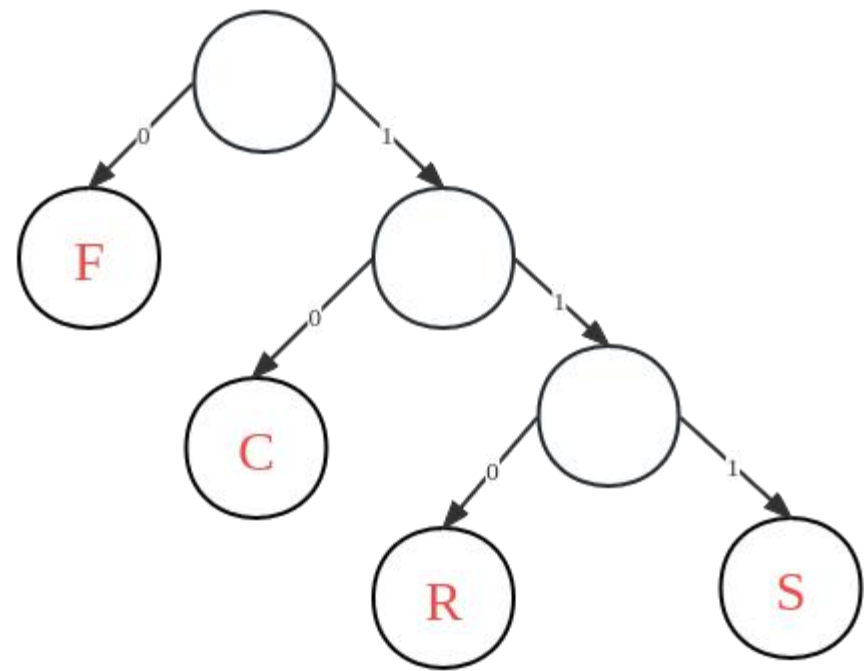


编码方式	F(50.0%)	C(25.0%)	R(12.5%)	S(12.5%)	编码长度期望
方式1	0	1	10	11	error
方式2	10	110	0	111	$2*0.5+3*0.25+1*0.125+3*0.125=2.25$
方式3	0	10	110	111	$1*0.5+2*0.25+3*0.125+3*0.125=1.75$

Huffman compression
哈夫曼编码

编码方式	F(50.0%)	C(25.0%)	R(12.5%)	S(12.5%)	编码长度期望
方式1	0	1	10	11	error
方式2	10	110	0	111	$2*0.5+3*0.25+1*0.125+3*0.125=2.25$
方式3	0	10	110	111	$1*0.5+2*0.25+3*0.125+3*0.125=1.75$

长度



- F : $-\log_2 0.5=1$
- C : $-\log_2 0.25=2$
- R : $-\log_2 0.125=3$
- S : $-\log_2 0.125=3$

信息熵就是最小编码长度期望

$$H(X) = - \sum_{i=1}^b P_i \log_2(P_i)$$

熵
信息熵
交叉熵

熵

- 服从某一特定概率分布事件的理论最小平均编码长度
- 熵在信息学中就是信息熵

$$H(X) = - \sum_{i=1}^b P_i \log_2(P_i)$$

$$H(x) = - \int P(x) \log_2 P(x) dx$$

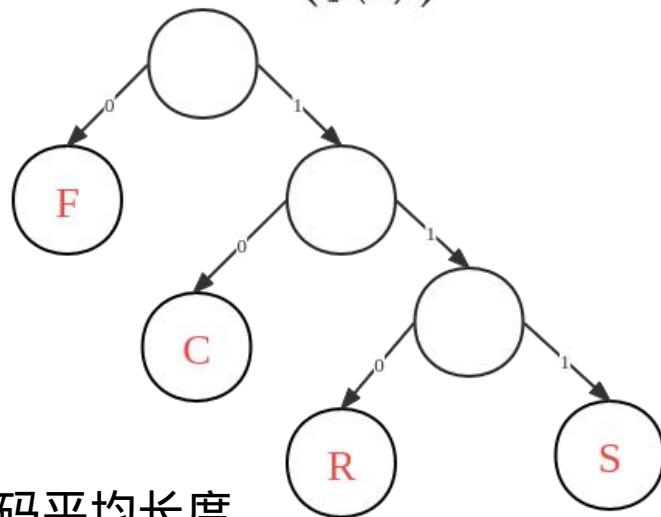
F(50.0%)	C(25.0%)	R(12.5%)	S(12.5%)
----------	----------	----------	----------

把来自一个分布 q 的消息使用另一个分布 p 的最佳代码传达的平均消息长度) 称为交叉熵。

one-hot	F	C	R	S
实际值 q	1	0	0	0
预测值 p	1/2	1/4	1/8	1/8

交叉熵= \sum 概率 * 对应字符长度

$$H_p(q) = \sum_x q(x) \log_2 \left(\frac{1}{p(x)} \right) = - \sum_x q(x) \log_2 p(x)$$



用 p 表示 q 所需最佳编码平均长度

$$H_p(q) = - \{ 1 * \log_2 0.5 + 0 * \log_2 0.25 + 0 * \log_2 0.125 + 0 * \log_2 0.125 \} = 1$$

$$H_q(q) = - \{ 1 * \log_2 1 + 0 * \log_2 0.25 + 0 * \log_2 0.125 + 0 * \log_2 0.125 \} = 0$$

$$H_q(p) = \infty$$

实际值p信息熵

$$H(p) = -\int p \log_2 p dx$$

预测值q信息熵

$$H(q) = -\int q \log_2 q dx$$

预测值q编码实际值p期望长度

$$H(p||q) = -\int p \log_2 q dx$$

KL散度(q编码p - p编码p)

$$KL(p||q) = H(p||q) - H(p)$$

$$= -\int p \log_2 q / p dx$$

p,q均为高斯分布

$$p \sim N(\mu_1, \sigma_1^2), q \sim N(\mu_2, \sigma_2^2)$$

$$\int \exp(-x^2) dx = \pi^{1/2}$$

$$\int x^2 \exp(-x^2) dx = \pi^{1/2} / 2$$

$$H(p(x)) = \int p(x) \log_2 p(x) dx$$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$KL(p, q) = -\int p(x) \log q(x) dx + \int p(x) \log p(x) dx$$

$$= \frac{1}{2} \log(2\pi\sigma_2^2) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} (1 + \log 2\pi\sigma_1^2)$$

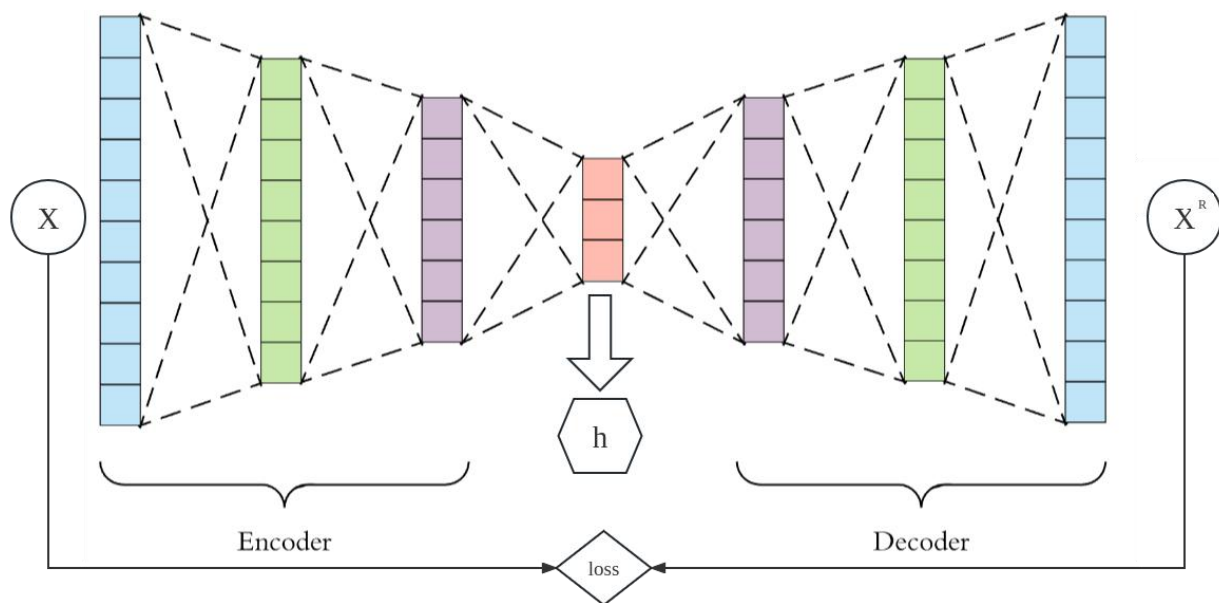
$$= \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

$$q \sim N(0, 1) \quad p(\mu, \delta)$$

$$KL(p, q) = -\log \delta + \delta / 2 + \mu^2 / 2 - 1 / 2$$

```
# D_KL(Q(z|X) || P(z)); calculate in closed form as both dist. are Gaussian
# here we assume that \Sigma is a diagonal matrix, so as to simplify the computation
KLD = 0.5 * torch.sum( torch.exp(log_var) + torch.pow(mu, 2) - 1. - log_var)
# 0.5 * { e^data + mu^2 - 1 - data }
# 0.5 * data + 0.5 * e^data + 0.5 * mu^2 - 0.5
```

AutoEncoder 自编码(AE)



自编码是一种非线性降维方式

- 目标：使隐藏层 h 包含高维输入 X 尽可能多的重要信息。

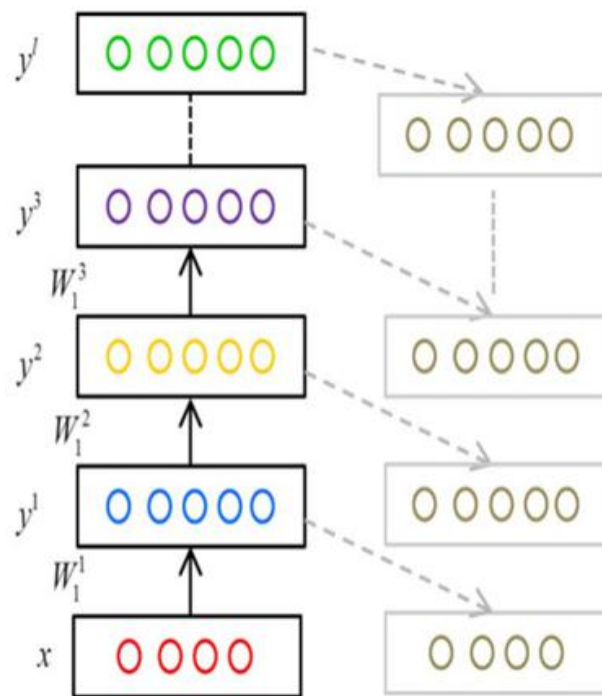
优化目标函数

$$\text{Minimize Loss} = \text{dist}(X, X^R)$$

优点：泛化性强，无监督，不需数据标注

缺点：针对异常识别场景，训练数据需要为正常数据。

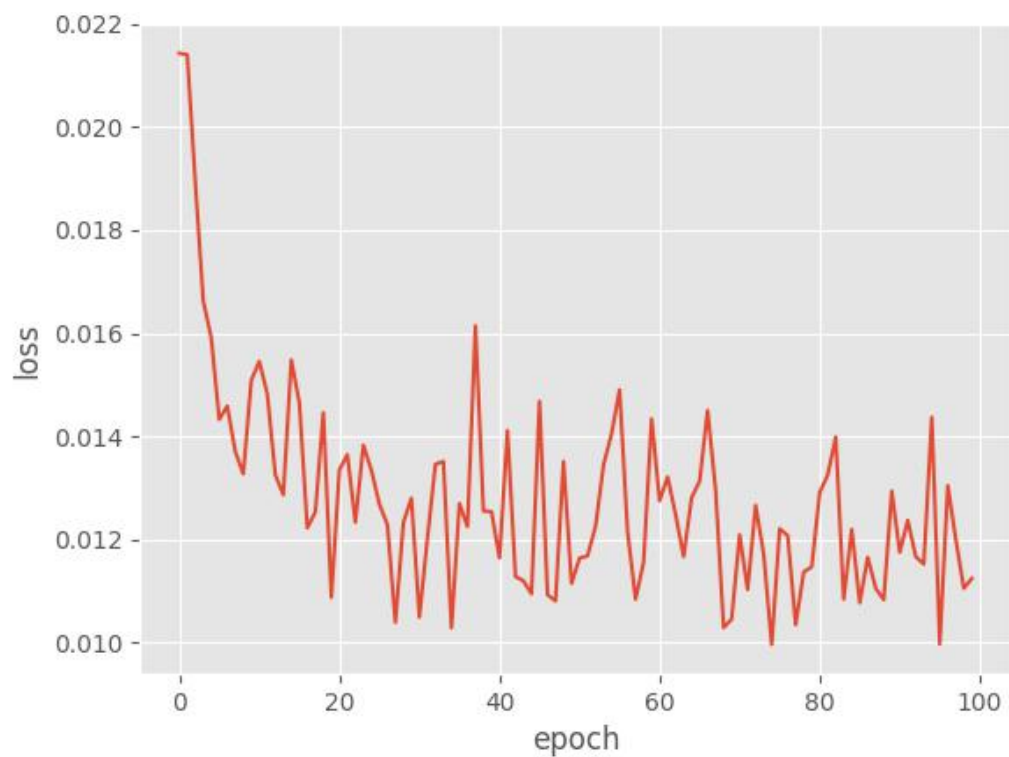
```
AE(  
  (encoder): Sequential(  
    (0): Linear(in_features=784, out_features=256, bias=True)  
    (1): ReLU()  
    (2): Linear(in_features=256, out_features=20, bias=True)  
    (3): ReLU()  
  )  
  (decoder): Sequential(  
    (0): Linear(in_features=20, out_features=256, bias=True)  
    (1): ReLU()  
    (2): Linear(in_features=256, out_features=784, bias=True)  
    (3): Sigmoid()  
  )  
)
```



变分自编码器 (VAE)

AutoEncoder
自编码器

- 实验：手写字母编码与重建
- 数据集：Minst
- 优化目标：原图和重建图均方误差



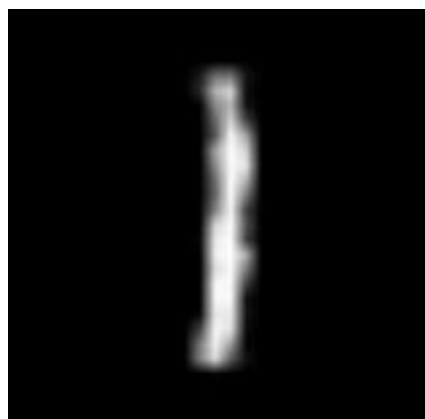
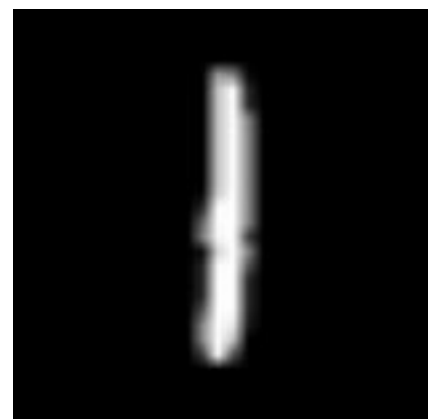
input

output

epoch 0



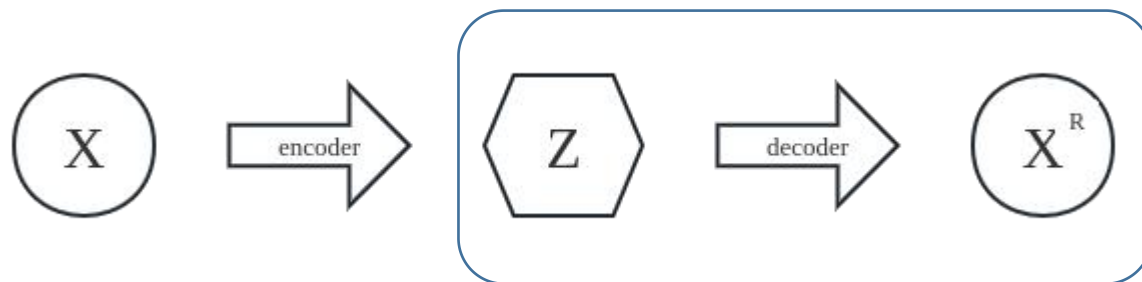
epoch90



变分自编码器 VAE

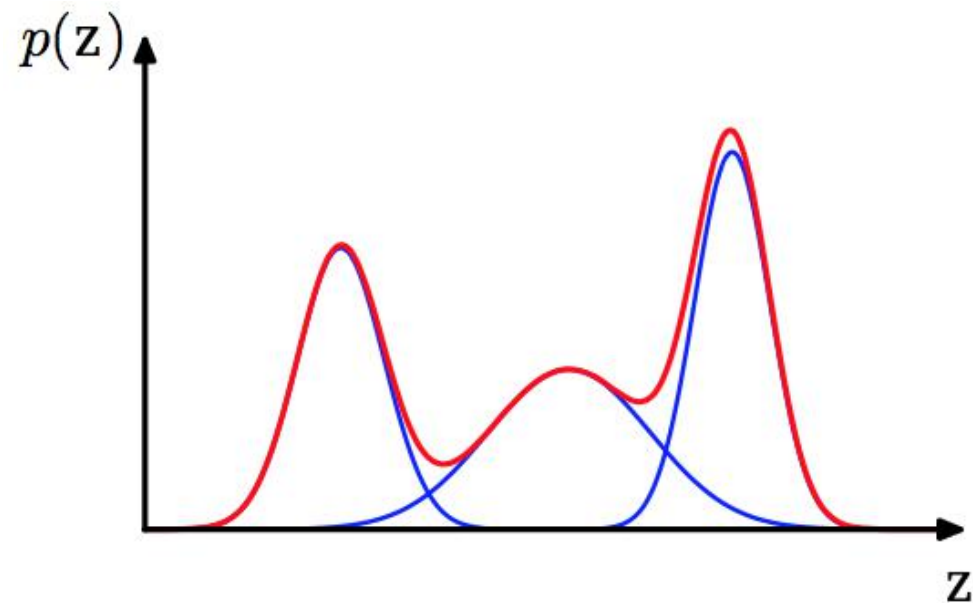
主要目标：构建从隐变量 Z 生成目标数据 X 的模型

假设：样本服从某些常见的分布(eg:正态)



$p(Z)$ ：先验分布，假设为正态分布

$p(X^R|Z)$ ：后验分布,由 Z 来生成 X 的模型



目标1：构建从隐变量 Z 生成目标数据 X 的模型

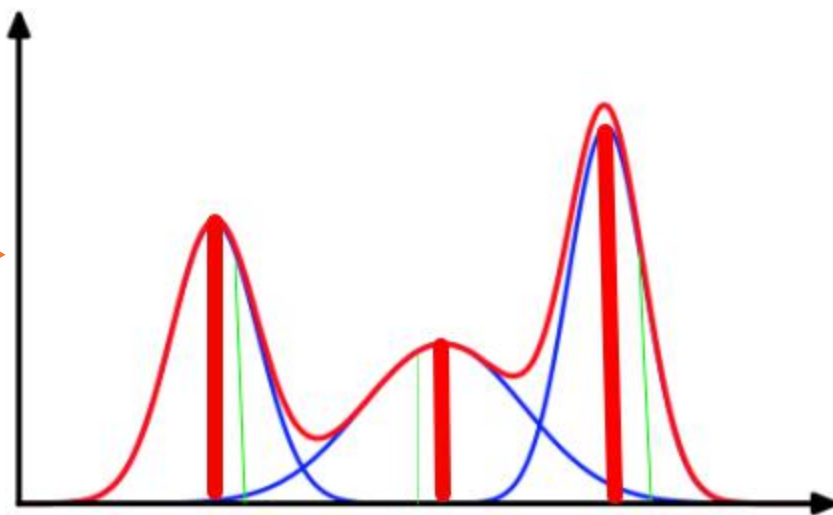
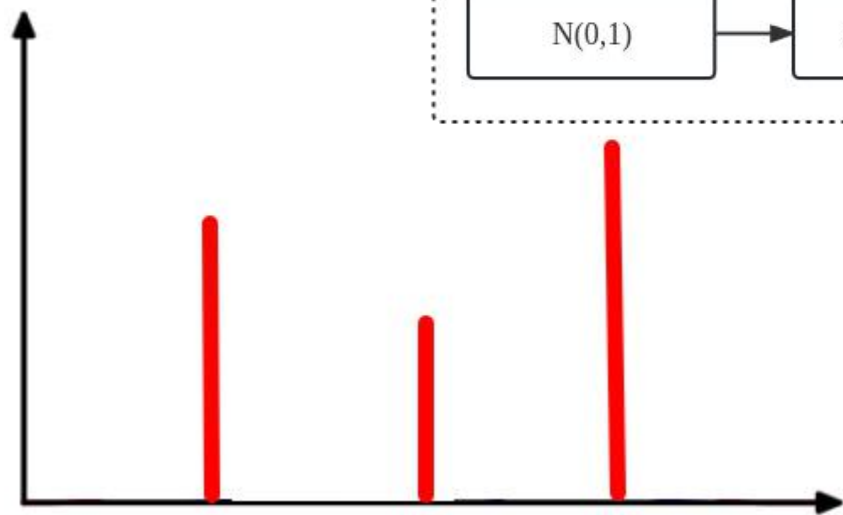
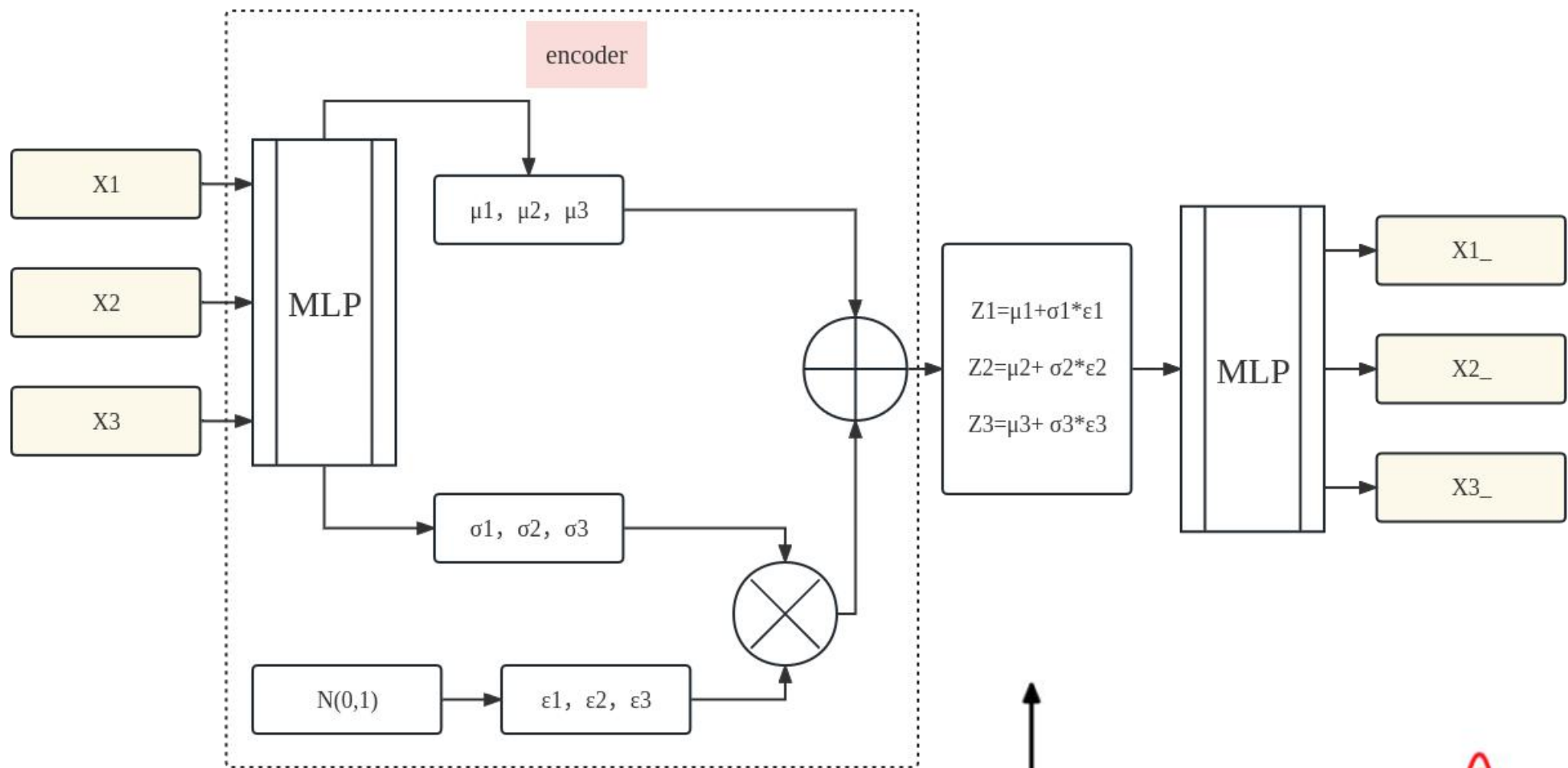
$$\text{Loss1} = \text{BCE}(X, X^R)$$

X 和 X^R 的二值交叉熵

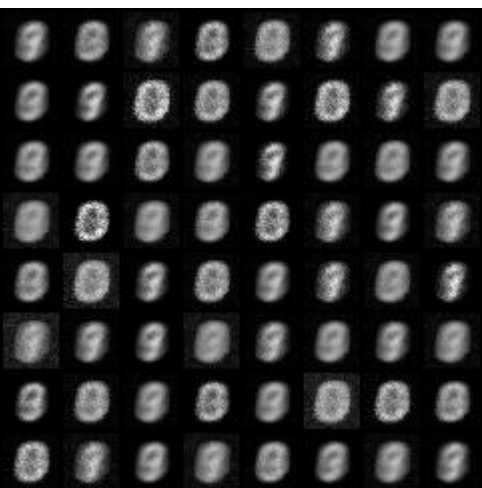
目标2：隐变量 Z 的分布与尽可能符合正态分布
(降低 Z 的KL散度)

KL散度(q编码p - p编码p)

$$\begin{aligned} \text{KL}(p||q) &= H(p||q) - H(p) \\ &= -\int p \log_2 q / p dx \end{aligned}$$



AE



VAE



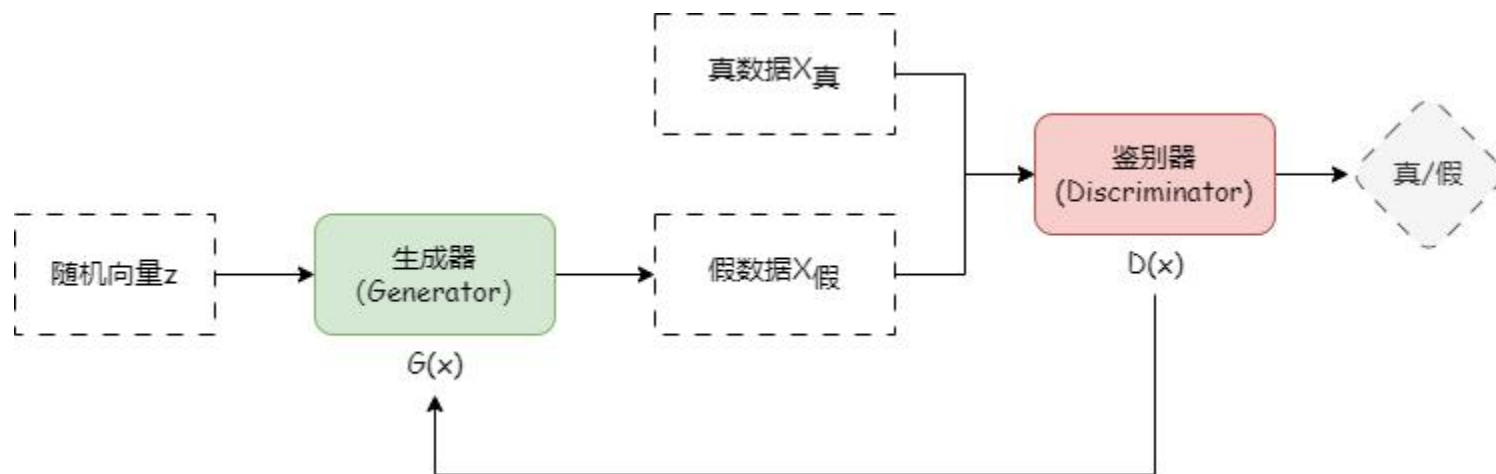
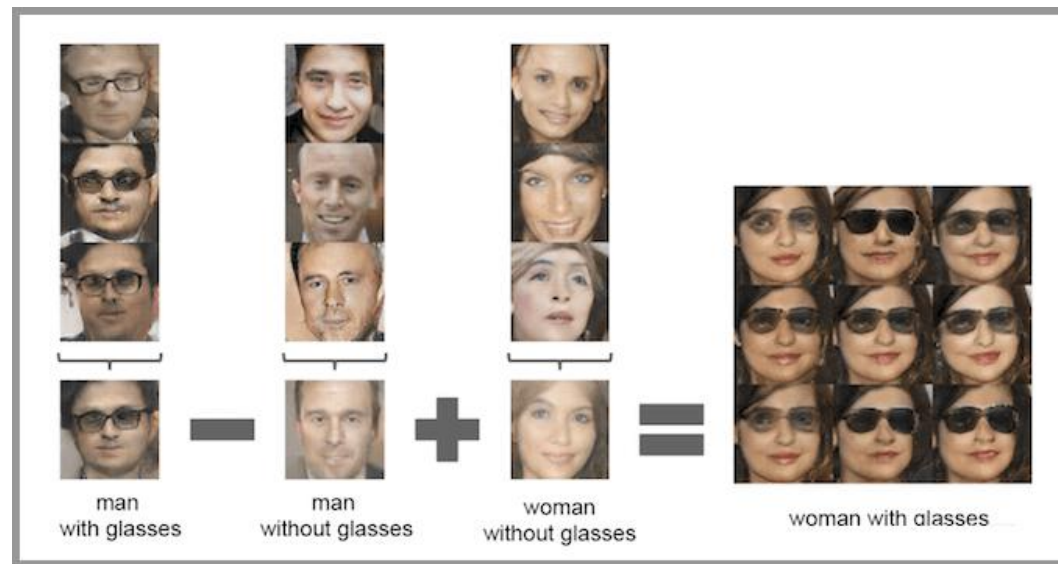
GAN

生成对抗网络(GAN, Generative adversarial network)

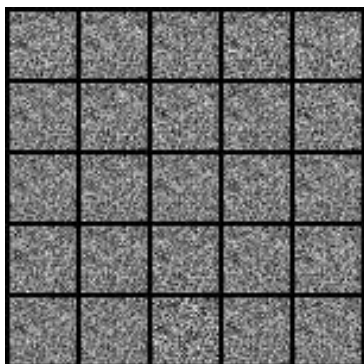
GAN其实是两个网络的组合：

- 生成网络(Generator)：
 - 负责生成模拟数据
 - 不断优化自己生成的数据让判别网络判断不出来
- 判别网络(Discriminator)：
 - 负责判断输入的数据是真实的还是生成的。
 - 优化自己让自己判断得更准确

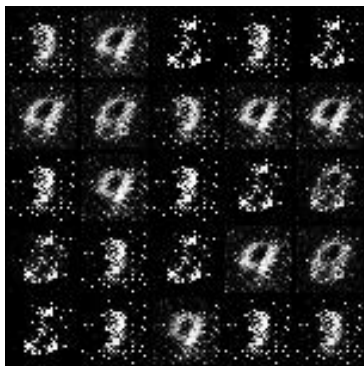
两个网络的关系形成对抗，因此叫对抗网络。



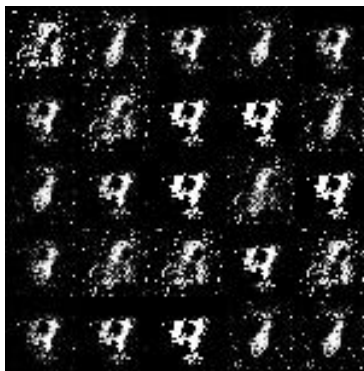
GAN



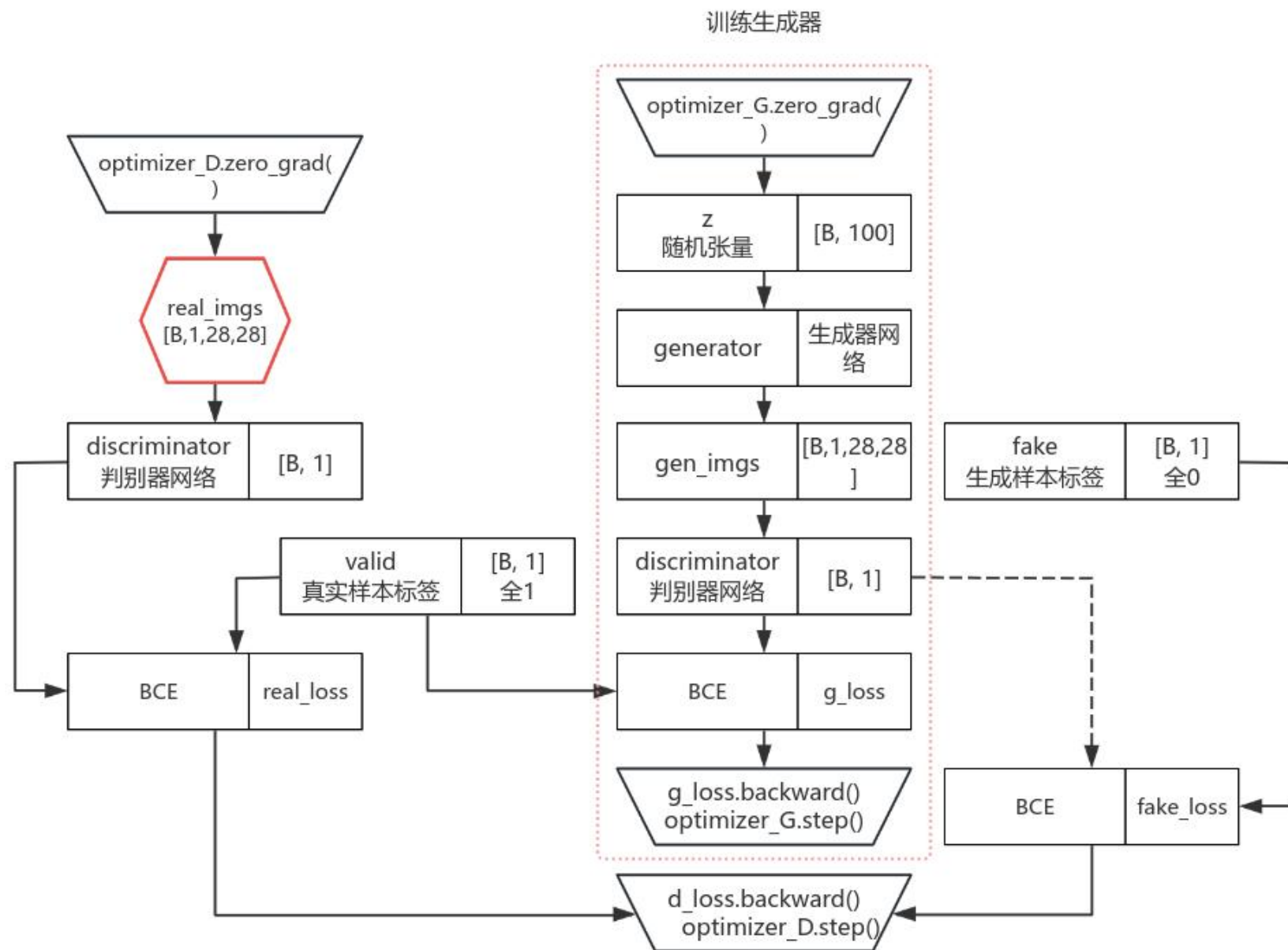
0



2000

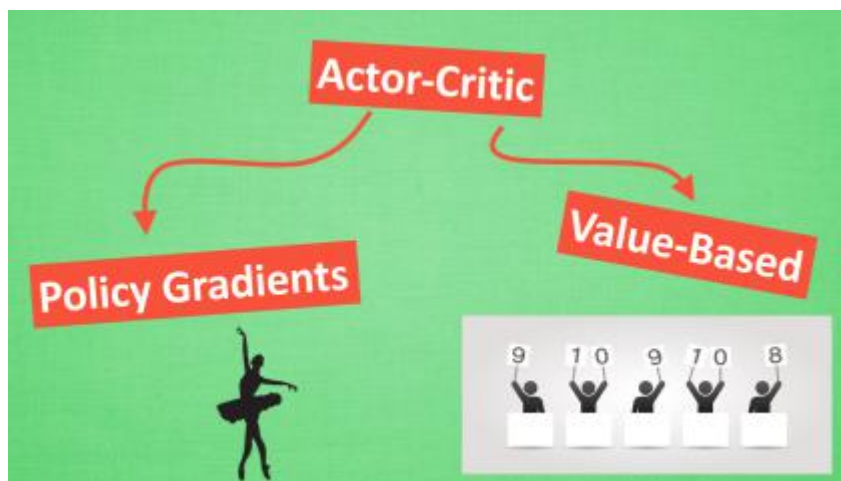


4000

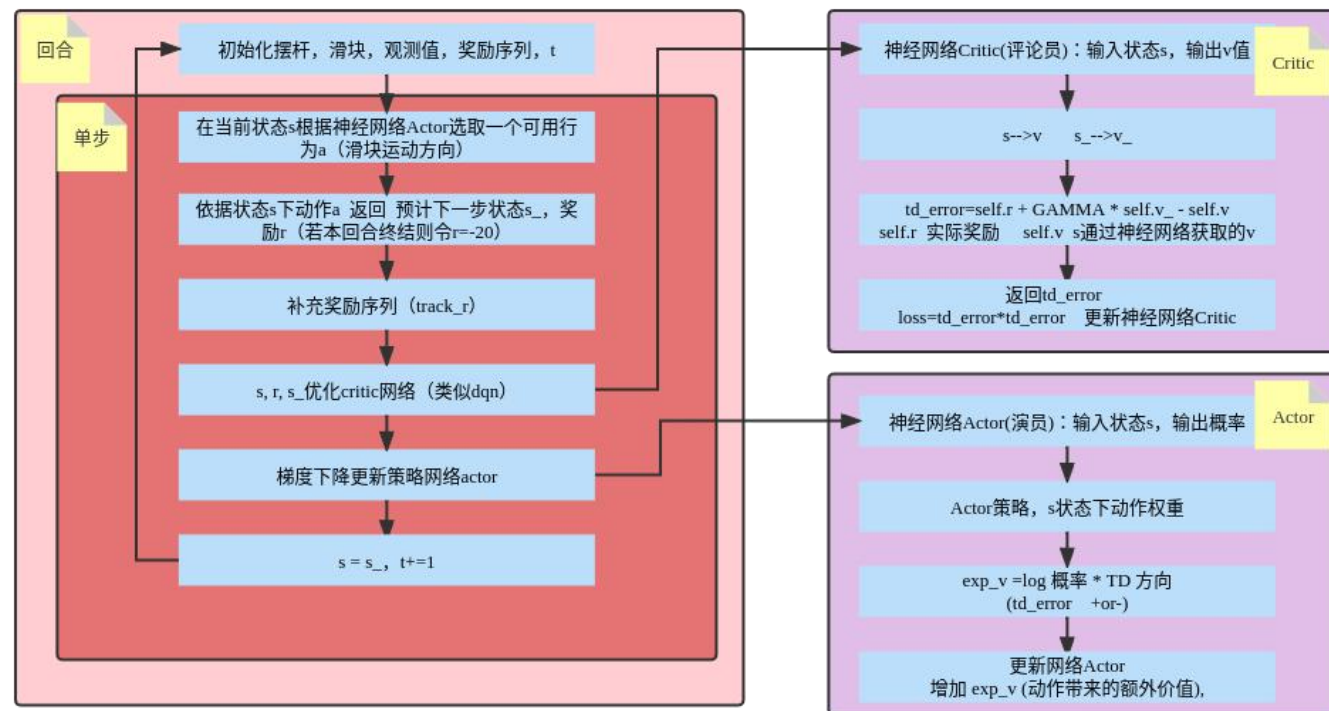


Actor-Critic 演员评论家

Actor (演员) 是策略函数 $\pi_{\theta}(s)$
Critic (评论家) 是值函数 $V^{\pi}(s)$



策略函数Actor，类似于策略梯度算法，但是没有Critic的（使用蒙特卡罗法来计算每一步的价值部分替代Critic的功能）。

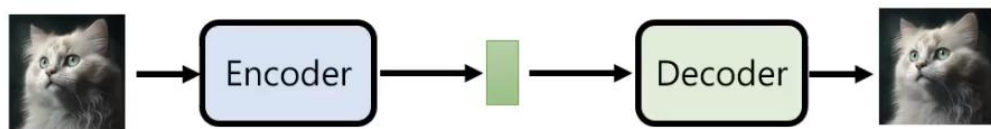


拟合值函数V区别于q表，可以近似做以下理解

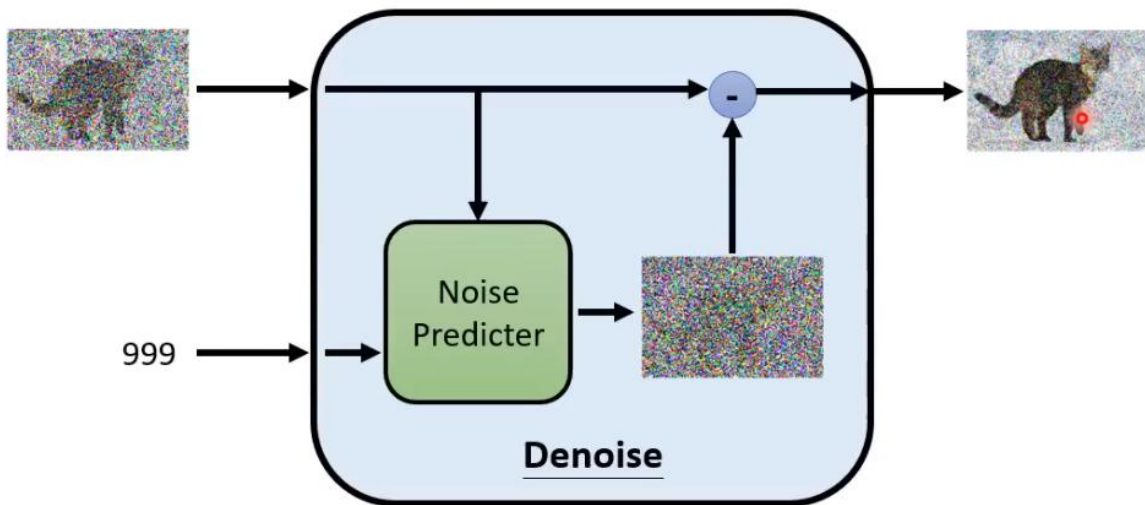
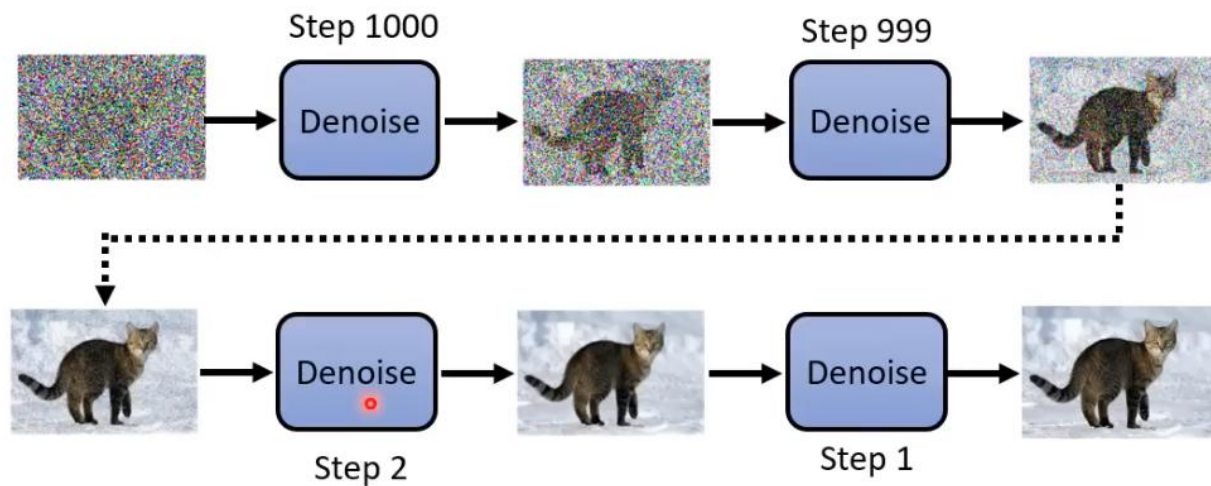
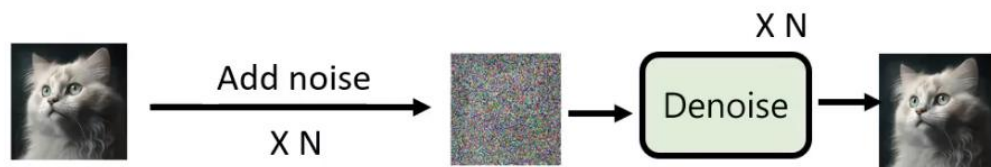
$$Q^{\pi}(s_t, a_t) \approx r(s_t, a_t) + V^{\pi}(s_{t+1})$$

difussion

VAE



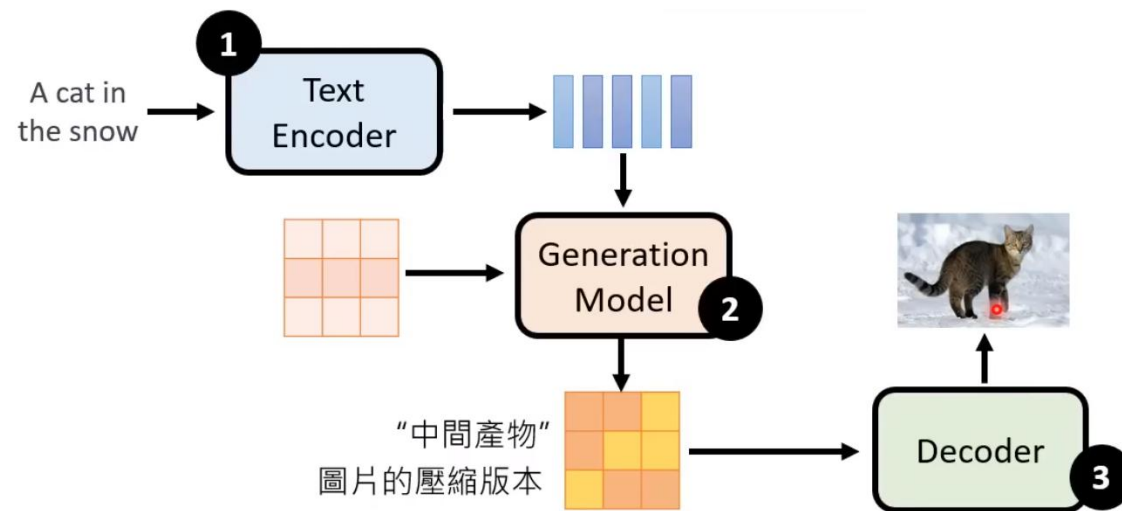
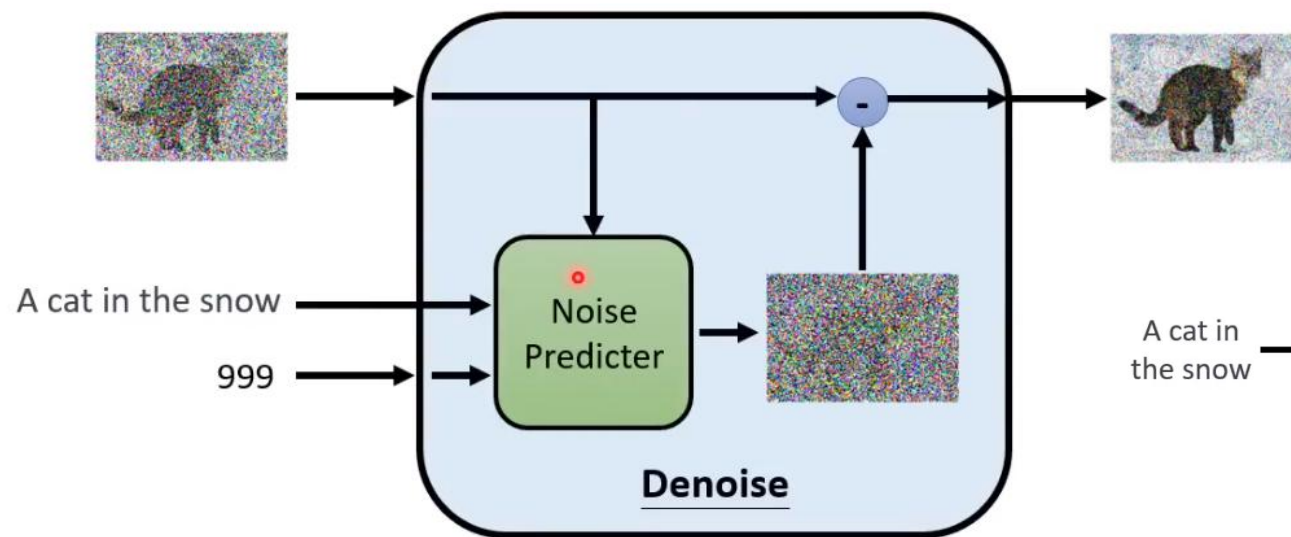
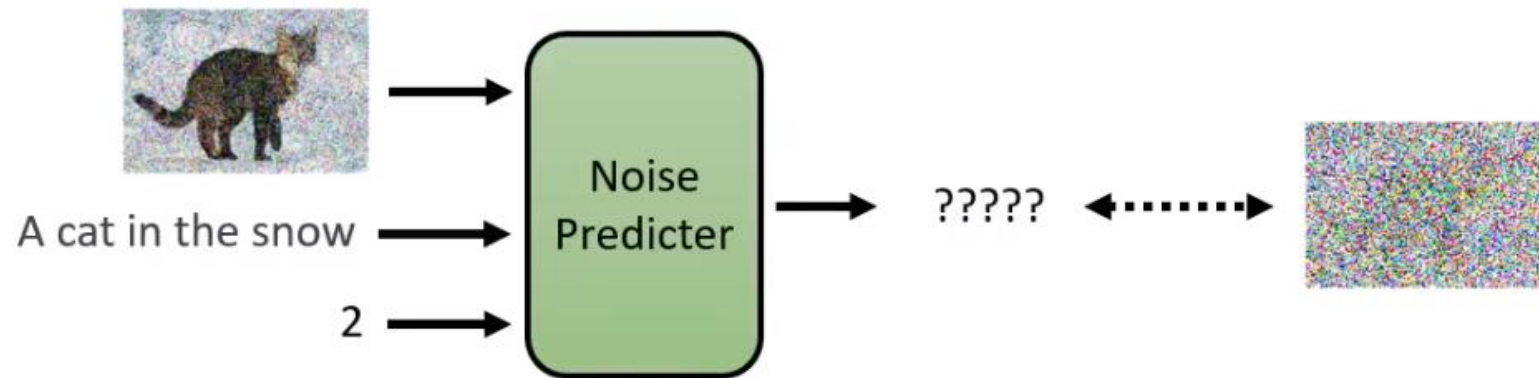
Diffusion



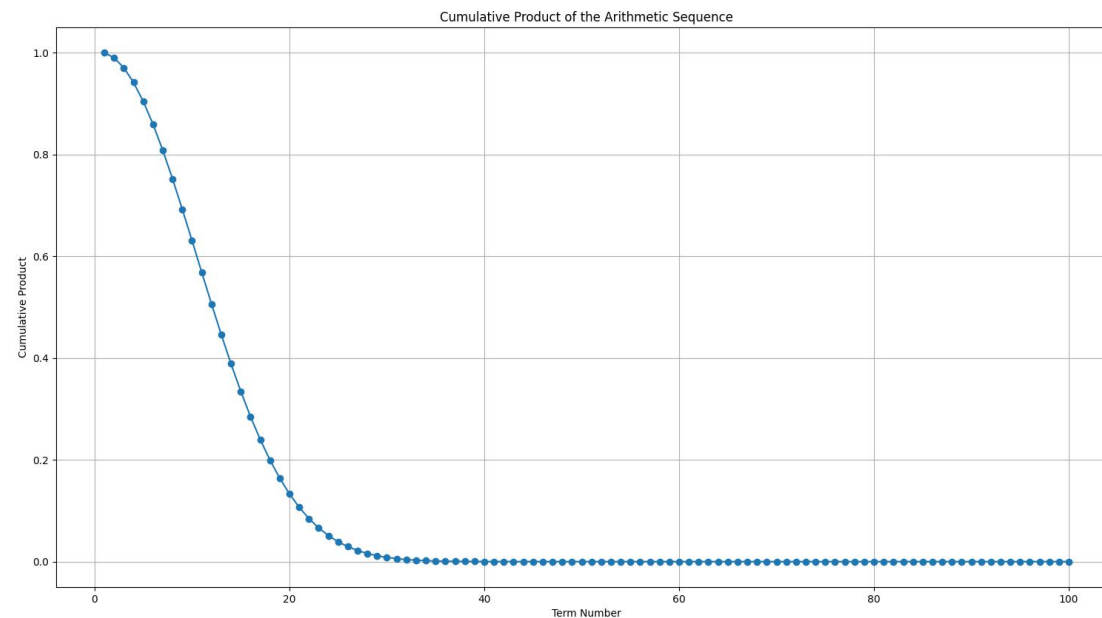
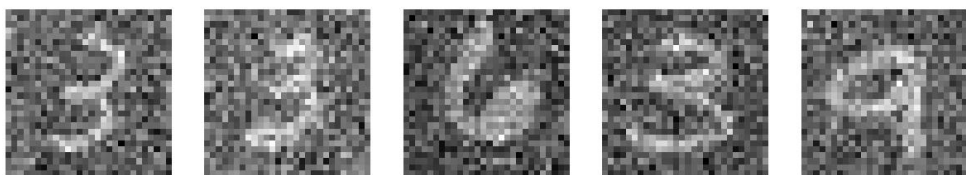
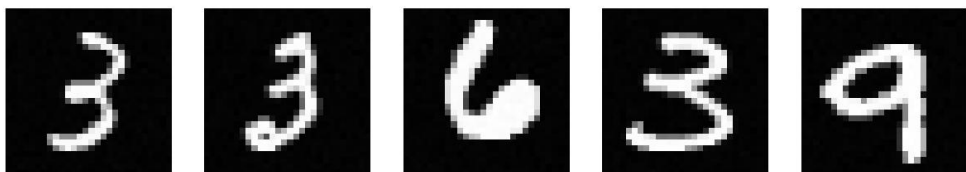
带噪声图+加噪step ---Noise Predictor ---> 噪声
带噪声图-噪声 -----> 新带噪声图（噪声更少）

Net的唯一作用就是预测噪声

diffusion

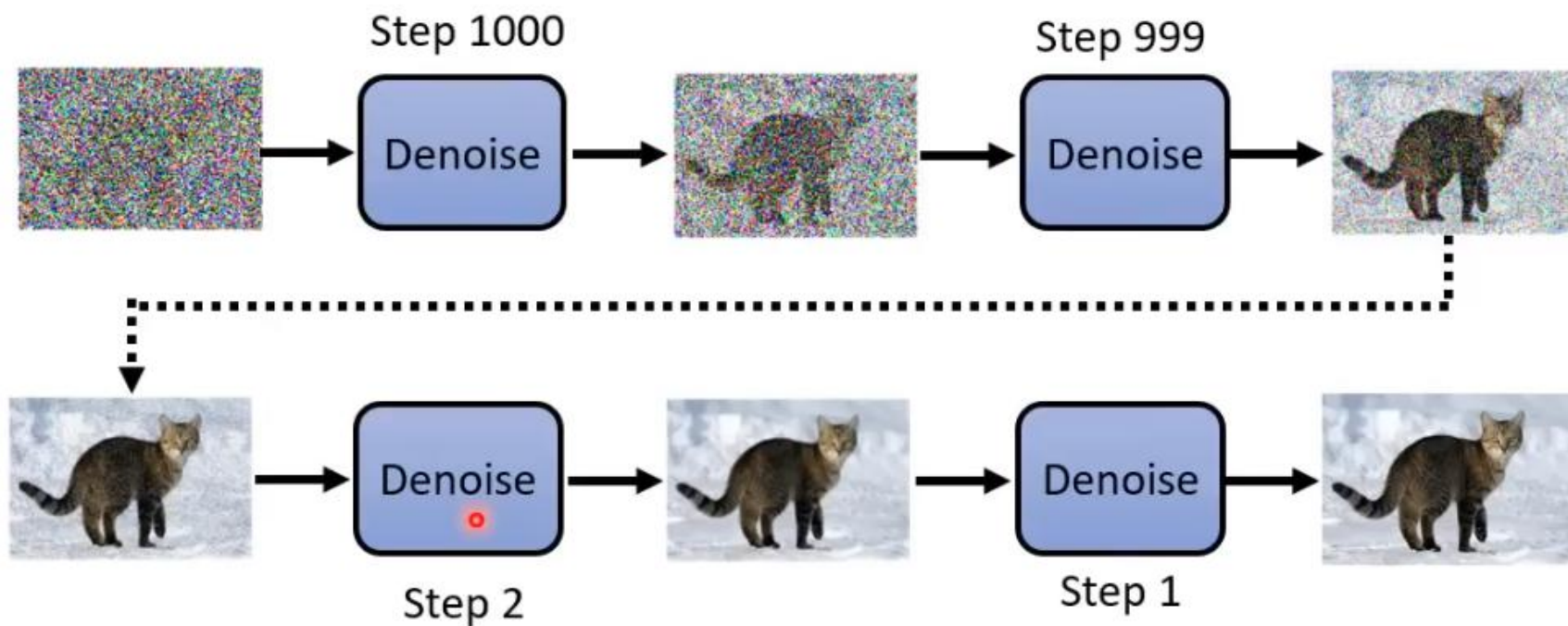


diffusion



β_t 从 $\beta_1 = 10^{-4}$ 到 $\beta_T = 0.02$ 线性增长。

diffusion



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

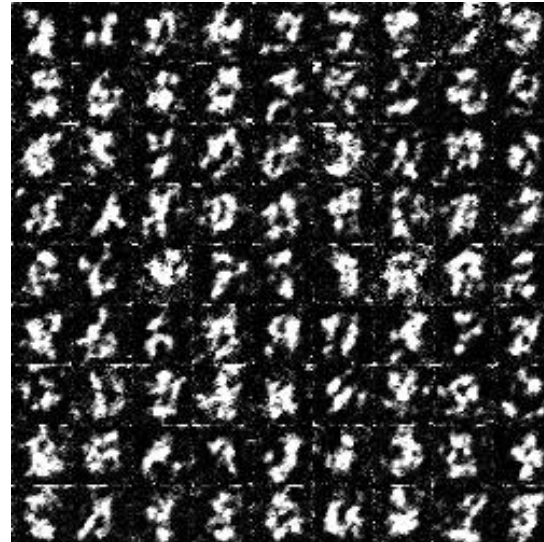
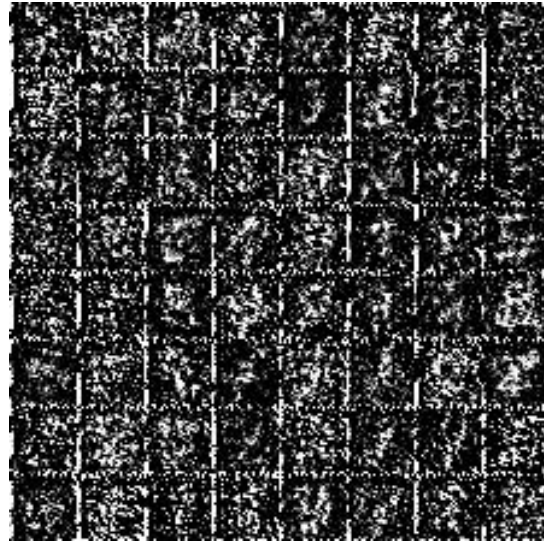
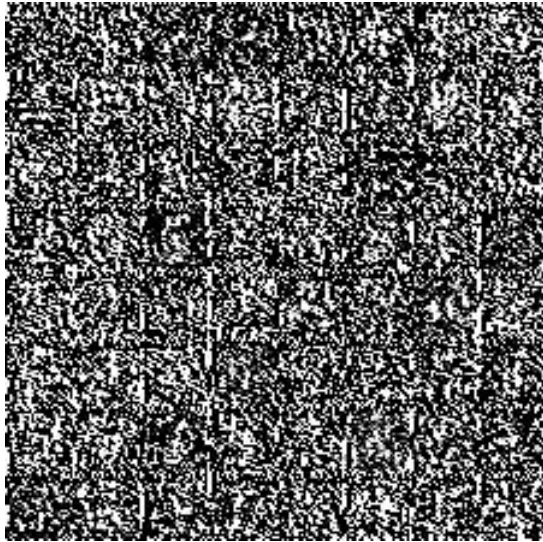
$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I})$$

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \mathbf{x}_0, \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \cdot \beta_t \mathbf{I})$$

其中，只有参数 β_t 是可调的。 $\bar{\alpha}_t$ 是根据 β_t 算出的变量

$$\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{i=1}^t \alpha_{i0}$$

diffusion



谢谢