

河北工业大学

# 毕业设计说明书

作者: 刘鹏 学号: 192241

学院: 人工智能与数据科学学院

系(专业): 计算机科学与技术

题目: 生成模型研究与中国画生成

指导者: 汪鹏 副教授

评阅者:

2023年5月22日

## 毕业设计中文摘要

### 生成模型研究与中国画生成

#### 摘要：

生成模型是人工智能领域的重要研究方向，研究生成模型，有助于人工智能更好地理解现实世界。近年来，生成模型在图像或文字生成方面取得了巨大突破。在图像领域，生成模型生成的图像质量已经可以与真实图像比肩，在文本生成方面，ChatGPT 已经可以与人们自然地交谈。

生成模型是近年来非常活跃的研究领域，对生成模型进行系统回顾总结有助于对其更好地研究。因此，本文对近年来的主要生成模型进行了较为系统地介绍，并根据是否直接定义概率密度函数，将生成模型分为显式密度模型和隐式密度模型。此外，对生成模型在中国画生成上的应用进行探索，使用降噪扩散模型生成中国画。

关键词： 生成模型 中国画 降噪扩散模型

## 毕业设计外文摘要

### Generative Models Research and Chinese Painting Generation

#### ABSTRACT

Generative model is an important research direction in the field of artificial intelligence. The research of generative model is helpful for artificial intelligence to understand the world better. In recent years, the generative model has made great breakthroughs in image and text generation. In the field of image, the quality of the images generated by the generative model can already be comparable to the real image. In terms of text generation, ChatGPT can already talk to people naturally.

Generative model is a very active research field in recent years, Systematic review and summary of generative model is helpful to better study it. Therefore, this paper systematically introduces the main generative models in recent years, and depending on whether the probability density function is directly defined, the generated models are divided into explicit density model and implicit density model. In addition, the application of the generative model in the generation of Chinese painting is explored, and the diffusion model is used to generate Chinese painting.

**Keywords:** Generative Models, Chinese Painting, Denoising Diffusion Model

## 目录

<b>1 絮论</b>	<b>1</b>
1.1 研究背景与意义 . . . . .	1
1.2 国内外研究现状 . . . . .	1
1.2.1 2014–2017 变分自编码器和生成对抗网络时代 . . . . .	1
1.2.2 2018–2019 Transformer 时代 . . . . .	2
1.2.3 2020–2023 大模型时代 . . . . .	2
1.3 研究内容与创新点 . . . . .	3
1.4 论文组织结构 . . . . .	3
<b>2 基本理论与方法</b>	<b>5</b>
2.1 生成模型介绍 . . . . .	5
2.2 显式密度模型 . . . . .	6
2.2.1 可求解密度模型 . . . . .	7
2.2.2 近似估计密度模型 . . . . .	9
2.3 隐式密度模型 . . . . .	26
2.3.1 生成对抗网络 . . . . .	26
2.4 生成模型评价指标 . . . . .	27
2.4.1 图灵测试 . . . . .	28
2.4.2 图像质量评估分数 . . . . .	28
<b>3 降噪扩散模型生成中国画</b>	<b>30</b>
3.1 模型概述 . . . . .	30
3.2 数据集 . . . . .	30
3.3 实验 . . . . .	32
3.3.1 实验 1-完整数据集生成中国画 . . . . .	33
3.3.2 实验 2-浅色中国画数据集大批量训练 . . . . .	34
3.3.3 实验 3-浅色中国画数据集小批量训练 . . . . .	35
3.3.4 实验 4-深色中国画数据集生成中国画 . . . . .	36
<b>结论</b>	<b>37</b>

参考文献	38
致谢	43
附录 A 基础知识	44
A.1 数学 . . . . .	44
A.1.1 线性代数 . . . . .	44
A.1.2 概率论 . . . . .	46
A.1.3 信息论 . . . . .	52
A.1.4 其他 . . . . .	54
A.2 物理学 . . . . .	57
A.2.1 玻尔兹曼分布 . . . . .	57
A.2.2 朗之万动力学 . . . . .	58
A.3 机器学习与深度学习 . . . . .	58
A.3.1 概率图模型 . . . . .	58

# 1 绪论

## 1.1 研究背景与意义

在当今时代，人工智能技术与我们日常生活的联系逐渐紧密。智能手机、智能家居等智能设施让我们的生活更加便利，影视作品、新闻报道中也常常出现与人工智能有关内容。理查德·费曼曾经说过：“我无法创造我所不理解的事物”，对生成模型的研究有助于让人工智能更加“理解”现实世界。中国画<sup>[1]</sup>是我国传统的绘画形式，其在内容和艺术创作上，体现了古人对自然、社会及与之相关联的政治、哲学、宗教、道德、文艺及自然等方面的认识。使用生成模型算法进行中国画创作，有助于继承和弘扬我国优秀传统文化，让中国画在新时代焕发新的光彩。

## 1.2 国内外研究现状

1950 年，艾伦·图灵提出著名的“图灵测试”<sup>[2]</sup>，给出了判定机器是否具有“智能”的试验方法，即机器是否能够模仿人类的思维方式来“生成”内容继而与人交互。某种程度上来说，人工智能从那时起就被寄予了用于内容创造的期许<sup>[3]</sup>。最初，人们尝试通过编码，将具体的规则赋予人工智能，但面临的各种困难使人们转变策略。不再是直接提供人为设定的规则，而是为人工智能提供数据资料，让人工智能从数据资料中“学习”所需的规则，也就是“机器学习”<sup>[4]</sup>。随着计算机运算能力的增加与数据量的增长，在机器学习中，让计算模型通过多个中间层来逐渐理解数据潜在表示的“深度学习”<sup>[5]</sup>，逐渐在计算机视觉、自然语言处理、语音识别等领域取得了突破。

深度学习与生成模型的结合，让人工智能生成内容迎来了新时代，生成内容百花齐放，效果逐渐逼真直至人类难以分辨。

### 1.2.1 2014–2017 变分自编码器和生成对抗网络时代

2013 年，变分自编码器<sup>[6]</sup>的提出使人们认识到，生成模型不仅可以生成简单的数字图像，还可以生成像人脸一样更复杂的图像。生成的图像可以随着隐空间取值的变化而变化。2014 年，生成对抗模型<sup>[7]</sup>的提出，为人们提供了一种通过对抗结构来解决生成模型问题的全新思路。在随后三年内，人们对生成对抗网络进行了扩展与改进，如对基础模型结构进行改进的 DCGAN<sup>[8]</sup>，对损失函数进行改进的 Wassertein GAN<sup>[9]</sup>，对

训练过程进行改进的 ProGAN<sup>[10]</sup>，还有使用生成对抗网络在新的领域进行应用，如图像与图像的转化<sup>[11][12]</sup>以及音乐的生成<sup>[13]</sup>。

在同时期，变分自编码器也有了一些重要的改进，如 VAE-GAN<sup>[14]</sup>和随后的 VQ-VAE<sup>[15]</sup>，以及在强化学习上的应用<sup>[16]</sup>。

在文本生成领域，自回归模型如 LSTMs 和 GRUs 仍然是主流研究方向，相同的自回归思路也应用在图像生成领域，如 PixelRNN<sup>[17]</sup>和 PixelCNN<sup>[18]</sup>引入了一种新的生成图像的方式。也有一些其他生成图像方式的尝试如 RealNVP<sup>[19]</sup>的出现，为之后规范化流模型打下了基础。

2017 年，Transformer<sup>[20]</sup>的提出使生成模型的研究转向 Transformers。

### 1.2.2 2018–2019 Transformer 时代

注意力机制是 Transformer 模型的核心，其使自回归模型不再需要循环层。随着只有 Transformer 解码器的 GPT<sup>[21]</sup>以及只有 Transformer 编码器的 BERT<sup>[22]</sup>的提出，Transformer 很快成为了研究的主流方向。之后，越来越大的模型在很多生成文本的任务上取得了很好的效果。此外，Transformer 也在音乐生成上取得了不错的效果<sup>[23]</sup>。

在这两年，一些重要的生成对抗网络模型的提出也巩固了其在图像生成领域的地位。如 SAGAN<sup>[24]</sup>和 BigGAN<sup>[25]</sup>将注意力机制引入生成对抗模型，取得了很好的效果，StyleGAN<sup>[26]</sup>和随后的 StyleGAN2<sup>[27]</sup>向人们展示了如何控制生成图像的风格与内容。

另一方面，基于分数的模型 NCSN<sup>[28]</sup>的提出，为扩散模型的出现打下了基础。

### 1.2.3 2020–2023 大模型时代

在此时期，一些模型将以往不同种类的模型混合，并在此基础上进一步改进。如 VQ-GAN<sup>[29]</sup>将生成对抗网络的判别器引入 VQ-VAE 模型结构，Vision Transformer<sup>[30]</sup>将 Transformer 应用在图像领域。2022 年，StyleGAN-XL<sup>[31]</sup>对 StyleGAN 的进一步改进，使其能够在更大的数据集上生成更高分辨率的图像。

于 2020 年提出的两个模型——DDPM<sup>[32]</sup>和 DDIM<sup>[33]</sup>，为之后的大规模图像生成模型打下基础。在图像生成领域，突然出现的扩散模型成为了生成对抗网络的竞争对手。扩散模型不仅可以生成高质量图像，相比于生成对抗网络也更容易训练。

在 2020 年同一时期，拥有 1750 亿参数的 GPT-3<sup>[34]</sup>发布，其可以生成关于任何主题的文本内容。在 2022 年，OpenAI 在 GPT-3 基础上发布网页应用 ChatGPT，让用户可以与 ChatGPT 自然地交流。

在 2021 年到 2022 年，一些在大规模语料库上训练的 Transformer 的变体模型也涌

现而出，如微软和英伟达发布的 Megatron-Turing NLG<sup>[35]</sup>，谷歌发布的 LaMDA<sup>[36]</sup>等。

在过去两年，由于 Transformer 系列模型在文本生成领域的成功，以及扩散模型在图像生成领域的应用，人们开始关注多模态内容生成的研究，如使用文本生成图像。如 OpenAI 在 2021 年发布的模型 DALL.E<sup>[37]</sup>，随后的 GLIDE<sup>[38]</sup>和 DALL.E 2<sup>[39]</sup>，谷歌发布的 Imagen<sup>[40]</sup>、Parti<sup>[41]</sup>和 MUSE 以及 Stability AI 发布的 Stable Diffusion<sup>[42]</sup>等。

### 1.3 研究内容与创新点

生成模型算法近年来发展较为迅速，但仍缺少资料对生成模型进行较为系统的探索与研究。如今主流生成模型有规范化流模型、自回归模型、基于能量的模型、变分自编码器、扩散模型和生成对抗网络，各类模型最先进的算法都可以取得很好的效果，但生成模型为什么这样设计，其真正原理是什么，各种生成模型之间有什么关系，理解这些问题有助于设计更好的生成模型，以达到更好的生成效果。本文为首次对各种生成模型原理进行介绍，并将各种框架纳入同一体系的，以有助于研究人员对生成模型进一步研究。

此外，中国画为传统艺术形式，但近年来相比于西方绘画技术的普及，中国画逐渐式微。本文对生成模型在中国画生成方面进行探索，以期望使用人工智能技术生成高质量中国画，继承与弘扬中华传统文化。

### 1.4 论文组织结构

第一章首先介绍研究背景与意义，随后介绍早期人工智能生成模型情况，再将近年来生成模型发展分为三个阶段，2014-2017 为变分自编码器和生成对抗网络时代，2018-2019 为 Transformer 时代 2020-2023 为大模型时代。随后介绍研究内容与创新点与论文组织结构。

第二章首先介绍生成模型算法基本理论与生成模型用途，再根据如何表示或近似似然函数将生成模型分为三类，分别为可求解的显示密度模型、近似估计的显示密度模型和隐式密度模型。其中可求解的显示密度模型与近似估计的显示密度模型都属于显式密度模型。如图1.1所示，可求解密度模型包含规范化流模型和自回归模型，近似估计密度模型包含基于能量的模型、变分自编码器和扩散模型。[2.2](#)对各显式密度模型进行了详细介绍，[2.3](#)对隐式密度模型生成对抗网络进行了详细介绍，最后，介绍了生成模型评价方法。

第三章使用扩散模型进行中国画生成，根据所使用数据集与训练超参数不同，生

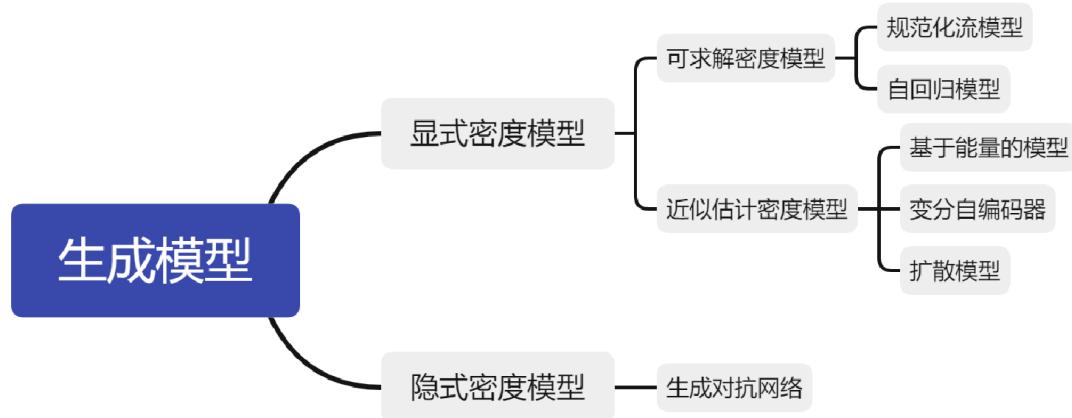


图 1.1 生成模型分类

成不同类型的中国画。

最后对各生成模型算法与中国画生成进行总结与展望。

此外，为保持文中推理证明的完整性，在附录中添加证明所用基础知识。

## 2 基本理论与方法

### 2.1 生成模型介绍

生成模型可以通过在数据集上训练，学习数据集中样本的概率分布 $p_{data}$ ，用生成模型所获得的 $p_{model}$ 来近似 $p_{data}$ ，训练完成获得 $p_{model}$ 之后，可以通过从 $p_{model}$ 中采样来生成与数据集中相似的样本。

生成模型不仅仅可以应用在近些年有较大突破的图片生成与聊天机器人，还包括很多更广泛的应用<sup>[43]</sup>，如：

- 生成模型的训练和采样，有助于表示和运用高维度概率分布。高维度概率分布在数学和工程领域应用广泛。
- 生成模型可以与强化学习相结合，如进行未来决策或对环境的模拟。
- 生成模型可以辅助半监督学习，如对无标签数据进行标注。
- 生成模型可以应用于多模态领域，如对同一输入，不仅仅生成文字，同时也可以生成图像。
- 从本质上来说，很多任务都需要从一些概率分布中进行采样，如提高图像分辨率，艺术创作，图像与图像之间的转换等。

很多生成模型应用极大似然估计的原理。极大似然估计的基本思想是，定义一个由参数 $\theta$ 确定的对概率分布的估计 $p_{model}(\mathbf{x}; \theta)$ ，之后，对训练集定义似然函数

$$\prod_{i=1}^m p_{model}(\mathbf{x}^{(i)}; \theta) \quad (2.1)$$

式2.1中， $m$ 为该训练集样本数， $\mathbf{x}^{(i)}$ 为训练集中样本。

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^m p_{model}(\mathbf{x}^{(i)}; \theta) \quad (2.2)$$

$$= \arg \max_{\theta} \log \prod_{i=1}^m p_{model}(\mathbf{x}^{(i)}; \theta) \quad (2.3)$$

$$= \arg \max_{\theta} \sum_{i=1}^m \log p_{model}(\mathbf{x}^{(i)}; \theta) \quad (2.4)$$

选取参数 $\theta^*$ 使似然函数式2.1，取得最大值，为方便计算，相比于将原似然函数进行最大化，可以将其转化之对数空间，这样可以将求积变为求和。在式2.3中，应用了对数函数为单调递增函数，不改变参数最大值取值的性质。获得的生成模型即为 $p_{model}(\mathbf{x}; \theta)$ 。

图2.1展示了二维极大似然估计过程，极大似然估计从训练集中采样，增大样本所在位置概率的同时保证概率密度总积分为 1。

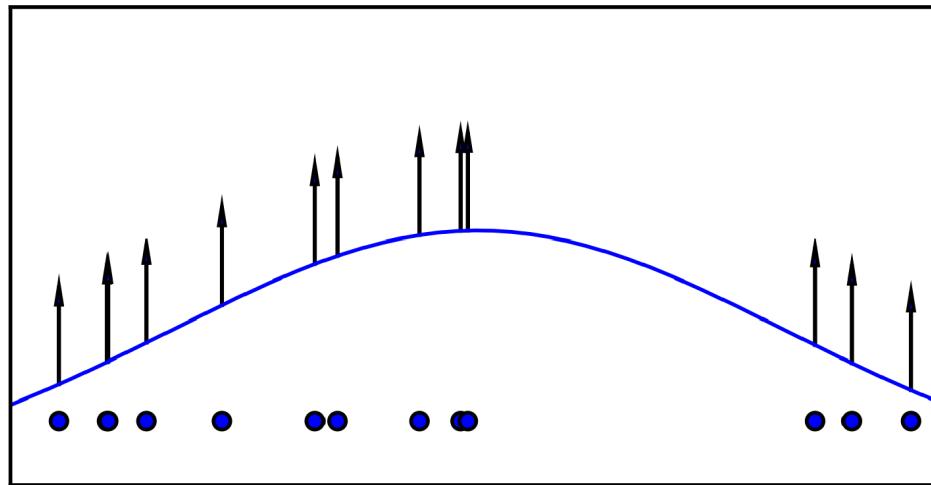


图 2.1 一维极大似然估计过程

从另一方面来说，对参数 $\theta^*$  进行极大似然估计与最小化生成模型与样本数据的实际概率分布的 KL 散度等价，即：

$$\theta^* = \arg \max_{\theta} D_{KL}(p_{data}(\mathbf{x}) || p_{model}(\mathbf{x}; \theta)) \quad (2.5)$$

可以根据如何表示或近似似然函数对生成模型进行分类。具体地说，主要可以分为以下三类：

- 可求解的显示密度模型，直接定义概率密度函数 $p_{model}(\mathbf{x}; \theta)$ ，且要求其似然函数可直接求解。如规范化流模型和自回归模型。
- 近似估计的显示密度模型，直接定义概率密度函数 $p_{model}(\mathbf{x}; \theta)$ ，且要求其似然函数可求得近似解。如基于能量的模型、变分自编码器和扩散模型。
- 隐式密度模型，不定义概率密度函数，而通过其他间接方式对概率分布 $p_{data}$  进行学习。如生成对抗网络。

## 2.2 显式密度模型

显式密度模型直接定义概率密度函数 $p_{model}(\mathbf{x}; \theta)$ ，之后即可以根据极大似然估计进行模型拟合。显式密度模型存在的主要问题是：很难设计既可以表示样本分布又容易求解的模型。有两种方法可以解决这个问题：

1. 设计易求解的模型结构，在此基础上提高模型表达能力，即可求解密度模型。
2. 设计模型后，通过近似方法求解似然函数，即近似估计密度模型。

### 2.2.1 可求解密度模型

#### 规范化流模型

规范化流模型通过一系列可逆变换方程，将简单的概率分布逐渐转化为复杂的概率分布，以希望能够拟合数据样本的概率分布。

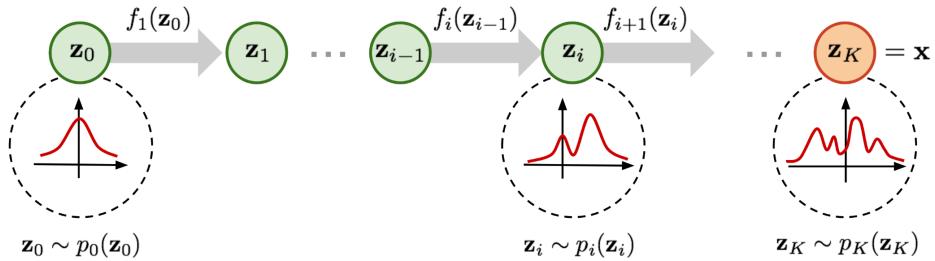


图 2.2 规范化流模型示意图

如图2.2所示，

$$z_{i-1} \sim p_{i-1}(z_{i-1}) \quad (2.6)$$

$$z_i = f_i(z_{i-1}) \quad (2.7)$$

由反函数定理，

$$z_{i-1} = f_i^{-1}(z_i) \quad (2.8)$$

由附录式A.87，

$$p_i(z_i) = p_{i-1}(f_i^{-1}(z_i)) \left| \det \frac{df_i^{-1}}{dz_i} \right| \quad (2.9)$$

为获得 $p_i(z_i)$  与 $p_{i-1}(z_{i-1})$  之间的关系，对式2.9进行变形：

$$p_i(z_i) = p_{i-1}(f_i^{-1}(z_i)) \left| \det \frac{df_i^{-1}}{dz_i} \right| \quad (2.10)$$

$$= p_{i-1}(z_{i-1}) \left| \det \left( \frac{df_i}{dz_{i-1}} \right)^{-1} \right| \quad (\text{根据式A.88}) \quad (2.11)$$

$$= p_{i-1}(z_{i-1}) \left| \det \frac{df_i}{dz_{i-1}} \right|^{-1} \quad (\text{根据式A.10}) \quad (2.12)$$

对式2.12进行对数化得：

$$\log p_i(z_i) = \log p_{i-1}(z_{i-1}) - \log \left| \det \frac{df_i}{dz_{i-1}} \right| \quad (2.13)$$

连续应用式2.13，对 $x$  不断变换可得其关于简单概率分布变量 $z$  的表达式，即：

$$x = z_K = f_K \circ f_{K-1} \circ \cdots \circ f_1(z_0) \quad (2.14)$$

$$\log p(\mathbf{x}) = \log p_K(\mathbf{z}_K) = \log p_{K-1}(\mathbf{z}_{K-1}) - \log \left| \det \frac{df_K}{d\mathbf{z}_{K-1}} \right| \quad (2.15)$$

$$= \log p_{K-2}(\mathbf{z}_{K-2}) - \log \left| \det \frac{df_{K-1}}{d\mathbf{z}_{K-2}} \right| - \log \left| \det \frac{df_K}{d\mathbf{z}_{K-1}} \right| \quad (2.16)$$

$$= \dots \quad (2.17)$$

$$= \log p_0(\mathbf{z}_0) - \sum_{i=1}^K \log \left| \det \frac{df_i}{d\mathbf{z}_{i-1}} \right| \quad (2.18)$$

所谓规范化流模型中的流指的是系列随机变量之间的代换，即不断应用  $\mathbf{z} = f_i(\mathbf{z}_{i-1})$ 。规范化流是指由一系列分布  $p_i$  组成的完整链式过程<sup>[44]</sup>。

由上述计算过程，转换函数  $f_i$  需要满足两个条件：

- 具备可逆性质；
- 雅可比矩阵容易计算。

### 自回归模型

自回归模型将生成问题简化成顺序问题，即通过之前的顺序值来预测下一个值。一般来说，自回归模型对于高维数据  $x$  将其联合概率分布化为条件概率的乘积的形式：

$$p(\mathbf{x}) = p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i | x_1, \dots, x_{i-1}) \quad (2.19)$$

对条件概率的模拟比直接对联合概率分布进行建模更加容易。

具体而言，为便于求解可以假定每一个变量只依赖于不超过一定数量的变量，比如两个变量，即：

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)\prod_{d=3}^D p(x_d|x_{d-1}, x_{d-2}) \quad (2.20)$$

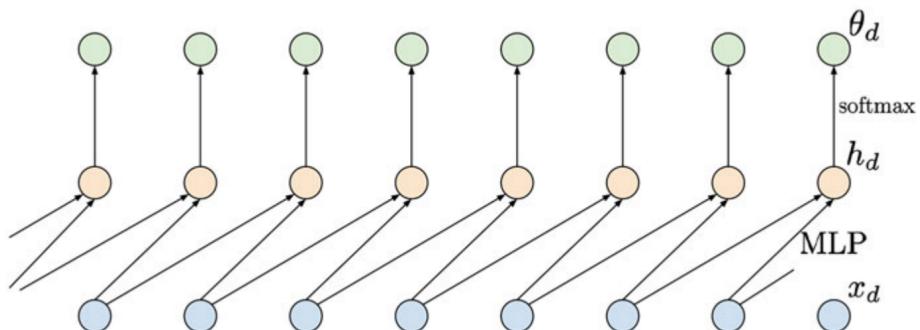


图 2.3 自回归模型依赖两变量示意图

可以使用多层感知机预测  $x_d$  的概率分布，图2.3表示使用多层感知机进行预测  $x_d$  的概率分布。最下面的蓝色结点表示输入，中间层橘色结点表示多层感知机对前两个数据

处理后的输出，最后绿色点表示通过归一化指数函数输出的概率  $p(x_d|x_{d-1}, x_{d-2})$ ，即  $\theta_d$ 。但是，这种假设，即每一个变量只依赖于不超过一定数量的变量，对模型造成了很大的限制。通过循环神经网络可以对以往信息进行更长期的保存<sup>[45]</sup>，即：

$$p(x_d|\mathbf{x}_{<d}) = p(x_d|RNN(x_{d-1}, h_{d-1})) \quad (2.21)$$

式2.21中， $h_d = RNN(x_d, h_{d-1})$ ， $h_d$  可以看作对所有历史信息的保存，可称为隐环境。图2.4表示使用循环神经网络 RNN 预测  $x_d$  的概率分布。最下面的蓝色结点表示输入，中间层橘色结点表示循环神经网络对前两个输入数据与隐环境处理后的输出，最后绿色结点表示通过归一化指数函数后输出的概率  $p(x_d|x_{d-1}, x_{d-2})$ ，即  $\theta_d$ 。

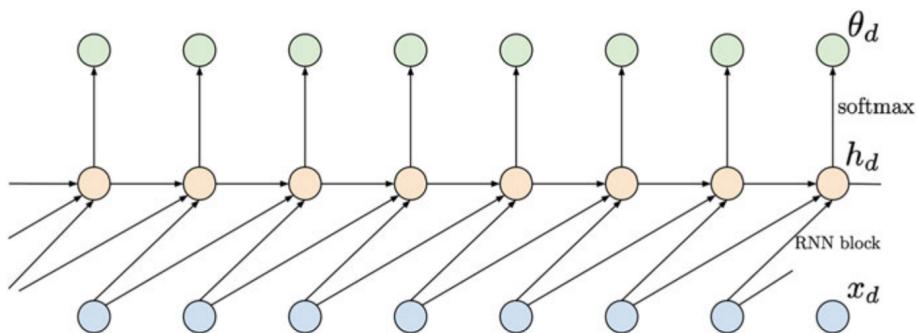


图 2.4 循环神经网络示意图

此外，transformer 也属于自回归模型，transformer 通过自注意力机制来对历史信息进行处理。

## 2.2.2 近似估计密度模型

### 基于能量的模型

基于能量的模型的诞生受到对物理系统建模的启发——一个事件的概率可以由玻尔兹曼分布式A.96表示<sup>[46]</sup>。如果一个神经网络  $E_\theta(\mathbf{x})$  只有一个输出神经元， $\theta$  表示神经网络的参数， $\mathbf{x}$  表示神经网络的输入，其输出结果为实值标量，那么有：

$$q_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z_\theta} , \text{ 其中 } Z_\theta = \begin{cases} \int_x \exp(-E_\theta(\mathbf{x})) d\mathbf{x} & \mathbf{x} \text{ 为连续变量} \\ \sum_x \exp(-E_\theta(\mathbf{x})) & \mathbf{x} \text{ 为离散变量} \end{cases} \quad (2.22)$$

式2.22中，指数函数保证所得概率大于 0，在  $E_\theta(\mathbf{x})$  前添加负号以表示  $E_\theta$  为能量函数：样本点概率取值越高则其能量越低，概率取值越低则具有更高的能量。 $Z_\theta$  是归一化项用来保证概率密度积分或和为 1，如式2.23所示：

$$\int_x q_\theta(\mathbf{x}) d\mathbf{x} = \int_x \frac{\exp(-E_\theta(\mathbf{x}))}{\int_{\hat{\mathbf{x}}} \exp(-E_\theta(\hat{\mathbf{x}})) d\hat{\mathbf{x}}} d\mathbf{x} = \frac{\int_x \exp(-E_\theta(\mathbf{x})) d\mathbf{x}}{\int_{\hat{\mathbf{x}}} \exp(-E_\theta(\hat{\mathbf{x}})) d\hat{\mathbf{x}}} = 1 \quad (2.23)$$

式2.22与式2.23中,  $q_\theta(\mathbf{x})$  为  $p(\mathbf{x})$  的近似, 通过训练, 可以使  $q_\theta(\mathbf{x})$  逐渐接近  $p(\mathbf{x})$ 。

对于基于能量的模型,  $E_\theta$  可以根据需要灵活选择。由于归一化常数  $Z_\theta$  未知, 无法直接计算对数似然函数, 且由于  $Z_\theta$  不一定能保证不变, 不可直接对未进行归一化的概率  $\exp(-E_\theta(\mathbf{x}_{train}))$  极大化, 即不一定保证训练样本点相比其他数据点有更高的概率出现。对于极大似然函数的计算, 可以通过对比散度方法来进行近似。通过比较不同数据点的似然来进行计算, 根据附录式A.45 可得负对数似然函数,

$$\nabla_\theta Loss(q_\theta(\mathbf{x})) = -\mathbb{E}_{p(\mathbf{x})} [\nabla_\theta \log q_\theta(\mathbf{x})] \quad (2.24)$$

$$= \mathbb{E}_{p(\mathbf{x})} [\nabla_\theta E_\theta(\mathbf{x})] - \mathbb{E}_{q_\theta(\mathbf{x})} [\nabla_\theta E_\theta(\mathbf{x})] \quad (2.25)$$

式2.24即为需要最小化的损失函数。对于式2.25, 从直观上理解, 第一项表示最小化数据集中样本点的能量, 以增加其概率; 第二项表示最大化随机生成的样本点的能量, 以减小其概率。图2.5中,  $f_\theta$  表示  $\exp(-E_\theta(\mathbf{x}))$ , 通过训练, 可以降低数据集中样本的能量函数值, 增大随机生成样本的能量函数值。

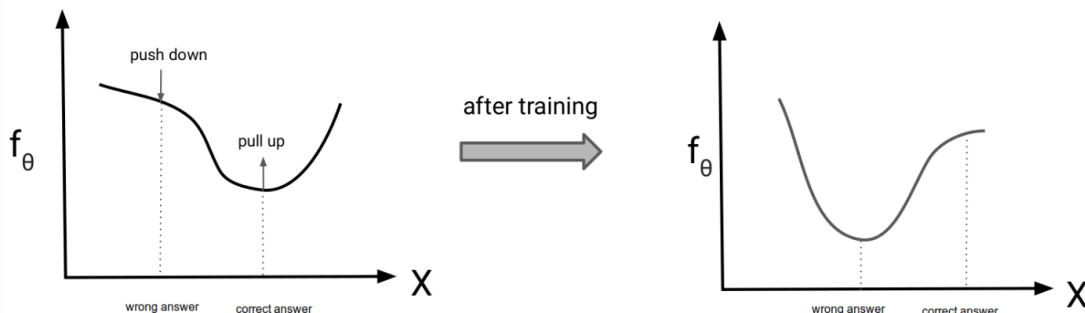


图 2.5 对比散度训练前后对比

式2.25中, 需要对概率分布  $q_\theta(\mathbf{x})$  采样。可以应用朗志万动力学与马尔科夫链蒙特卡洛结合的方法采样, 即从随机数据点开始, 根据能量函数的梯度  $\nabla_x E_\theta(\mathbf{x})$ , 使数据点的值向能量函数降低的方向移动。此外, 为避免陷入局部最小值, 需要在每一次更新梯度时添加噪声  $\omega \sim \mathcal{N}(0, \sigma)$ 。理论上讲, 根据梯度进行足够多次数据点取值更新, 即可以准确地从概率分布  $q_\theta(\mathbf{x})$  采样, 但在实际中, 通常将马尔科夫链的链长限制为  $K$ , 即进行  $K$  次取值更新。算法1为从基于能量的模型采样的算法。

### 变分自编码器

自编码器是一个包含了两部分的神经网络, 其编码器可以将原始高维输入映射到低维隐变量空间, 解码器可以将隐变量空间中的低维表示恢复成原始输入<sup>[47]</sup>。通过使用自编码器, 可以对数据进行更高效地压缩。

### 算法 1 从基于能量的模型采样

```

1: 从高斯分布或均匀分布中采样  $\tilde{\mathbf{x}}^0$ 
2: for 采样步骤  $k = 1$  到  $K$  do
3:    $\tilde{\mathbf{x}}^k \leftarrow \tilde{\mathbf{x}}^{k-1} - \eta \nabla_{\mathbf{x}} E_{\theta}(\tilde{\mathbf{x}}^{k-1}) + \omega$ , 其中  $\omega \sim \mathcal{N}(0, \sigma)$ 
4: end for
5:  $\mathbf{x}_{sample} \leftarrow \tilde{\mathbf{x}}^K$ 

```

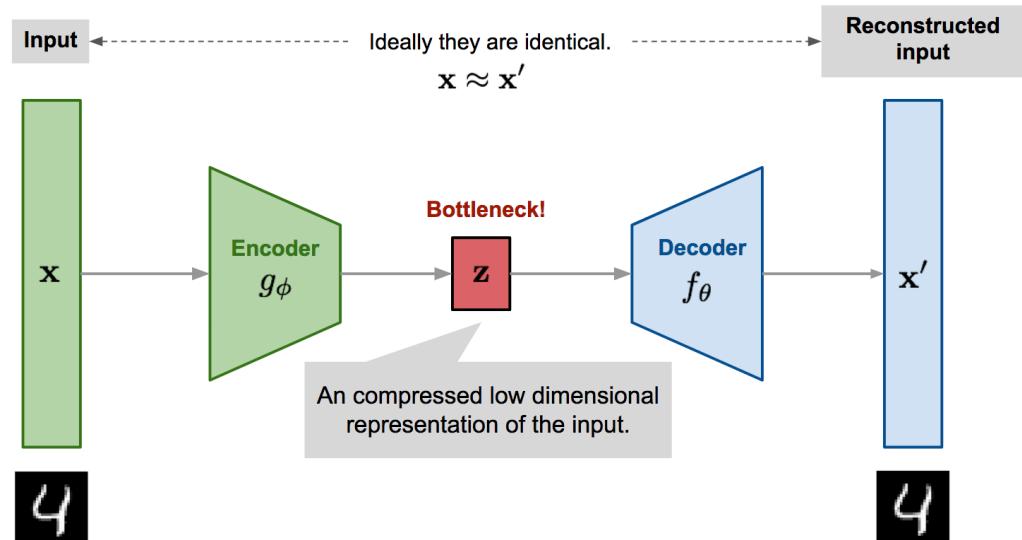


图 2.6 自编码器示意图

图2.6中,  $\mathbf{x}$  表示样本,  $\mathbf{x}'$  表示对原始样本的重构,  $\mathbf{z}$  表示样本在隐变量空间的表示,  $g_{\phi}(\cdot)$  表示由参数 $\phi$  确定的编码器函数,  $f_{\theta}(\cdot)$  表示由参数 $\theta$  确定的解码器函数。

$$\mathbf{z} = g_{\phi}(\mathbf{x}) \quad (2.26)$$

$$\mathbf{x}' = f_{\theta}(g_{\phi}(\mathbf{x})) \quad (2.27)$$

参数 $(\theta, \phi)$  可以通过比较原始样本值  $\mathbf{x}$  与重构样本值  $\mathbf{x}'$  来学习, 如使用均方误差:

$$L_{AE}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}^{(i)} - f_{\theta}(g_{\phi}(\mathbf{x}))) \quad (2.28)$$

为避免过拟合, 降噪自编码器<sup>[48]</sup>对原始样本添加随机噪声:

$$\hat{\mathbf{x}}^{(i)} \backsim \mathcal{M}_{\mathcal{D}}(\hat{\mathbf{x}}^{(i)} | \mathbf{x}^{(i)}) \quad (2.29)$$

$$L_{DAE}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}^{(i)} - f_{\theta}(g_{\phi}(\hat{\mathbf{x}}^{(i)}))) \quad (2.30)$$

式2.29中,  $\hat{\mathbf{x}}^{(i)}$  表示对原始样本值  $\mathbf{x}$  添加随机噪声或者进行掩码处理后的值,  $\mathcal{D}$  为原始样本所在数据集,  $\mathcal{M}_{\mathcal{D}}$  表示从原始样本到  $\hat{\mathbf{x}}^{(i)}$  的映射。式2.30与式2.28相比, 不再

将重构样本与原始样本  $\mathbf{x}$  进行比较，而是与添加随机噪声或进行掩码处理后的  $\hat{\mathbf{x}}^{(i)}$  进行比较。图2.7为降噪自编码器结构图。

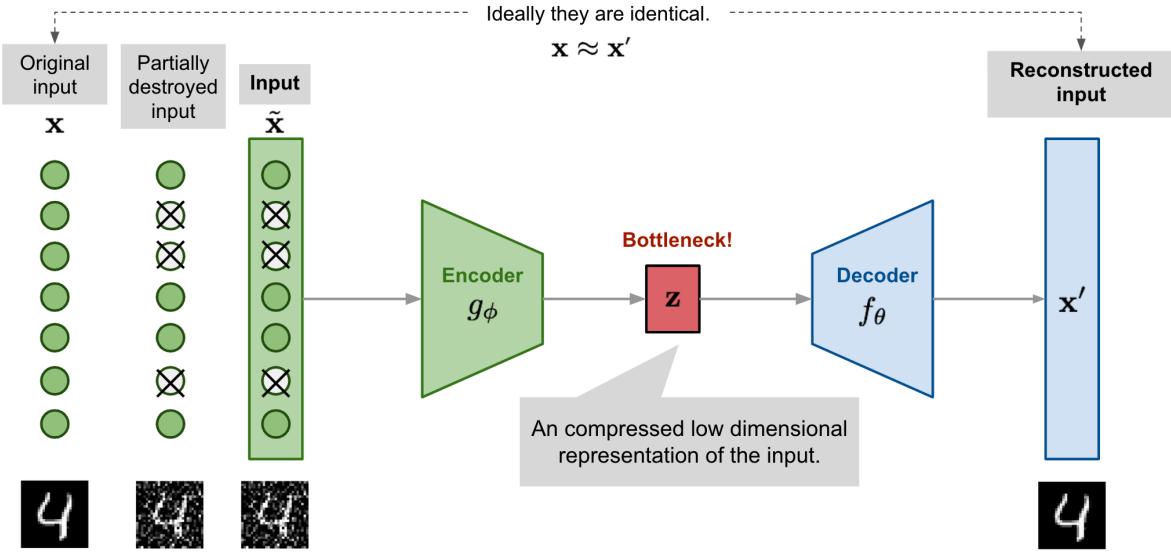


图 2.7 降噪自编码器结构

相比于自编码器和降噪自编码器，变分自编码器并不将样本映射到一个固定的向量而是映射到一个由参数  $\theta$  确定的概率分布  $p_\theta$ 。输入样本  $\mathbf{x}$  和隐变量  $\mathbf{z}$  的关系可以定义为：

- 先验概率  $p_\theta(\mathbf{z})$ ；
- 似然函数  $p_\theta(\mathbf{x} | \mathbf{z})$ ；
- 后验概率  $p_\theta(\mathbf{z} | \mathbf{x})$ 。

若知道概率分布  $p_\theta$  的真实参数值  $\theta^*$ ，则生成样本点某一维度的值  $\mathbf{x}^{(i)}$ （如图片样本的一个像素值）需要两个步骤：

1. 从先验分布  $p_{\theta^*}(\mathbf{z})$  中采样获得  $\mathbf{z}^{(i)}$ ；
2. 从条件概率  $p_{\theta^*}(\mathbf{x} | \mathbf{z} = \mathbf{z}^{(i)})$  中生成值  $\mathbf{x}^{(i)}$ 。

最优参数值  $\theta^*$  可以通过最大化生成真实数据样本的概率获得，即：

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^n p_\theta(\mathbf{x}^{(i)}) \quad (2.31)$$

取对数可得：

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n \log p_\theta(\mathbf{x}^{(i)}) \quad (2.32)$$

式2.32中， $p_\theta(\mathbf{x}^{(i)})$  可以表示为：

$$p_\theta(\mathbf{x}^{(i)}) = \int p_\theta(\mathbf{x}^{(i)} | \mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z} \quad (2.33)$$

但由于某些隐变量无法积分或计算开销巨大，式2.33无法直接计算。可用证据下界式A.57来近似对数似然函数。使用证据下界可以获得对后验概率 $p_\theta(z | x)$ 的近似，即由参数 $\phi$ 确定的 $q_\phi(z | x)$ 。图2.8中，近似函数 $q_\phi(z | x)$ 定义了编码器，条件概率 $p_\theta(x | z)$ 定义了解码器。

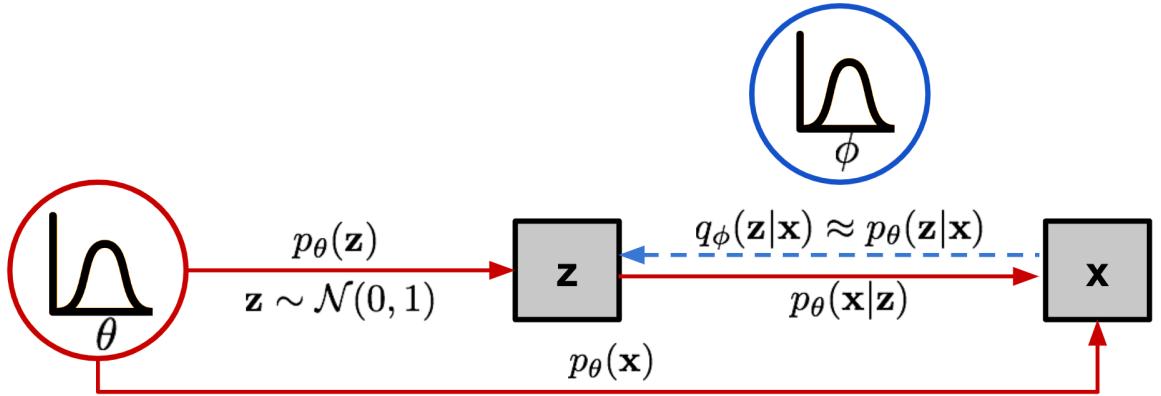


图 2.8 使用 ELBO 近似变分自编码器对数似然函数

对证据下界式A.57变形可得：

$$\mathbb{E}_{q_\phi(z|x)} \left[ \log \frac{p(x, z)}{q_\phi(z | x)} \right] = \mathbb{E}_{q_\phi(z|x)} \left[ \log \frac{p_\theta(x | z)p(z)}{q_\phi(z | x)} \right] \quad (2.34)$$

$$= \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x | z)] + \mathbb{E}_{q_\phi(z|x)} \left[ \log \frac{p(z)}{q_\phi(z | x)} \right] \quad (2.35)$$

$$= \underbrace{\mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x | z)]}_{\text{重构项}} - \underbrace{D_{KL}(q_\phi(z | x) || p(z))}_{\text{先验匹配项}} \quad (2.36)$$

式2.36中，重构项度量了解码器相对于变分分布 $q_\phi(z | x)$ 的重构似然函数 $p_\theta(x | z)$ ，从而使学到的分布可以对隐变量有效地建模，以便于可以通过 $p_\theta(x | z)$ 进行原样本重构。先验匹配项度量了学到的变分分布与隐变量先验分布的相似度，从而保证编码器能够真正学到隐变量分布而不是记忆原样本到隐变量的映射。最大化证据下界等价于最大化重构项和最小化先验匹配项。

变分自编码器中，通常编码器都为具有对角协方差矩阵的多变量高斯分布，隐变量的先验分布为多变量标准高斯分布：

$$q_\phi(z | x) = \mathcal{N}(z; \boldsymbol{\mu}_\phi(x, \sigma_\phi^2(x)\mathbf{I})) \quad (2.37)$$

$$p(z) = \mathcal{N}(z; \mathbf{0}, \mathbf{I}) \quad (2.38)$$

因此，由式2.37和式2.38可以计算式2.36中的先验匹配项。而式2.36中的重构项可

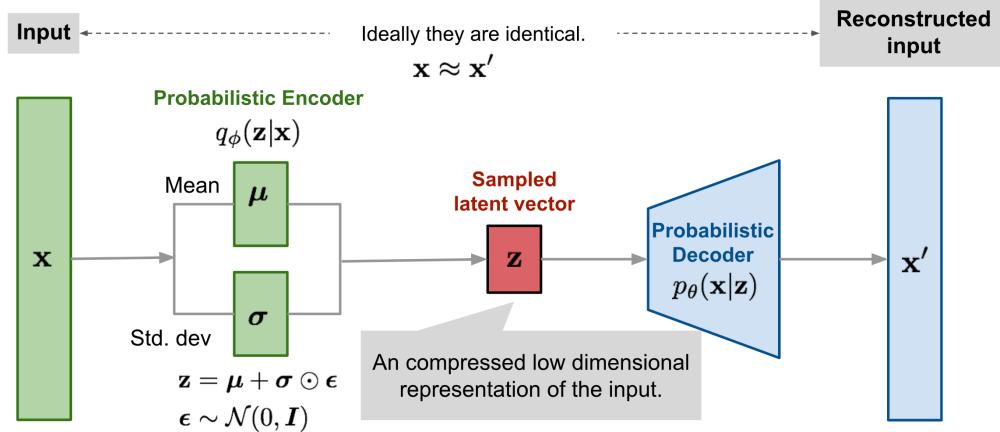


图 2.9 变分自编码器

以由蒙特卡洛方法进行估计，即：

$$\arg \max_{\phi, \theta} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x} | \mathbf{z})] - D_{KL}(q_\phi(\mathbf{z} | \mathbf{x}) || p(\mathbf{z})) \quad (2.39)$$

$$\approx \arg \max_{\phi, \theta} \sum_{l=1}^L \log p_\theta(\mathbf{x} | \mathbf{z}^{(l)}) - D_{KL}(q_\phi(\mathbf{z} | \mathbf{x}) || p(\mathbf{z})) \quad (2.40)$$

式2.40中， $\mathbf{z}^{(l)}_{l=1}^L$  为关于数据集中每个样本  $\mathbf{x}$  从  $q_\phi(\mathbf{z} | \mathbf{x})$  中获得的采样。

但由于通过随机过程获得的采样  $\mathbf{z}^{(l)}$  不可微，无法计算梯度，根据重参数方法式A.66可得：

$$\mathbf{z} = \boldsymbol{\mu}_\phi(\mathbf{x}) + \boldsymbol{\sigma}_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon} \quad \text{其中 } \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I}) \quad (2.41)$$

式2.41中， $\odot$  为对应元素分别相乘。图2.10中，重参数方法使损失梯度不可反向传播的过程变为可以反向传播。

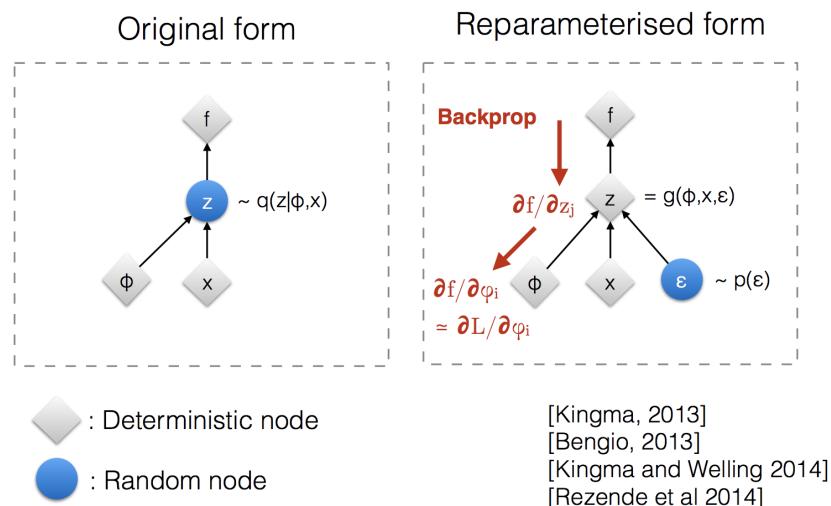


图 2.10 重参数方法示意图

式2.40中，变分自编码器使用重参数方法和蒙特卡洛方法，对证据下界关于 $\phi$  和 $\theta$ 同时进行优化。训练完成后，即可直接从隐变量分布 $p(\mathbf{z})$  中进行采样，再经由解码器获得生成样本 $\mathbf{x}'$ 。

层级变分自编码器是对变分自编码器的扩展<sup>[49]</sup>，相比于变分自编码器，层级变分自编码器可以有多层隐变量。层级变分自编码器假设浅层隐变量由更深层的隐变量生成。对于有 $T$  层隐变量的层级变分自编码器，每一个隐变量都依赖于先前所有隐变量，而其一个特例为马尔可夫层级变分自编码器。马尔可夫层级变分自编码器中，生成过程为马尔可夫链，解码时每一个隐变量 $\mathbf{z}_t$  只依赖于之前一个隐变量 $\mathbf{z}_{t+1}$ ，图2.11为马尔可夫层级变分自编码器示意图。

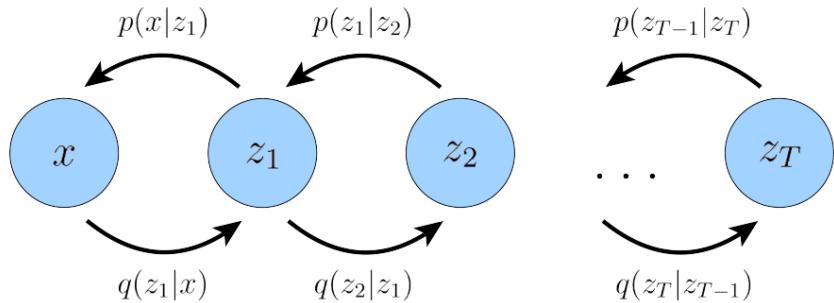


图 2.11 马尔可夫层级变分自编码器示意图

马尔可夫层级变分自编码器的联合概率分布可以表示为：

$$p(\mathbf{x}, \mathbf{z}_{1:T}) = p(\mathbf{x}_T)p_\theta(\mathbf{x} | \mathbf{z}_1) \prod_{t=2}^T p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t) \quad (2.42)$$

其后验概率可以表为：

$$q_\phi(\mathbf{z}_{1:T} | \mathbf{x}) = q_\phi(\mathbf{z}_1 | \mathbf{x}) \prod_{t=2}^T q_\phi(\mathbf{z}_t | \mathbf{z}_{t-1}) \quad (2.43)$$

证据下界可以扩展为：

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}, \mathbf{z}_{1:T}) dz \quad (\text{根据式A.49}) \quad (2.44)$$

$$= \log \int p(\mathbf{x}, \mathbf{z}_{1:T}) \frac{q_\phi(\mathbf{z}_{1:T} | \mathbf{x})}{q_\phi(\mathbf{z}_{1:T} | \mathbf{x})} dz \quad (2.45)$$

$$= \log \int \frac{p(\mathbf{x}, \mathbf{z}_{1:T}) q_\phi(\mathbf{z}_{1:T} | \mathbf{x})}{q_\phi(\mathbf{z}_{1:T} | \mathbf{x})} dz \quad (2.46)$$

$$= \log \int q_\phi(\mathbf{z}_{1:T} | \mathbf{x}) \frac{p(\mathbf{x}, \mathbf{z}_{1:T})}{q_\phi(\mathbf{z}_{1:T} | \mathbf{x})} dz \quad (2.47)$$

$$= \log \mathbb{E}_{q_\phi(\mathbf{z}_{1:T} | \mathbf{x})} \left[ \frac{p(\mathbf{x}, \mathbf{z}_{1:T})}{q_\phi(\mathbf{z}_{1:T} | \mathbf{x})} \right] \quad (\text{根据期望定义式A.21}) \quad (2.48)$$

$$\geq \mathbb{E}_{q_\phi(z_{1:T} | \mathbf{x})} \left[ \log \frac{p(\mathbf{x}, z_{1:T})}{q_\phi(z_{1:T} | \mathbf{x})} \right] \quad (\text{根据杰森不等式 A.89}) \quad (2.49)$$

将式2.42与2.43代入式2.49可得：

$$\mathbb{E}_{q_\phi(z_{1:T} | \mathbf{x})} \left[ \log \frac{p(\mathbf{x}, z_{1:T})}{q_\phi(z_{1:T} | \mathbf{x})} \right] = \mathbb{E}_{q_\phi(z_{1:T} | \mathbf{x})} \left[ \log \frac{p(z_T) p_\theta(\mathbf{x} | z_1) \prod_{t=2}^T p_\theta(z_{t-1} | z_t)}{q_\phi(z_1 | \mathbf{x}) \prod_{t=2}^T q_\phi(z_t | z_{t-1})} \right] \quad (2.50)$$

在扩散模型中，式2.50可以进一步分解更具有意义的成分。

此外，由于自编码器与降噪自编码器并不直接定义似然函数，可以将自编码器和降噪自编码器视为隐式密度模型。而变分自编码器由于使用证据下界对似然函数进行近似，变分自编码器为近似估计的显示密度模型。

### 扩散模型

扩散模型，是近年来在生成图像领域最有影响力的模型之一。在很多评价基准上，扩散模型已经取得了相比生成对抗网络更好的成绩。就如 2017——2020 年生成对抗网络的流行，在 2022 年，扩散模型也被广泛地用于各项生成任务。

2015 年，扩散模型被从物理学中借鉴引入至深度学习领域<sup>[50]</sup>。2019 年，基于分数的模型 NCSN<sup>[28]</sup>的提出，以及 2020 年对基于分数的模型训练方法的改进，都为扩散模型的诞生打下了良好的基础。

扩散模型可以理解为加一定限制的马尔可夫层级变分自编码器：

1. 隐变量维度与样本维度相同；
2. 每一时间（层级）隐变量编码器的结构为预先定义的线性高斯模型，并非学习获得，即将前一层隐变量值作为高斯分布的均值；
3. 隐变量编码器高斯分布的参数随着时间（层级）的变化而变化，且保证最终的隐变量分布为标准高斯分布。

图2.12为扩散模型示意图， $\mathbf{x}_0$  表示真实样本，如自然图像， $\mathbf{x}_T$  表示完全的高斯噪声， $\mathbf{x}_t$  表示真实样本 $\mathbf{x}_0$  添加噪声后所得的隐变量。 $q(\mathbf{x}_t | \mathbf{x}_{t-1})$  为将前一状态作为均值的高斯分布。

由第一个假设，将真实样本和隐变量均由 $\mathbf{x}_t$  表示，其中 $t = 0$  表示真实样本， $t \in [1, T]$  表示相应的 $t$  层隐变量，则扩散模型的后验与马尔可夫层级变分自编码器相似，根据式2.43得：

$$q(\mathbf{x}_{1:T} | \mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1}) \quad (2.51)$$

由第二个假设，编码器中隐变量服从以前一隐变量为均值的高斯分布，与马尔可夫结构变分自编码器不同的是，编码器在每一时间步 $t$  的结构并非学习获得，而是预先

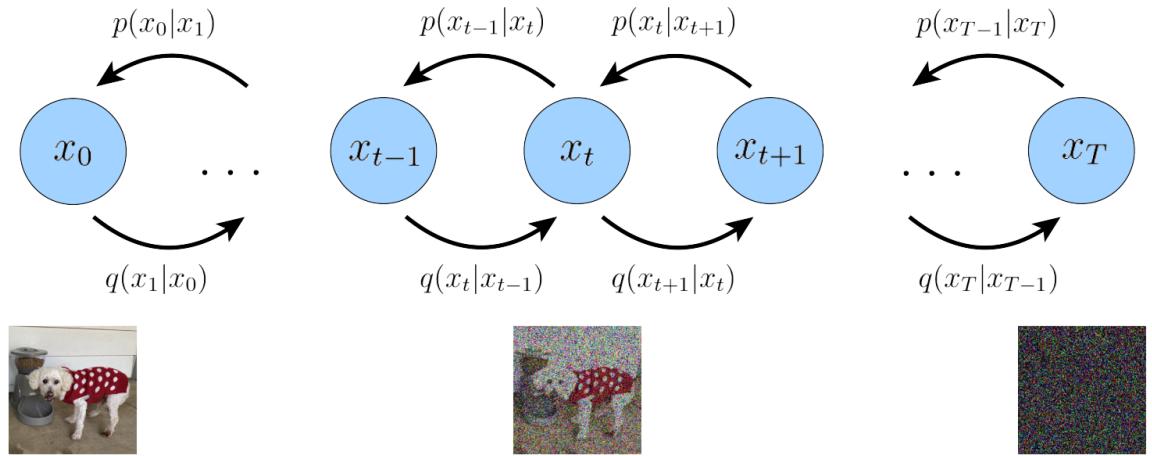


图 2.12 扩散模型示意图

设定的线性高斯模型，均值和方差可以设为超参数<sup>[32]</sup>，或经学习获得<sup>[51]</sup>。设高斯编码器均值为  $\mu_t(\mathbf{x}_t) = \sqrt{a_t} \mathbf{x}_{t-1}$ ，方差为  $\Sigma_t(\mathbf{x}_t) = (1 - \alpha_t) \mathbf{I}$ ，以使隐变量的方差大小相近。为使模型更灵活， $\alpha_t$  的值可以随着层级深度  $t$  的变化而变化。编码转换过程可以表示为：

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}(\mathbf{x}_{t-1}), (1 - \alpha_t)\mathbf{I}) \quad (2.52)$$

由第三个假设，预先定义或可学习获得的  $\alpha_t$  可以随着时间的变换而变化，且最深层隐变量  $p(\mathbf{x}_T)$  的分布为标准高斯分布，则马尔可夫层级变分自编码器的联合概率分布式<sup>2.42</sup>可以改写为：

$$p(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) \quad (2.53)$$

式<sup>2.53</sup>中， $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ 。假设三表明，扩散模型逐渐向图片添加噪声直至图片变为完全的高斯噪声，即如图<sup>2.12</sup>所示。

相比于马尔可夫层级变分自编码器，编码器分布  $q(\mathbf{x}_t | \mathbf{x}_{t-1})$  不再由参数  $\phi$  定义，而完全服从预先定义了均值与方差的高斯分布。因此，为能够生成新的样本点，扩散模型更注重对条件概率  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$  的学习。对扩散模型优化后，采样过程即为首先从高斯噪声  $p(\mathbf{x}_T)$  中采样，随后连续地进行  $T$  次去噪变换  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$ ，最终获得新的样本点  $\mathbf{x}_0$ 。

类比于层级变分自编码器，扩散模型可以通过最大化证据下界来优化，即：

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \quad (2.54)$$

$$= \log \int p(\mathbf{x}_{0:T}) \frac{q(\mathbf{x}_{1:T} | \mathbf{x}_0)}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} d\mathbf{x}_{1:T} \quad (2.55)$$

$$= \log \int q(\mathbf{x}_{1:T} | \mathbf{x}_0) \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} d\mathbf{x}_{1:T} \quad (2.56)$$

$$= \log \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \quad (2.57)$$

$$\geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \quad (2.58)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \quad (2.59)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) p_\theta(\mathbf{x}_0|\mathbf{x}_1) \prod_{t=2}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_T|\mathbf{x}_{T-1}) \prod_{t=1}^{T-1} q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \quad (2.60)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) p_\theta(\mathbf{x}_0|\mathbf{x}_1) \prod_{t=1}^{T-1} p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})}{q(\mathbf{x}_T|\mathbf{x}_{T-1}) \prod_{t=1}^{T-1} q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \quad (2.61)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_T|\mathbf{x}_{T-1})} \right] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \prod_{t=1}^{T-1} \frac{p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \quad (2.62)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [\log p(\mathbf{x}_T)] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_T|\mathbf{x}_{T-1})} \right] \\ + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \sum_{t=1}^{T-1} \log \frac{p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \quad (2.63)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [\log p(\mathbf{x}_T)] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_T|\mathbf{x}_{T-1})} \right] \\ + \sum_{t=1}^{T-1} \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \quad (2.64)$$

$$= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_{T-1}, \mathbf{x}_T|\mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_T|\mathbf{x}_{T-1})} \right] \\ + \sum_{t=1}^{T-1} \mathbb{E}_{q(\mathbf{x}_{T-1}, \mathbf{x}_t, \mathbf{x}_{t+1}|\mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \quad (2.65)$$

$$= \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]}_{\text{重构项}} - \underbrace{\mathbb{E}_{q(\mathbf{x}_{T-1}|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_T|\mathbf{x}_{T-1})||p(\mathbf{x}_T))]}_{\text{先验匹配项}} \\ - \sum_{t=1}^{T-1} \underbrace{\mathbb{E}_{q(\mathbf{x}_{t-1}, \mathbf{x}_{t+1}|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_t|\mathbf{x}_{t-1})||p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}))]}_{\text{一致检验项}} \quad (2.66)$$

式2.66中：

- $\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]$  项为重构项，是对样本点关于第一层隐变量的对数条件概率的预测，与一般变分自编码器式2.36中的重构项类似，可由蒙特卡洛模拟近似计算与优化。
- $\mathbb{E}_{q(\mathbf{x}_{T-1}|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_T|\mathbf{x}_{T-1})||p(\mathbf{x}_T))]$  项为先验匹配项，当最后一层隐变量的概率分布为高斯分布时达到最小，先验匹配项由于没有需要学习的参数，因此无需进行优化，假设当  $T$  足够大，则最后一层隐变量的概率分布为高斯分布，先验匹配

项的值为0。

- $\mathbb{E}_{q(x_{t-1}, x_{t+1} | x_0)} [D_{KL}(q(\mathbf{x}_t | \mathbf{x}_{t-1}) \| p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1}))]$  项为一致检验项，一致检验项使扩散模型的前向过程和反向过程在 $\mathbf{x}_t$ 保持一致，即降噪过程获得的 $p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1})$  应与高斯分布 $q(\mathbf{x}_t | \mathbf{x}_{t-1})$  尽可能一致。

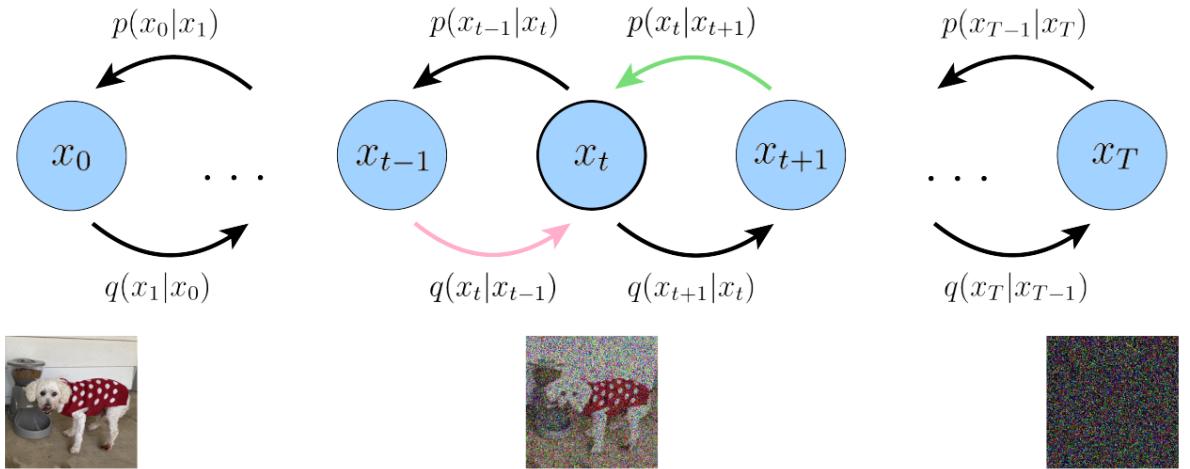


图 2.13 扩散模型证据下界一致检验项示意图

图2.13中，对于中间层隐变量 $\mathbf{x}_t$ ，可以通过减小绿色箭头所代表的条件概率 $p(\mathbf{x}_t | \mathbf{x}_{t+1})$  与粉色箭头代表的高斯分布 $q(\mathbf{x}_t | \mathbf{x}_{t-1})$  的差异来优化，由于需要对所有时间步骤 $t$  进行优化，扩散模型的优化时间取决于一致检验项。

式2.66将证据下界分解为不同的期望项，因此可以使用蒙特卡洛模拟来获得近似解，但由于一致检验项在每个时间步计算关于两个随机变量 $\mathbf{x}_{t-1}, \mathbf{x}_{t+1}$  的期望，其蒙特卡洛估计的方差可能相比每个时间步只关于一个随机变量的项更高。此外，由于需要对 $T - 1$  步一致检验项进行求和，当 $T$  越大时，证据下界估计值的方差也越大。

由于马尔可夫性质， $\mathbf{x}_t$  不依赖于 $\mathbf{x}_0$ ，可将编码过程改写为 $q(\mathbf{x}_t | \mathbf{x}_{t-1}) = q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0)$ ，则根据贝叶斯定理可得：

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t | \mathbf{x}_0)}{q(\mathbf{x}_{t-1} | \mathbf{x}_0)} \quad (2.67)$$

根据式2.67，可由式2.58得：

$$\log p(\mathbf{x}) \quad (2.68)$$

$$\geq \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \right] \quad (2.69)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{\prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right] \quad (2.70)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) p_\theta(\mathbf{x}_0 | \mathbf{x}_1) \prod_{t=2}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_1 | \mathbf{x}_0) \prod_{t=2}^T q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right] \quad (2.71)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)\prod_{t=2}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_1|\mathbf{x}_0)\prod_{t=2}^T q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0)} \right] \quad (2.72)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \prod_{t=2}^T \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0)} \right] \quad (2.73)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \prod_{t=2}^T \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}} \right] \quad (2.74)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \log \prod_{t=2}^T \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)} \right] \quad (2.75)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \sum_{t=2}^T \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)} \right] \quad (2.76)$$

$$\begin{aligned} &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} \right] \\ &\quad + \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)} \right] \end{aligned} \quad (2.77)$$

$$\begin{aligned} &= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_T|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} \right] \\ &\quad + \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_t,\mathbf{x}_{t-1}|\mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)} \right] \end{aligned} \quad (2.78)$$

$$\begin{aligned} &= \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]}_{\text{重构项}} - \underbrace{D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T))}_{\text{先验匹配项}} \\ &\quad - \sum_{t=2}^T \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]}_{\text{降噪匹配项}} \end{aligned} \quad (2.79)$$

式2.79相比式2.66，每一项均为最多只关于一个随机变量的期望。类比对式2.66的解释，式2.79有如下解释：

- $\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]$  项为重构项，与变分自编码器证据下界中的重构项类似。可以蒙特卡洛模拟方法近似估计与优化。
- $D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T))$  项为先验匹配项，表明不断添加噪声后的图像与高斯分布的匹配程度。先验匹配项在扩散模型假设下为0。
- $\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]$  为降噪匹配项，扩散模型用降噪变换  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$  来近似真实降噪变换  $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ 。通过最小化降噪匹配项，可以使两个降噪变换逐渐接近。

图2.14为使用式2.79优化扩散模型示意图，使用贝叶斯定理计算真实降噪过程  $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ ，并计算其关于近似降噪过程  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$  的 KL 散度，即尽可能使绿色箭头所

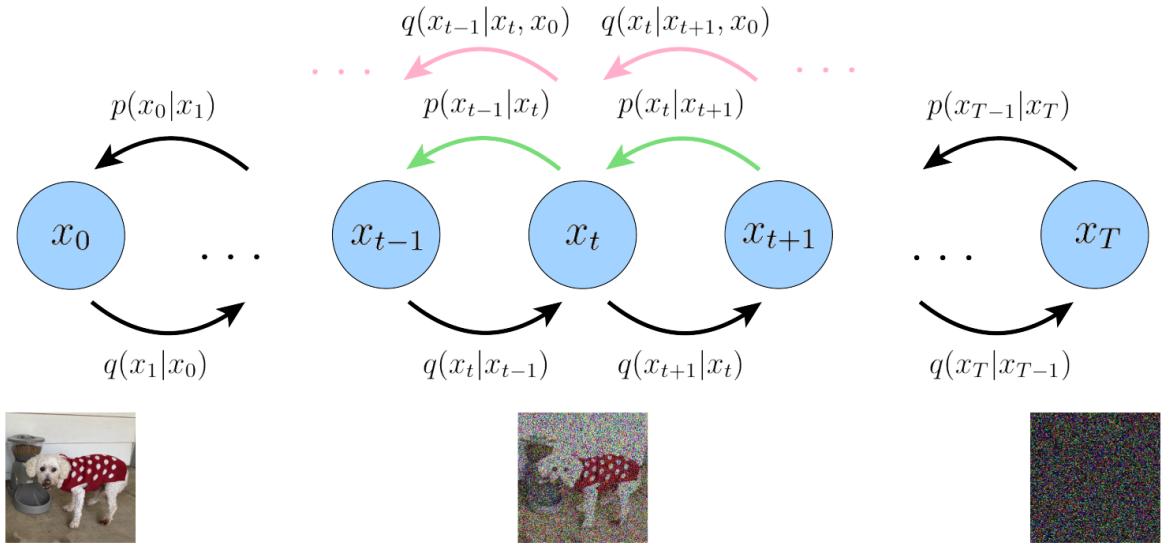


图 2.14 扩散模型证据下界降噪匹配项单变量期望示意图

代表分布与粉色箭头所代表分布一致。

KL 散度项  $D_{KL}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t))$  由于需要同时对多个时间步的编码器进行学习，难以进行最小化。为使 KL 散度项更易优化，根据贝叶斯定理有：

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1} | \mathbf{x}_0)}{q(\mathbf{x}_t | \mathbf{x}_0)} \quad (2.80)$$

根据式 2.52，可得：

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0) = q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}(\mathbf{x}_{t-1}), (1 - \alpha_t)\mathbf{I}) \quad (2.81)$$

由于扩散模型编码器为线性高斯模型，根据重参数方法，对于  $\mathbf{x}_t \sim q(\mathbf{x}_t, \mathbf{x}_{t-1})$  可得：

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon} \text{ 其中, } \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I}) \quad (2.82)$$

对于  $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}, \mathbf{x}_{t-2})$  可得：

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_{t-1}}\boldsymbol{\epsilon} \text{ 其中, } \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I}) \quad (2.83)$$

可以使用重参数方法对  $q(\mathbf{x}_t | \mathbf{x}_0)$  进行递归推导，假设有  $2T$  个随机噪声变量  $\{\boldsymbol{\epsilon}_t^*, \boldsymbol{\epsilon}_t\}_{t=0}^{2T} \stackrel{iid}{\sim} \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$ ，则对于任意采样  $\mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_0)$ ，有：

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}_{t-1}^* \quad (2.84)$$

$$= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}^*) + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}_{t-1}^* \quad (2.85)$$

$$= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}^* + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}_{t-1}^* \quad (2.86)$$

$$= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\sqrt{\alpha_t - \alpha_t\alpha_{t-1}}^2 + \sqrt{1 - \alpha_t}^2}\boldsymbol{\epsilon}_{t-2} \quad \text{根据式 A.19} \quad (2.87)$$

$$= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t \alpha_{t-1} + 1 - \alpha_t} \boldsymbol{\epsilon}_{t-2} \quad (2.88)$$

$$= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}} \boldsymbol{\epsilon}_{t-2} \quad (2.89)$$

$$= \dots \quad (2.90)$$

$$= \sqrt{\prod_{i=1}^t \alpha_i} \mathbf{x}_0 + \sqrt{1 - \prod_{i=1}^t \alpha_i} \boldsymbol{\epsilon}_0 \quad (2.91)$$

$$= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_0 \quad (2.92)$$

$$\sim \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}) \quad (2.93)$$

与式2.93类似，有：

$$\mathbf{x}_{t-1} \sim \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0, (1 - \bar{\alpha}_{t-1}) \mathbf{I}) \quad (2.94)$$

由式2.80、式2.81、式2.93和式2.94，可得：

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \quad (2.95)$$

$$= \frac{q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0) q(\mathbf{x}_{t-1} | \mathbf{x}_0)}{q(\mathbf{x}_t | \mathbf{x}_0)} \quad (2.96)$$

$$\propto \exp\left\{-\frac{1}{2} \left[ \frac{(\mathbf{x}_t - \sqrt{\alpha_t} \mathbf{x}_{t-1})^2}{1 - \alpha_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0)^2}{1 - \bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{x}_0)^2}{1 - \bar{\alpha}_t} \right]\right\} \quad (2.97)$$

$$= \exp\left\{-\frac{1}{2} \left[ \frac{-2\sqrt{\alpha_t} \mathbf{x}_t \mathbf{x}_{t-1} + \alpha_t \mathbf{x}_{t-1}^2}{1 - \alpha_t} + \frac{(\mathbf{x}_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_{t-1} \mathbf{x}_0)^2}{1 - \bar{\alpha}_{t-1}} + C(\mathbf{x}_t, \mathbf{x}_0) \right]\right\} \quad (2.98)$$

$$\propto \exp\left\{-\frac{1}{2} \left[ -\frac{2\sqrt{\alpha_t} \mathbf{x}_t \mathbf{x}_{t-1}}{1 - \alpha_t} + \frac{\alpha_t \mathbf{x}_{t-1}^2}{1 - \alpha_t} + \frac{\mathbf{x}_{t-1}^2}{1 - \bar{\alpha}_{t-1}} - \frac{2\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_{t-1} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right]\right\} \quad (2.99)$$

$$= \exp\left\{-\frac{1}{2} \left[ \left( \frac{\alpha_t}{1 - \alpha_t} + \frac{1}{1 - \bar{\alpha}_{t-1}} \right) \mathbf{x}_{t-1}^2 - 2 \left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right) \mathbf{x}_{t-1} \right]\right\} \quad (2.100)$$

$$= \exp\left\{-\frac{1}{2} \left[ \frac{\alpha_t(1 - \bar{\alpha}_{t-1}) + 1 - \alpha_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \mathbf{x}_{t-1}^2 - 2 \left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right) \mathbf{x}_{t-1} \right]\right\} \quad (2.101)$$

$$= \exp\left\{-\frac{1}{2} \left[ \frac{\alpha_t - \bar{\alpha}_t + 1 - \alpha_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \mathbf{x}_{t-1}^2 - 2 \left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right) \mathbf{x}_{t-1} \right]\right\} \quad (2.102)$$

$$= \exp\left\{-\frac{1}{2} \left[ \frac{1 - \bar{\alpha}_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \mathbf{x}_{t-1}^2 - 2 \left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right) \mathbf{x}_{t-1} \right]\right\} \quad (2.103)$$

$$= \exp\left\{-\frac{1}{2} \left( \frac{1 - \bar{\alpha}_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \right) \left[ \mathbf{x}_{t-1}^2 - 2 \frac{\left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right)}{\frac{1 - \bar{\alpha}_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}} \mathbf{x}_{t-1} \right]\right\} \quad (2.104)$$

$$= \exp\left\{-\frac{1}{2} \left( \frac{1 - \bar{\alpha}_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \right) \left[ \mathbf{x}_{t-1}^2 - 2 \frac{\left( \frac{\sqrt{\alpha_t} \mathbf{x}_t}{1 - \alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{1 - \bar{\alpha}_{t-1}} \right) (1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_{t-1} \right]\right\} \quad (2.105)$$

$$= \exp\left\{-\frac{1}{2} \left(\frac{1}{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}\right) \left[\mathbf{x}_{t-1}^2 - 2\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{1-\bar{\alpha}_t}\mathbf{x}_{t-1}\right]\right\} \quad (2.106)$$

$$\propto \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{1-\bar{\alpha}_t}}_{\mu_q(\mathbf{x}_t, \mathbf{x}_0)}, \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I}}_{\Sigma_q(t)}) \quad (2.107)$$

式2.98中,  $C(\mathbf{x}_t, \mathbf{x}_0)$  为由  $\mathbf{x}_t, \mathbf{x}_0, \alpha$  组成的相对于  $\mathbf{x}_{t-1}$  的常数项, 可与式2.106经过配方法式A.82, 获得式2.107。即  $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$  为正态分布, 均值  $\mu_q(\mathbf{x}_t, \mathbf{x}_0)$  为  $\mathbf{x}_t, \mathbf{x}_0$  的函数, 方差  $\Sigma_q(t)$  为  $\alpha$  的函数,  $\alpha$  为已知的关于时间步的确定量, 可以为超参数或由神经网络学习获得。根据式2.107, 可得:

$$\Sigma_q(t) = \sigma_q^2(t)\mathbf{I} = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I} \quad (2.108)$$

为使近似降噪转换  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$  与真实降噪转换  $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$  更加接近, 可以使  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$  服从高斯分布。此外, 由于  $\alpha$  在转换过程中为固定值, 可以令近似降噪转换的方差与真实降噪转换的方差相同, 即  $\Sigma_q(t) = \sigma_q^2(t)\mathbf{I}$ 。而对于高斯分布的均值, 由于真实降噪转换的高斯分布均值  $\mu_q(\mathbf{x}_t, \mathbf{x}_0)$  为  $\mathbf{x}_t, \mathbf{x}_0$  的函数, 近似降噪转换  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$  却不含有  $\mathbf{x}_0$ , 可将近似降噪转换高斯分布的均值设为  $\mu_\theta(\mathbf{x}_t, t)$ 。

根据两高斯分布 KL 散度计算式A.77, 由于近似降噪转换的方差与真实降噪转换的方差相同, 最小化 KL 散度即转换为最小化两分布的均值, 即:

$$\arg \min_{\theta} D_{KL}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)) \quad (2.109)$$

$$= \arg \min_{\theta} D_{KL}(\mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_q, \Sigma_q(t)) \| \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta, \Sigma_q(t))) \quad (2.110)$$

$$= \arg \min_{\theta} \frac{1}{2} \left[ \log \frac{|\Sigma_q(t)|}{|\Sigma_q(t)|} - d + \text{tr}(\Sigma_q(t)^{-1}\Sigma_q(t)) + (\boldsymbol{\mu}_\theta - \boldsymbol{\mu}_q)^\top \Sigma_q(t)^{-1}(\boldsymbol{\mu}_\theta - \boldsymbol{\mu}_q) \right] \quad (2.111)$$

$$= \arg \min_{\theta} \frac{1}{2} [\log 1 - d + d + (\boldsymbol{\mu}_\theta - \boldsymbol{\mu}_q)^\top \Sigma_q(t)^{-1}(\boldsymbol{\mu}_\theta - \boldsymbol{\mu}_q)] \quad (2.112)$$

$$= \arg \min_{\theta} \frac{1}{2} [(\boldsymbol{\mu}_\theta - \boldsymbol{\mu}_q)^\top \Sigma_q(t)^{-1}(\boldsymbol{\mu}_\theta - \boldsymbol{\mu}_q)] \quad (2.113)$$

$$= \arg \min_{\theta} \frac{1}{2} [(\boldsymbol{\mu}_\theta - \boldsymbol{\mu}_q)^\top \sigma_q^2(t)\mathbf{I}^{-1}(\boldsymbol{\mu}_\theta - \boldsymbol{\mu}_q)] \quad (2.114)$$

$$= \arg \min_{\theta} \frac{1}{2(\sigma_q^2(t))} [\|\boldsymbol{\mu}_\theta - \boldsymbol{\mu}_q\|_2^2] \quad (2.115)$$

式2.115中,  $\boldsymbol{\mu}_q$  为  $\mu_q(\mathbf{x}_t, \mathbf{x}_0)$  的缩写,  $\boldsymbol{\mu}_\theta$  为  $\mu_\theta(\mathbf{x}_t, t)$  的缩写。由式2.107, 真实降噪转换分布的均值为:

$$\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\alpha_t(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{1-\bar{\alpha}_t} \quad (2.116)$$

为使近似降噪转换的均值  $\mu_\theta(\mathbf{x}_t, t)$  与真实降噪转换的均值  $\mu_q(\mathbf{x}_t, \mathbf{x}_0)$  的均值尽可能一致, 由于近似降噪转换的均值  $\mu_\theta(\mathbf{x}_t, t)$  也包含  $\mathbf{x}_t$ , 可令两均值具有相似形式, 即:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{\alpha_t(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}1 - \alpha_t\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)}{1 - \bar{\alpha}_t} \quad (2.117)$$

式2.117中,  $\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)$  为由噪声图像  $\mathbf{x}_t$  和时间索引  $t$  预测原始样本  $\mathbf{x}_0$  的神经网络。则优化问题可转化为:

$$\arg \min_{\theta} D_{KL}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)) \quad (2.118)$$

$$= \arg \min_{\theta} D_{KL}(\mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q(t)) \| \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_q(t))) \quad (2.119)$$

$$= \arg \min_{\theta} \frac{1}{2(\sigma_q^2(t))} [\|\boldsymbol{\mu}_\theta - \boldsymbol{\mu}_q\|_2^2] \quad (2.120)$$

$$= \arg \min_{\theta} \frac{1}{2(\sigma_q^2(t))} \left[ \left\| \frac{\alpha_t(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}1 - \alpha_t\mathbf{x}_0}{1 - \bar{\alpha}_t} \right. \right. \quad (2.121)$$

$$\left. \left. - \frac{\alpha_t(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}1 - \alpha_t\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)}{1 - \bar{\alpha}_t} \right\|_2^2 \right] \quad (2.122)$$

$$= \arg \min_{\theta} \frac{1}{2(\sigma_q^2(t))} \left[ \left\| \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)}{1 - \bar{\alpha}_t} - \frac{\sqrt{\bar{\alpha}_{t-1}}1 - \alpha_t\mathbf{x}_0}{1 - \bar{\alpha}_t} \right\|_2^2 \right] \quad (2.123)$$

$$= \arg \min_{\theta} \frac{1}{2(\sigma_q^2(t))} \left[ \left\| \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{1 - \bar{\alpha}_t} (\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0) \right\|_2^2 \right] \quad (2.124)$$

$$= \arg \min_{\theta} \frac{1}{2(\sigma_q^2(t))} \frac{\bar{\alpha}_{t-1}(1 - \alpha_t)^2}{1 - \bar{\alpha}_t^2} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.125)$$

由式2.125, 对扩散模型的优化可以转化为对神经网络的训练, 该神经网络可以从任意时刻噪声图像来预测真实样本。此外, 对式2.79中的降噪匹配求和项, 可以通过最小化所有时刻的期望  $\eta$  来近似, 即:

$$\arg \min_{\theta} \mathbb{E}_{t \sim U\{2, T\}} [\mathbb{E}_{q(\mathbf{x}_t | \mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t))] ] \quad (2.126)$$

噪声参数  $\alpha_t$  可设置为超参数, 也可参与训练学习获得。可以使用以  $\eta$  为参数的神经网络  $\hat{\alpha}_\eta(t)$  来获得  $\alpha_t$ , 但由于每个时间步  $t$  都需计算  $\alpha_t$ , 效率较低, 可将式2.108代入式2.125, 可得:

$$\frac{1}{2(\sigma_q^2(t))} \frac{\bar{\alpha}_{t-1}(1 - \alpha_t)^2}{1 - \bar{\alpha}_t^2} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.127)$$

$$= \frac{1}{2 \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}} \frac{\bar{\alpha}_{t-1}(1 - \alpha_t)^2}{1 - \bar{\alpha}_t^2} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.128)$$

$$= \frac{1}{2} \frac{1 - \bar{\alpha}_t}{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \frac{\bar{\alpha}_{t-1}(1 - \alpha_t)^2}{1 - \bar{\alpha}_t^2} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.129)$$

$$= \frac{1}{2} \frac{\bar{\alpha}_{t-1}(1 - \alpha_t)}{(1 - \bar{\alpha}_{t-1})(1 - \bar{\alpha}_t)} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.130)$$

$$= \frac{1}{2} \frac{\bar{\alpha}_{t-1} - \bar{\alpha}_t}{(1 - \bar{\alpha}_{t-1})(1 - \bar{\alpha}_t)} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.131)$$

$$= \frac{1}{2} \frac{\bar{\alpha}_{t-1} - \bar{\alpha}_t \bar{\alpha}_{t-1} + \bar{\alpha}_t \bar{\alpha}_{t-1} - \bar{\alpha}_t}{(1 - \bar{\alpha}_{t-1})(1 - \bar{\alpha}_t)} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.132)$$

$$= \frac{1}{2} \frac{\bar{\alpha}_{t-1}(1 - \bar{\alpha}_t) - \bar{\alpha}_t(1 - \bar{\alpha}_{t-1})}{(1 - \bar{\alpha}_{t-1})(1 - \bar{\alpha}_t)} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.133)$$

$$= \frac{1}{2} \left( \frac{\bar{\alpha}_{t-1}(1 - \bar{\alpha}_t)}{(1 - \bar{\alpha}_{t-1})(1 - \bar{\alpha}_t)} - \frac{\bar{\alpha}_t(1 - \bar{\alpha}_{t-1})}{(1 - \bar{\alpha}_{t-1})(1 - \bar{\alpha}_t)} \right) [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.134)$$

$$= \frac{1}{2} \left( \frac{\bar{\alpha}_{t-1}}{1 - \bar{\alpha}_{t-1}} - \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.135)$$

根据式2.93,  $q(\mathbf{x}_t | \mathbf{x}_0)$  为服从高斯分布  $\mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I})$ , 则根据信噪比定义式A.80可得:

$$SNR(t) = \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \quad (2.136)$$

根据式2.136可将式2.135改写为:

$$\frac{1}{2(\sigma_q^2(t))} \frac{\bar{\alpha}_{t-1}(1 - \alpha_t)^2}{1 - \bar{\alpha}_t^2} [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.137)$$

$$= \frac{1}{2} \left( \frac{\bar{\alpha}_{t-1}}{1 - \bar{\alpha}_{t-1}} - \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.138)$$

$$= \frac{1}{2} (SNR(t-1) - SNR(t)) [\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (2.139)$$

根据信噪比定义, 信噪比越高, 则包含更多的信息, 信噪比越低, 则噪声越多。扩散模型的信噪比随着时间步  $t$  的增加而单调减小, 即  $\mathbf{x}_t$  随着时间推移而逐渐包含更多的噪声, 直至在  $t = T$  时变为标准高斯分布。

为简化式2.139, 可使用神经网络来模拟任意时间的信噪比, 并与扩散模型同时训练。即:

$$SNR(t) = \exp(-\omega_\eta(t)) \quad (2.140)$$

式2.140中,  $\omega_\eta(t)$  为单调递增函数,  $-\omega_\eta(t)$  为单调递减函数, 进行指数化以使信噪比为正数。此时, 使用神经网络模拟任意时间信噪比, 式2.126需要同时对  $\eta$  进行优化。

此外, 由式2.136、式2.140和式A.84, 可得:

$$\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} = \exp(-\omega_\eta(t)) \quad (2.141)$$

$$\bar{\alpha}_t = sigmoid(-\omega_\eta(t)) \quad (2.142)$$

$$1 - \bar{\alpha}_t = sigmoid(\omega_\eta(t)) \quad (2.143)$$

式2.142和式2.143可用于式2.106中使用重参数方法从原始样本 $\mathbf{x}_0$ 生成任意时刻添加噪声后的图像 $\mathbf{x}_t$ 。

## 2.3 隐式密度模型

隐式密度模型不定义概率密度函数 $p_{model}(\mathbf{x}; \boldsymbol{\theta})$ , 不通过最大化似然函数来训练, 而通过其他方式与实际概率密度函数 $p_{data}$ 进行交互。

### 2.3.1 生成对抗网络

生成对抗网络可以视为根据博弈论设计的一场游戏, 包含两个机器学习模型, 机器学习模型通常采用神经网络。

其中一个神经网络为生成器, 其隐式地确定了概率分布 $p_{model}$ , 生成器不需要对概率分布 $p_{model}$ 进行评估。根据一个先验分布 $p(\mathbf{z})$ 以及生成函数方程 $G(\mathbf{z}; \boldsymbol{\theta}^{(G)})$ , 生成器可以从概率分布 $p_{model}$ 中采样, 其中可学习参数 $\boldsymbol{\theta}^{(G)}$ 确定了生成器在博弈中采取的策略, 先验分布 $p(\mathbf{z})$ 经常为相对而言无特殊结构的分布, 比如高维高斯分布或超立方体均匀分布, 从这类分布中采样的 $\mathbf{z}$ 为随机噪声。生成器的主要任务是通过学习将随机噪声 $\mathbf{z}$ 转化为能够以假乱真的样本。

另一个神经网络为辨别器, 辨别器可以通过辨别函数 $D(\mathbf{x}; \boldsymbol{\theta}^{(D)})$ , 判断一个样本 $x$ 是从训练集中采样的真实样本或是由生成器生成的伪造样本。

在生成对抗网络游戏中, 每一个玩家都有其本身的损失函数, 生成器的损失函数为 $J^G(\boldsymbol{\theta}^{(G)}, \boldsymbol{\theta}^{(D)})$ , 辨别器的损失函数为 $J^D(\boldsymbol{\theta}^{(G)}, \boldsymbol{\theta}^{(D)})$ 。每一个玩家都希望能够最小化其损失函数。辨别器的损失函数使其能够更好地辨别一个样本是真实样本还是伪造样本, 生成器的损失函数使其能够制造出可以迷惑辨别器的伪造样本, 使辨别器将伪造样本误认为真实样本。

生成对抗网络可以形象的表示成警察和造假者的博弈, 造假者尝试制造假币, 而警察在尽力逮捕造假者的同时, 还要保证正规货币的流通。随着警察和造假者的博弈, 造假者的“造假”能力逐渐提高, 直至警察无法分辨。对于生成对抗网络的训练过程, 可以比喻为造假者在警察中有内线, 其可以向造假者提供警方辨别真假的方法, 以让造假者制造更具有迷惑性的假币。

图2.15为生成对抗网络训练过程, 生成对抗网络的训练包括两部分: 对生成网络的训练和对辨别网络的训练。训练的过程包括不断地从数据集中获得真实样本和由生成网络生成伪造样本, 辨别网络的训练与其他辨别深度神经网络的训练类似。在图2.15的

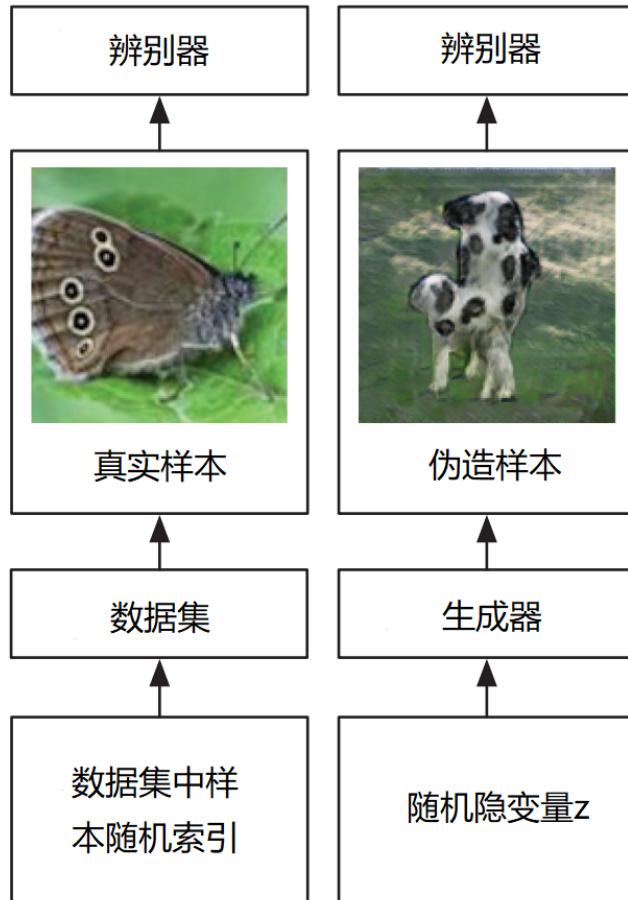


图 2.15 生成对抗网络训练过程

左侧，向辨别器输入从数据集中获取的真实样本，其对应的标签为真；在图2.15的右侧，向辨别器输入由生成网络生成的伪造样本，其对应的标签为假。从隐变量先验分布中采样获得随机向量 $z$ ，再将该随机变量 $z$  输入生成器，即可得伪造样本 $\mathbf{x} = G(z)$ 。生成器生成函数 $G$ 由神经网络表示，其可以将随机无结构的向量 $z$  转换为伪造样本 $\mathbf{x}$ ，且该伪造样本要尽可能与训练数据集中的样本在统计意义上难以分辨。通过反向传播算法，可以借助辨别器输出相对于对辨别器输入的梯度来训练生成器。生成网络训练的方向为，使其生成的伪造样本能够更多地被辨别器判定为真实样本。辨别器的训练与其他辨别网络训练类似，唯一区别是标记为伪造样本类别的概率分布，可以随着生成器的训练而不断变化。

## 2.4 生成模型评价指标

在生成模型研究中，为了证明一种方法比另一种方法更好，往往需要有一个评价标准，即生成模型评价指标。

### 2.4.1 图灵测试

为了让生成模型生成的图像更加真实，可以借鉴图灵测试，即让人们判断生成图像的质量与真实图像相比如何。但一些模型由于过拟合问题，可能仅仅对原始样本进行记忆，也可以生成真实度很高的图像。

### 2.4.2 图像质量评估分数

Inception 分数<sup>[52]</sup>是为判断生成对抗网络生成图像的质量而提出的一种计算分数，生成图像的多样性和质量都会对影响 Inception 分数。Inception 分数通过计算标签关于生成图像的条件概率  $p(y | \mathbf{x})$  来评估图像质量。Inception 分数有以下假设：

- 某标签关于生成图像的条件分布  $p(y | \mathbf{x})$  应当具有较低的信息熵，即越真实的图像越有可能属于某一预设类别。
- 可以生成多种类别图像的模型，其边缘分布  $\int p(y | \mathbf{x} = G(\mathbf{z})) dz$  应当具备更高的信息熵，即生成的图像应当均匀地属于不同类别。

$$IS = \exp(\mathbb{E}_{\mathbf{x} \sim p_G} D_{KL}(p(y | \mathbf{x}) || p(y))) \quad (2.144)$$

式2.144中， $\mathbf{x}$  表示生成器生成的样本。 $p(y | \mathbf{x})$  表示标签  $y$  关于生成样本  $\mathbf{x}$  的条件概率。 $p(y)$  表示关于标签  $y$  的边缘分布。 $D_{KL}(P || Q)$  表示 KL 散度，即式A.70。进行指数化以便于比较。Inception 分数越高，则图片生成的质量越好。

Inception 分数的不足之处是没有使用真实样本的统计性质。

FID<sup>[53]</sup> (Fréchet Inception Distance) 使用在 ImageNet 数据集上预训练的 Inception V3 模型，计算生成图像和真实图像特征向量之间的距离，越低的分数表明两组图像越相似，分数为 0 则代表两组图像完全相同。假设两组图像在 Inception V3 特征层特征向量的均值分别为  $\mathbf{m}_{model}, \mathbf{m}_{real}$ ，方差分别为  $\mathbf{C}_{model}, \mathbf{C}_{real}$ ，则：

$$FID = \|\mathbf{m}_{model} - \mathbf{m}_{real}\|_2^2 + Tr(\mathbf{C}_{model} + \mathbf{C}_{real} - 2\sqrt{(\mathbf{C}_{model}\mathbf{C}_{real}))}) \quad (2.145)$$

KID<sup>[54]</sup> (Kernel Inception Distance) 计算生成图像和真实图像特征向量之间最大平均距离的平方，特征向量也由与训练 Inception 模型获得。根据式A.93，有

$$MMD^2(X, Y) = \frac{1}{m(m-1)} \sum_{i \neq j}^m k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i \neq j}^n k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^m \sum_{j=1}^n k(x_i, y_j) \quad (2.146)$$

式2.146中， $m$  是生成图像的样本量， $n$  是真实图像的样本量，每个  $x$  和  $y$  均是 Inception 网络的特征表示层的 2048 维向量，且

$$k(x, y) = \frac{1}{d} x^\top y + 1 \quad (2.147)$$

相比于 FID，KID 不假定特征向量概率分布为高斯分布<sup>[55]</sup>。

### 3 降噪扩散模型生成中国画

#### 3.1 模型概述

降噪扩散模型，也称为扩散模型，在训练时分为添加噪声和去除噪声两部分，首先，逐渐将随机噪声添加至图像，且对添加的噪声进行记录，最终获得均匀的噪声图像。之后，对该噪声图像进行逐渐去噪，期望最终获得原始图像。训练完成后，可以将任意的随机噪声，转化为与训练集中图像类似的图像。算法2为扩散模型训练算法，算法3为扩散模型采样算法。

---

#### 算法 2 扩散模型训练算法

---

```

1: repeat
2:    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
3:    $t \sim \text{Uniform}(\{1, \dots, T\})$ 
4:    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:   关于参数 $\theta$  进行梯度下降，梯度为 $\nabla_{\theta} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta}(\mathbf{x}_t, t)\|^2$ 
6: until 收敛

```

---



---

#### 算法 3 扩散模型采样算法

---

```

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

```

---

扩散模型使用 U 型网络，U 型网络的输入和输出维度相同。U 型网络中，使用了上采样和下采样、残差连接、正弦位置嵌入以及注意力模块，在注意力层前进行了组规范。

#### 3.2 数据集

原始完整数据集<sup>[56]</sup>为从如下博物馆获取的高质量传统中国画：

- 普林斯顿大学艺术博物馆 362 张;
- 哈佛大学艺术博物馆 101 张;
- 纽约大都会博物馆 428 张;
- 史密森尼弗里尔艺术画廊 1301 张。

共 2194 张高质量传统中国画，裁剪后大小均为  $512 \times 512$ ，图3.1为完整数据集中国画采样。

在去除边缘留白较大的图像以及颜色过深的图像后，对原始数据集根据颜色深浅进行分类，得到包含 1159 张中国画的浅色中国画数据集，以及包含 792 张中国画的深色中国画数据集。图3.2为浅色中国画数据集采样。图3.3为深色中国画数据集采样。

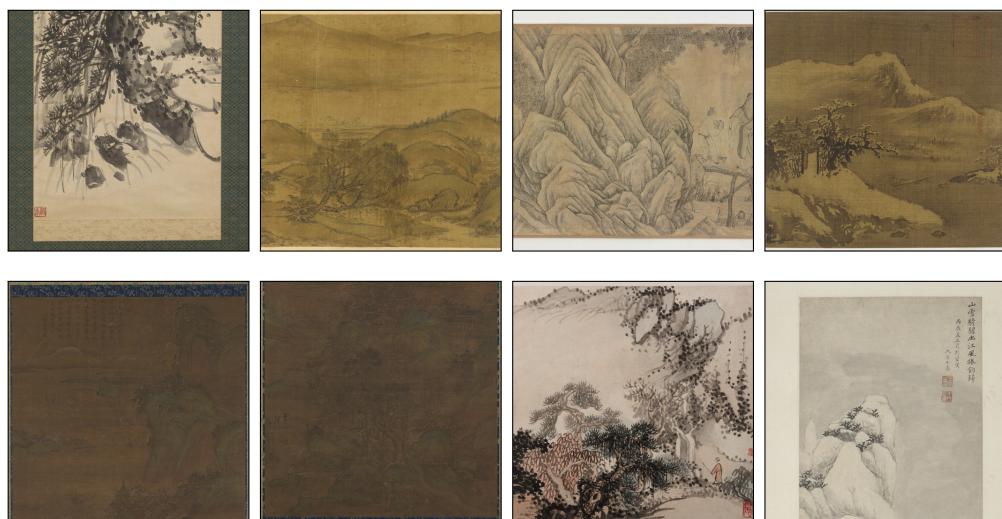


图 3.1 完整数据集中国画样本



图 3.2 浅色中国画数据集样本



图 3.3 深色中国画数据集样本

### 3.3 实验

分别使用完整数据集、浅色中国画数据集、深色中国画数据集进行训练，以生成高质量中国画，其中使用浅色中国画数据集依次进行了大批量和小批量训练。

各实验根据实际情况不同，使用不同 GPU 型号，不同实验共同的环境配置如下：

- 操作系统：Ubuntu 20.04
- Python 编译器器：Python 3.8
- Pytorch 框架：1.12
- CUDA 版本：11.3
- 编辑器：Visual Studio Code

不同实验扩散模型相同参数设置如下：

- 图像大小：128\*128
- 扩散步骤：1000 步
- 采样步骤：250 步
- 损失函数类型：L1 距离
- 学习率： $1 \times 10^{-5}$
- 梯度累计步骤：2

### 3.3.1 实验 1-完整数据集生成中国画

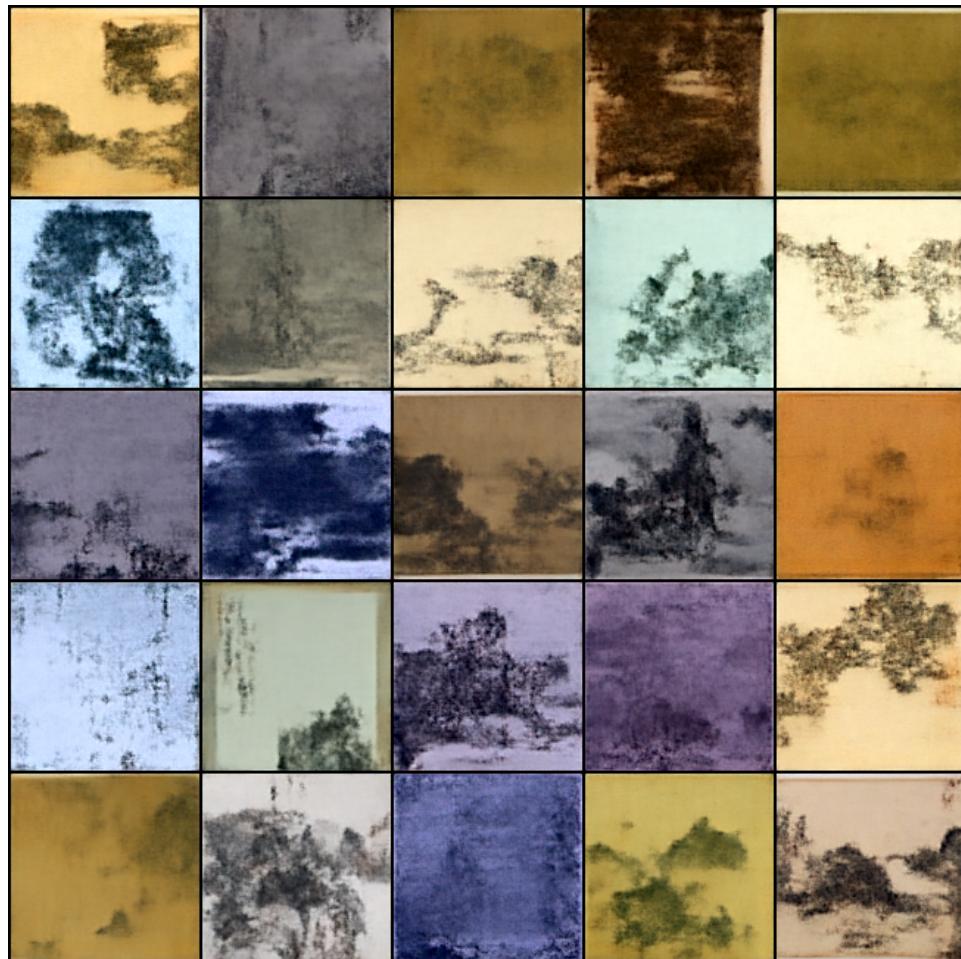


图 3.4 实验 1-完整数据集生成中国画采样图

#### 实验配置

使用完整数据集，在 A100- 80GB GPU 上，以 128 为训练批量训练约 15 小时。

#### 实验结果与分析

实验一结果如图3.4所示。为减小比对模型不同训练阶段图像生成质量的误差，采样 20 次，共获取 20 张图片并合并显示。由于完整数据集中图片颜色丰富，模型需要同时学习不同颜色深度中国画，在计算资源有限情况下模型收敛需要更长时间，此外，不同深度中国画混合训练容易造成中国画颜色混合，降低图片生成质量，为加快模型收敛与提高图片生成质量，使用浅色中国画数据集和深色中国画数据集分别训练。

### 3.3.2 实验 2-浅色中国画数据集大批量训练

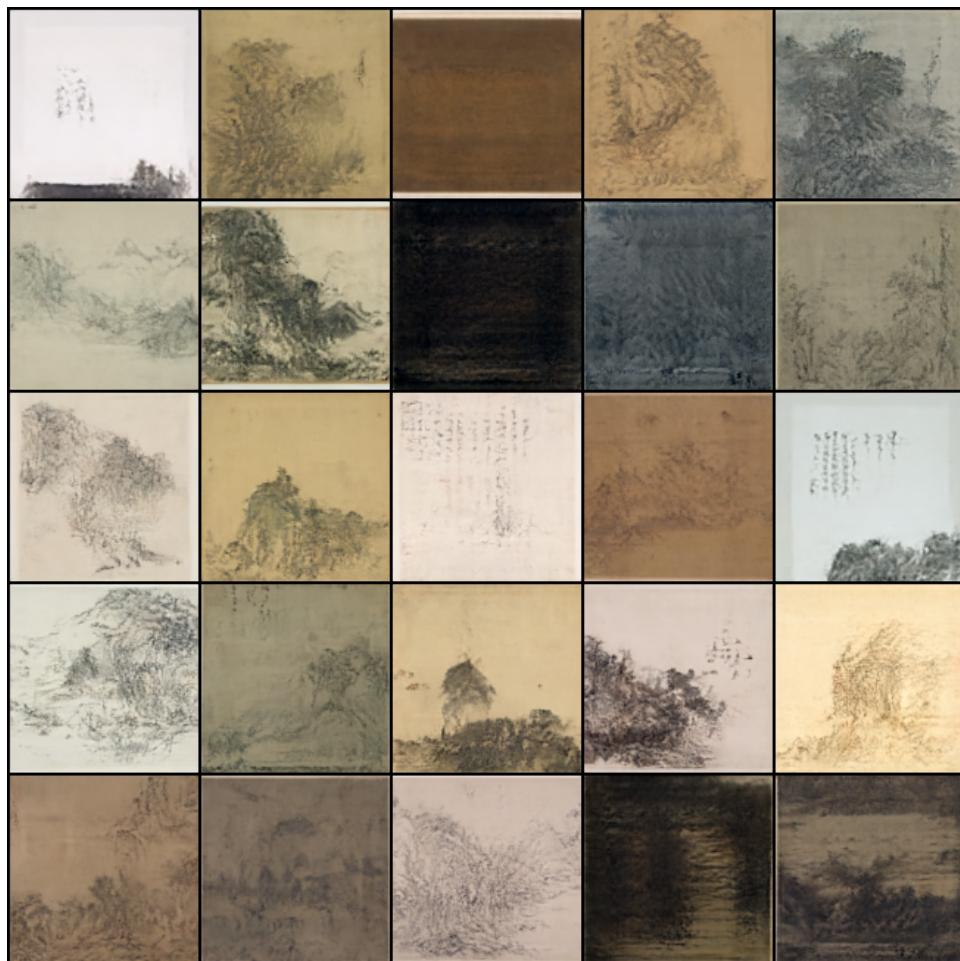


图 3.5 实验 2-浅色中国画数据集大批量训练生成中国画采样图

#### 实验配置

使用浅色中国画数据集，在 A100- 40GB GPU 上，以 64 为训练批量，训练约 48 小时。

#### 实验结果与分析

图3.5为训练 70 轮后的采样图像，相比于实验 1 使用完整数据集所生成的图像，大部分图像颜色较浅，图像质量明显提升。第四行<sup>1</sup>图像取得了与中国画非常相似的效果，一些图像如(3,3), (3,5)，已经可以生成与中国画中的落款相似的文字符号。同时有一些图像如(2,3), (5,4)，未成功生成中国画，增大数据集容量、或使用数据增强技术，有望改善扩散模型无法将隐变量噪声空间映射到中国画空间的情况。

<sup>1</sup>以图像表格上方为第一行，左侧为第一列，坐标表示为（行，列）

### 3.3.3 实验 3-浅色中国画数据集小批量训练

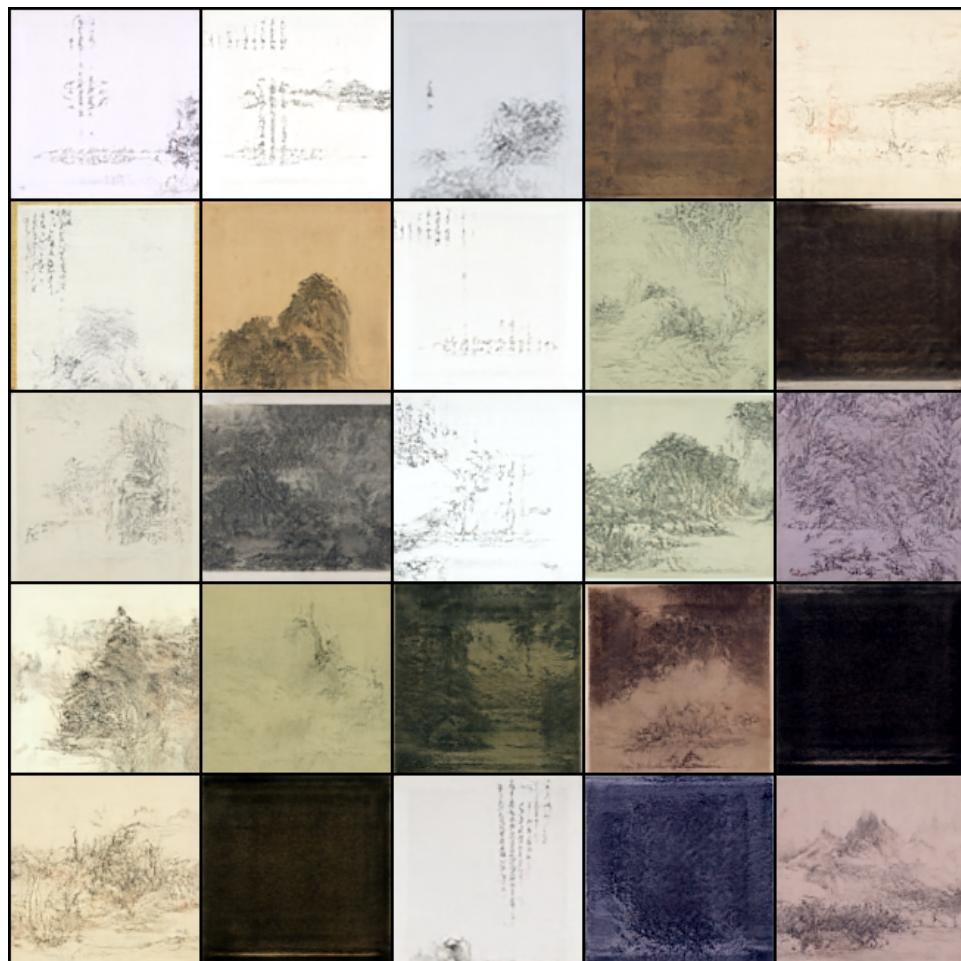


图 3.6 实验 3-浅色中国画数据集小批量训练生成中国画采样图

#### 实验配置

使用浅色中国画数据集，在 3090- 24GB GPU 上，以 32 为训练批量训练约 48 小时。

#### 实验结果与分析

图3.6为训练 80 轮后的采样图像，相比于实验 2，实验三仅仅改变了训练批量，但产生了更多的无法成功生成中国画的情况，如子图(2, 5), (4, 2), (4, 5), (5, 2), (5, 4) 生成颜色过深图像，子图(1, 1), (1, 2), (1, 5), (2, 3), (3, 3) 生成颜色过浅图像。对比实验 2 与实验 3 可得，更大的训练批量有助于生成更高质量的中国画。

### 3.3.4 实验 4-深色中国画数据集生成中国画

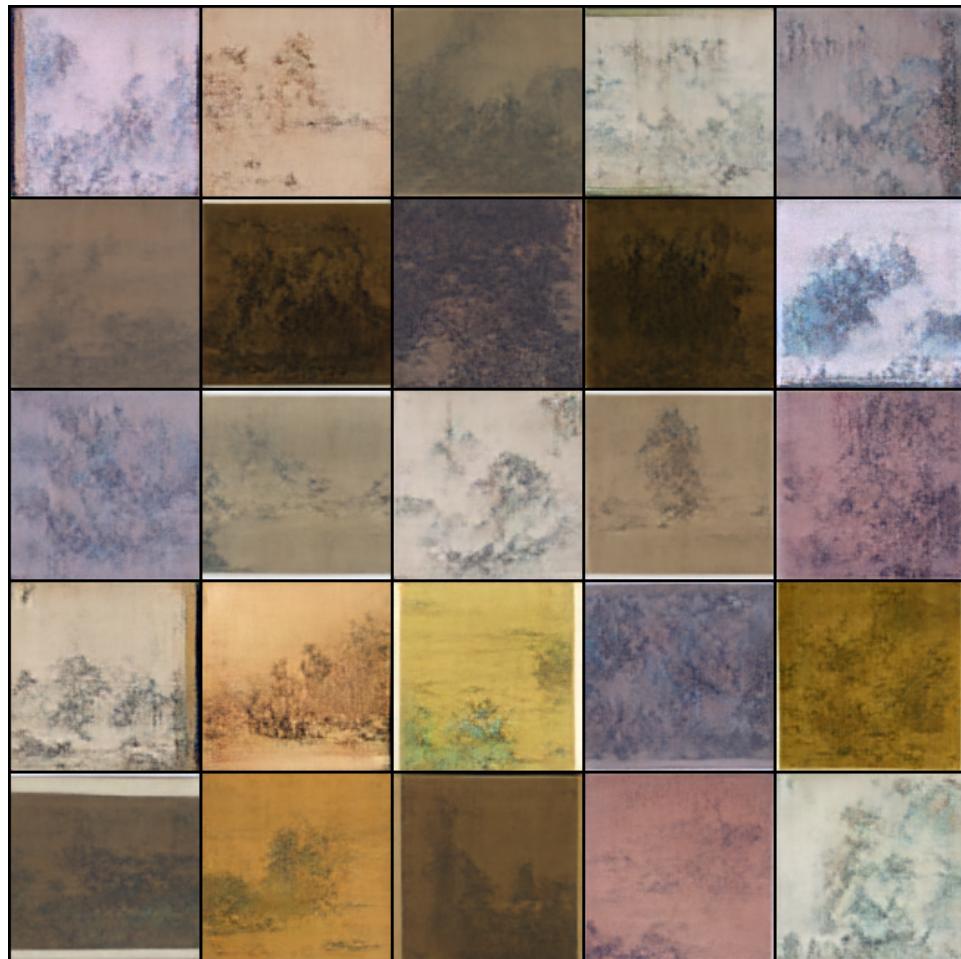


图 3.7 实验 4-深色中国画数据集生成中国画采样图

#### 实验配置

使用深色中国画数据集，在 A100- 40GB GPU 上，以 64 为训练批量训练约 36 小时。

#### 实验结果与分析

图3.7为训练 33 轮后的采样图像。实验 4 与实验 2 类似，训练批量都为 64，都在 A100- 40GB GPU 上训练。不同的是，实验 4 使用深色中国画数据集，而实验 2 使用浅色中国画数据集。实验 4 相对于实验 1，得到的图像与中国画更相似；相比于实验 2，生成的中国画的颜色更加丰富。

## 结论

本文对各种主流生成模型进行了探索，对生成模型进行了分类，随后将扩散模型应用在中国画生成领域。

可以根据是否直接定义概率密度函数，将生成模型分为显式密度模型和隐式密度模型。显式密度模型可以直接表示似然函数，其中，似然函数可以直接求解的模型主要有规范化流模型和自回归模型，通过近似方法求解似然函数的模型主要有基于能量的模型、变分自编码器和扩散模型。隐式密度模型不通过似然函数进行训练，而使用其他方式与数据分布进行交互，生成对抗网络是隐式密度模型的主要代表。

使用扩散模型生成中国画，在计算资源有限的情况下，根据中国画不同的风格训练不同的扩散模型，有助于提升中国画生成质量与加快模型收敛速度，提高训练批量也有助于提高生成中国画的质量。将生成模型应用于中国画生成有助于继承和弘扬中华优秀传统文化。

近年来生成模型发展迅速，但有关生成模型整体性介绍的资料仍然缺少，而对生成模型领域的整体性把握，有助于以系统的思维设计生成模型算法，有利于对生成模型的研究。更广泛地说，2006 年《模式识别与机器学习》一书成为机器学习领域经典书籍，2016《深度学习》一书促进了深度学习的知识普及与发展，生成模型领域仍缺少一本系统性书籍来促进生成模型的研究与应用，一本关于“生成模型”的书籍是值得期待的。此外，未来可考虑获取更多中国画或使用数据增强技术，以增大训练集来获得更好的图像生成效果。此外，还可考虑将生成模型应用于其他传统文化，如唐诗宋词、古典乐曲、皮影戏等，以继承和弘扬中华优秀传统文化。

## 参考文献

- [1] 百度百科. 中国画[Z]. <https://baike.baidu.com/item/%E4%B8%AD%E5%9B%BD%E7%94%BB/197394>. 2023 年 4 月 28 日引用.
- [2] MACHINERY C. Computing machinery and intelligence-AM Turing[J]. Mind, 1950, 59(236): 433.
- [3] 中国信通院, 京东探索研究院. 人工智能生成内容 (AIGC) 白皮书[Z]. 2022.
- [4] GOODFELLOW I, BENGIO Y, COURVILLE A. Deep learning[M]. MIT press, 2016.
- [5] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. nature, 2015, 521(7553): 436-444.
- [6] KINGMA D P, WELLING M. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114, 2013.
- [7] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.
- [8] RADFORD A, METZ L, CHINTALA S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.
- [9] ARJOVSKY M, CHINTALA S, BOTTOU L. Wasserstein generative adversarial networks[C]//International conference on machine learning. 2017: 214-223.
- [10] KARRAS T, AILA T, LAINE S, et al. Progressive growing of gans for improved quality, stability, and variation[J]. arXiv preprint arXiv:1710.10196, 2017.
- [11] ISOLA P, ZHU J Y, ZHOU T, et al. Image-to-image translation with conditional adversarial networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1125-1134.
- [12] ZHU J Y, PARK T, ISOLA P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2223-2232.

- [13] DONG H W, HSIAO W Y, YANG L C, et al. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment [C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 32. 2018.
- [14] LARSEN A B L, SØNDERBY S K, LAROCHELLE H, et al. Autoencoding beyond pixels using a learned similarity metric[C]//International conference on machine learning. 2016: 1558-1566.
- [15] RAZAVI A, VAN DEN OORD A, VINYALS O. Generating diverse high-fidelity images with vq-vae-2[J]. Advances in neural information processing systems, 2019, 32.
- [16] HA D, SCHMIDHUBER J. World models[J]. arXiv preprint arXiv:1803.10122, 2018.
- [17] VAN DEN OORD A, KALCHBRENNER N, KAVUKCUOGLU K. Pixel recurrent neural networks[C]//International conference on machine learning. 2016: 1747-1756.
- [18] VAN DEN OORD A, KALCHBRENNER N, ESPEHOLT L, et al. Conditional image generation with pixelcnn decoders[J]. Advances in neural information processing systems, 2016, 29.
- [19] DINH L, SOHL-DICKSTEIN J, BENGIO S. Density estimation using real nvp[J]. arXiv preprint arXiv:1605.08803, 2016.
- [20] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [21] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training[J]. journal, 2018.
- [22] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
- [23] HUANG C Z A, VASWANI A, USZKOREIT J, et al. Music transformer[J]. arXiv preprint arXiv:1809.04281, 2018.
- [24] ZHANG H, GOODFELLOW I, METAXAS D, et al. Self-attention generative adversarial networks[C]//International conference on machine learning. 2019: 7354-7363.

- [25] BROCK A, DONAHUE J, SIMONYAN K. Large scale GAN training for high fidelity natural image synthesis[J]. arXiv preprint arXiv:1809.11096, 2018.
- [26] KARRAS T, LAINE S, AILA T. A style-based generator architecture for generative adversarial networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 4401-4410.
- [27] KARRAS T, LAINE S, AITTALA M, et al. Analyzing and improving the image quality of stylegan[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 8110-8119.
- [28] SONG Y, ERMON S. Generative modeling by estimating gradients of the data distribution[J]. Advances in neural information processing systems, 2019, 32.
- [29] ESSER P, ROMBACH R, OMMER B. Taming transformers for high-resolution image synthesis[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 12873-12883.
- [30] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.
- [31] SAUER A, SCHWARZ K, GEIGER A. Stylegan-xl: Scaling stylegan to large diverse datasets[C]//ACM SIGGRAPH 2022 conference proceedings. 2022: 1-10.
- [32] HO J, JAIN A, ABBEEL P. Denoising diffusion probabilistic models[J]. Advances in Neural Information Processing Systems, 2020, 33: 6840-6851.
- [33] SONG J, MENG C, ERMON S. Denoising diffusion implicit models[J]. arXiv preprint arXiv:2010.02502, 2020.
- [34] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners [J]. Advances in neural information processing systems, 2020, 33: 1877-1901.
- [35] SMITH S, PATWARY M, NORICK B, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model[J]. arXiv preprint arXiv:2201.11990, 2022.
- [36] THOPPILAN R, DE FREITAS D, HALL J, et al. Lamda: Language models for dialog applications[J]. arXiv preprint arXiv:2201.08239, 2022.

- [37] RAMESH A, PAVLOV M, GOH G, et al. Zero-shot text-to-image generation[C]// International Conference on Machine Learning. 2021: 8821-8831.
- [38] NICHOL A, DHARIWAL P, RAMESH A, et al. Glide: Towards photorealistic image generation and editing with text-guided diffusion models[J]. arXiv preprint arXiv:2112.10741, 2021.
- [39] RAMESH A, DHARIWAL P, NICHOL A, et al. Hierarchical text-conditional image generation with clip latents[J]. arXiv preprint arXiv:2204.06125, 2022.
- [40] SAHARIA C, CHAN W, SAXENA S, et al. Photorealistic text-to-image diffusion models with deep language understanding[J]. Advances in Neural Information Processing Systems, 2022, 35: 36479-36494.
- [41] YU J, XU Y, KOH J Y, et al. Scaling autoregressive models for content-rich text-to-image generation[J]. arXiv preprint arXiv:2206.10789, 2022.
- [42] ROMBACH R, BLATTMANN A, LORENZ D, et al. High-resolution image synthesis with latent diffusion models[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 10684-10695.
- [43] GOODFELLOW I. Nips 2016 tutorial: Generative adversarial networks[J]. arXiv preprint arXiv:1701.00160, 2016.
- [44] WENG L. Flow-based Deep Generative Models[J/OL]. [lilianweng.github.io](https://lilianweng.github.io/posts/2018-10-13-flow-models/), 2018. <https://lilianweng.github.io/posts/2018-10-13-flow-models/>.
- [45] TOMCZAK J M. Deep Generative Modeling[M]. Springer Cham, 2022.
- [46] LIPPE P. UvA Deep Learning Tutorials[Z]. <https://uvadlc-notebooks.readthedocs.io/en/latest/>. 2022.
- [47] WENG L. From Autoencoder to Beta-VAE[J/OL]. [lilianweng.github.io](https://lilianweng.github.io/posts/2018-08-12-vae/), 2018. <https://lilianweng.github.io/posts/2018-08-12-vae/>.
- [48] VINCENT P, LAROCHELLE H, BENGIO Y, et al. Extracting and composing robust features with denoising autoencoders[C]//Proceedings of the 25th international conference on Machine learning. 2008: 1096-1103.
- [49] LUO C. Understanding Diffusion Models: A Unified Perspective[J]. ArXiv, 2022, abs/2208.11970.

- [50] SOHL-DICKSTEIN J, WEISS E, MAHESWARANATHAN N, et al. Deep unsupervised learning using nonequilibrium thermodynamics[C]//International Conference on Machine Learning. 2015: 2256-2265.
- [51] KINGMA D P, SALIMANS T, POOLE B, et al. Variational Diffusion Models[Z]. 2023. arXiv: 2107.00630 [cs.LG].
- [52] SALIMANS T, GOODFELLOW I, ZAREMBA W, et al. Improved techniques for training gans[J]. Advances in neural information processing systems, 2016, 29.
- [53] HEUSEL M, RAMSAUER H, UNTERTHINER T, et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium[J]. Advances in neural information processing systems, 2017, 30.
- [54] BIŃKOWSKI M, SUTHERLAND D J, ARBEL M, et al. Demystifying mmd gans [J]. arXiv preprint arXiv:1801.01401, 2018.
- [55] BETZALEL E, PENSO C, NAVON A, et al. A Study on the Evaluation of Generative Models[Z]. 2022. arXiv: 2206.10935 [cs.LG].
- [56] XUE A. End-to-End Chinese Landscape Painting Creation Using Generative Adversarial Networks[Z]. 2020. arXiv: 2011.05552 [cs.CV].

## 致谢

首先，感谢指导教师汪鹏老师，感谢汪鹏老师给予的关于毕业设计的灵活度，很幸运能在汪鹏老师指导下做出这样一份有意义的毕业设计，衷心感谢汪鹏老师的指导和教诲。

其次，感谢河北工业大学的教师们，尤其是智能学院的老师们，感谢老师们授予宝贵知识，感谢老师们给予的无私帮助。

再要感谢河北工业大学，回首大学时光，最重要的转折点莫过于转专业至计算机专业，感谢学校给予的宝贵转专业机会，让我能够学习更感兴趣的知识。感谢学校提供的优良学习环境。

最后要感谢我的父母，感谢父母的付出与支持。

## 附录 A 基础知识

### A.1 数学

#### A.1.1 线性代数

##### 对角矩阵

除主对角线之外的元素皆为0 的矩阵。

##### 矩阵乘法

如果  $A$  是一个  $l \times m$  矩阵,  $B$  是一个  $m \times n$  矩阵, 则  $AB$  是一个  $l \times n$  矩阵。

##### 矩阵的迹

矩阵的迹为矩阵所有对角元素之和, 即:

$$Tr(A) = \sum_i A_{i,i} \quad (\text{A.1})$$

##### 范数

范数可以表示向量的大小, 范数定义为:

$$\|\boldsymbol{x}\|_p = \left( \sum_i |x_i|^p \right)^{\frac{1}{p}} \quad (\text{A.2})$$

式A.2中,  $p \in \mathbb{R}, p \geq 1$

$L^1$  范数为:

$$\|\boldsymbol{x}\|_1 = \sum_i |x_i| \quad (\text{A.3})$$

最大范数为:

$$\|\boldsymbol{x}\|_\infty = \max_i |x_i| \quad (\text{A.4})$$

矩阵的 Frobenius 范数为:

$$\|\boldsymbol{x}\|_F = \sqrt{\sum_{i,j} A_{i,j}^2} = \sqrt{Tr(\boldsymbol{A}\boldsymbol{A}^T)} \quad (\text{A.5})$$

### 矩阵行列式

矩阵的行列式是一个关于方形矩阵（方阵）内所有元素的标量函数值。一个  $n \times n$  的矩阵  $M$  的行列式为：

$$\det M = \det \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \sum_{j_1 j_2 \cdots j_n} (-1)^{\tau(j_1 j_2 \cdots j_n)} a_{1j_1} a_{2j_2} \cdots a_{nj_n} \quad (\text{A.6})$$

式 A.6 中，求和符号  $\sum$  的下标  $j_1 j_2 \cdots j_n$  表示集合  $1, 2, \dots, n$  的全排列，即共有  $n$  项； $\tau(j_1 j_2 \cdots j_n)$  表示排列  $j_1 j_2 \cdots j_n$  的符号，排列的符号定义为式 A.83。

方阵的行列式可以用来判断矩阵  $M$  是否可逆：如果  $\det M = 0$ ，那么矩阵不可逆。

矩阵乘积的行列式等于矩阵行列式的乘积，即：

$$\det(AB) = \det(A) \det(B) \quad (\text{A.7})$$

### 可逆矩阵

如果一个  $n \times n$  的矩阵  $A$  可逆，那么存在  $n \times n$  矩阵  $B$  使得：

$$AB = BA = I_n \quad (\text{A.8})$$

式 A.8 中， $I_n$  为单位矩阵，并且使用矩阵乘法。

不可逆的方阵又称为奇异矩阵或退化矩阵。

对于可逆矩阵有：

$$\det(M) \det(M^{-1}) = \det(MM^{-1}) = \det(I) = 1 \quad (\text{A.9})$$

因此，

$$\det(M^{-1}) = (\det(M))^{-1} \quad (\text{A.10})$$

### 雅可比矩阵

假设某函数  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ，从  $\mathbf{x} \in \mathbb{R}^n$  映射到向量  $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$ ，这个函数的一阶偏导矩阵称为雅可比矩阵，是一个  $m \times n$  的矩阵。其第  $i$  行，第  $j$  列的值为  $J_{ij} = \frac{\partial f_i}{\partial x_j}$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (\text{A.11})$$

### A.1.2 概率论

#### 概率密度函数

一个连续随机变量 $X$  的概率可以由概率密度函数表示:

$$p(X \in A) = \int_A f_X(x) dx \quad (\text{A.12})$$

#### 概率质量函数

一个离散随机变量 $X$  的概率可以由概率质量函数表示:

$$p_X(x) = p(\{X = x\}) \quad (\text{A.13})$$

#### 正态分布

若实值随机变量 $X$  服从正态分布(高斯分布), 则其概率密度函数为:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad (\text{A.14})$$

记为:  $X \sim \mathcal{N}(\mu, \sigma^2)$

两高斯分布之和仍为高斯分布, 假设 $X, Y$  为两独立随机变量, 且

$$X \sim \mathcal{N}(\mu_X, \sigma_X^2) \quad (\text{A.15})$$

$$Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2) \quad (\text{A.16})$$

$$(A.17)$$

则 $X, Y$  之和仍为高斯分布, 即:

$$Z = X + Y \quad (\text{A.18})$$

$$Z \sim \mathcal{N}(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2) \quad (\text{A.19})$$

#### 期望

当 $x \sim p(x)$ , 函数 $f(x)$  的期望为:

$$\mathbb{E}_{x \sim p}[f(x)] = \sum_x p(x)f(x) \quad (\text{A.20})$$

或

$$\mathbb{E}_{x \sim p}[f(x)] = \int p(x)f(x) dx \quad (\text{A.21})$$

## 协方差

随机变量  $X, Y$  的协方差为:

$$\text{cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \quad (\text{A.22})$$

## 极大似然估计

假设  $\theta$  为未知参数 (标量或者矢量), 对于服从联合概率质量函数  $p_X(\mathbf{x}; \theta)$  的一组观测向量  $X = X_1, \dots, X_n$ , 假设我们有  $X$  的具体的观测值  $\mathbf{x} = (x_1, \dots, x_n)$ 。那么, 其极大似然估计是未知参数  $\theta$  的一个取值, 该取值能够使函数  $p_X(x_1, \dots, x_n; \theta)$  取得最大值。

$$\hat{\theta}_n = \arg \max_{\theta} p_X(\mathbf{x}; \theta) = \arg \max_{\theta} p_X(x_1, \dots, x_n; \theta) \quad (\text{A.23})$$

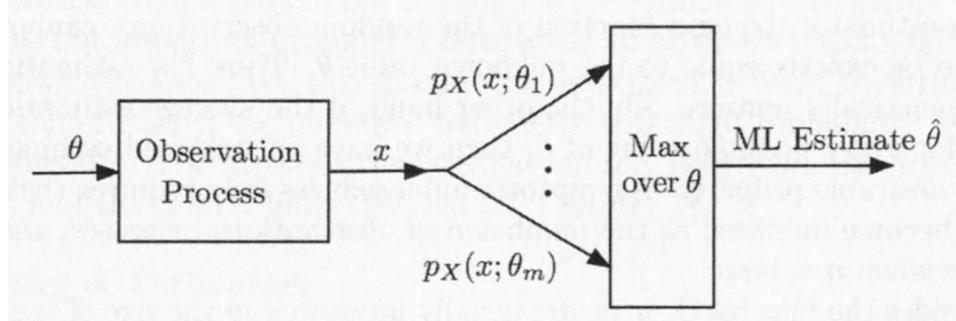


图 A.1 极大似然估计示意图

图 A.1 中, 假设  $X$  为离散变量, 未知参数  $\theta$  可以从  $\theta_1, \dots, \theta_m$  中选取。给定观测值  $X = \mathbf{x}$ , 对于每个  $\theta_i$  取值, 都可以计算  $p_X(\mathbf{x}; \theta_i)$ 。使函数  $p_X(\mathbf{x}; \theta)$  取得最大值的  $\theta_i$  即为极大似然估计  $\theta$ 。

在很多情况下, 都假定观测向量中的每一个  $X_i$  为互相独立的, 因此, 似然函数通常可以改写为:

$$p_X(x_1, \dots, x_n; \theta) = \prod_{i=1}^n p_{X_i}(x_i; \theta) \quad (\text{A.24})$$

为了分析与计算方便, 可以将其改写为对数似然函数:

$$\log p_X(x_1, \dots, x_n; \theta) = \log \prod_{i=1}^n p_{X_i}(x_i; \theta) = \sum_{i=1}^n \log p_{X_i}(x_i; \theta) \quad (\text{A.25})$$

当  $X$  为连续变量时, 由概率密度函数替换概率质量函数可得:

$$\log f_X(x_1, \dots, x_n; \theta) = \log \prod_{i=1}^n f_{X_i}(x_i; \theta) = \sum_{i=1}^n \log f_{X_i}(x_i; \theta) \quad (\text{A.26})$$

值得注意的是, 对于  $X$  的观测值  $\mathbf{x}$ , 其似然函数  $p_X(\mathbf{x}; \theta)$  并不是指未知参数取值为  $\theta$  的概率, 而是指在未知参数取值为  $\theta$  时,  $X$  的观测值为  $\mathbf{x}$  的概率。

### 边缘似然

边缘似然是似然函数在参数空间上的积分，表示生成观测样本的概率，因此，边缘似然也被称为模型证据，简称为证据。

### 全概率公式

全概率公式将对一复杂事件 A 的概率求解问题转化为了在不同情况下发生的简单事件的概率的求和问题。

若事件  $B_1, B_2, \dots, B_n$  构成一个完备事件组且都有正概率，则有，

$$p(A) = p(AB_1) + p(AB_2) + \dots + p(AB_n) \quad (\text{A.27})$$

$$= p(A|B_1)p(B_1) + p(A|B_2)p(B_2) + \dots + p(A|B_n)p(B_n) \quad (\text{A.28})$$

### 贝叶斯定理

若事件  $B_1, B_2, \dots, B_n$  构成一个完备事件组且都有正概率，则有，

$$p(B_i|A) = \frac{p(A|B_i)p(B_i)}{p(A)} \quad (\text{A.29})$$

$$= \frac{p(A|B_i)p(B_i)}{p(A|B_1)p(B_1) + p(A|B_2)p(B_2) + \dots + p(A|B_n)p(B_n)} \quad (\text{A.30})$$

在式A.29与A.30中， $B_i$  通常表示一个命题，如“硬币正面朝上的次数占投掷次数的 50%”； $A$  通常表示事实，如“连续多次投掷硬币的结果”。 $p(B_i)$  表示 $B_i$  的先验概率，先验概率为不考虑事实 $A$  时，人们对事件 $B_i$  的相信程度，其也包含了人们对于 $B_i$  的先验知识。 $p(A|B_i)$  为似然函数，表示当命题 $B_i$  发生时，事实 $A$  发生的概率。“似然”表达了事件 $A$  对命题 $B_i$  的支撑程度。 $p(B_i|A)$  为 $B_i$  的后验概率，表示考虑事实 $A$  后，命题 $B_i$  发生的概率。贝叶斯定理根据事实 $B_i$ ，对先验概率 $p(B_i)$  进行更新。

### 贝叶斯推理

贝叶斯推理可以由先验概率和似然函数得出后验概率，其中先验概率和似然函数皆由统计模型关于观测数据产生。

$$p(H | E) = \frac{p(E | H)p(H)}{p(E)} \quad (\text{A.31})$$

式A.31中：

- $H$  为假设，其概率受数据（以下称为证据）影响。
- $p(H)$  为先验概率，是在获得数据 $E$  前，对假设 $H$  概率的估计， $E$  为当前证据。

- $E$  为证据，即未参与先验概率  $p(H)$  计算的数据。
- $p(H | E)$  为后验概率，给出证据  $E$  后， $H$  为真的概率，是关于  $H$  的函数。
- $p(E | H)$  为似然函数，给出假设  $H$  后观测到  $E$  的概率，是关于  $E$  的函数。
- $p(E)$  为边缘似然，也称为模型证据。表示从先验概率中获得观测样本的概率。

### 蒙特卡洛方法

蒙特卡洛方法是依赖于随机采样来获得数值结果的一种计算方法。其底层思想是，用随机方法来解决原理上具备确定性的问题。在物理和数学上，当其他方法不可用时，常常采用蒙特卡洛方法。蒙特卡洛方法主要应用于三类问题：优化、数值积分与从概率分布中采样。

当无法精确求和或者计算积分时，通常使用蒙特卡洛采样方法来近似。基本思想为，将其和或者积分视为某分布的期望，通过相应的计算来近似该期望。如：

$$s = \sum_{\mathbf{x}} p(\mathbf{x}) f(\mathbf{x}) = E_p[f(\mathbf{x})] \quad (\text{A.32})$$

$$s = \int p(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} = E_p[f(\mathbf{x})] \quad (\text{A.33})$$

式 A.32 中， $p(\mathbf{x})$  为随机变量  $\mathbf{x}$  的概率分布，式 A.33 中， $p(\mathbf{x})$  为随机变量  $\mathbf{x}$  的概率密度。

### 随机微分方程

随机微分方程（Stochastic differential equation）是添加了一项或多项随机项的微分方程。

### 对比散度

用函数  $f(x; \boldsymbol{\theta})$  来为数据点的概率分布建模，其中， $x$  为模型的输入， $\boldsymbol{\theta}$  为模型的参数，且要保证概率积分为 1 的性质，即：

$$p(x; \boldsymbol{\theta}) = \frac{f(x; \boldsymbol{\theta})}{Z(\boldsymbol{\theta})} \quad (\text{A.34})$$

式 A.34 中， $Z(\boldsymbol{\theta})$  为划分函数：

$$Z(\boldsymbol{\theta}) = \int f(x; \boldsymbol{\theta}) dx \quad (\text{A.35})$$

假定数据点集合为  $\mathbf{x} = x_1, \dots, x_K$ ，则其似然函数为：

$$p(\mathbf{x}; \boldsymbol{\theta}) = \prod_{k=1}^K \frac{f(x_k; \boldsymbol{\theta})}{Z(\boldsymbol{\theta})} \quad (\text{A.36})$$

极大化似然函数A.36等价于最小化负对数似然函数，即能量函数 $E(\mathbf{x}; \boldsymbol{\theta})$ :

$$E(\mathbf{x}; \boldsymbol{\theta}) = \log Z(\boldsymbol{\theta}) - \frac{1}{K} \sum_{i=1}^K \log f(x_i; \boldsymbol{\theta}) \quad (\text{A.37})$$

对式A.37关于参数 $\boldsymbol{\theta}$ 求偏导，即：

$$\frac{\partial E(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial \log Z(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} - \frac{1}{K} \sum_{i=1}^K \frac{\partial \log f(x_i; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \quad (\text{A.38})$$

$$= \frac{\partial \log Z(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} - \mathbb{E}_{p_{data}} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (\text{A.39})$$

$$= \frac{1}{Z(\boldsymbol{\theta})} \frac{\partial Z(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} - \mathbb{E}_{p_{data}} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (\text{A.40})$$

$$= \frac{1}{Z(\boldsymbol{\theta})} \frac{\partial \int f(x; \boldsymbol{\theta}) dx}{\partial \boldsymbol{\theta}} - \mathbb{E}_{p_{data}} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (\text{A.41})$$

$$= \frac{1}{Z(\boldsymbol{\theta})} \int \frac{\partial f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} dx - \mathbb{E}_{p_{data}} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (\text{A.42})$$

$$= \frac{1}{Z(\boldsymbol{\theta})} \int f(x; \boldsymbol{\theta}) \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} dx - \mathbb{E}_{p_{data}} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (\text{A.43})$$

$$= \int p(x; \boldsymbol{\theta}) \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} dx - \mathbb{E}_{p_{data}} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (\text{A.44})$$

$$= \mathbb{E}_{p(x; \boldsymbol{\theta})} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] - \mathbb{E}_{p_{data}} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (\text{A.45})$$

式A.45即为对比散度的梯度，对于其等号右侧第一项

$$\mathbb{E}_{p(x; \boldsymbol{\theta})} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (\text{A.46})$$

可以通过多次循环使用马尔科夫链蒙特卡洛采样来将训练集数据转化为从 $p(x; \boldsymbol{\theta})$ 分布中的采样。假设 $\mathbf{x}^n$ 表示对训练样本数据 $\mathbf{x}$ 使用 $n$ 次马尔科夫链蒙特卡洛采样获得数据，可令 $\mathbf{x}^0 = \mathbf{x}$ ，即得：

$$\frac{\partial E(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbb{E}_{x^\infty} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] - \mathbb{E}_{x^0} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (\text{A.47})$$

对于式A.47，在机器学习中，即使使用一次马尔科夫链蒙特卡洛采样也可以取得很好得效果，参数更新式为：

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta \left( \mathbb{E}_{x^0} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] - \mathbb{E}_{x^1} \left[ \frac{\partial \log f(x; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \right) \quad (\text{A.48})$$

### 证据下界

显式密度模型需要表示出概率密度函数，假定观测变量 $\mathbf{x}$ 和隐变量 $\mathbf{z}$ 构成联合概率分布 $p(\mathbf{x}, \mathbf{z})$ ，概率密度函数 $p(\mathbf{x})$ 可表示为：

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} \quad (\text{A.49})$$

或使用概率链式法则：

$$p(\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z} | \mathbf{x})} \quad (\text{A.50})$$

对于式 A.49，复杂模型无法对所有隐变量  $\mathbf{z}$  进行积分；对于式 A.50，因无法获得实际后验分布  $p(\mathbf{z} | \mathbf{x})$  而也无法直接计算。结合式 A.49 与式 A.50，有：

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} \quad (\text{根据式 A.49}) \quad (\text{A.51})$$

$$= \log \int p(\mathbf{x}, \mathbf{z}) \frac{q_\phi(\mathbf{z} | \mathbf{x})}{q_\phi(\mathbf{z} | \mathbf{x})} d\mathbf{z} \quad (\text{A.52})$$

$$= \log \int \frac{p(\mathbf{x}, \mathbf{z}) q_\phi(\mathbf{z} | \mathbf{x})}{q_\phi(\mathbf{z} | \mathbf{x})} d\mathbf{z} \quad (\text{A.53})$$

$$= \log \int q_\phi(\mathbf{z} | \mathbf{x}) \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} | \mathbf{x})} d\mathbf{z} \quad (\text{A.54})$$

$$= \log \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} \left[ \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} | \mathbf{x})} \right] \quad (\text{根据期望定义式 A.21}) \quad (\text{A.55})$$

$$\geq \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} | \mathbf{x})} \right] \quad (\text{根据杰森不等式 A.89}) \quad (\text{A.56})$$

可获得观测变量  $\mathbf{x}$  对数似然函数的证据下界 (Evidence Lower BOund, ELBO)：

$$\mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} | \mathbf{x})} \right] \quad (\text{A.57})$$

其中， $q_\phi(\mathbf{z} | \mathbf{x})$  是由参数  $\phi$  确定的近似变分分布，最大化证据下界即优化参数  $\phi$ 。最大化证据下界可以取得与极大似然估计相近的效果。

此外，对证据下界的另一种证明方式能够体现最大化证据下界的原因，

$$\begin{aligned} \log p(\mathbf{x}) &= \log p(\mathbf{x}) \int q_\phi(\mathbf{z} | \mathbf{x}) d\mathbf{z} \quad (1 = \int q_\phi(\mathbf{z} | \mathbf{x}) d\mathbf{z}) \\ &= \int q_\phi(\mathbf{z} | \mathbf{x}) (\log p(\mathbf{x})) d\mathbf{z} \quad (\text{不改变积分值}) \end{aligned} \quad (\text{A.58})$$

$$\begin{aligned} &= \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} [\log p(\mathbf{x})] \quad (\text{期望定义式 A.21}) \\ &\quad (\text{A.60}) \end{aligned}$$

$$\begin{aligned} &= \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z} | \mathbf{x})} \right] \quad (\text{根据式 A.50}) \\ &\quad (\text{A.61}) \end{aligned}$$

$$\begin{aligned} &= \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z}) q_\phi(\mathbf{z} | \mathbf{x})}{p(\mathbf{z} | \mathbf{x}) q_\phi(\mathbf{z} | \mathbf{x})} \right] \quad (1 = \frac{q_\phi(\mathbf{z} | \mathbf{x})}{q_\phi(\mathbf{z} | \mathbf{x})}) \\ &\quad (\text{A.62}) \end{aligned}$$

$$= \mathbb{E}_{q_\phi(z|x)} \left[ \log \frac{p(\mathbf{x}, z)}{q_\phi(z|\mathbf{x})} \right] + \mathbb{E}_{q_\phi(z|x)} \left[ \log \frac{q_\phi(z|\mathbf{x})}{p(z|\mathbf{x})} \right] \quad (\text{期望拆分}) \quad (\text{A.63})$$

$$= \mathbb{E}_{q_\phi(z|x)} \left[ \log \frac{p(\mathbf{x}, z)}{q_\phi(z|\mathbf{x})} \right] + D_{KL}(q_\phi(z|\mathbf{x}) || p(z|\mathbf{x})) \quad (\text{散度定义式 A.70}) \quad (\text{A.64})$$

$$\geq \mathbb{E}_{q_\phi(z|x)} \left[ \log \frac{p(\mathbf{x}, z)}{q_\phi(z|\mathbf{x})} \right] \quad (\text{根据式 A.76}) \quad (\text{A.65})$$

根据式 A.64，对数似然函数  $\log p(\mathbf{x})$  即为证据下界与近似后验  $q_\phi(z|\mathbf{x})$  和真实后验  $p(z|\mathbf{x})$  的 KL 散度，KL 散度项即为式 A.56 中被移除的项。由于对数似然函数  $\log p(\mathbf{x})$  与证据下界的差值仅为非负的 KL 散度项，因此，证据下界的值不可能超过对数似然函数  $\log p(\mathbf{x})$  的值。通过最小化 KL 散度项，即可使变分后验  $q_\phi(z|\mathbf{x})$  更接近于真实后验  $p(z|\mathbf{x})$ ，但由于无法获得真实后验  $p(z|\mathbf{x})$ ，无法直接对 KL 散度项最小化。而注意到，式 A.64 左侧证据下界项中， $p(\mathbf{x}, z)$  与参数  $\phi$  无关，对其关于变量  $z$  计算边缘概率所得  $p(\mathbf{x})$  也因此与参数  $\phi$  无关，即  $\log p(\mathbf{x})$  与参数  $\phi$  无关。因此，在改变参数  $\phi$  时，证据下界与 KL 散度项的和为定值，对证据下界项的最大化即代表了 KL 散度项的最小化。对证据下界项的优化程度可以体现模型对隐变量后验概率的拟合程度，证据下界项优化程度越高，近似后验则越接近真实后验。此外，由于证据下界是对模型证据  $\log p(\mathbf{x})$  的近似，模型经过训练后，证据下界也可以作为对观测数据或生成数据似然的估计。

### 重参数方法

从均值为  $\mu$ ，方差为  $\sigma^2$  的正态分布  $x \sim \mathcal{N}(\mu, \sigma^2)$  中采样过程可以改写为

$$x = \mu + \sigma \epsilon \quad \text{其中 } \epsilon \sim \mathcal{N}(0, 1) \quad (\text{A.66})$$

使用重参数方法，从任意高斯分布取样可以变为从标准高斯分布取样，即将标准高斯分布采样取值根据  $\sigma$  对进行伸缩变换，再根据  $\mu$  进行平移变换。

### A.1.3 信息论

#### 信息论基础

对信息论最直觉的理解是，某一事件发生携带的信息量，与其发生的概率成反相关。也就是说，低概率事件的发生，要比更高概率事件的发生携带更多的信息。如“今

天早上有日食”相比“今天早上太阳照常升起”携带更多的信息。因此，对事件携带信息的量化也需要满足这种直觉。即：

- 更高概率发生的事件携带更少的信息，必定发生的事件不携带信息。
- 更低概率发生的事件携带更多的信息。
- 相互独立的事件携带的信息具有可加性。

由此，可以定义事件  $x$  携带的信息为：

$$I(x) = -\log p(x) \quad (\text{A.67})$$

式 A.67 中， $\log$  表示以自然数  $e$  为底数的自然对数。

### 香农熵

香农熵可以量化事件概率分布中所含的不确定度：

$$H(x) = \mathbb{E}_{x \sim p}[I(x)] = -\mathbb{E}_{x \sim p}[\log p(x)] \quad (\text{A.68})$$

### Kullback-Leibler (KL) 散度

对于一随机变量  $x$  的两个概率分布  $p(x)$  和  $q(x)$ ，KL 散度可以测量这两个概率分布的不同程度。

$$D_{KL}(p \parallel q) = \mathbb{E}_{x \sim p}[\log \frac{p(x)}{q(x)}] \quad (\text{A.69})$$

$$= \mathbb{E}_{x \sim p}[\log p(x) - \log q(x)] \quad (\text{A.70})$$

对式 A.69 进行变形：

$$-D_{KL}(p \parallel q) = -(\mathbb{E}_{x \sim p}[\log \frac{p(x)}{q(x)}]) \quad (\text{A.71})$$

$$= \sum_x p(x) \log \frac{q(x)}{p(x)} \quad (\text{A.72})$$

$$\leq \sum_x p(x) \left( \frac{q(x)}{p(x)} - 1 \right) \quad (\text{A.73})$$

$$= \sum_x q(x) - \sum_x p(x) \quad (\text{A.74})$$

$$= 1 - 1 \quad (\text{A.75})$$

$$= 0 \quad (\text{A.76})$$

因此，KL 散度具有非负性。

对于两高斯分布，其 KL 散度为：

$$\begin{aligned} & D_{KL}(\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_x, \Sigma_x) \| \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_y, \Sigma_y)) \\ &= \frac{1}{2} \left[ \log \frac{|\Sigma_y|}{|\Sigma_x|} - d + \text{tr}(\Sigma_y^{-1} \Sigma_x) + (\boldsymbol{\mu}_y - \boldsymbol{\mu}_x)^\top \Sigma_y^{-1} (\boldsymbol{\mu}_y - \boldsymbol{\mu}_x) \right] \end{aligned} \quad (\text{A.77})$$

### 交叉熵

交叉熵与 KL 散度相近，相比于 KL 散度，交叉熵增加了概率分布  $p(x)$  的香农熵  $H(p)$ ：

$$H(p, q) = H(p) + D_{KL}(p \| q) \quad (\text{A.78})$$

$$= -\mathbb{E}_{x \sim p} \log q(x) \quad (\text{A.79})$$

因为被省略的项  $H(p)$  与  $q(x)$  无关，所以关于  $q(x)$  最小化交叉熵  $H(p, q)$  等价于最小化 KL 散度  $D_{KL}(p \| q)$ 。

### 信噪比

信噪比（Signal-to-noise ratio, SNR, S/N）为有价值信号和背景噪声的比值。

$$SNR = \frac{\mu^2}{\sigma^2} \quad (\text{A.80})$$

式 A.80 中， $\mu$  为均值， $\sigma$  为方差。

## A.1.4 其他

### 配方法

若有抛物线  $y = ax^2 + bx + c (a \neq 0)$ ，则可使用配方法求其顶点与对称轴，

$$y = ax^2 + bx + c \quad (\text{A.81})$$

$$= a\left(x + \frac{b}{2a}\right)^2 + \frac{4ac - b^2}{4a} \quad (\text{A.82})$$

由式 A.82，抛物线  $y = ax^2 + bx + c (a \neq 0)$  的对称轴为  $x = -\frac{b}{2a}$ ，顶点坐标为  $(-\frac{b}{2a}, \frac{4ac - b^2}{4a})$ 。

## 排列

排列指对一个集合中所有元素的一种排列。如果集合 $\mathbb{X}$ 确定了集合内部元素的顺序，其每一种排列 $\sigma$ 可以看成对原序列内部元素进行多次置换，置换会增加集合内逆序对（两元素与原确定顺序不同）的个数，排列 $\sigma$ 的符号即为该排列下逆序对的个数。

排列的符号由 $sgn(\sigma)$ 表示，可以表示为：

$$sgn(\sigma) = (-1)^{N(\sigma)} \quad (\text{A.83})$$

其中 $N(\sigma)$ 表示排列 $\sigma$ 中逆序对的个数、

## S 型函数

S型函数的曲线为S型，常见的S型函数如逻辑斯蒂函数为：

$$sigmoid(x) = \frac{1}{1 + \exp(-x)} = \frac{\exp(x)}{\exp(x) + 1} \quad (\text{A.84})$$

## 换元定理在概率中的应用

若有随机变量 $z \sim \pi(z)$ ，一一映射函数 $x = f(z)$ ，函数 $f$ 可逆，即 $z = f^{-1}(x)$ 。根据概率分布的定义有：

$$\int p(x) dx = \int \pi(z) dz = 1 \quad (\text{A.85})$$

对于单元变量有：

$$p(x) = \pi(z) \left| \frac{dz}{dx} \right| = \pi(f^{-1}(x)) \left| \frac{df^{-1}}{dx} \right| = \pi(f^{-1}(x)) \left| (f^{-1})'(x) \right| \quad (\text{A.86})$$

对于多元变量形式 $\mathbf{z} \sim \pi(\mathbf{z})$ ,  $\mathbf{x} = f(\mathbf{z})$ ,  $\mathbf{z} = f^{-1}(\mathbf{x})$ 有：

$$p(\mathbf{x}) = \pi(\mathbf{z}) \left| \det \frac{d\mathbf{z}}{d\mathbf{x}} \right| = \pi(f^{-1}(\mathbf{x})) \left| \det \frac{df^{-1}}{d\mathbf{x}} \right| \quad (\text{A.87})$$

在式A.87中， $\det \frac{\partial f^{-1}}{\partial x}$  表示函数 $f^{-1}$ 的雅可比矩阵行列式。

## 反函数定理应用

对于 $y = f(x)$ 和 $x = f^{-1}(y)$ ,

$$\frac{df^{-1}(y)}{dy} = \frac{dx}{dy} = \left( \frac{dy}{dx} \right)^{-1} = \left( \frac{df(x)}{dx} \right)^{-1} \quad (\text{A.88})$$

## 微分方程

微分方程是包含了一项或多项函数以及它们的导数的方程。

### 杰森不等式

$$\mathbb{E}[g(x)] \geq g(\mathbb{E}[x]) \quad (\text{A.89})$$

### 指示函数

一个集合的子集的指示函数，将集合中属于该子集的元素映射为 1，其他元素映射为 0。假设集合  $X$  有子集  $Y$ ，则：

$$1_A(x) := \begin{cases} 1 & ,x \in A \\ 0 & ,x \notin A \end{cases} \quad (\text{A.90})$$

### 狄拉克 $\delta$ 函数

狄拉克  $\delta$  函数，也被称为单位脉冲，是在实数域上除 0 以外点的值处处为 0，且在整个实数域上积分为 1 的广义函数或分布。

### 狄拉克测量

狄拉克测量将集合的大小指定为含有特定元素  $x$  的数量。假设集合  $X$  具有可观测子集  $A$ ，且  $x \in X$ ，则有：

$$\delta_{(x)}(A) = 1_A(x) = \begin{cases} 0 & ,x \notin A \\ 1 & ,x \in A \end{cases} \quad (\text{A.91})$$

式 A.91 中， $1_A$  为指示函数。

### 点评估

点评估是狄拉克测量积分的描述，假设  $x \in X$ ，则关于点  $y$  的点评估为：

$$\int_X f(x) d u = f(y) \quad (\text{A.92})$$

### 泛函

泛函是指以函数构成的向量空间为定义域，实数为值域的“函数”，往往被称为“函数的函数”。在泛函分析中，泛函也用来指一个从任意向量空间到标量域的映射。

### 向量空间

向量空间是一个内部元素具备可加性和可乘性的集合。

### 内积空间

内积空间是具有内积操作的向量空间。

### 欧几里得向量空间

欧几里得向量空间是在实数域上的有限维度内积空间。

### 希尔伯特空间

希尔伯特空间将欧几里得向量空间扩展至无限维度。

### 再生核希尔伯特空间

在泛函分析中，再生核希尔伯特空间是函数的希尔伯特空间，其函数的点评估为线性连续泛函。当两个函数 $f, g$  在再生核希尔伯特空间中在范数上相近，即 $\|f - g\|$  值趋近于 0，则对于所有 $x$ ， $\|f(x) - g(x)\|$  趋近于 0，反之不一定成立。

### 最大均值差异

最大均值差异以特征的嵌入均值（Mean Embeddings）来表示距离，假设 $p, q$  为集合 $\mathcal{X}$  上的概率分布，特征映射 $\phi : \mathcal{X} \rightarrow \mathcal{H}$ ，其中 $\mathcal{H}$  为再生核希尔伯特空间，则有：

$$MMD(P, Q) = \|\mathbb{E}_{X \sim P}[\phi(X)] - \mathbb{E}_{Y \sim Q}[\phi(Y)]\|_{\mathcal{H}} \quad (\text{A.93})$$

## A.2 物理学

### A.2.1 玻尔兹曼分布

在统计力学与数学中，玻尔兹曼分布（也成为吉布斯分布）给出了一个系统处在特定状态的概率，其值与该状态的能量和该系统的热力学温度相关：

$$p_i \propto \exp\left(-\frac{\varepsilon_i}{kT}\right) \quad (\text{A.94})$$

式A.96中,  $p_i$  表示系统处在状态*i* 的概率,  $\varepsilon_i$  式该状态的能量,  $k$  为玻尔兹曼常数,  $T$  为热力学温度。 $\propto$  表示其成正比。其比例系数为 $\frac{1}{Q}$ , 其中

$$Q = \sum_{i=1}^M \exp\left(-\frac{\varepsilon_i}{kT}\right) \quad (\text{A.95})$$

结合式A.94与式A.95的

$$p_i = \frac{1}{Q} \exp\left(-\frac{\varepsilon_i}{kT}\right) = \frac{\exp\left(-\frac{\varepsilon_i}{kT}\right)}{\sum_{i=1}^M \exp\left(-\frac{\varepsilon_i}{kT}\right)} \quad (\text{A.96})$$

## A.2.2 朗之万动力学

在物理学中, 朗之万动力学是一种对分子系统运动的数学建模方式, 是一种采样方式。朗之万模拟是一种蒙特卡洛模拟。

## A.3 机器学习与深度学习

### A.3.1 概率图模型

机器学习算法通常涉及到大量随机变量的概率分布。在通常情况下, 这些变量只与很少部分的变量互相影响, 使用单一的函数方程来描述全体变量的联合概率分布不利于进行计算, 而将联合概率分布分解为条件概率有助于减少参数的数量。如对于三个随机变量 $a, b, c$ , 其联合概率分布可以分解为:

$$p(a, b, c) = p(a)p(b | a)p(c | a) \quad (\text{A.97})$$

当用“图”来表示这种对联合概率分布的分解时, 即为概率图模型。

概率图模型使用图 $\mathcal{G}$  来对变量进行建模, 图中的每一个节点表示一个随机变量, 每一条边连接两个随机变量, 表示两个变量的概率分布可以直接互相表示。

概率图模型可以分为有向图模型和无向图模型。

#### 有向图模型

有向图模型中的边为有向边。有向图模型中, 每一个随机变量 $x_i$  都有一个概率因子 $p_{\mathcal{G}}(x_i)$  来表示其条件分布, 即:

$$p(\mathbf{x}) = \prod_i p(x_i | p_{\mathcal{G}}(x_i)) \quad (\text{A.98})$$

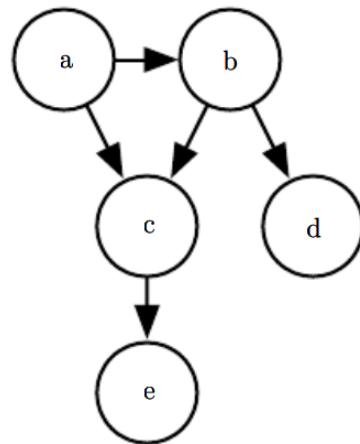


图 A.2 有向图模型实例

图A.2中， $a, b, c, d, e$  为随机变量，该有向图模型的联合概率分布为：

$$\begin{aligned} p(a, b, c, d, e) &= p(a \mid p_{\mathcal{G}}(a)) + p(b \mid p_{\mathcal{G}}(b)) + p(c \mid p_{\mathcal{G}}(c)) + p(d \mid p_{\mathcal{G}}(d)) + p(e \mid p_{\mathcal{G}}(e)) \\ &\quad (A.99) \end{aligned}$$

$$= p(a)p(b \mid a)p(c \mid a, b)p(d \mid b)p(e \mid c) \quad (A.100)$$

### 无向图模型

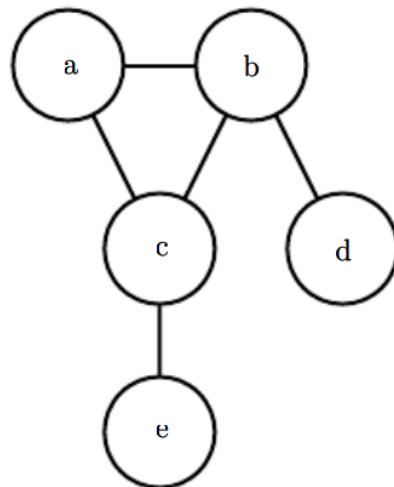


图 A.3 无向图模型实例

无向图模型中的边为无向边。无向图中任意两个节点都直接相连的部分为一个团 $\mathcal{C}^{(i)}$ ，每一个团都可定义一个相关方程 $\phi^{(i)}(\mathcal{C}^{(i)})$ 。可以由归一化后的团相关方程来表

示无向图模型的联合概率分布，即：

$$p(\mathbf{x}) = \frac{1}{Z} \prod_i \phi^{(i)}(\mathcal{C}^{(i)}) \quad (\text{A.101})$$

图A.3中， $a, b, c, d, e$  为随机变量。该有向图模型的联合概率分布为：

$$p(a, b, c, d, e) = \frac{1}{Z} \phi^{(1)}(a, b, c) \phi^{(2)}(b, d) \phi^{(3)}(c, e) \quad (\text{A.102})$$