华南师范大学

人工智能导论
课程项目报告

设计题目　基于概率扩散模型的图像生成

学生姓名　温志森 邓实诚 施楷鑫 张亮 许培炫

专业班级　21 级计算机科学 1 班

指导教师　曹阳

2023 年 5 月 27 日

一、 引言

概率扩散模型首先于 2015 年被提出 [6]，于 2020 年被完善 [2]，其基本思想是逐步删除图片中的噪声生成人类可以理解的信息片段，比如语音和图片。自从首次提出至今，概率扩散模型就被广泛应用于图像生成、音乐生成以及语音生成等生成式任务之中。Midjourney、DALLE2、Stable Diffusion 等图片生成模型更是以其优秀的性能表现被应用于实际的生产活动中。同时，以 Stable Diffusion 为代表的开源模型更是为研究者们提供了一个优秀的研究平台，使得研究者们可以在其基础上进行更多的研究工作。目前的扩散模型社区的研究者和爱好者的大部分工作都是在 Stable Diffusion 的基础上进行的，比如使用 Stable Diffusion 进行 LoRA 训练。并且，Stable Diffusion 的爱好者开发了开源的 WebUI 项目，使得研究者们可以在不了解模型原理的情况下直接使用图形化界面使用 Stable Diffusion 进行 LoRA 训练。

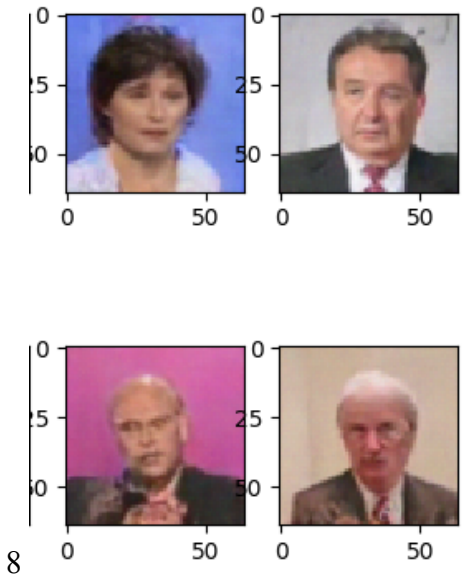


图 1　模型生成的人脸照片

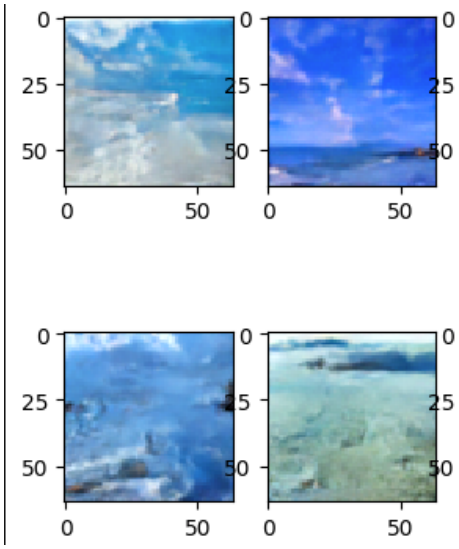


图 2　模型生成的风景照片

区别于使用开源的预训练模型，本小组成员希望更深入地理解概率扩散模型背后的数学原理、代码编写以及训练流程。我们的工作涉及概率统计的公式理解与推导、模型训练环境的安装与部署、模型的训练参数调优等一系列对本小组成员具有挑战性的任务。因此，本小组成员希望能够从零开始学习扩散模型的原理，并着手编写代码从头开始完成一个扩散模型的“最小可行方案”，具体包括下面的任务：

1. 直观感受概率扩散模型的基本思想。概率扩散模型的的数学原理比较复杂。对概率扩散模型的工作原理有一个感性的认知有助于我们深入理解模型背后的数学思想。

2. 数学公式的理解。概率扩散模型的原始论文包括大量的数学推导过程。理解这些数学公式是理解模型工作原理的基础。

3. 布置深度学习模型的训练环境。

4. 从数学公式出发，编写训练模型的代码。

5. 寻找合适的数据集，训练模型。

6. 测试模型的性能。

通过完成上面的六个任务，目标是提高本小组成员的对概率扩散模型的理解，提高对深度学习的基石概率论的理解，提高小组成员的代码编写能力。

二、 图国内外研究现状

概率扩散模型是一种生成模型，它通过不断地向数据添加噪声，然后学习逆向的去噪过程，从而生成与训练数据相似的数据。概率扩散模型的灵感来自非平衡热力学，它使用马尔可夫链来映射到高维的隐空间。

在 2015 年，斯坦福大学的研究人员首次提出了扩散模型这个概念 [6]。这篇文章确定了扩散模型的理论框架，但是直到 2020 年 DDPM 的出现，扩散模型才真正开始流行起来 [2]。在 [2] 中，DDPM 的作者提出了针对扩散模型的训练和推理算法。算法包括前向过程和逆向过程两个部分。在前向过程中逐步地向数据添加噪声，然后学习逆向的去噪过程；在逆向过程中，从正态分布中抽取样本图片，标记为 $\mathbf{x}_T$，然后使用模型预测 $\mathbf{x}_T$ 相对于 $\mathbf{x}_{T-1}$ 的噪声 ϵ_T，得到 $\mathbf{x}_{T-1} = \mathbf{x}_T - \epsilon_T$ 。前向过程可以形式化的表述为:

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

方程 3表示给定 $\mathbf{x}_{t-1}$，$\mathbf{x}_t$ 的概率分布，其中 $\sqrt{1-\beta_t}\mathbf{x}_{t-1}$ 表示正态分布的均值，$\beta_t\mathbf{I}$ 表示正态分布的方差，β_t 是线性递增的噪声步长。逆向过程可以形式化的表述为:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{2}$$

方程 5表示给定 $\mathbf{x}_t$，$\mathbf{x}_{t-1}$ 的概率分布。$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ 和 $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ 是神经网络。在 2021 年，来自 OpenAI 的研究者对 DDPM 进行了改进，提出将参数 β_t 由线性变化改为非线形变化，同时除了要使用神经网络预测噪声的均值，还要预测噪声的方差 [4]。这种改进提高了模型生成图片的质量。在此之后，又有许多研究者将对 DDPM 进行了改进，并将 DDPM 应用到了不同的领域，例如视频生成、音频生成。由于本文的目标仅是学习 DDPM 的基本原理，因此不再对 DDPM 的改进进行详细的介绍。

三、 模型和算法

在本节中，我们将对 [2] 这篇论文中 DDPM 的模型和算法进行详细的介绍。首先，我们将介绍 DDPM 的模型结构，然后介绍 DDPM 的训练和推理算法。

3.1 模型结构

DDPM 使用 U-net 模型作为预测图片噪声的网络。U-net 最早于 2015 年被踢出来，用于医学图像分割 [5]，它的模型结构如图3所示。可以看到，U-net 分为编码器和解码器两个部分，编码器逐步压缩图片特征的分辨率用于提取图片的特征，解码器采用与编码器相反的过程用于将提取到的特征映射到图片空间。编码器和解码器之间使用残差链接提高模型的性能。在 [5] 中，模型的输入图片，模型的输出是图片的分割结果。但在在 DDPM 中，模型的输入是图片，模型的输出是图片的噪声的均值。预测的过程可以表示为:

$$\epsilon_t = f_\theta(\mathbf{x}_t, t) \tag{3}$$

更多关于 U-net 的信息，请参照原论文 [5]。

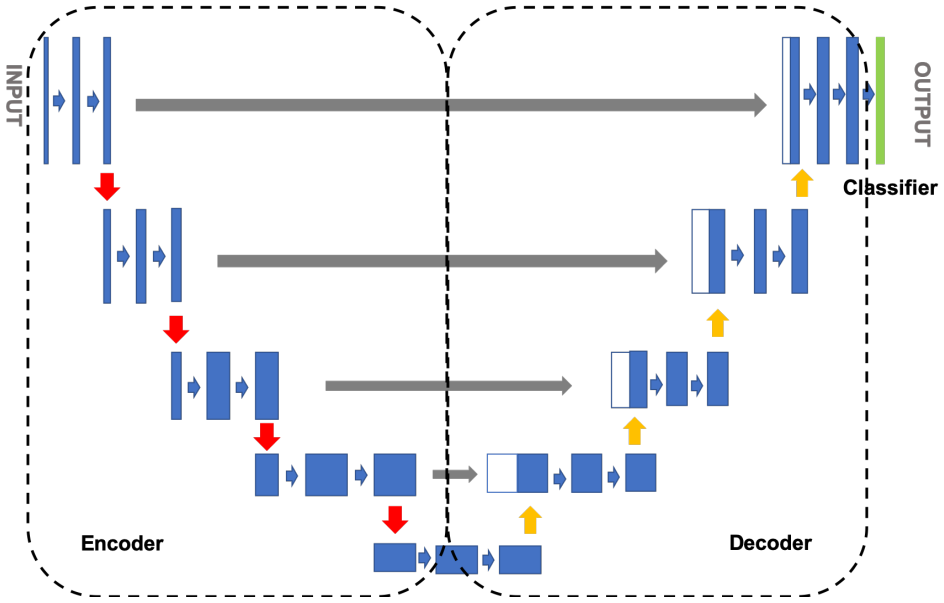


图 3　DDPM 的模型结构

3.2 训练和推理算法

本小节希望使用通俗的语言描述 DDPM 的训练和推理算法的基本思路。DDPM 包括的训练包括两个过程，分别是前向过程和反向过程，如图4所示。

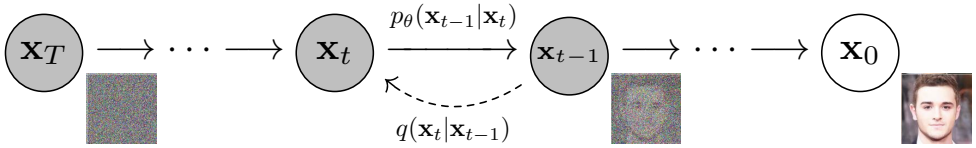


图 4　DDPM 的前向和反向过程

我们的目标是从一个完全是随机噪声的图片出发，生成一张有意义的照片。为了实现这个目标，我们可以逐步地从照片中去除噪声，直到噪声完全消失。这个过程可以形

式化的表述为公式3。这公式表示给定得到第 t 时刻的图片 $\mathbf{x}_t$，如果我们可以知道照片中的噪声 ϵ_t，那么我们就可以得到第 $t-1$ 时刻的图片 $\mathbf{x}_{t-1}$，因为 $\mathbf{x}_t = \mathbf{x}_{t-1} + \epsilon$。

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{4}$$

但是我们无法直接得到噪声 ϵ_t，因为噪声 ϵ_t 是随机的。所以我们需要通过神经网络，即 U-net，预测第 t 步相对于第 t-1 步的噪声。因此我们的目标就变成了训练神经网络，使得神经网络能够预测出噪声 ϵ_t。神经网络中的参数即是公式**??**中的 θ。在训练过程中，我们需要随机选取第 t 步的图片 $\mathbf{x}_t$，然后通过神经网络预测出噪声 ϵ_t，最后通过公式5计算出第 t-1 步的图片 $\mathbf{x}_{t-1}$。得到第 t 步照片的方法是每次从训练集中随机选取一张图片，逐步的往照片中添加符合正态分布的照片，直到第 t 步。给定第 $t-1$ 步照片，得到的第 t 步骤片的过程可以使用下面的公式表示:

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \epsilon_t \tag{5}$$

其中 ϵ_t 是符合标准正态分布的随机噪声，β_t 是一个固定值用来控制噪声的大小。显然，如果只使用公式5，那么从第 0 步得到第 t 步的照片，我们需要重复 t 次公式5。这样的计算过程是非常耗时的，因此我们需要使用更加高效的方法。我们可以使用下面的公式来计算第 t 步的照片:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{6}$$

其中 $\alpha_t = 1 - \beta_t$，$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$。具体的推导过程不再展开。总之，通过使用公式6，我们可以通过一次计算得到第 t 步的照片，而不需要重复 t 次公式5，这极大地提高了模型计算的速度。

总之，DDPM 的训练和推理过程可以总结为下面的算法流程:

Algorithm 1 Training

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: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

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

图 5　DDPM 的训练及推理算法。图片来自于 [2]。

四、 实验结果及分析

4.1 深度学习平台

概率扩散模型是一种深度学习模型，使用神经网络作为模型的架构。神经网络的优化过程需要使用反向传播算法和梯度下降算法。而概率扩散模型包含有全连接神经网络层、卷积神经网络层、批归一化层等功能和结构各异的网络层，具有复杂的网络结构和函数依赖关系。这导致手动编写概率扩散模型的反向传播算法和梯度下降算法成为了一项复杂而庞大的工程。

深度学习框架为开发者准备了包括全连接层、卷积层、注意力机制层在内常用网络构件。同时，深度学习框架通过搭建计算图的方式自动帮助开发者实现反向传播算法，使得开发者只需要编写一行代码即可以实现网络的梯度更新和参数优化。另外，通过使用深度学习框架，开发者还可以利用 GPU 等高性能芯片加速的模型的训练过程。

主流的深度学习框架包括 Pytorch(Meta)、Tensorflow(Google)、MindSpore(华为)。在此次项目中，我们选择使用百度公司开发的 PaddlePaddle 框架。PaddlePaddle 的 API 设计较为直观，易于理解。它的高层 API 具有良好的封装性，使得开发者可以更快速地搭建模型。重要的是，百度的飞桨平台为开发者提供了免费的 GPU 资源。开发者可以在飞桨平台上使用 PaddlePaddle 框架编写模型的训练代码，使用平台的提供的 GPU 资源训练模型。

4.2 训练过程

我们使用上述的 DDPM 模型对两个不同的数据集进行了实验，分别是风景照数据集 [3] 和人脸数据集 [1]。两个数据集分别含有 4300 张风景照片和 3462 张人脸照片。下面是我们的代码训练流程：

1. 使用 paddle 提供的 ImageFolder 接口加载磁盘中的图片数据。
2. 使用 paddlevision 提供的 transforms 对图片进行预处理。预处理操作包括：将图片缩放到统一大小、随机裁剪、转化为 Tensor、归一化等。
3. 使用 paddle 提供的 DataLoader 接口生成可迭代的数据集。
4. 定义 Unet 模型，用来预测噪声。
5. 定义 Adam 优化器，用来优化 Unet 模型的参数。
6. 定义 MSE 损失函数，用来衡量预测噪声的准确性。
7. 开始迭代训练。
8. 使用训练好的 Unet 模型生成图片。

下面的表 1是训练过程中的一些重要参数。可以看到，两个数据集训练参数基本相同，只有训练时间有所不同。我们使用英伟达 V100 16G 显卡在风景照片数据集上训练

150 个 epoch，大约需要 4 个小时；使用英伟达 V100 32G 显卡在人脸照片数据集上训练了 100 个 epoch，大约需要 2 个小时。

表 1　训练参数

数据集	epoch	训练时间	batchsize	learning rate	image size	noise step
风景照片	150	4 hours	24	1.5e-4	80	500
人脸照片	1000	10 hours	24	1.5e-4	80	500

4.3 模型输出及分析

图6和图7分别是我们在风景照片数据集和人脸照片数据集上训练得到的模型生成的图片。可以看到，模型在人脸照片数据集上的生成效果较好，生成的图片清晰可辨，人脸轮廓清晰，五官分明。但是在风景照片数据集上的生成效果较差，生成的图片模糊不清，只有大致的色块，无法辨认出具体的物体。这既是因为风景照片数据集的图片大多都是大片蓝色 (天空)、绿色 (树木、草原) 和白色 (云朵) 色块，具有的明显的特征比较少，也是因为我们在风景照片数据集上训练的 epoch 数较少。

当然，相比于 Stable Diffusion 和 DALLE 2 模型生成的照片，我们的模型生成的照片还是有很大的差距的。这其中的原因特别多，但是可是归结为下面的几种情况：

1. 数据集的问题。我们使用的数据集都是比较小的数据集，而且数据集中的照片往往具有单一性，比如风景照片数据集中的照片大多都是蓝天、绿树、白云，人脸照片数据集中的照片大多都是正脸、侧脸、半身照。这样的数据集对于模型的训练效果不利。

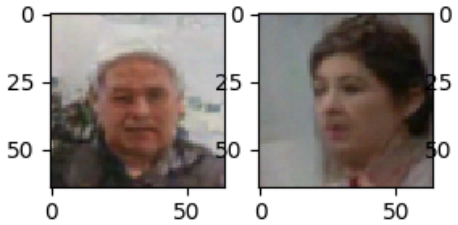
2. 训练时间的问题。我们的模型训练时间都比较短，训练时间越长，生成图片的质量越高。

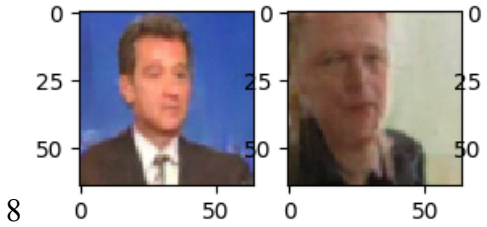
3. 训练参数的问题。我们的模型训练参数都是根据经验设置的，没有经过大量的实验来确定最优参数。另外，由于算力的问题，我们的 batchsize，特别是 image size 无法设置的太大，这也会影响模型的训练效果。

4. 算法的问题。我们的模型是基于 [2] 的，在这之后的 Stable Diffusion 和 DALLE 2 模型在算法和训练策略上都有了很大的改进，这也提高生成图片的质量。

5. 预训练模型的问题。在使用 Stable Diffusion 进行训练时，往往是使用经过大量训练的基模型进行迁移学习，这样可以大大提高模型的训练效果。但是在本项目中我们并没有使用预训练模型，这也是影响模型训练效果的一个特别重要的因素。

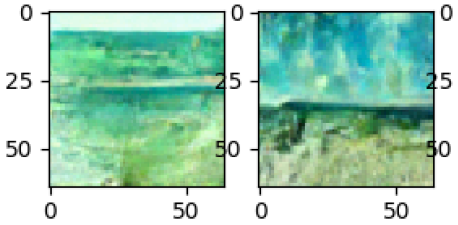
总之，本项目的目的是为了学习概率生成模型的原理，而不是为了生成高质量的图片，所以我们的模型生成的图片质量不高也是可以理解和接受的。





8

图 6　生成的人脸照片



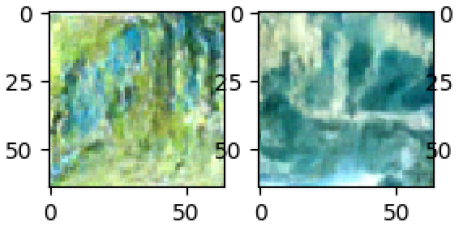


图 7　生成的风景照片

五、 结论

本次项目我们实现了一个基于概率生成模型的图片生成模型。我们首先介绍了概率生成模型的基本原理，然后介绍了我们使用的模型——Unet 模型，以及模型的训练策略。接着我们介绍了我们使用的两个数据集，并且介绍了我们的模型的训练参数。最后我们展示了我们的模型生成的图片，并且对模型的生成效果进行了分析。

参考文献

[1] aistudio. 人脸关键点识别. https://aistudio.baidu.com/aistudio/datasetdetail/69065. 2021-11-11.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 33:6840–6851, 2020.

[3] kaggle. Landscape pictures. https://www.kaggle.com/datasets/arnaud58/landscape-pictures. 2020.

[4] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In International Conference on Machine Learning, pages 8162–8171. PMLR, 2021.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pages 234–241. Springer, 2015.

[6] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In International Conference on Machine Learning, pages 2256–2265. PMLR, 2015.