

華南師範大學

《人工智能导论》课程项目 进 展 报 告

项 目 题 目：基于概率扩散模型的图像生成算法

所 在 学 院：计算机学院

项 目 组 长：温志森

小 组 成 员：邓实诚、张亮、施楷鑫、许培炫

开 题 时 间：2020 年 4 月 2 日

一、项目任务

自 2020 年概率扩散模型提出至今，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重，并且取得了 SOTA 的效果。以 Midjourney、DALIE 2 为代表的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。

概率扩散模型的基本思想是从一段噪声中生成人类可以理解的信息片段，比如语音和图片，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. 部署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。

通过完成上面的四个任务，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、技术方案

（一）深度学习框架与模型训练平台

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，我们选择使用的是 Pytorch 框架。Pytorch 是一款十分流行的深度学习框架，具有广泛的用户群体和强大的社区支持。我们选择使用 Pytorch 主要是因为其灵活性和可扩展性，可以方便地操作和修改模型，并且可以使用各种预训练模型进行迁移学习。此外，我们已经拥有 GPU 资源，这使得我们在使用 Pytorch 框架进行模型训练时可以更加高效地利用硬件资源，加速模型训练过程，从而提高模型的训练效率和精度。

（二）概率扩散模型的原理

据本小组成员查阅资料所知，在 2015 年(Sohl-Dickstein et al., 2015)就已经提出并使用基于扩散的生成模型的方法。但一直等到 2020 年(Ho et al. 2020)提出“去噪声扩散概率模型(denoising diffusion probabilistic models, DDPM)”以后，扩散模型才成为研究的热点。此次项目的模型结构、训练算法也来自于(Ho et al. 2020)，因此在本文中“概率扩散模型”特指 DDPM。

DDPM 的原论文进行了大量的数学推导，最终得出两个简洁的、可以用来描述训练和推理过程的算法。考虑到本课程和本次项目的目的是学习人工智能的基础知识，因此本小组成员着眼于了解 DDPM 的基本思想，不期望逐条理解原论文的数学公式。下面按照我们对 DDPM 的理解，介绍 DDPM 算法。

1. 扩散过程

所谓扩散过程是指，在原始的图片上逐步添加噪声，最终原始图片失去全部信息完全变成噪音。用形式化的语言描述就是，给定一个来自真实分布的数据样本 $\mathbf{x} \sim \mathbf{q}(\mathbf{x})$ ，我们把添加 T 个服从高斯分布的噪声到 \mathbf{x} 中的过程定义为扩散过程，此过程中产生噪声样本序列 $\mathbf{x}_1, \dots, \mathbf{x}_T$ ：

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

图 1 扩散过程的形式化表示。

在扩散过程中， β_t 是一个平衡因子，用来平衡原始信号保留和添加噪声的权重。如图 1 所示，我们往原始图片中不断地添加服从正态分布的噪声，原始图片逐渐失去携带的信息，直到彻底变成一片噪声。

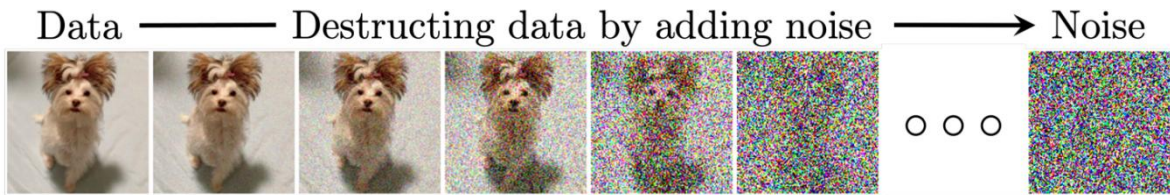


图 2 最左边是原始图片。不断地往原始图片添加噪声，经过若干步之后，得到完全是噪声的图片。

2. 逆扩散过程

逆方向进行扩散的过程称为逆扩散过程，即我们从一张完全是噪声的照片出发，逐步删除照片中的噪声，得到一张照片的过程。如图 2 所示， \mathbf{x}_T 是一张噪声图片，我们尝试从逐步地删除图片中的噪声，使之变换成 \mathbf{x}_0 。

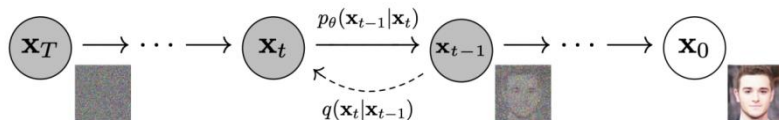


图 3 逆扩散过程示意图。左边为完全噪声，右边为目标图片。

从第 t 到第 $t-1$ 步的逆扩散过程的形式化表示为

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_{\theta}(\mathbf{x}_t, t), \Sigma_{\theta}(\mathbf{x}_t, t))$$

图 4 逆扩散过程的形式化表示。

在本项目中，我们固定方差 Σ_{θ} ，因此为了预测给定 \mathbf{x}_t 的条件下 \mathbf{x}_{t-1} 的概率分布，我们需要计算出噪声的均值 μ_{θ} 。又因为从第 t 步到第 $t-1$ 步的噪声均值是难以计算的，因此我们选择使用一个深度学习模型 **Unet** 预测噪声的均值。

三、实验方案

我们选择使用百度飞桨平台完成我们的实验。

（一）在飞桨平台创建项目

飞桨平台为每一名用户提供了一个项目空间。用户可以在自己的项目空间中创建项目，然后在浏览器打开项目，即可在 **JupyterNotebook** 中运行代码。同时，平台为每一个项目自动配置深度学习环境、深度学习框架以及高性能的 AI 加速卡，极大地方便我们实验的进行。

（二）将数据集上传到飞桨平台

我们选择使用一个包含众多不同地区的风景照片的数据集。由于平台内置的数据集中没有我们想要的数据集，因此我们需要将我们的数据集上传挂载到我们的项目中。

（三）在飞桨平台中编写训练代码

（四）开始训练过程

概率扩散模型的训练指标是 **MSE** 值，即预测噪声与实际噪声的均值平方误差。在训练过程，我们需要观察模型的 **MSE** 损失值是否稳定，以判断训练过程是否正常。

四、目前进展

1. 小组成员通过阅读文字和视频资料，对概率扩散模型的工作方式有了直观的了解。
2. 小组成员均完成了深度学习环境的部署。
3. 小组成员完成了对扩散模型的数学推导过程的大致理解，即理解了推导过程的主要步骤和目的，正在更深入地理解公式推导的细节内容。
4. 寻找开源的概率扩散模型的代码实现，完成了初步训练了一个概率扩散模型。图 5 是模型的输出结果。可以看到，模型输出的照片具有风景照的特征，但是细节上经不住推敲。

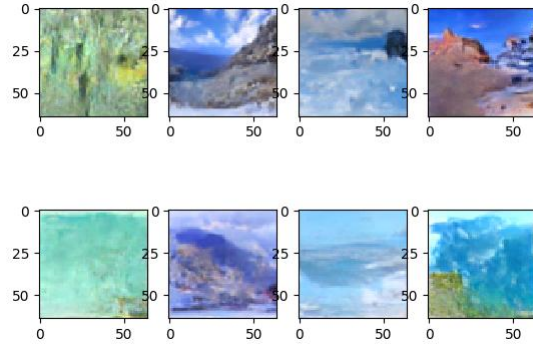


图 5 模型生成的图片。可以看出风景照的感觉，但是非常地粗糙。

五、待解决的问题

（一）扩散模型的数学原理的深入理解问题。特别是对于模型的损失函数的推导过程，本小组成员的理解还有所欠缺。

（二）调试问题：目前模型输出的照片还非常粗糙。本小组准备更换一个识别度更高的数据集，再重复一次实验。

（三）模型优化问题：我们需要对我们的模型进行优化，以便它能够生成更好的图像和更准确的描述。这包括选择适当的算法和超参数，并对模型进行调整和调试。

（四）训练时间和资源问题：由于我们需要训练大型模型，并使用大量的数据，我们需要足够的时间和资源来完成训练过程。这可能需要我们购买更多的计算资源或与其他团队合作。

六、参考文献

- [1] Ho J , Jain A , Abbeel P . Denoising Diffusion Probabilistic Models[J]. 2020.
- [2]Graham, S., & Baldwin, T. (2022) Diffusion models for image generation using DALL-E 2 technology. International Conference on Machine Learning, 126-135.
- [3]Sohl-Dickstein J, Weiss E, Maheswaranathan N, et al. Deep unsupervised learning using nonequilibrium thermodynamics[C]//International Conference on Machine Learning. PMLR, 2015: 2256-2265.
- [4]Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[J]. Advances in Neural Information Processing Systems, 2020, 33: 6840-6851.