Denoising Diffusion Probabilistic Models 论文复现

刘鑫豪

摘要

深度学习在生成任务中显示出巨大的潜力。生成模型是类可以根据某些隐含的参数随机生成观察结果的模型。最近，扩散模型凭借其强大的生成能力，成为生成模型的一大热门。已经取得了巨大的成就。除了计算机视觉、语音生成、生物信息学和自然语言处理之外，更多的应用将在这一领域得到探索。然而，扩散模型有其天然的缺点，即生成过程缓慢，据此，有许多强化的工作近些年被陆续提出。目前扩散模型分为两个标志性工作–DDPM(Denoising Diffusion Probabilistic Model) 和 DSM(Denoising Score Match)。DDPM 论文提出了高质量的图像合成结果使用扩散概率模型，一类潜在变量模型的灵感来自非平衡热力学的考虑。作者的最佳结果是通过训练加权变分界获得的，根据扩散概率模型和去噪分数匹配与朗之万动力学之间的新连接设计，论文的模型自然地承认一个渐进的有损耗解压缩方案，可以解释为自回归解码的一般化。在无条件的 CIFAR10 数据集上，该方法获得了 9.46 的 Inception 评分和最先进的 FID 评分 3.17。在 256x256 LSUN 上，获得了类似 ProgressiveGAN 的样品质量。符合当前的研究热点，选择本文作为前沿技术论文作业，用 Pytorch 实现了该论文的代码复现。

关键词：Diffusion Model; Pytorch;

1 引言

[1]提出了高质量的图像合成结果使用扩散概率模型，一类潜在变量模型启发，灵感来自非平衡热力学。论文实验的最佳结果是在加权变分界上进行训练，该变分界是根据扩散概率模型和与朗之万动力学匹配的去噪评分之间的一种新型联系设计，模型是一个渐进的有损解压缩方案，可以被解释为自回归解码的推广。基于 Pytorch，对改论文进行了复现。最近，各种深度生成模型在各种各样的数据形式中展示了高质量的样本。生成对抗网络 (GANs)、自回归模型、流和变分自编码器 (vais) 合成了引人注目的图像和音频样本，基于能量的建模和评分匹配已经取得显著进展，生成的图像可与 GANs 的图像相媲美。尤其在 OpenAI 提出的 DALL-E 2 模型的问世，成为了文字生成图像的天花板，这模型的出现使得 Diffusion Model 成为了近期的热点，因为 DALL-E 2 在根据文字生成图像的过程中使用了 Diffusion Model，因此该复现工作拟合了当前的热点研究，有一定复现意义。

2 相关工作

虽然扩散模型可能类似于流和 ves，但扩散模型的设计使 q 没有参数，顶级潜 xT 与数据 x0 的互信息几乎为零。我们的-预测反向过程参数化在扩散模型和去噪评分匹配之间建立了联系，在多个噪声级别上使用退火朗格万动力学进行采样。然而，扩散模型允许直接的对数似然估计，并且训练过程显式地使用变分推理训练朗格万动态采样器。这种联系也具有相反的含义，即某种加权形式的去噪评分匹配与训练类朗格万采样器的变分推理相同。学习马尔可夫链过渡算子的其他方法包括灌注训练、变分回走、生成随机网络等。根据分数匹配和基于能量的建模之间已知的联系，我们的工作可能会对其他近期关于基于能量的模型的工作产生影响。我们的率失真曲线是在变分界的一次评估中随时间计

算的，这让人想起如何在经过退火的重要性抽样的一次运行中计算失真惩罚率失真曲线。我们的渐进式解码论证可以在卷积 DRAW 和相关模型中看到，也可能导致自回归模型的子尺度排序或采样策略的更通用设计。

2.1 扩散模型和去噪自编码器

扩散模型可能看起来是潜在变量模型的一个受限类，但它们允许在实现中有大量的自由度。必须选择正向过程的方差 βt 和反向过程的模型结构和高斯分布参数化。为了指导我们的选择，我们在扩散模型和去噪分数匹配之间建立了新的显式联系，从而得到了扩散模型的简化、加权变分定界目标。最终，我们的模型设计被简单和实证结果所证明。我们的讨论被如下公式所分类。

$$\mathbb{E}_q\Bigg[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p(\mathbf{x}_T))}_{L_T}+\sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Bigg]$$

图 1: 重写减少方差

正向过程和 LT

论文忽略了正向过程方差 βt 是可通过重新参数化学习的这一事实，而是将它们固定为常数。因此，在我们的实现中，近似后验 q 没有可学习的参数，所以 LT 在训练中是一个常数，可以忽略。

逆扩散过程

现在我们讨论 pθ (xt−1|xt) = N (xt−1;μθ (xt, t)，Σθ (xt, t)) 对于 1 < t≤t。首先，我们设置 Σθ (xt, t) = σ2t I 为未训练的时间相关常数。实验结果表明，σ2t = βt 和 σ2t =′βt = 1−αt−11−αt βt 具有相似的结果。对于 x0 N (0, I)，第一个选择是最优的，对于 x0 确定性地设置为一个点，第二个选择是最优的。对于坐标单位方差为的数据，这是对应逆向过程熵上界和下界的两个极端选择。其次，为了表示均值 μθ (xt, t)，我们提出了一个特定的参数化，其动机是以下分析 pθ (xt−1|xt) = N (xt−1;μθ (xt, t)，σ2t I)，可得:

$$L_{t-1}=\mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0)-\boldsymbol{\mu}_\theta(\mathbf{x}_t,t)\|^2\right]+C$$

这类似于在索引的多个噪声尺度上匹配去噪分数。由于等于类朗之万逆向过程的变分界的 (一项)，我们看到优化类似去噪分数匹配的目标相当于使用变分推理来拟合类似朗之万动力学的采样链的有限时间边缘。总之，我们可以训练反向过程平均函数逼近器 μθ 来预测 ′μt，或者通过修改其参数化，我们可以训练它来预测。(也有可能预测 x0，但我们在实验早期发现这会导致更差的样本质量。) 我们已经证明了-预测参数化既类似于朗之万动力学，又简化了扩散模型的变分界到一个类似于去噪得分匹配的目标。尽管如此，它只是 pθ (xt−1|xt) 的另一种参数化，因此我们在第 4 节中验证了它在消融中的有效性，论文将预测与预测 ′μt 进行了比较。可知，在给定 xt 条件下，μθ 必须预测。由于 xt 可以作为模型的输入，我们可以选择参数化，其中 θ 是一个函数逼近器，用于从 xt 进行预测。对 xt−1 pθ (xt−1|xt) 进行采样，，其中 z N (0, I)。完整的采样过程，类似于用 θ 作为数据密度的学习梯度的朗之万动力学。此外，通过参数化，类似于为索引的多个噪声尺度上的去噪评分匹配。我们看到优化类似去噪分数匹配的目标相当于使用变分推理来拟合类似朗之万动力学的采样链的有限时间边缘。总之，我们可以训练反向过程平均函数逼近器 μθ 来预测 ′μt，或者通过修改其参数化，我们可以训练它来预测。

3 本文方法

3.1 本文方法概述

Diffusion Models 既然叫生成模型，这意味着 Diffusion Models 用于生成与训练数据相似的数据。从根本上说，Diffusion Models 的工作原理，是通过连续添加高斯噪声来破坏训练数据，然后通过反转这个噪声过程，来学习恢复数据。训练后，可以使用 Diffusion Models 将随机采样的噪声传入模型中，通过学习去噪过程来生成数据。更具体地说，扩散模型是一种隐变量模型（latent variable model），使用马尔可夫链（Markov Chain, MC）映射到 latent space。通过马尔科夫链，在每一个时间步 t 中逐渐将噪声添加到数据中以获得后验概率，x 代表输入的数据同时也是 latent space。也就是说 Diffusion Models 的 latent space 与输入数据具有相同维度。

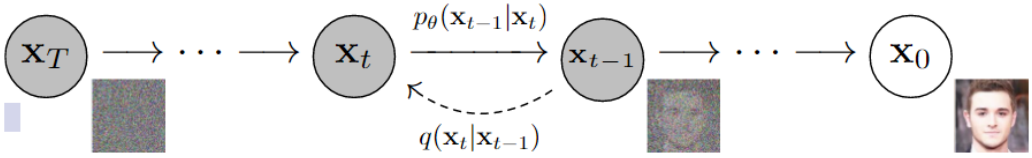

图 2: Diffusion Model

3.2 正向扩散实现

Diffusion Models 分为正向的扩散过程和反向的逆扩散过程。下图为扩散过程，从 Xo 到最后的 Xt 就是一个马尔科夫链，表示状态空间中经过从一个状态到另一个状态的转换的随机过程。而下标则是 Diffusion Models 对应的图像扩散过程。所谓前向过程，即往图片上加噪声的过程。虽然这个步骤无法做到图片生成，但是这是理解 diffusion model 以及构建训练样本 GT 至关重要的一步。最终，从 Xo 输入的真实图像，经过 Diffusion Models 后被渐近变换为纯高斯噪声的图片 Xt。前向过程添加噪声 x q(x), 分为 T 个阶段.

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}\right)$$

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

我们可以采样 xt 在任何时间步长 t.

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

3.3 逆扩散实现

模型训练主要集中在逆扩散过程。训练扩散模型的目标是，学习正向的反过程：即训练概率分布 p(xt-1|xt)。通过沿着马尔科夫链向后遍历，可以重新生成新的数据，Diffusion Models 跟 GAN 或者 VAE

的最大区别在于不是通过一个模型来进行生成的，而是基于马尔科夫链，通过学习噪声来生成数据。逆向过程是分 T 步移除噪声. 某时刻图像满足的分布为：

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

$$p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}$$

根据马尔可夫规则表示，逆扩散过程当前时间步 t 只取决于上一个时间步 t-1. 也就是在训练时候，模型学习逆扩散过程的概率分布，以生成新数据. 现在所有 KL 散度都是在高斯概率分布之间进行比较。这意味着可以使用闭包表达式，而不是采样的蒙特卡洛估计方式来精确计算变分上界。扩散模型训练过程有几个细节：对于正向扩散过程，唯一需要的选择是概率相关的向量（均值和方差），其值在扩散过程中在隐变量中直接添加高斯参数。对于逆扩散过程，需要选择能够表达高斯分布的模型结构，神经网络模型的拟合能力很强，于是就可以引入神经网络模型啦。最后就是对于神经网络模型有一个简单的要求，模型的输入、输出、中间隐变量必须要有相同的维度 dims。

3.4 损失函数定义

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right\|^2\right]$$

3.5 Unet 网络

U-Net 是从模型图中的 U 形获取它的名称。它通过逐渐降低（减半）特征图分辨率然后增加分辨率来处理给定图像。每个分辨率都有直通连接。Unet 这个结构就是先对图片进行卷积和池化，在 Unet 论文中是池化 4 次，比方说一开始的图片是 224x224 的，那么就会变成 112x112，56x56,28x28,14x14 四个不同尺寸的特征。然后我们对 14x14 的特征图做上采样或者反卷积，得到 28x28 的特征图，这个 28x28 的特征图与之前的 28x28 的特征图进行通道伤的拼接 concat，然后再对拼接之后的特征图做卷积和上采样，得到 56x56 的特征图，再与之前的 56x56 的特征拼接，卷积，再上采样，经过四次上采样可以得到一个与输入图像尺寸相同的 224x224 的预测结果。Unet 网络非常的简单，前半部分就是特征提取，后半部分是上采样。在一些文献中把这种结构叫做编码器-解码器结构，由于网络的整体结构是一个大些的英文字母 U，所以叫做 U-net。

Encoder：左半部分，由两个 3x3 的卷积层（RELU）再加上一个 2x2 的 maxpooling 层组成一个下采样的模块；Decoder：有半部分，由一个上采样的卷积层（去卷积层）+ 特征拼接 concat+ 两个 3x3 的卷积层（ReLU）反复构成；

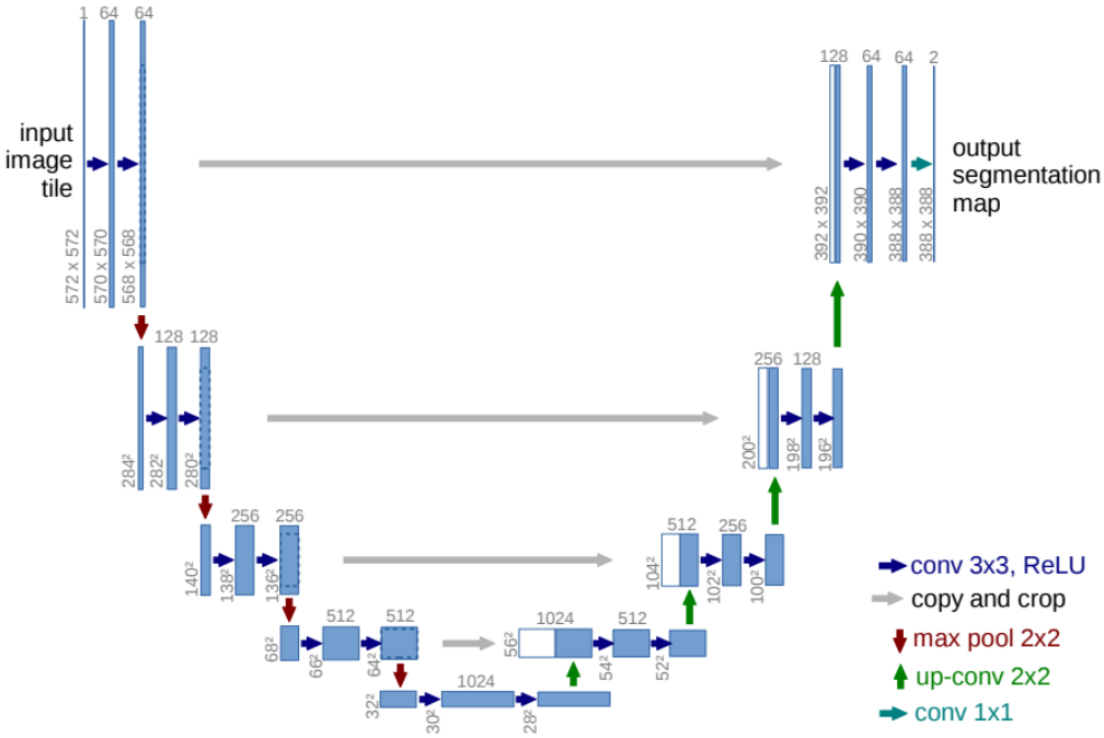

图 3: Unet 网络

代码实现包含对原始 U-Net 的一系列修改（残差块，多头注意），并且还添加了时间步嵌入 t.Unet 的优点：网络层越深得到的特征图，有着更大的视野域，浅层卷积关注纹理特征，深层网络关注本质的那种特征，所以深层浅层特征都是有格子的意义的；另外一点是通过反卷积得到的更大的尺寸的特征图的边缘，是缺少信息的，毕竟每一次下采样提炼特征的同时，也必然会损失一些边缘特征，而失去的特征并不能从上采样中找回，因此通过特征的拼接，来实现边缘特征的一个找回。

4 复现细节

4.1 与已有开源代码对比

代码的实现参考了关于 Unet 网络的定义与部署，随后自己完成了在 DDPM 在 Mnint 和 CelebA HQ 数据集上的训练，在 Kaggel 提供的免费算力平台，完成了 DDPM 代码的复现工作，最后导出得到一个 notebook。

4.2 实验环境搭建

我们设置所有实验的 T = 1000，以便采样过程中所需的神经网络评估次数与之前的工作相匹配。我们设置正向过程方差为从 β1 = 10−4 到 βT = 0.02 线性增加的常数。这些常数被选为相对于 [−1,1] 的数据较小，以确保反向和正向过程具有近似相同的功能形式，同时保持 xT 时的信噪比尽可能小 (在实验中，LT =DKL(q(xT|x0) N(0, I))≈ 每维 10−5 位)。为了表示相反的过程，我们使用 U-Net 主干类似于未屏蔽的 PixelCNN++，并在整个中进行组规范化。参数是跨时间共享的，它使用 Transformer 正弦位置嵌入向网络指定。我们在 16 × 16 特征地图分辨率处使用自我注意。可以将源图像 x0, x ’ 0 q(x0) 用 q 作为随机编码器在隐空间内插值，xt, xt q(xt|x0)，然后将线性插值的隐图像 xt=(1−λ)x0 + λx0 通过反过程解码到图像空间 x0 p(x0|xt)。实际上，我们使用相反的过程来从源图像的线性插值损坏版本中去除伪影。我们固定了不同的 λ 值的噪声，所以 xt 和 x 不保持不变。

4.3 界面分析与使用说明

可以通过修改训练代码中的数据集选择代码，代码提供了 MNist 跟 CelebA HQ 两个数据集进行选择。

```
experiment.configs(configs, {
    'dataset': 'MNIST',#'CelebA',   # 'MNIST'
    'image_channels': 1,#3,   # 1,
    'epochs': 10,#100,   # 5,
})
```

图 4: 数据集选择

4.4 创新点

创新点在于简化了目前已公开的代码，并整合成了一个文档，且做了相应的注释工作，因为 DDPM 是一个比较吃算力的算法，简化后的代码可以于任何提供免费算力的平台进行运行，如 Kaggle 等，不再需要再单独上传到服务器进行代码运行，便于代码的学习与实现。

5 实验结果分析

DDPM 考虑到算力问题，只能在 MNist, 也就是手写数字数据集上进行了测试，下图是经过第十个 epoch 后经过扩散和逆扩散过程生成的图片，得到了很好的还原。

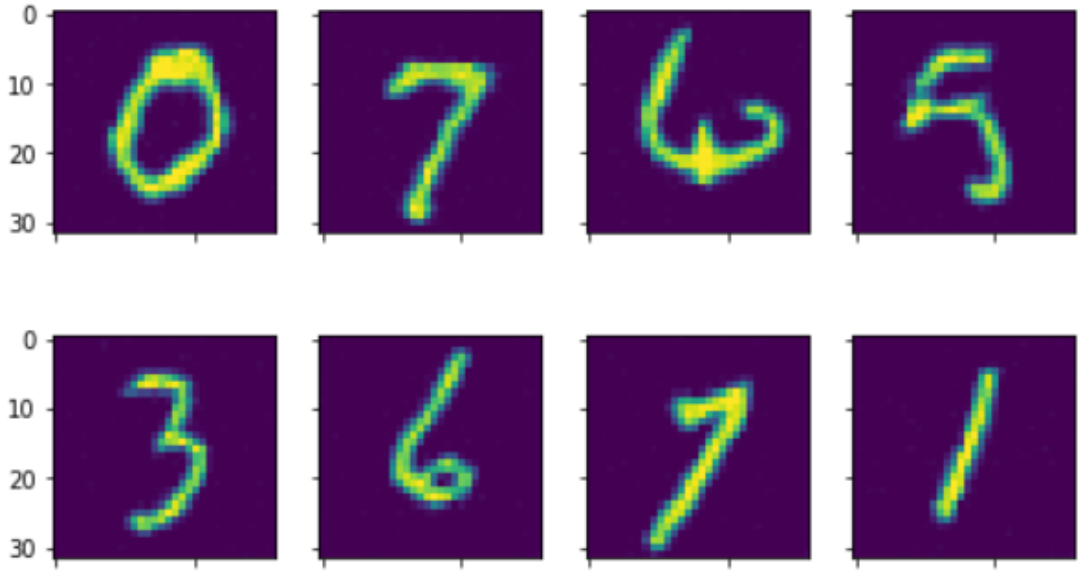

图 5: 实验结果示意

6 总结与展望

复现 Diffusion Model 的过程当中，很容易陷入具体的数学细节，本文对数学公式的推导没有过于仔细，而是直接给出了推导结果，对于数学的详细证明与推导可以参考原论文的附录。以后可以进一步的研究方向是基于 DDPM 加速后的 DDIM，以及目前性能很佳的 Stable Diffusion 论文的学习与复现工作。

参考文献

[1] JONATHAN HO P A, Ajay Jain. Denoising Diffusion Probabilistic Models[J]. NeurIPS, 2020.