

实验指导：生成式模型

一、实验背景

生成式模型是机器学习模型中的一类，主要作用是生成和训练数据性质相似的新样本。这类模型通过学习数据的统计规律与潜在结构，能产出类似训练数据的新数据，在图像生成、文本生成、音乐生成等诸多领域都有着重要应用。

本次实验基于PaddlePaddle框架，开展AE（自编码器）、VAE（变分自编码器）、Diffusion Model（扩散模型）相关的训练与推理实践。通过动手操作，结合课堂所学知识，能更深入地理解生成式模型。

二、实验目标

1. 编写AE相关代码，探索自编码器将图像嵌入隐空间并还原图像的完整过程。
2. 编写VAE相关代码，对比与AE的差异，学习VAE从分布采样并生成图像的方法。
3. 编写Diffusion Model相关代码，透彻理解扩散模型加噪、去噪的流程。

三、实验内容

任务一：AE生成手写数字

补全AutoEncoder类中的编码器（encoder）和解码器（decoder）部分，具体要求如下：

- **编码器（encoder）网络结构顺序：**
 - 2D卷积层（input_channel=3, output_channel=16, kernel_size=3, stride=1, padding=1）
 - ReLU激活层
 - BatchNorm归一化层
 - MaxPool池化层（kernel_size=2, stride=2）
 - 2D卷积层（output_channel=32, kernel_size=3, stride=1, padding=1）
 - ReLU激活层
 - BatchNorm归一化层
 - MaxPool池化层（kernel_size=2, stride=2）
- **解码器（decoder）网络结构顺序：**
 - 转置2D卷积层（output_channel=16, kernel_size=2, stride=2）
 - ReLU激活层
 - BatchNorm归一化层
 - 转置2D卷积层（output_channel=3, kernel_size=2, stride=2）
 - Sigmoid激活层

编写过程中，注意数据维度的变化，若对转置卷积存在疑问，可参考相关资料。

任务二：VAE生成手写数字

1. **完成重参数化操作**：依据公式 $z = \mu + \sigma \odot \epsilon$ 实现，其中 μ 是隐变量的均值， σ 是标准差， ϵ 是从标准正态分布 $N(0, 1)$ 中采样得到的。思考采用该重参数化操作，而非直接从分布 $N(\mu, \sigma^2)$ 采样的原因。
2. **完成KL损失函数**：参考公式 $D_{KL}(q(z|x)|p(z)) = \frac{1}{2} \sum_{j=1}^J (\sigma_j^2 + \mu_j^2 - 1 - \log(\sigma_j^2))$ 编写，这里 $p(z)$ 服从标准正态分布， $q(z|x)$ 代表隐变量分布， J 是隐变量维度。思考KL散度在生成过程中的作用。
3. **优化VAE**：修改损失函数、网络结构或者参数，不管你用什么方法，只要最终采样生成的汉字清晰可辨即可，以下内容是一些提示。

我们已经为你提供了一个简单的VAE。运行它，你会发现生成的手写数字效果并不是很理想。因此在本实验中，我们希望你能够优化当前的变分自编码器。VAE的训练过程涉及两个损失函数：重构损失和KL散度损失。

关于重构损失

`white_region_loss_coef` 和 `black_region_loss_coef` 的作用是为图像的不同区域赋予不同的损失权重。例如，在处理黑色区域（图像背景）时，可能不需要太高的重构精度，而对图像中的白色区域（重要特征）则希望模型更加关注。而 `white_threshold` 是用来区分“白色区域”和“黑色区域”的阈值。这个阈值定义了哪些像素应该被视为“白色区域”，并因此得到更高的重构损失。

你可以调整以上三个参数，查看模型的效果是否有改善~

同时，你可以尝试不同的损失函数（这个可能对最终的生成效果影响很大），例如：

```
white_region = paddle.where(x > white_threshold)
white_recon_loss = F.smooth_l1_loss(reconstruction[white_region],
x[white_region], reduction='sum')
black_region = paddle.where(x < white_threshold)
black_recon_loss = F.smooth_l1_loss(reconstruction[black_region],
x[black_region], reduction='sum')
recon_loss = white_region_loss_coef * white_recon_loss + black_region_loss_coef *
black_recon_loss
```

对于不同的损失函数和聚合方式，训练的稳定性和学习率的选择可能会有所不同，记得适当调整

`learning_rate` 和 `num_epochs` 等参数~

关于KL散度损失

KL散度损失被加权为 `beta * kl_div`，你可以自行尝试来选择更合适的 `beta` 值。（调整 `beta` 值使得潜在空间更加有结构，帮助模型生成更具多样性的样本）

当然，你还可以尝试将编码器和解码器的网络结构进行一些改变。

在最后提供了隐变量可视化的代码，通过这个部分你可以直观感受AE和VAE的区别，如果看不出区别，可以观察不同数字种类之间的过渡情况，也即观察类与类的混合情况以及图中的空白区域情况，要记住VAE是通过KL散度约束了其隐变量的分布。

任务三：扩散模型原理

这部分只需填写三个函数，代码中已以图片形式展示公式，并给出注释提示，理解原理后认真填写，难度不大。