HW2-图像生成

WGAN

网络结构与训练方法

详细结构请参考net.txt文件，以下简略说明，省略BN，激活函数等

- 生成器，输入(128,)
  - Linear(128, 4*4*128)
  - 带上采样的卷积块1，形状变为8*8
  - 带上采样的卷积快2，形状变为16*16
  - 带上采样的卷积快3，形状变为32*32
  - 卷积层，通道由128变为3
- 辨别器 (3,32,32)
  - 带下采样的卷积块1，形状变为16*16
  - 带下采样的卷积块2，形状变为8*8
  - 不带下采样的卷积块3
  - 不带下采样的卷积块4
  - 此处没有具体的层，但在execute过程中进行了mean操作，相当于图像大小变为1

    - 线性层，投影到1维

训练过程

- 采用Adam，lr=2e-4, betas=(0, 0.9)
- 使用改进的WGAN-DIV，参数k,p取为2，6
- 步数10000，辨别器每步训5次

1. 训练集获得真实图片，采样噪声输入G得到fake
2. 二者均输入D得到分数
3. 计算散度，回传
4. 上述过程重复5次
5. 采样噪声输入G
6. 对得到的fake输入D得到los
7. 回传更新G

训练结果的变化

请见`gifs/wgan.gif`

最终结果与分析

- 清晰度较高
- 确实类似真实图片，如果不考虑CIFAR的几个类别标签，确实需要认真看才能发现是生成的图片
- 种类较为丰富，没出现明显的特征坍塌

Beta-VAE

网络结构与训练方法

详细结构请参考net.txt文件，以下简略说明，省略BN，激活函数等

- 编码器 输入(n, 3, 32, 32)
  - conv(in=3, out=32, k=4, s=2, p=1) 16*16
  - conv(in=32, out=64, k=4, s=2, p=1) 8*8
  - conv(in=64, out=128, k=4, s=2, p=1) 8*8
  - conv(in=128, out=128, k=4, s=2, p=1) 4*4
  - conv(in=128, out=256, k=4, s=1, p=0) 1*1
  - linear(256, 1024)
  - linear(1024, 2048)
- 解码器 输入(n, z_dim=100)

    - linear(100, 1024)
    - linear(1024, 4096=256*4*4)
    - convtrans(256, 256, 4, 2, 1) 8*8
    - convtrans(256, 256, 3, 1, 1) 8*8
    - convtrans(256, 128, 4, 2, 1) 16*16
    - convtrans(128, 128, 3, 1, 1) 16*16
    - convtrans(128, 64, 4, 2, 1) 32*32
    - convtrans(64, 3, 3, 1, 1) 32*32

训练

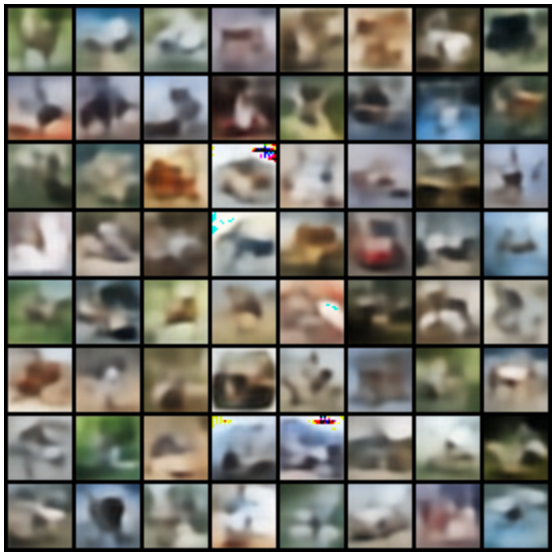
- Adam: lr=1e-4, betas=(0.9, 0.999)
- beta采用2，4，6

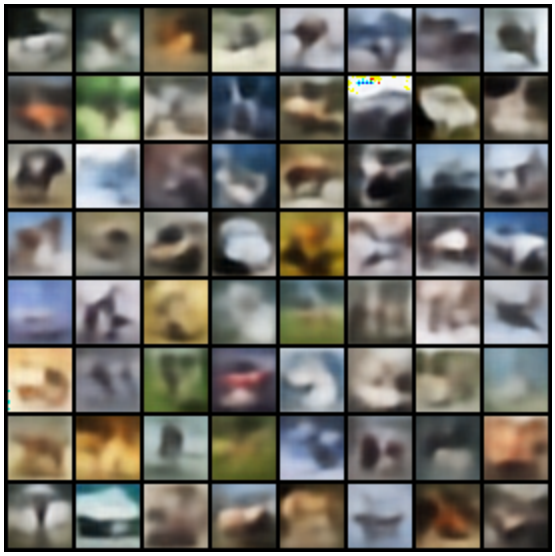
1. 通过编码器得到分布的参数
2. 采样后经过解码器得到重构输出
3. 计算KL散度以及MSE
4. 更新

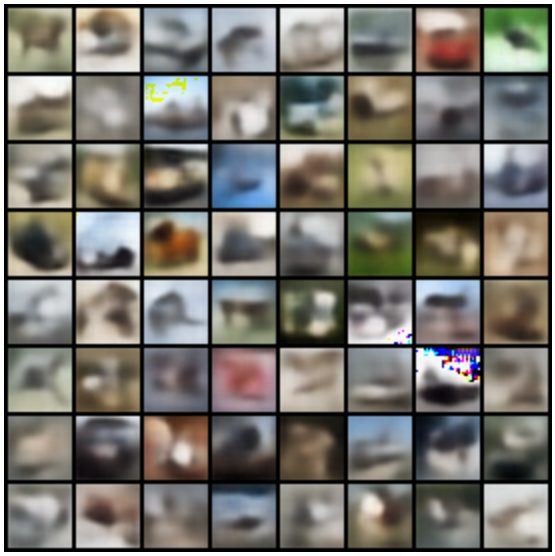
训练结果的变化

请见`gifs/n_vae.gif`

最终结果与分析



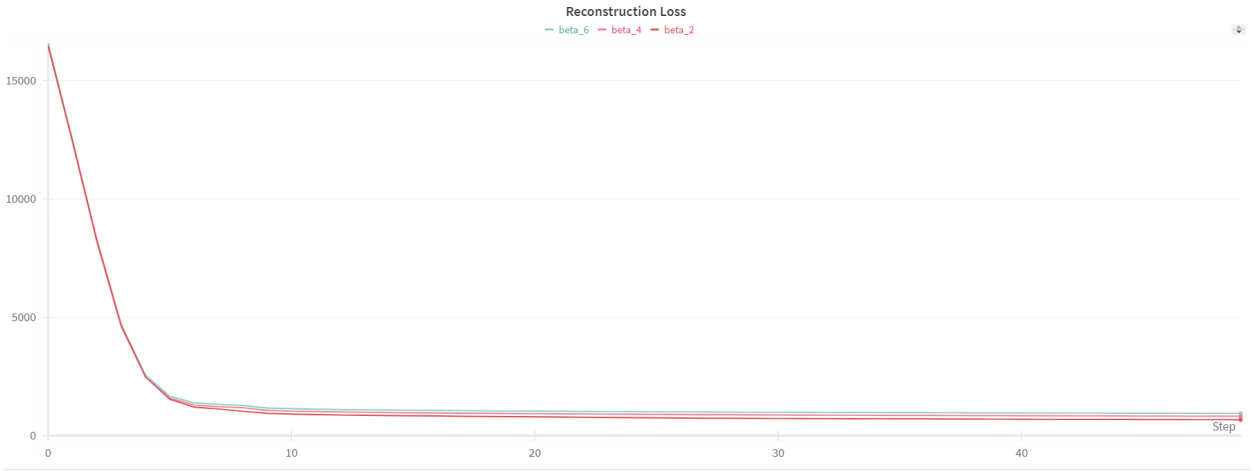


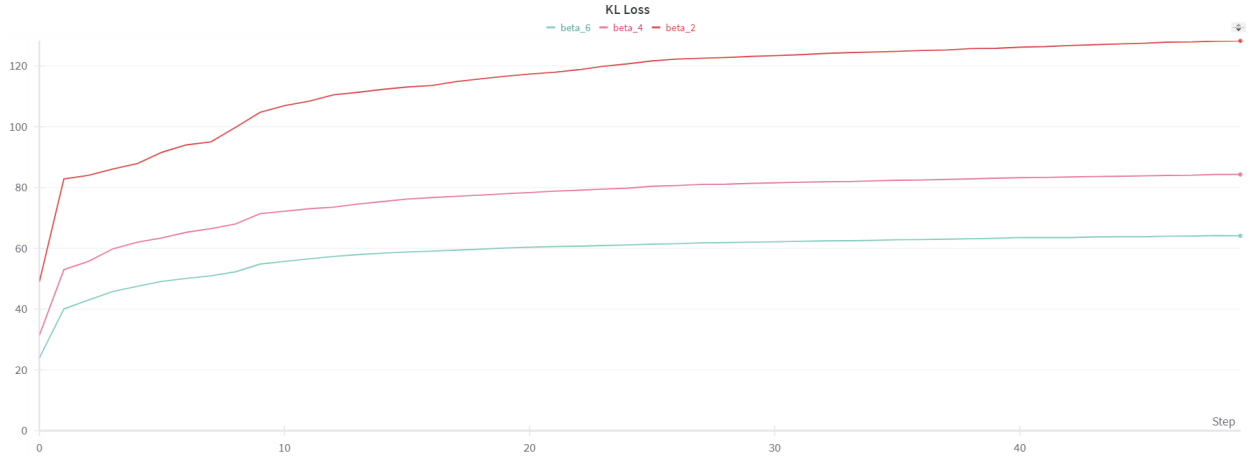


从上到下，从左到右依次是beta为2,4,6时的结果

- 结果不那么清晰，比较模糊，就像是高斯滤波之后一样——分析原因可能是MSE损失下，这样的模糊的图片（像是滤波后的结果）loss比较小
- 没有明显的坍塌问题

不同beta的比较





分析上面的训练loss以及训练结果

- beta越大，KL占的比重越大，优化效果越好，实际上鼓励更加有效的隐编码，鼓励图片之间分离。（我的理解是在CIFAR下就更加倾向于某一个特定的label，而不是又像A又像B）
- beta越大，MSE占比越小，上面也可看出较大的beta导致重构优化效果略差，看图也发现beta更大的似乎更模糊
- 实际上beta代表着图像质量以及分离程度之间的一种权衡。beta越大，可能生成越真实越好的图片，但是可能分布上比较奇怪，同时含有几种label对应的图片的特点。