ESRGAN

1 Reading Notes

由于之前的SRGAN产生的假图像总会存在一定的伪影，为了解决这个问题使得生成的图片更加真实，本文通过对SRGAN的三个主要成分（网络结构，对抗的损失以及感知损失）分别进行了改进，最终得到增强后的ESRGAN。

首先，对网络结构的改进指的是引入一种Residual-in-Residual Dense Block (RRDB)的结构，并除去BN层，这可以使得训练更深表达能力更强的网络，同时残差部分引入下一部分前，会乘上一个0 1的系数去防止不稳定性，而BN层的去除是实验判断的，去除之后对结果的影响不大并且可以训练更深的网络。其次，对抗的损失的改进是基于Relativisitic的思想，是去判断一副真的图片是不是比假的更真，或假的图片比真实图片假多少，这从后续的表达式中可以看出。最后还引入了感知损失，因为之前使用的评价标准PSNR更多针对像素进行的，这和我们人类对图片真假的评判方式不同，因此引入一种感知损失用于使其更加符合现实，这一点想法可以借鉴用于我现在的工作，在原有的像素级损失上引入感知损失。

2 Code about Network and Loss

对抗损失：

$$L_D^{Ra} = -\mathbb{E}_{x_r}\left[\log\left(D_{Ra}\left(x_r, x_f\right)\right)\right] - \mathbb{E}_{x_f}\left[\log\left(1 - D_{Ra}\left(x_f, x_r\right)\right)\right] \tag{1}$$

其中，$D_{Ra}\left(x_r, x_f\right) = \sigma\left(C\left(x_r\right) - \mathbb{E}_{x_f}\left[C\left(x_f\right)\right]\right)$ 借用了相对平均判别器RaD的想法。$C(x)$表示判别器的输出，$\mathbb{E}_x$表示对mini-batch内该类型数据取平均。对应代码如下所示，L_D^{Ra}即为代码中的$loss_D$，等于两部分损失相加，而公式中的前半部分即为代码中的$loss_real$，后半部分为代码中的$loss_fake$。

```
loss_D = (loss_real + loss_fake) / 2
loss_real = criterion_GAN(pred_real - pred_fake.mean(0, keepdim=True),
    valid)
loss_fake = criterion_GAN(pred_fake - pred_real.mean(0, keepdim=True), fake
    )
criterion_GAN = torch.nn.BCEWithLogitsLoss()
```

生成损失：

$$L_G = L_{percep} + \lambda L_G^{Ra} + \eta L_1 \tag{2}$$

其中，$L_G^{Ra} = -\mathbb{E}_{x_r}\left[\log\left(1 - D_{Ra}\left(x_r, x_f\right)\right)\right] - \mathbb{E}_{x_f}\left[\log\left(D_{Ra}\left(x_f, x_r\right)\right)\right]$ 表示生成器中的对抗损失，$L_1 = \mathbb{E}_{x_i}\left\|G\left(x_i\right) - y\right\|_1$ 表示输出图片与真值的l_1损失。L_{percep}表示感知损失，就是利用VGG网络提取出来的特征图去计算误差l_1损失，之前是在激活后计算，本文用于激活前。对应代码为：

```
loss_G = loss_content + opt.lambda_adv * loss_GAN + opt.lambda_pixel *
    loss_pixel
loss_pixel = criterion_pixel(gen_hr, imgs_hr)
criterion_pixel = torch.nn.L1Loss()
loss_GAN = criterion_GAN(pred_fake - pred_real.mean(0, keepdim=True), valid
    )
loss_content = criterion_content(gen_features, real_features)
criterion_content = torch.nn.L1Loss()
```

如上所示，L_G^{Ra} 分为3部分，第一部分为感知损失，计算假的特征图与真值特征图的$l_1$loss，第二部分为生成器的对抗损失，第三部分为输出图片与真值的pixel loss。

网络部分的代码为：

```
generator = GeneratorRRDB(opt.channels, filters=64, num_res_blocks=opt.
    residual_blocks)
discriminator = Discriminator(input_shape=(opt.channels, *hr_shape))
feature_extractor = FeatureExtractor()
```

网络部分分为生成器网络、判别器网络，以及固定不优化的特征提取器网络(VGG19)。