

$$A \xrightarrow{G} B' \xrightarrow{F} A' \qquad LA \cdot |F(G(b)) - \pi|$$

$$B \xrightarrow{F} A' \xrightarrow{G} B'$$

通过 A' 与 A 的 loss 及传递给 F 及 T 得到 G ,
 A' 与 A 的 loss 及传递给 F 及 T 得到 G ,
 A' 与 A 的 loss 及传递给 F 及 T 得到 G ,

使 G 成为 B' 可以重建成 A' , 并与固定 A 一致。

而D则可使B'与B更近。

127

$$\text{parallelogram} \xrightarrow{F} \text{rectangle} \xrightarrow{G} \text{parallelogram}$$

是而 猫 \rightarrow 狗, 狗的不确定度, 所以 CycleGAN, 又值保持顺序

是而 猫 \rightarrow 狗, 狗部的石南会咬, 所以 *cyckon*, *sh*

为什么要G和F两个生成器

我们之前已经说过，CycleGAN的原理可以概述为：将一类图片转换成另一类图片。也就是说，现在有两个样本空间，X和Y，我们希望把X空间中的样本转换成Y空间中的样本。

因此，实际的目标就是学习 $X \rightarrow Y$ 的映射。我们设这个映射为G。它对应着GAN中的生成器，G可以将X中的图片X转换为Y中的图片G(x)。对于生成的图片，我们还需要GAN中的判别器来判别它是否为真实图片，由此构成对抗生成网络。设这个判别器为 D_Y 。这样的话，根据这里的生成器和判别器，我们就可以构造一个GAN损失，表达式为：

$$L_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)} [\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)} [\log (1 - D_Y(G(x)))]$$

这个损失实际上和原始的GAN损失是一模一样的，但单纯的使用这一个损失是无法进行训练的。原因在于，映射G完全可以将所有x都映射为Y空间中的同一张图片，使损失无效化。我们再假设一个映射F，它可以将Y空间中的图片y转换为X中的图片F(y)。同样的我们为F也引入一个判别器 D_X ，由此可以同样定义一个GAN的损失：

$$L_{GAN}(F, D_X, Y, X) = \mathbb{E}_{x \sim p_{data}(x)} [\log D_X(x)] + \mathbb{E}_{y \sim p_{data}(y)} [\log (1 - D_X(F(y)))]$$

CycleGAN同时学习G和F两个映射，并要求将X的图片转换到Y空间后，应该还可以转换回来。这样就杜绝模型把所有X的图片都转换为Y空间中的同一张图片了。对此，作者又提出了所谓的“循环一致性损失” (cycle consistency loss)。

而循环一致性损失(两个生成器的loss加起来)就定义为：

$$L_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)} [\|G(F(y)) - y\|_1]$$

需要 $X \xrightarrow{G} Y$ ，但若G将所有X映射到同一个Y，则不行。

所以要 $Y \xrightarrow{F} X$ ，

再利用循环一致，使可正常学习

