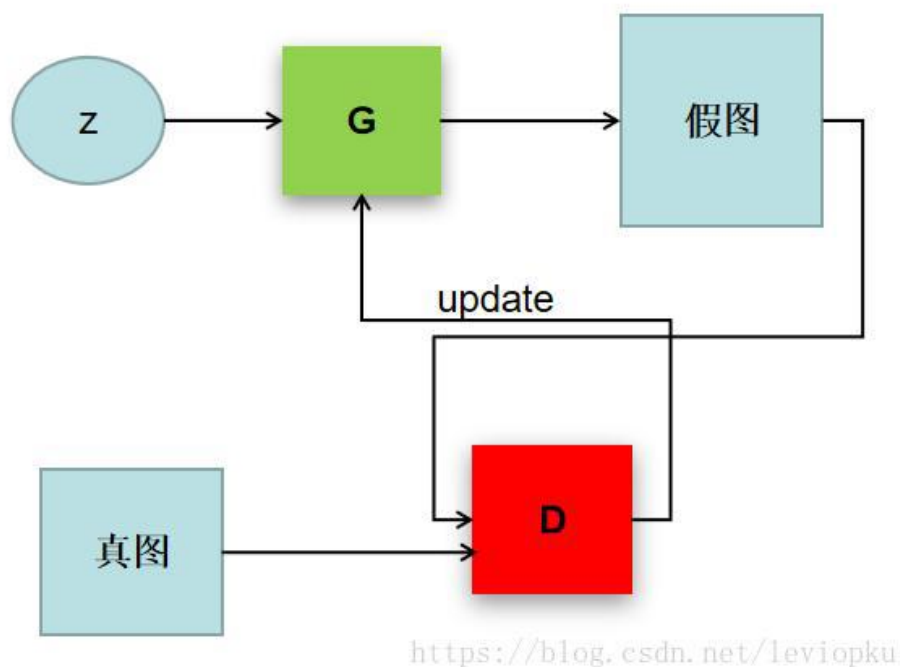


理解 GAN 的两大护法  $G$  和  $D$

$G$  是 generator, 生成器: 负责凭空捏造数据出来

$D$  是 discriminator, 判别器: 负责判断数据是不是真数据

这样可以简单的看作是两个网络的博弈过程。在最原始的 GAN 论文里面,  $G$  和  $D$  都是两个多层感知机网络。首先, 注意一点, GAN 操作的数据不一定非得是图像数据, 不过为了更方便解释, 我在这里用图像数据为例解释以下 GAN:



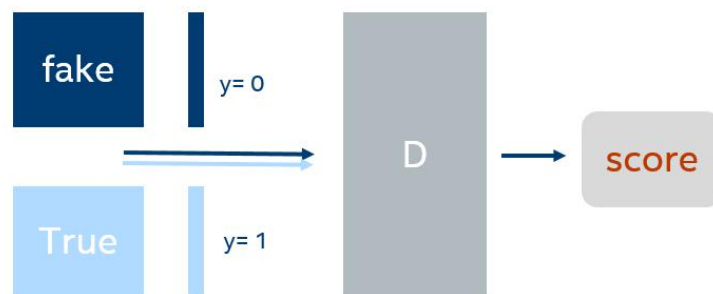
稍微解释以下上图,  $z$  是随机噪声(就是随机生成的一些数, 也就是 GAN 生成图像的源头)。

$D$  通过真图和假图的数据(相当于天然 label), 进行一个二分类神经网络训练(想各位必再熟悉不过了)。  $G$  根据一串随机数就可以捏造一个“假图像”出来, 用这些假图去欺骗  $D$ ,  $D$  负责辨别这是真图还是假图, 会给出一个 score。比如,  $G$  生成了一张图, 在  $D$  这里得分很高, 那证明  $G$  是很成功的; 如果  $D$  能有效区分真假图, 则  $G$  的效果还不太好, 需要调整参数。 GAN 就是这么一个博弈的过程。

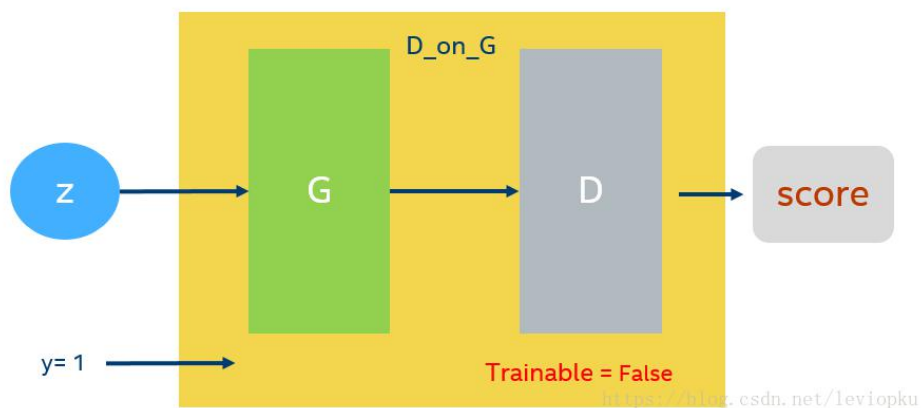
那么，*GAN* 是怎么训练呢？

根据 *GAN* 的训练算法，我画一张图：

### When training D:



### When training G:



*GAN* 的训练在同一轮梯度反传的过程中可以细分为 2 步，先训练 *D* 再训练 *G*；注意不是等所有的 *D* 训练好以后，才开始训练 *G*，因为 *D* 的训练也需要上一轮梯度反传中 *G* 的输出值作为输入。

当训练 *D* 的时候，上一轮 *G* 产生的图片，和真实图片，直接拼接在一起，作为  $x$ 。然后根据，按顺序摆放 0 和 1，假图对应 0，真图对应 1。然后就可以通过， $x$  输入生成一个 *score*（从 0 到 1 之间的数），通过 *score* 和  $y$  组成的损失函数，就可以进行梯度反传了。（我在图片上举的例子是  $batch = 1$ ， $len(y)=2*batch$ ，训练时通常可以取较大的 *batch*）

当训练  $G$  的时候，需要把  $G$  和  $D$  当作一个整体，我在这里取名叫做 ' $D_{on\_G}$ '。这个整体(下面简称  $DG$  系统)的输出仍然是  $score$ 。输入一组随机向量，就可以在  $G$  生成一张图，通过  $D$  对生成的这张图进行打分，这就是  $DG$  系统的前向过程。 $score=1$  就是  $DG$  系统需要优化的目标， $score$  和  $y=1$  之间的差异可以组成损失函数，然后可以反向传播梯度。注意，这里的  $D$  的参数是不可训练的。这样就能保证  $G$  的训练是符合  $D$  的打分标准的。这就好比：如果你参加考试，你别指望能改变老师的评分标准

需要注意的是，整个  $GAN$  的整个过程都是无监督的(后面会有监督性  $GAN$  比如  $cGAN$ )，怎么理解这里的无监督呢？

这里，给的真图是没有经过人工标注的，你只知道这是真实的图片，比如全是人脸，而系统里的  $D$  并不知道来的图片是什么玩意儿，它只需要分辨真假。 $G$  也不知道自己生成的是什  
么玩意儿，反正就是学真图片的样子骗  $D$ 。

正由于  $GAN$  的无监督，在生成过程中， $G$  就会按照自己的意思天马行空生成一些“诡异”的图片，可怕的是  $D$  还能给一个很高的分数。比如，生成人脸极度扭曲的图片。这就是无监督目的性不强所导致的，所以在同年的  $NIPS$  大会上，有一篇论文 *conditional GAN* 就加入了监督性进去，将可控性增强，表现效果也好很多。

来自 <<https://www.pianshen.com/article/2495690365/?jsessionid=A929EB195A785EF5D377012CF21FB7FF>>