1. GAN的基本思想

GAN全称对抗生成网络，顾名思义是生成模型的一种，而他的训练则是处于一种对抗博弈状态中的。下面举例来解释一下GAN的基本思想。

*假如你是一名篮球运动员，你想在下次比赛中得到上场机会。*

*于是在每一次训练赛之后你跟教练进行沟通：*

*你：教练，我想打球*

*教练：（评估你的训练赛表现之后）... 算了吧*

*（你通过跟其他人比较，发现自己的运球很差，于是你苦练了一段时间）*

*你：教练，我想打球*

*教练：... 嗯 还不行*

*（你发现大家投篮都很准，于是你苦练了一段时间的投篮）*

*你：教练，我想打球*

*教练： ... 嗯 还有所欠缺*

*（你发现你的身体不够壮，被人一碰就倒，于是你去泡健身房）*

*......*

*通过这样不断的努力和被拒绝，你最终在某一次训练赛之后得到教练的赞赏，获得了上场的机会。*

值得一提的是在这个过程中，所有的候选球员都在不断地进步和提升。因而教练也要不断地通过对比场上球员和候补球员来学习分辨哪些球员是真正可以上场的，并且要“观察”得比球员更频繁。随着大家的成长教练也会会变得越来越严格。

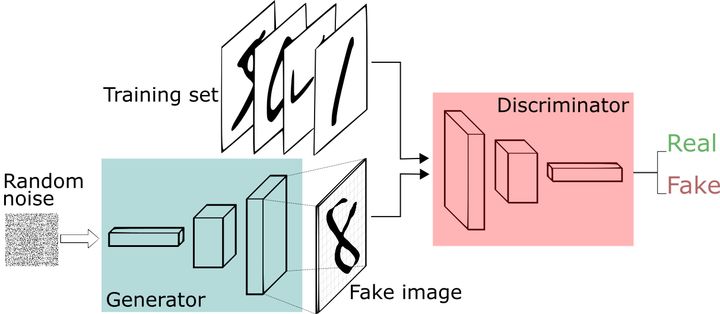
2. GAN浅析

2.1 GAN的基本结构

GAN的主要结构包括一个生成器G（Generator）和一个判别器D（Discriminator）。

在上面的例子中的球员就相当于生成器，我们需要他在球场上能有好的表现。而球员一开始都是初学者，这个时候就需要一个教练员来指导他们训练，告诉他们训练得怎么样，直到真的能够达到上场的标准。而这个教练就相当于判别器。

下面我们举另外一个手写字的例子来进行进一步窥探GAN的结构。



我们现在拥有大量的手写数字的数据集，我们希望通过GAN生成一些能够以假乱真的手写字图片。主要由如下两个部分组成：

1、定义一个模型来作为生成器（图三中蓝色部分Generator），能够输入一个向量，输出手写数字大小的像素图像。2、定义一个分类器来作为判别器（图三中红色部分Discriminator）用来判别图片是真的还是假的（或者说是来自数据集中的还是生成器中生成的），输入为手写图片，输出为判别图片的标签。

其他解释：

首先这个网络模型（定义在上面），先传入生成器中，然后生成器生成图片之后，把图片传入判别器中，标签此刻传入的是1,真实的图片，但实际上是假图，此刻判别器就会判断为假图，然后模型就会不断调整生成器参数，此刻的判别器的参数被设置为为不可调整，d.trainable=False，所以为了不断降低loss值，模型就会一直调整生成器的参数，直到判别器认为这是真图。此刻判别器与生成器达到了一个平衡。也就是说生成器产生的假图，判别器已经分辨不出来了。所以继续迭代，提高判别器精度，如此往复循环，直到生成连人都辨别不了的图片。

## **2.2 GAN的训练方式**

前面已经定义了一个生成器（Generator）来生成手写数字，一个判别器（Discrimnator）来判别手写数字是否是真实的，和一些真实的手写数字数据集。那么我们怎样来进行训练呢？

Tips: 这里的生成器可以是任意可以输出图片的模型，比如最简单的全连接神经网络，又或者是反卷积网络等。这里大家明白就好。

这里输入的向量我们将其视为携带输出的某些信息，比如说手写数字为数字几，手写的潦草程度等等。由于这里我们

对于输出数字的具体信息不做要求，只要求其能够最大程度与真实手写数字相似（能骗过判别器）即可。所以我们使用随机生成的向量来作为输入即可，这里面的随机输入最好是满足常见分布比如均值分布，高斯分布等。

Tips: 假如我们后面需要获得具体的输出数字等信息的时候，我们可以对输入向量产生的输出进行分析，获取到哪些维度是用于控制数字编号等信息的即可以得到具体的输出。而在训练之前往往不会去规定它。

## **2.2.2** 关于判别器

对于判别器不用多说，往往是常见的判别器，输入为图片，输出为图片的真伪标签。

Tips: 同理，判别器与生成器一样，可以是任意的判别器模型，比如全连接网络，或者是包含卷积的网络等等。

**2.2.3** 如何训练

