

7月23日

一、李宏毅2021春机器学习课程第6.1节：生成式对抗网络 GAN (一)

1 能够作为生成器的神经网络 GAN

生成式对抗网络 (GAN, Generative Adversarial Networks) 是一种深度学习模型，是近年来复杂分布上无监督学习最具前景的方法之一。模型通过框架中（至少）两个模块：生成模型 (Generative Model) 和判别模型 (Discriminative Model) 的互相博弈学习产生相当好的输出。

generative adversarial network，它的缩写是GAN，中文名称生成式对抗网络，这也是不少人在还没有接触机器学习之前都有听说过的一个很出名的模型。它其实有很多各式各样的变形，你可以在网路上找到一个GAN的[动物园](#)：

All Kinds of GAN ... <https://github.com/hindupuravinash/the-gan-zoo>

GAN	<ul style="list-style-type: none">• SeUDA - Semantic-Aware Generative Adversarial Nets for Unsupervised Domain Adaptation Segmentation
ACGAN	<ul style="list-style-type: none">• SG-GAN - Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption (github)
BGAN	<ul style="list-style-type: none">• SG-GAN - Sparsely Grouped Multi-task Generative Adversarial Networks for Facial Attr
CGAN	<ul style="list-style-type: none">• SGAN - Texture Synthesis with Spatial Generative Adversarial Networks
DCGAN	<ul style="list-style-type: none">• SGAN - Stacked Generative Adversarial Networks (github)
EBGAN	<ul style="list-style-type: none">• SGAN - Steganographic Generative Adversarial Networks
fGAN	<ul style="list-style-type: none">• SGAN - SGAN: An Alternative Training of Generative Adversarial Networks
GoGAN	<ul style="list-style-type: none">• SGAN - CT Image Enhancement Using Stacked Generative Adversarial Networks and Tr
⋮	<ul style="list-style-type: none">• Segmentation Improvement
⋮	<ul style="list-style-type: none">• sGAN - Generative Adversarial Training for MRA Image Synthesis Using Multi-Contrast
	<ul style="list-style-type: none">• SiftingGAN - SiftingGAN: Generating and Sifting Labeled Samples to Improve the Rem
	<ul style="list-style-type: none">• Classification Baseline in vitro
	<ul style="list-style-type: none">• SiGAN - SiGAN: Siamese Generative Adversarial Network for Identity-Preserving Face H
	<ul style="list-style-type: none">• SimGAN - Learning from Simulated and Unsupervised Images through Adversarial Trai
	<ul style="list-style-type: none">• SisGAN - Semantic Image Synthesis via Adversarial Learning

Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, Shakir Mohamed, “Variational Approaches for Auto-Encoding Generative Adversarial Networks”, arXiv, 2017

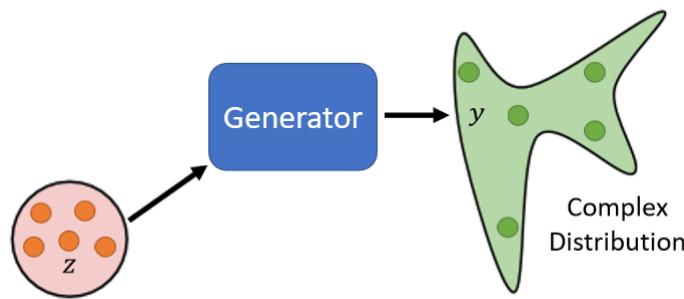
²We use the Greek α prefix for α -GAN, as AEGAN and most other Latin prefixes seem to have been taken
<https://deephunt.in/the-gan-zoo-79597dc8c347>.

9

2 动漫人物头像生成

直接通过一个例子来介绍什么是GAN，以及GAN要做什么，怎么实现的。假设我们现在的任务是让机器生成二次元人物的头像，假设现在是unconditional generation，就是没有输入x，只有输入的一个随机变量z

- Unconditional generation



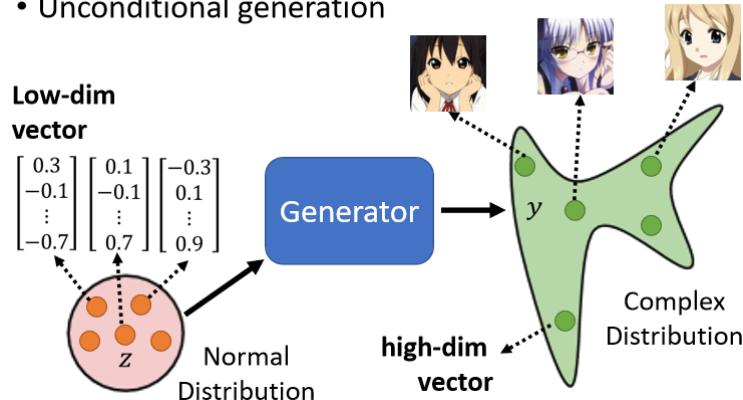
那之后我们在讲到 **conditional generation** 的时候，我们会再把x加回来，那输入的这个z是什么呢？

实际上我们可以假设**z是从一个normal distribution里采样出来的向量**，这个向量通常会是一个 **low-dimensional** 的向量，一般定位50, 100维，**它的大小是由你自己决定的**。

那现在我们从一个normal distribution里面采样出一个向量z，然后输入给GAN，GAN就给你一个对应的输出，那我们希望对应的输出就是一个二次元人物的脸。而**一张图片就是一个非常高维的向量**，所以**generator实际上做的就是产生一个非常高维的向量**。

当你输入的向量不同的时候，你的输出就会跟着改变，所以你从这个normal distribution里面采样到不同的z，那么每次输出的y也就不一样，但我们希望不管采样到什么z，输出来的都是动漫人物的脸。

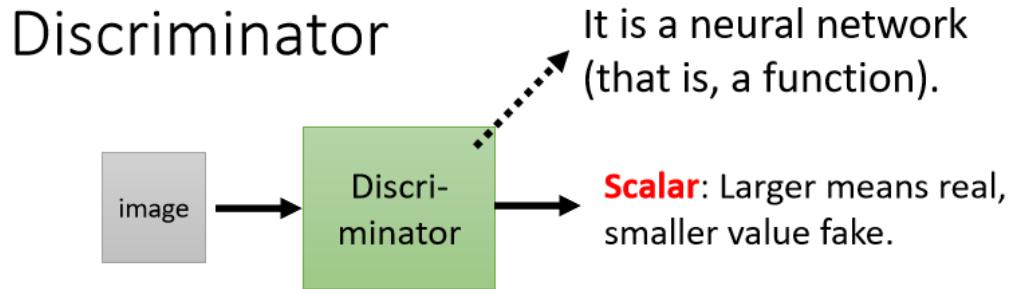
- Unconditional generation



3 判别器 (Discriminator)

在GAN里面一个特别的地方就是，除了generator以外，我们要多训练一个**discriminator**。

discriminator的作用是，输入一张图片，输出是一个数值，这个**数值越大就代表现在输入的这张图片越像是真实的二次元人物的图像**。



而discriminator的架构完全是你自己设计的，你可以用CNN，也可以用transformer等等，只要能够产生出你想要的输入输出，就可以了。

在这个例子里面，因为discriminator的输入是一张图片，很显然选择CNN很比较有优势，毕竟CNN在处理影像上有很多优点。

4 从自然选择看GAN的基本思想

为什么除了生成器之外，我们还需要多训练一个判别器呢，这里其实GAN的基本思想可以从生物进化的角度来看，我这里复述一下李宏毅老师举的一个比较有趣的例子：



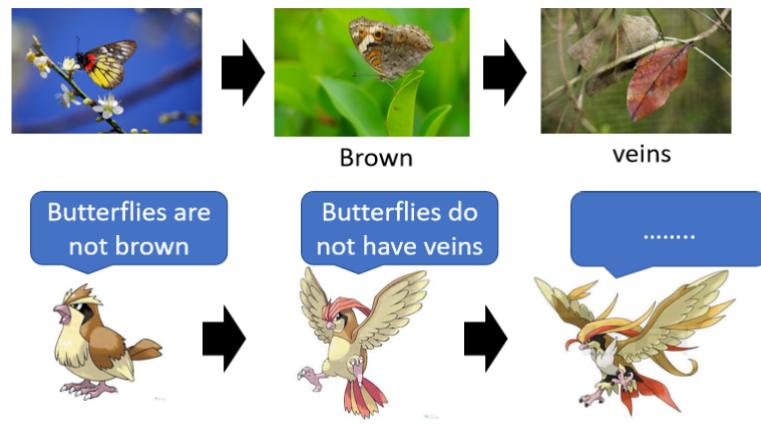
上面这张图似乎没什么特别的哈？不就是一个树枝和一片枯叶嘛，但其实这不是一片枯叶，这是枯叶蝶的拟态，枯叶蝶长得跟枯叶非常像，因此它可以躲避天敌。但枯叶蝶的祖先其实并不是长得像枯叶一样，也许他们原来也是五彩斑斓的，但为什么他们变成长得像枯叶一样，是因为有天择的压力。



Butterflies are not brown

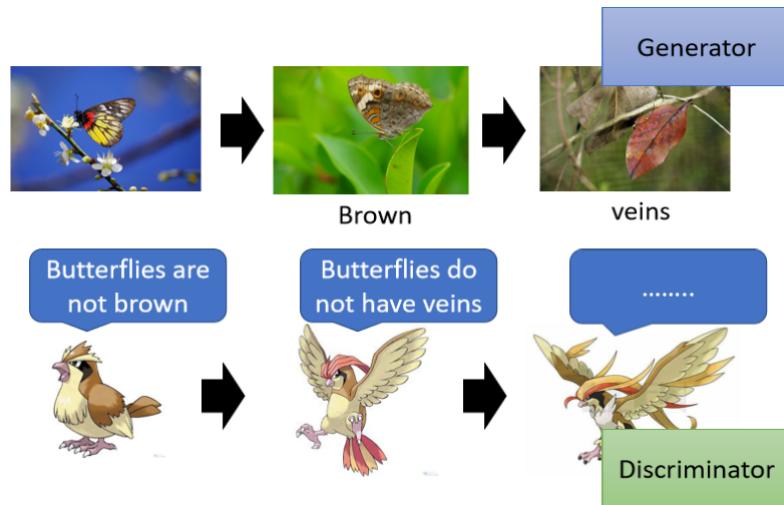


这个不是普通的麻雀，这个是波波（一种宝可梦），波波会吃枯叶蝶的祖先，在天择的压力之下，枯叶蝶就变成棕色的。因为波波它只会吃彩色的东西，它看到彩色的东西知道是蝴蝶，就把它吃掉，那看到棕色的东西，波波就不会去吃它。



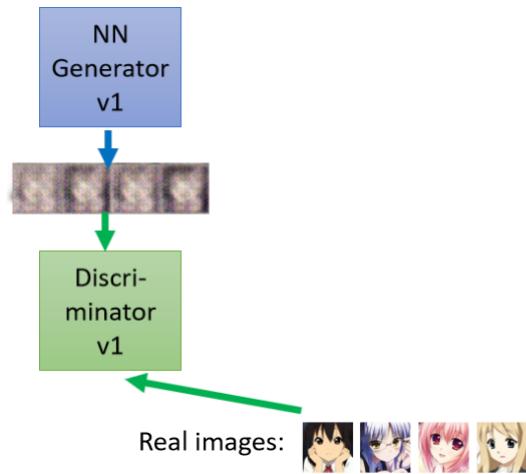
但是逐渐这样下去，只会吃彩色蝴蝶的波波慢慢的找不到足够的食物，也会被大自然淘汰了，在自然选择中获胜的波波都进化了，它们进化成了比比鸟，比比鸟在判断一个蝴蝶能不能吃的时候不会只看颜色，它会看它的纹路，它知道说没有叶脉的是蝴蝶，有叶脉的才是真正的枯叶。

在天择的压力之下，枯叶蝶就产生了拟态，产生了叶脉，想要骗过比比鸟，但是比比鸟它也有可能会再进化成大比鸟，那大比鸟可能可以分辨枯叶蝶跟枯叶的不同。



那这个是一个物种演化的故事，对应到GAN中的相关概念，枯叶蝶就是generator，那波波就是discriminator。

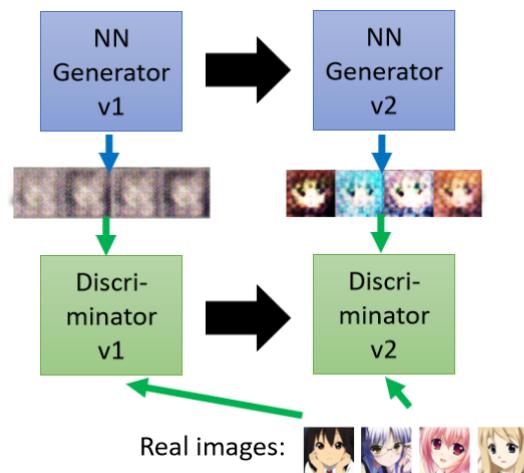
回到我们之前的例子中来，现在我们generator要做的事情，是画出二次元的人物头像，那generator学习画出二次元人物头像的过程是这样的：



第一代的generator它的参数几乎是完全随机的，所以它根本就不知道到底要怎么画二次元的人物，所以它画出来的东西就是一些莫名其妙的东西。

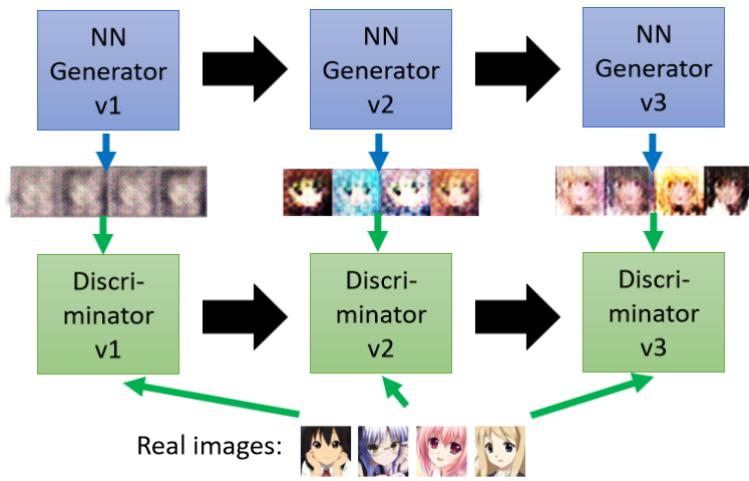
那**discriminator**学习的目标是**分辨generator的输出与真正的动漫头像的不同**，在现在的状况下可能非常的容易，对discriminator来说它只要看图片里面有没有两个黑黑的圆球，有眼睛就是真正的二次元人物，没有眼睛就是generator产生出来的东西。

接下来**generator就调整它的里面的参数**，Generator就**进化了**，它调整它里面的参数，调整的目标是为了骗过discriminator，假设discriminator判断一张图片是不是真实的依据是有没有眼睛，那generator就产生眼睛出来，以期能够骗过discriminator：



所以第二代的generator可以产生眼睛，这样就可以骗过第一代的discriminator，但是**discriminator也是会进化的**，第二代的discriminator会试图分辨generator产生的图片，跟真实图片之间的差异，它可能会发现说，generator产生的图片都没有头发也没有嘴巴，真实图片是有头发的也有嘴巴的。

接下来第三代的generator就会想办法去骗过第二代的discriminator，既然第二代的discriminator是看有没有头发和嘴巴来判断是不是真正的二次元人物，那第三代的generator就会把嘴巴加上去。



那discriminator也会逐渐的进步，它会越来越严苛，在这样的左右互搏之后，我们期望Generator产生的图片可以越来越像真实二次元的人物。

可以看到我们的generator跟discriminator中间有一个**对抗的关系**，所以就用了**adversarial**这个词语，总而言之，**generator跟discriminator既是对抗的，也是互相成就的**，其中任何一方太弱的话，都会导致双方都训练不起来。

5 GAN 的具体实现过程

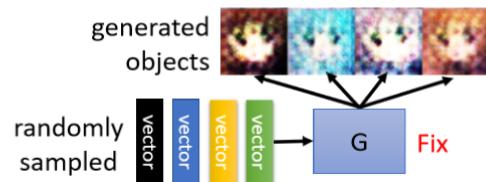
接下来介绍一下GAN的具体实现过程是怎样的，generator和discriminator，他们就是两个network，我们假设**generator跟discriminator的参数都已经初始化过了**。

步骤一：固定 generator G 的参数，只更新discriminator D

初始化完以后，接下来训练的第一步是，固定住你的**generator**的参数，只训练你的**discriminator**。

- Initialize generator and discriminator
- In each training iteration:

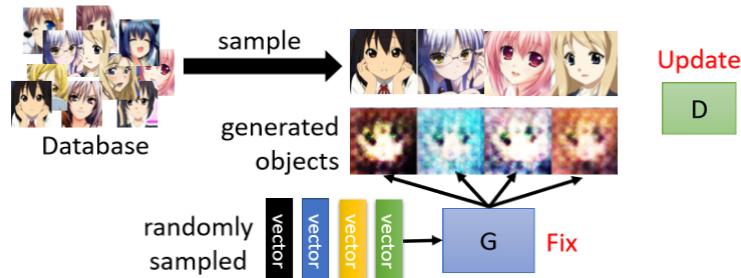
Step 1: Fix generator G, and update discriminator D



因为一开始**generator**的参数都是随机初始化的，并且我们又固定住了**generator**的参数，那它的输出完全都是乱七八糟的图片。

那假设我们有一个database，这个database里面有很多二次元人物的头像，从这个图库里面去sample一些二次元人物的头像出来，用这些真正的二次元人物头像，跟generator产生出来的结果，去训练你的discriminator。

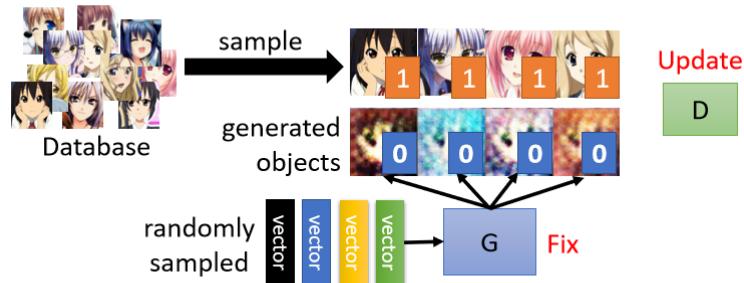
Step 1: Fix generator G, and update discriminator D



Discriminator learns to assign high scores to real objects and low scores to generated objects.

discriminator训练的目标是要分辨真正的二次元人物跟generator产生出来的二次元人物之间的差异，具体一点来说，实际上我们可能会把这些真正的人物都标1，Generator产生的图片都标0。

Step 1: Fix generator G, and update discriminator D



Discriminator learns to assign high scores to real objects and low scores to generated objects.

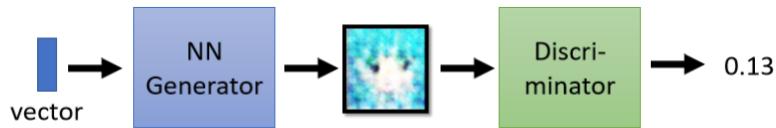
接下来对于discriminator来说，这就是一个**分类的问题**，你就把真正的人脸当作类别1，Generator产生的图片当作类别2，然后训练一个classifier就结束了。或者看做一个**回归的问题**，输出的值越接近1，就代表越接近真实图片；而越接近0，就代表越接近假的图片，这两种办法都可以。

步骤二：固定 discriminator D 的参数，只更新generator G

我们训练完discriminator以后，接下来**固定住discriminator，改为训练generator**。

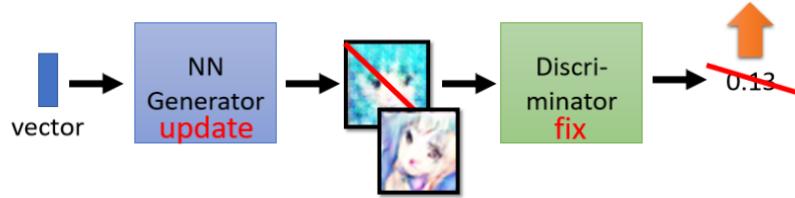
我们要让generator想办法去骗过discriminator，因为刚才discriminator已经学会分辨真图跟假图的差异，generator产生的图片如果可以骗过discriminator，那在discriminator足够强大的情况下，生成的图片就可以假乱真了。

Generator learns to “fool” the discriminator



实际的操作方法是这样的，你有一个generator，generator从gaussian distribution sample出来一个向量作为输入，然后输出一个图片的向量。

接下来我们把这个图片输入到Discriminator里面，Discriminator会给这个图片一个分数，分数越高表示越接近真实图片。

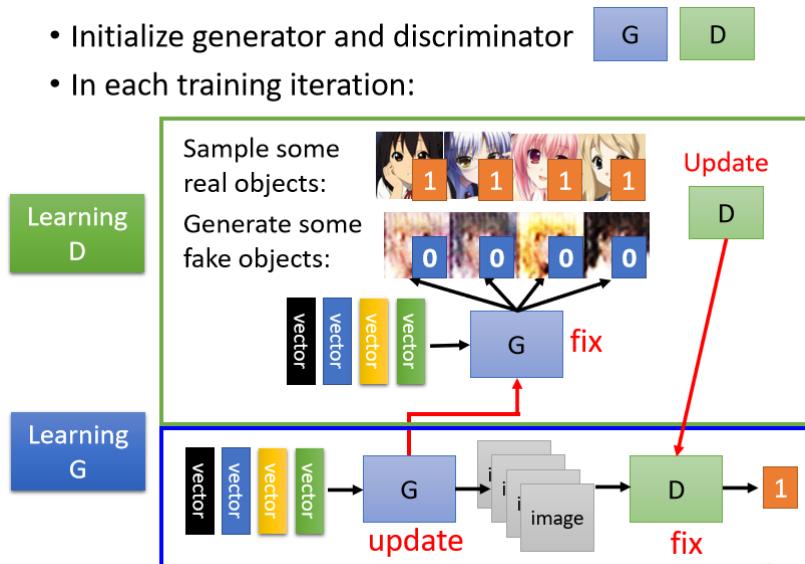


那Generator训练的目标就是要Discriminator输出的值越大越好，如果Generator调整参数之后输出的图片可以蒙骗Discriminator，也就是Discriminator会给予高分，那意味着Generator产生的图片是比较真实的。

所以现在讲了两个步骤

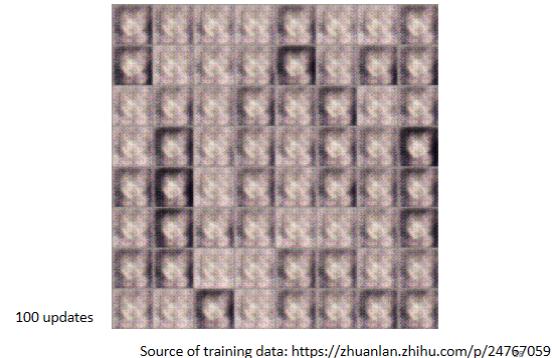
- 第一个步骤：固定generator，训练discriminator
- 第二个步骤：固定discriminator，训练generator

接下来就是**重复这两个步骤反复的训练**discriminator和generator，期待discriminator跟generator都可以做得越来越好，直到generator产生图片的效果能让我们比较满意。



6 动漫头像生成的具体实验结果

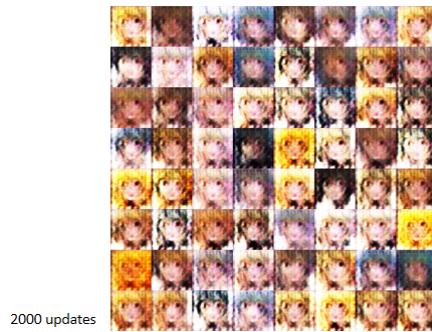
以下的结果是李宏毅老师在17年的时候做的， dataset的地址: <https://zhuanlan.zhihu.com/p/24767059>，训练了100个来回之后的结果如下：



这时generator还不知道在做些什么，但训练了1000个回合后的结果如下：



discriminator 和 generator 各自训练这样反复一千次以后，机器就产生了眼睛，机器知道说人脸就是要两个眼睛，所以它就把眼睛标上去，训练到两千次的时候，你发现嘴巴就出来了：



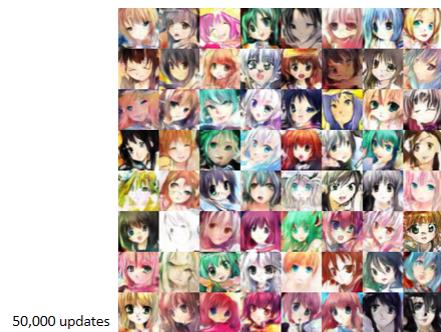
训练到五千次的时候，已经开始有一点人脸的样子了，而且你发现说机器学到了动画人物啊，就是要有那个水汪汪的大眼睛，所以它给每个人的眼睛呢都涂得非常的大：



接下来是训练两万个回合的结果：



然后是训练五万个回合的结果：



那你会发现这些生成的动漫人物大体上还不错，只是有一些比较崩坏，如果你有真的非常好的资料的话，也许你可以做出真的很好的结果。

这里有一个链接：

<https://www.gwern.net/images/gan/stylegan/2019-02-11-stylegan-danbooru2017faces-interpolation.mp4>

这个是用StyleGAN做的，那用StyleGAN可以做到这个地步：



可以看到效果已经相当好了，完全辨别不出来是机器自己产生的，这个结果还是很惊人的。

除了产生动画人物以外，当然也可以产生真实的人脸，有一个技术叫做progressive GAN，它可以产非常高清的人脸，你可能会问产生人脸有什么用呢，我去路边拍一个人产生出来的照片不是更像真的吗？

但是用**GAN**你可以产生你没有看过的人脸，甚至根本不存在的人脸。这就是GAN的神奇之处。