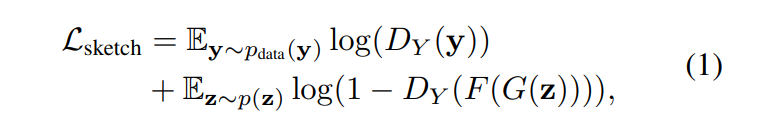
**复现论文：Sketch Your Own GAN(起草你自己的GAN对抗生成网络)**

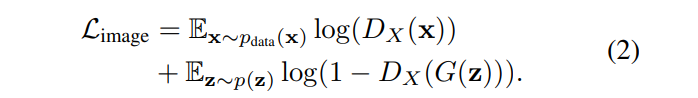
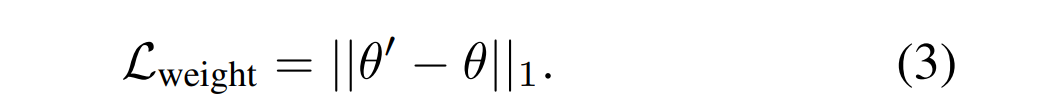
夏远林 PB19020632

注：由于这篇论从零开始训练的模型的dataset大小为40g(数据库形式)，160w张猫的图片，解压缩后大小至少为120g(我解压缩了12w张占用了11g)，规模太大，所以不能完美复现，故而采用了部分复现，主体任务为了解网络结构和运行机理。

* **一 ：文章贡献和创新点**

 个人认为，文章的主要贡献有2个，其一是他们拓展并优化了GAN对抗生成网络的目标函数，从原始的

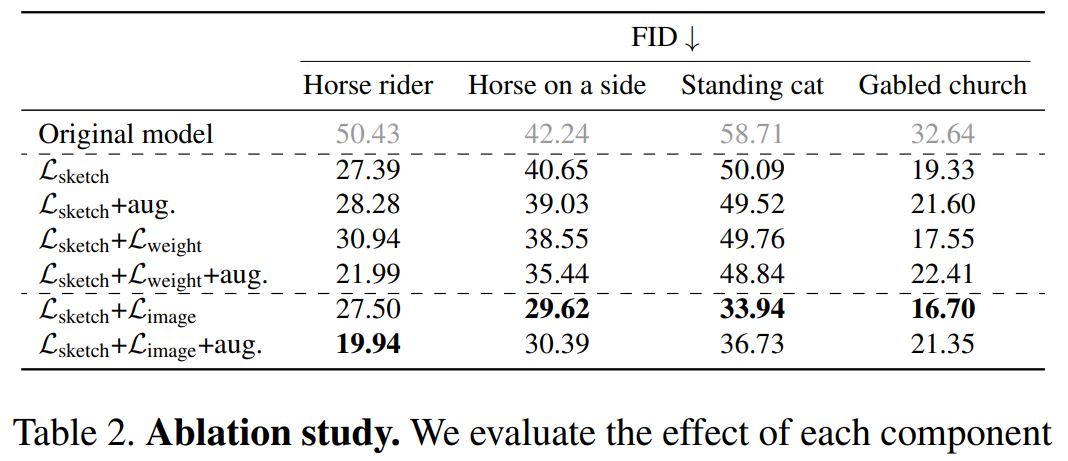
变为由下列三种函数线性或乘法组成的目标函数:



并在之后取得了较为良好的效果。

另一个贡献就是验证了数据增强策略(data augment)在实践中的应用效果。

具体效果如下图所示：

其中，FID(Fréchet Inception Distance) 是计算真实图像和生成图像的特征向量之间距离的一种度量。FID 分数常被用于评估由生成对抗网络（GAN）生成的图像的质量，较低的分数与较高质量的图像有很高的相关性。也就是分数越低越好。那么从上述图中可以看到，使用(3)和(1)相较于仅使用(1)并不会提高图像质量，但是同时使用数据增强和(1)and(3)会有比较明显的效果提升。综合最后两个结果来看，作者团队提供的方法是很有效果的。

个人认为，他们的创新点是使得用户只需要提供一个草图便可以描绘出很真实的图片，在艺术创作领域可能会卓有成效，同时可以提供研究GAN网络生成图像的新思路。

* **二 ： 代码结构描述**

整体project的代码结构可以分为4个部分:

[1]:数据预处理(prepare\_lsun.py&dataset.py)

[2]:具体模型训练部分，包括训练(gan\_model.py&gan\_trainer.py)和评价(evaluation.py)

[3]:结果可视化和展示，也就是util文件夹内的3个文件(html.py&util.py&visualizer.py)

[4]:训练好的模型对应的生成(generate.py)

下面将详细描述dataset.py和训练部分的文件内容以及架构。

* [1]:dataset.py

这个文件就是一个类:ImagePathDataset和几个函数

具体解释为载入数据部分(ImagePathDataset)和归一化数据部分(create\_dataloader),其中归一化是为了便于神经网络进行统一分析。

* [2]:gan\_model.py

这个文件定义了GANModel这个类，利用GPU进行运算，其中包含:

1.“向前”（也就是计算当前的loss和优化方向，利用梯度下降）

2.构造优化器

3.定义损失函数

4.定义梯度下降的方向

5.获得各种参数和资源(get系列)

6.各种处理和相应的结果输出(数值输出和可视化输出)

* [3]:gan\_trainer.py

这个文件定义了GANTrainer这个类，主要作用是粘合上面gan\_model.py定义的各种方法和函数，包括:

1.运行各种方法by one step

2.加载和输出各种参数和数据

也就是封装了一下GANMODEL

* **三：训练过程描述与结果分析和猜测**

首先，在安装完全部的依赖和相应的cudnn以及cuda之后，根据论文描述进行第一步的generate。

# generates samples from the "standing cat" model.

python generate.py --ckpt weights/photosketch\_standing\_cat\_noaug.pth --save\_dir output/samples\_standing\_cat

# generates samples from the cat face model in Figure. 1 of the paper.

python generate.py --ckpt weights/by\_author\_cat\_aug.pth --save\_dir output/samples\_teaser\_cat

# generates samples from the customized ffhq model.

python generate.py --ckpt weights/by\_author\_face0\_aug.pth --save\_dir output/samples\_ffhq\_face0 --size 1024 --batch\_size 20

猫在石头上

描述已自动生成黄色的猫

描述已自动生成随后生成的图片保存到output当中，例如如下：

男人看着前面

描述已自动生成戴眼镜的女人在微笑

描述已自动生成猫在雪地上

描述已自动生成灰色的猫

描述已自动生成

此时可以看到，效果还是不错的，但观察第四张猫猫图片发现，生成的有些许怪异，我个人认为是由于使用的dataset只有12w张图片的缘故。

随后，复现论文提供的预训练好的模型，具体保存在

ThirdExp\相关文件的副本.原始运行的模型repo不在这里\pretrained\_cat

这个文件夹内，在web\index中，可以看到效果较为良好，但是仍旧存在部分猫猫图有些扭曲和异常，同样怀疑是由于dataset容量不是162w张图的缘故。

紧接着从零开始复现模型，(注：由于论文工作中他们至少训练了100w个epoch，可囿于我的显卡和电脑不足以支持这种长时间的训练和这种大规模的集群训练，所以我只训练了前4500个epoch作为参考)详情保存在

ThirdExp\相关文件的副本.原始运行的模型repo不在这里\no\_pretrained\_cat

这个文件夹内。通过查看web\index内容可以发现，随着epoch逐渐增加(500个epoch记录一次)，输出的图片复杂度明显升高，但是远达不到pretrained的模型的精度和真实度，这也证明了如果需要获得较为真实和精准的GAN网络需要大规模长时间的训练和校验。

以下是pretrained的2000个epoch总计21次loss的记录以及从零开始的模型的4500个epoch总计46次loss的记录：

Pretrained模型：图形用户界面

描述已自动生成

图形用户界面, 应用程序

描述已自动生成从零开始的模型：

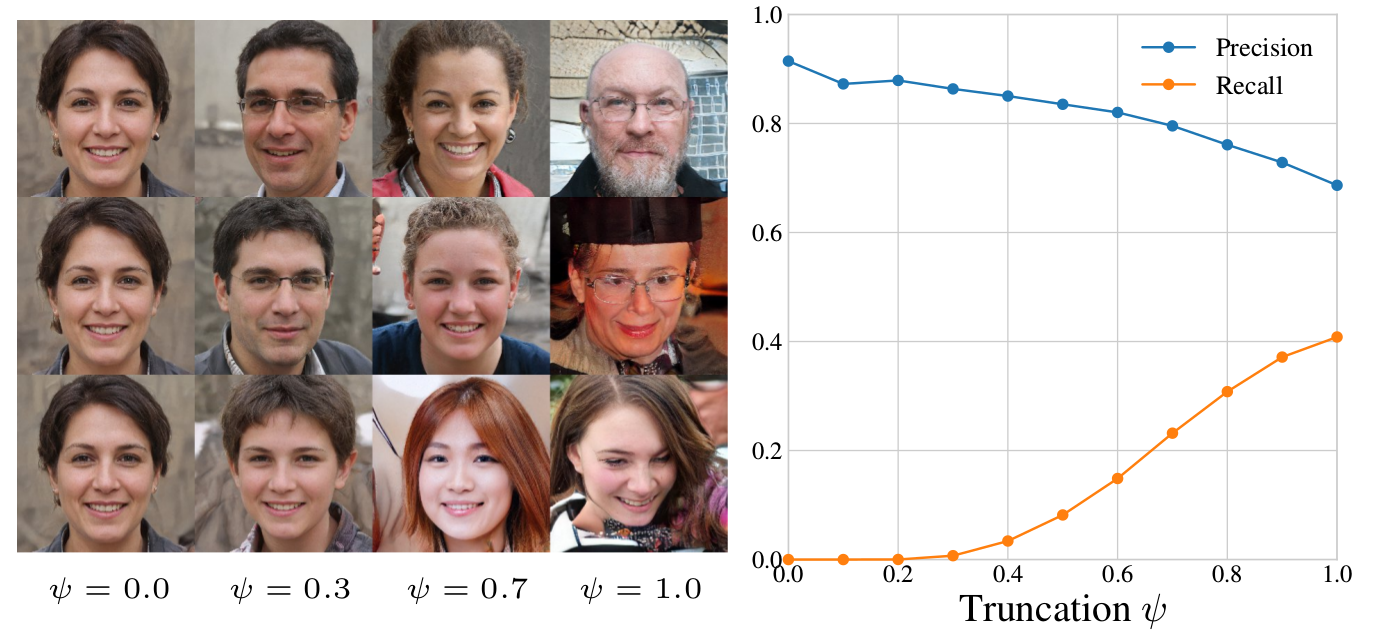
此时观察各种输出参数不难看到，pretrained的模型已经是一个平稳序列了，但是从零开始的模型明显存在趋势项，D\_fake\_image和D\_real\_image长时间为0，并且G\_sketch以及G\_image的数量级远大于pretrained的模型的数量级。

个人认为，出现这些差距的因素就是训练时间不足，迭代次数不足。毕竟对抗生成网络相当于左右手互搏，需要长时间的训练才能起到相应的效果。

**超参数的理解**：

在这篇论文中，最重要的超参数莫过于

parser.add\_argument("--l\_image", type=float, default=0, help="strength of image regularization loss")

在论文中，他的值是0.7，他的主要作用是对输入的image进行变换，强度与这个值正相关，在经过大量的查阅之后发现，这个参数的值会明显改变输入image的图像，并有如下结果:

可以看到，这个参数的提升会降低precision，提高召回率，体现在这篇文章当中就是会降低训练精度，但是会提高真实度。具体原因并未在论文中有详细叙述，但是从作者对于他们的目标函数的描述来看：



这个东西会影响目标函数中image这个函数的权重，也正好符合代码中的注释的描述。

* **四：复现过程中的感悟和吐槽**

不得不说，搞人工智能必须得进实验室，我光光是捯饬torch和tensorflow以及对应的cudnn和cuda环境就搞了3天，天天就研究这个版本那个版本，没有师哥师姐或者导师的指导真的很难熬，再者，就算熬过去了，要复现的模型可能也不适合这个版本的环境，还需要反复的研究和修改，又是一件折磨人的事情。搞好这个了，到了具体复现的时候，下载数据又是一个问题，学校的网络经不起这种大规模的下载，光光是下载猫猫的数据集我就下载了一个下午，断断续续的，最后还是花的热点。而且这还没完，等全都做完了预备工作，真正开始跑模型炼丹的时候，显卡和硬件桎梏又来了，他们实验室的工作环境可能是好几块3080Ti，还是桌面版的，我的就只有2080，还是笔记本的，算力差的不是一个量级，搞得我只能复现几千个epoch，想复现更多电脑不允许了。（搞人工智能还是个花钱的东西）最关键的是，人家的程序不一定适合你，许多参数你都看不懂，你得去查你得去研究才能勉强知道什么意思，这东西需要的理论知识是成体系成框架的，远不是一个课一个学期能搞明白的（所以为什么要复现模型，教一点理论知识让我们自己写一个简单的模型不是更好？）

如果说感悟的话，那我必须得说：学习真的就是自学，没有人教你，也没有人有义务教你，你需要合理利用各种搜索引擎和渠道获取知识理解知识，并且对于人工智能大数据这类的前言学科，你需要保持一颗学徒的心，并且有足够的耐心应对各种糟心的事情（包括不仅限于环境搭建，显存爆炸，存储容量不足，下载速度过慢，数组溢出，内存溢出等等）。

**接下来的详细内容详见视频**