

数据集下载

有一些数据集会被放在Google Drive上，国内网络下载非常容易失败（例如FFHQ）

一个解决办法如下: (需要科学上网)

1.先把别人的分享链接保存到自己的Google Drive上，接下来进入Colab

2.在colab中挂载自己的云盘，利用

```
!cp -r 分享的数据集 ./
```

将数据集拷贝到当前工作空间中

3.创建一个新的文件夹，利用一段代码将分享的数据集拷贝到新创建的文件夹中

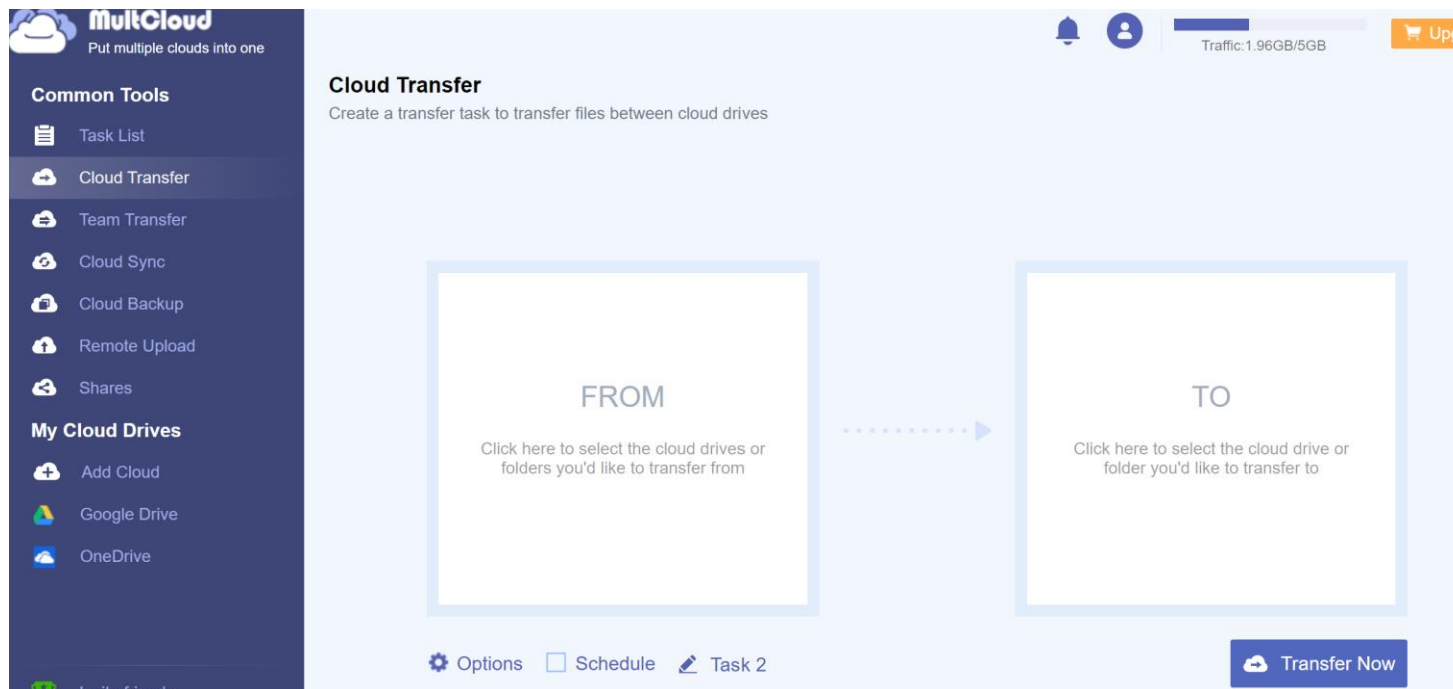
```
import os

def copy_dir(src_path, target_path):
    if os.path.isdir(src_path) and os.path.isdir(target_path):
        filelist_src = os.listdir(src_path)
        for file in filelist_src:
            path = os.path.join(os.path.abspath(src_path), file)
            if os.path.isdir(path):
                path1 = os.path.join(os.path.abspath(target_path), file)
                if not os.path.exists(path1):
                    os.mkdir(path1)
                copy_dir(path, path1)
            else:
                with open(path, 'rb') as read_stream:
                    contents = read_stream.read()
                path1 = os.path.join(target_path, file)
                with open(path1, 'wb') as write_stream:
                    write_stream.write(contents)
        return True

    else:
        return False
```

```
copy_dir('分享的数据集路径', '新创建的文件夹路径')
```

4.登录multcloud.com，注册一个账号，然后将Google drive和OneDrive或百度网盘添加到账户中，然后进行文件互传，将Google drive中的数据集传到这两个云盘中去。（速度还是很慢，1.5g传了12个小时。。。好歹可以下载）



5.传输完成后进行下载
百度网盘速度慢，操作简单
OneDrive速度快，需要设置从云端和本机同步。

代码编写

当训练代码编写完成后进行训练的过程中需要多次调整超参数，如果利用修改代码的方式效率很低。可以使用命令行传参的形式快速调整超参数。

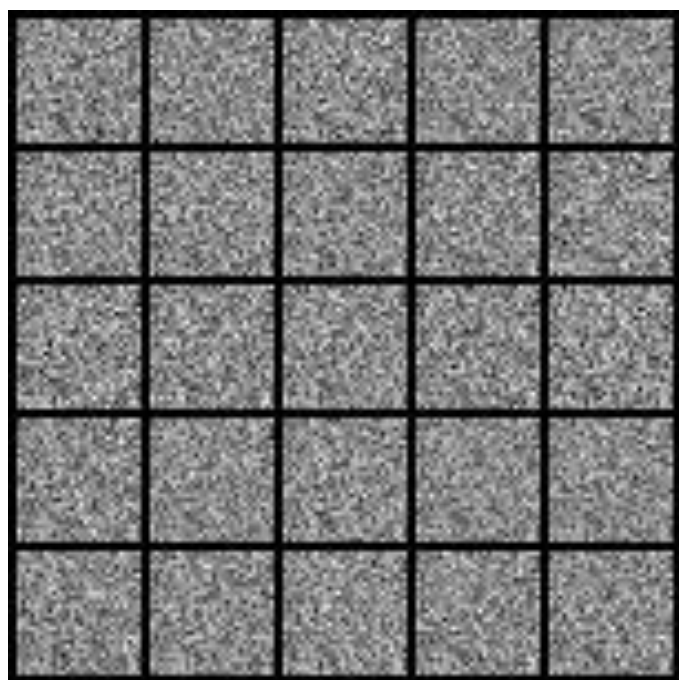
```
parser = argparse.ArgumentParser()
parser.add_argument("--n_epochs", type=int, default=100, help="number of epochs of training")
parser.add_argument("--batch_size", type=int, default=64, help="size of the batches")
parser.add_argument("--lr", type=float, default=0.0002, help="adam: learning rate")
parser.add_argument("--b1", type=float, default=0.5, help="adam: decay of first order momentum of gradient")
parser.add_argument("--b2", type=float, default=0.999, help="adam: decay of first order momentum of gradient")
parser.add_argument("--n_cpu", type=int, default=15, help="number of cpu threads to use during batch generation")
parser.add_argument("--latent_dim", type=int, default=100, help="dimensionality of the latent space")
parser.add_argument("--img_size", type=int, default=128, help="size of each image dimension")
parser.add_argument("--channels", type=int, default=3, help="number of image channels")
parser.add_argument("--sample_interval", type=int, default=100, help="interval between image samples")
opt = parser.parse_args()
```

然后在终端执行代码时利用如下的格式就可以用想用的超参数进行训练了（--后跟参数名称，空格然后跟参数值）

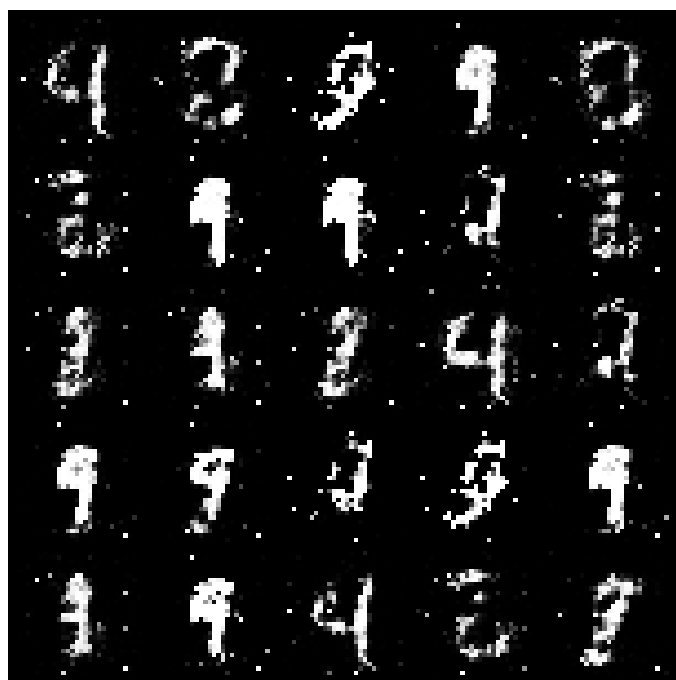
```
(py_37) root@interactive56174:/opt/data/private/hyk-gans/stylegan2-pytorch-master/stylegan2_pytorch# stylegan2_pytorch --data /opt/data/private/hyk-ga
g-prob 0.25 --aug-types [translation,cutout,color] --batch-size 16 --calculate-fid-every 500 --num-train-steps 50000 --log
```

训练结果

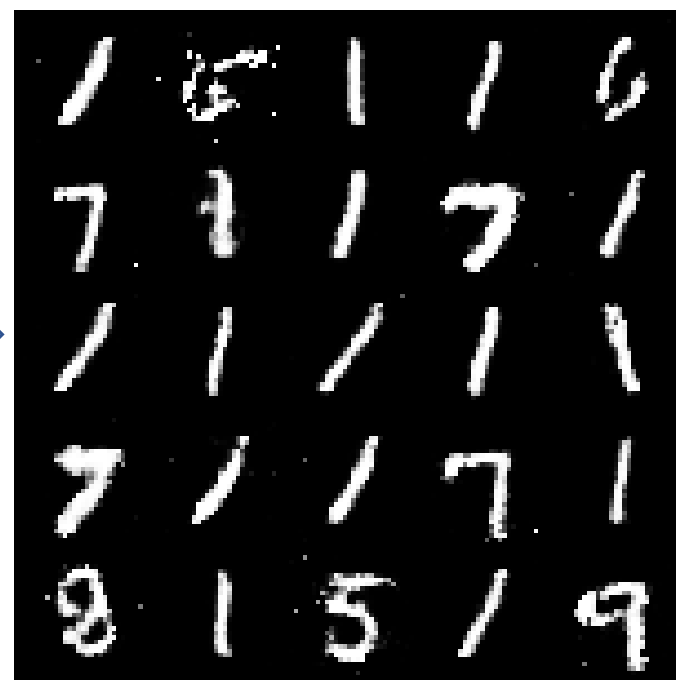
首先是最基础的GAN 训练MNIST数据集



0



29200



184800

然后是128*128的FFHQ数据集，由于70000张图片同时训练训练时间太长，就挑了10000张来进行训练，100个EPOCH，效果很差，推测是迭代次数太少的原因，时间关系没有继续跑



50



25000



47100

接下来是StyleGan2的结果
还是10000张128*128的FFHQ， 跑50000个iteration

