实验 5-生成式对抗网络实现

学号：1190201215
姓名：冯开来

一、 实验目的：

生成式对抗网络实现
隐空间语义方向搜索

二、 实验环境：

详细可见文件 requirements.txt
Python==3.9; Pytorch==1.11

三、 实验内容：

1.超参数定义

```
parser = argparse.ArgumentParser("lab4")
parser.add_argument("--device", default="cuda:0")
parser.add_argument("--model", default="WGAN-GP")  # GAN WGAN WGAN-GP
parser.add_argument("--epochs", default=1000)
parser.add_argument("--batch-size", default=2000)
parser.add_argument("--seed", default=42)
parser.add_argument("--dataset", default="points")  # points
parser.add_argument("--output-path", default="./result/")
parser.add_argument("--hidden-size", default=128)
parser.add_argument("--input-size", default=128)
parser.add_argument("--CLAMP", default=0.1)
parser.add_argument("--optimizer", default="rmsprop")  # adam sgd rmsprop
parser.add_argument("--draw", default=False, help="draw the loss and process")
```

这里默认由 GPU 进行加速训练。
模型支持 GAN、WGAN、WGAN-GP。
优化器支持 RMSprop、Adam、SGD。
路径 output-path 包含了不同模型、不同优化器下的拟合分布图，还保存了不同类别的最佳模型。

2.数据集构造

本次实验构造数据集相比上一次简单了很多。
直接从文件中读入，转换为 array 后转换为 tensor 后就完成了。通过 DataLoader 和 batch-size 构造出训练集和验证集，还是比较容易的。

3.搭建网络结构

本次实验需要实现三个网络结构，相比 lab4 任务量少了很多。
本质上三个网络结构是一样的，只有在训练的过程中计算损失的方法不一样。对抗生成网络主要有两个部分，一个是生成器，一个是判别器，生成器不断进化让自己生成的图片更加逼真，判别器不断通过甄别图片使自己的判别能力提升。在生成器中，本次实验是要拟合分布，所以生成的

就是一个个点，是一个二维的，所以最后的 output 是通过全连接层输出一个二维的张量。对于判别器而言，最后生成的应该是一个一维的概率值，并且还需要通过 sigmoid 函数。具体代码如下图：

```python
class Generator(nn.Module):
    def __init__(self):
        super(Generator, self).__init__()
        self.net = nn.Sequential(
            # z: [b, 2] => [b, 2]
            nn.Linear(10, 128),
            nn.ReLU(True),
            nn.Linear(128, 256),
            nn.ReLU(True),
            nn.Linear(256, 512),
            nn.ReLU(True),
            nn.Linear(512, 2),
        )

    def forward(self, z):
        output = self.net(z)
        return output
```

生成器网络结构

```python
class Discriminator(nn.Module):
    def __init__(self):
        super(Discriminator, self).__init__()
        self.net = nn.Sequential(
            # [b, 2] => [b, 1]
            nn.Linear(2, 128),
            nn.LeakyReLU(),
            nn.Linear(128, 256),
            nn.LeakyReLU(),
            nn.Linear(256, 128),
            nn.LeakyReLU(),
            nn.Linear(128, 1),
            nn.Sigmoid()
        )

    def forward(self, x):
        output = self.net(x).view(-1)
        return output
```

判别器网络结构

4. 优化器和损失函数

本次实验选用的优化器可以使用 Adam 和 Sgd。

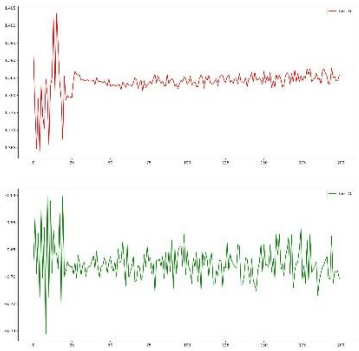
本次实验所用的损失函数并不是真正意义上的损失函数，因为生成模型希望生成的概率期望越大越好，先当于其负数越小越好，也就可以理解为其负数就是损失。

5. 训练过程、测试过程及实验结果

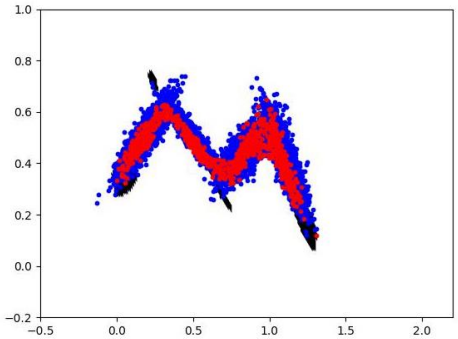
本次训练过程区别三个网络的重点在于判别器的改造。在 GAN 网络中，给定一个样本是属于真实还是虚假的，判别器的目标函数为最小化交叉熵。具体公式课件和网络上都有。而对于 WGAN 网络来说，其用 Wasserstein 距离替代了 JS 散度，需要修改的地方就是满足 K-Lipschitz 连续，近似的方法就是限制参数的范围，使关于 x 的偏导数的模都小于某个上界。对于 WGAN-GP 来说，它的不同之处在于对于损失函数加了一个 gradient_penalty 函数，具体的计算公式也都有现成的，这里不推导了。

下面是不同网络的 loss 曲线以及拟合效果图片：

GAN 网络 + RMSprop：

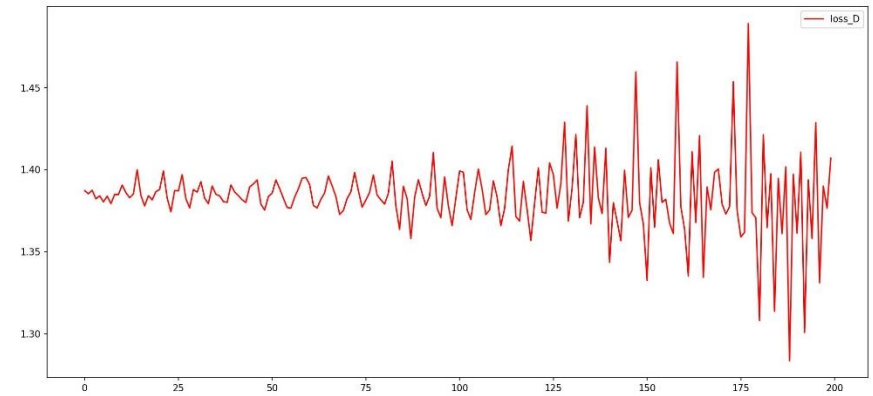


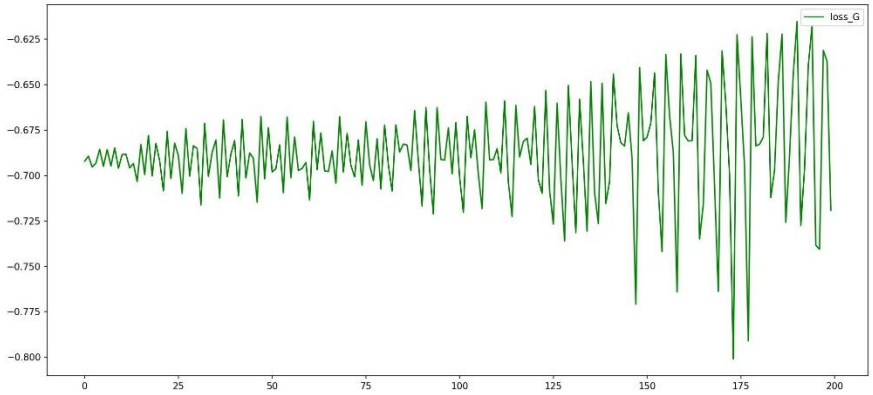
GAN + RMSprop D 和 G 的损失曲线



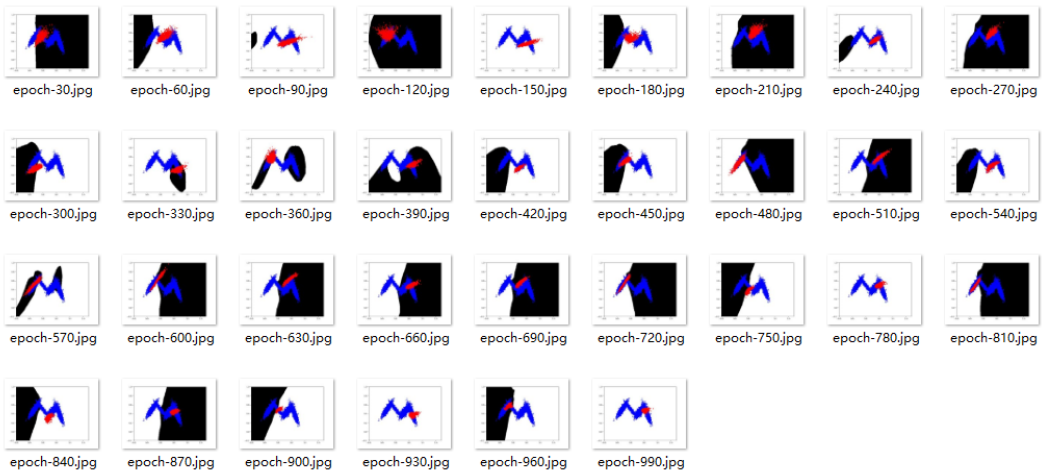
epoch==990 时拟合效果

WGAN + RMSprop：





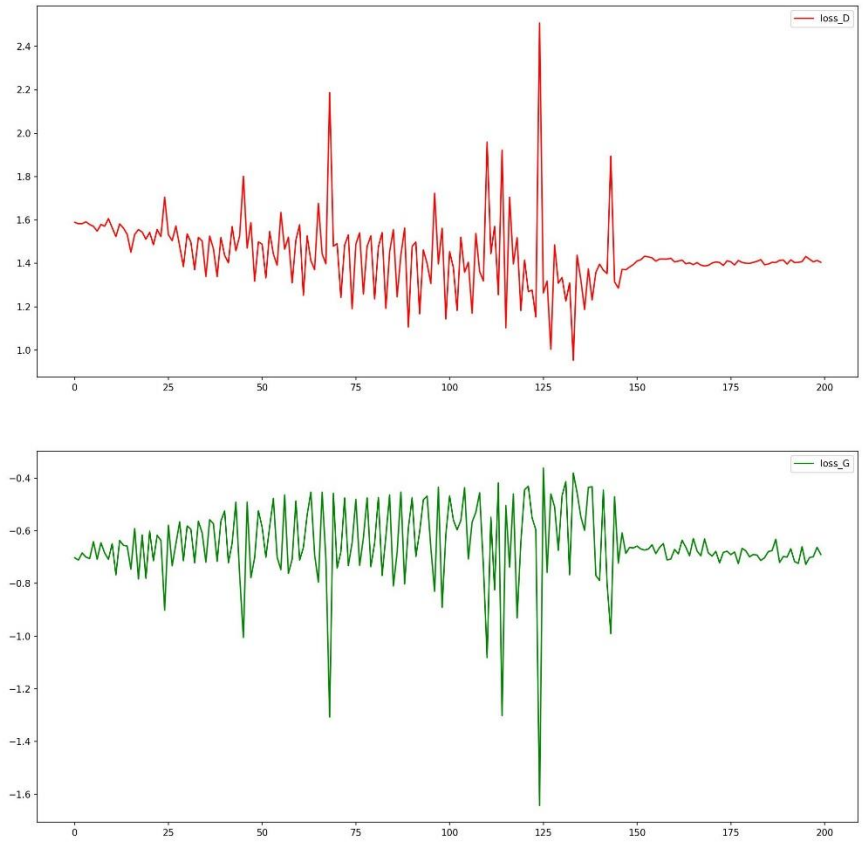
WGAN loss 曲线



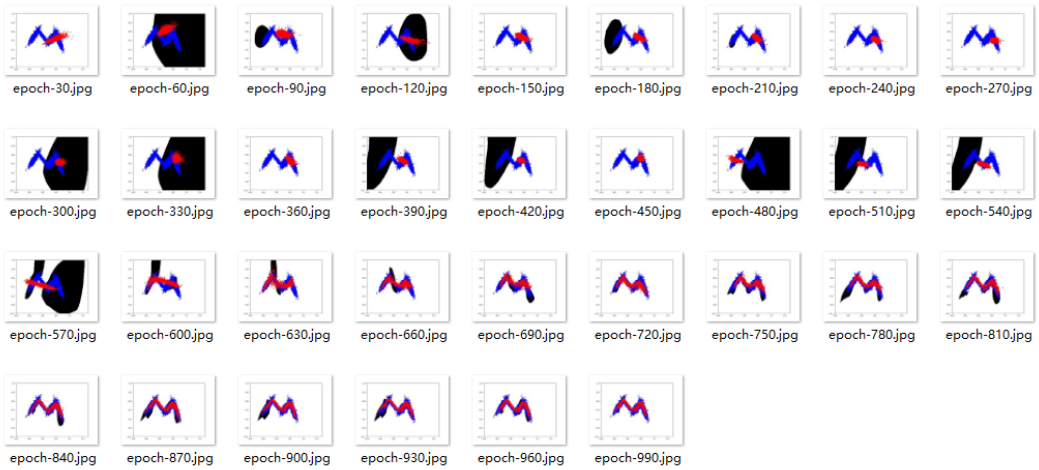
WGAN 拟合过程

（黑色部分是该点通过判别器概率高于 0.5 的点，反正则为白色）

WGAN-GP + RMSprop：



WGAN-GP loss 曲线

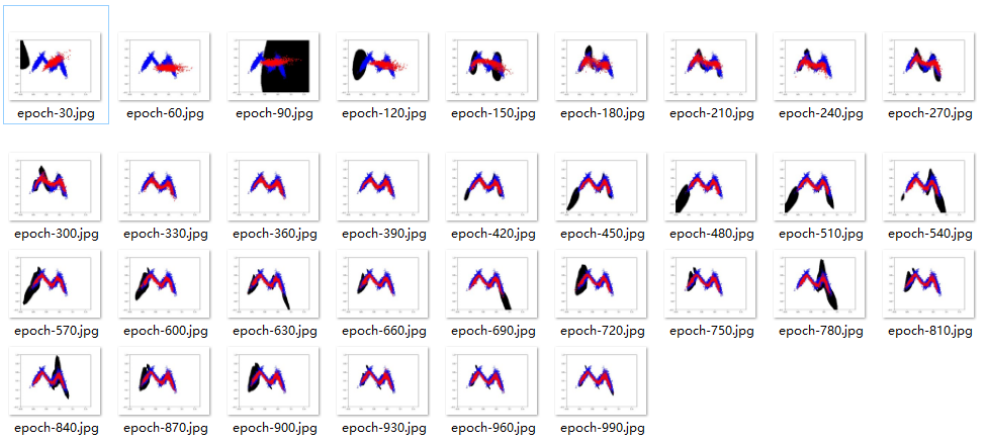


WGAN-GP 拟合过程

（黑色部分是该点通过判别器概率高于 0.5 的点，反正则为白色）

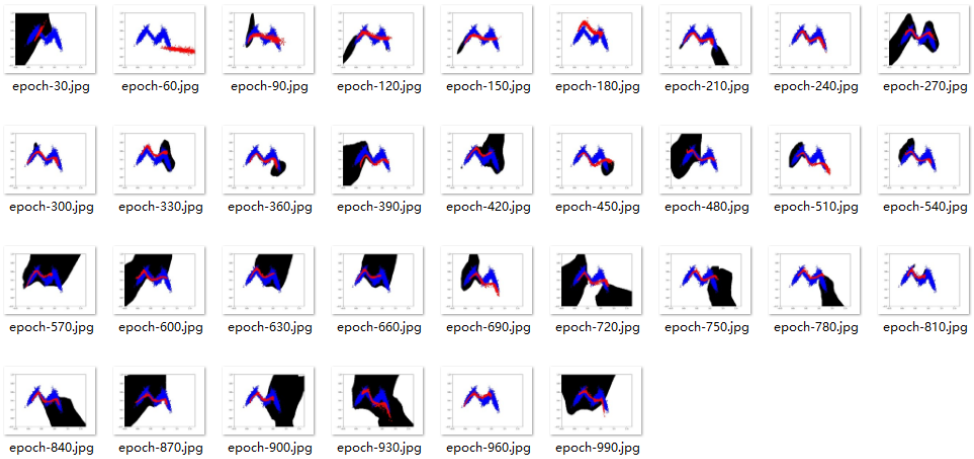
下面是对比不同优化器的拟合效果：

GAN + RMSprop：



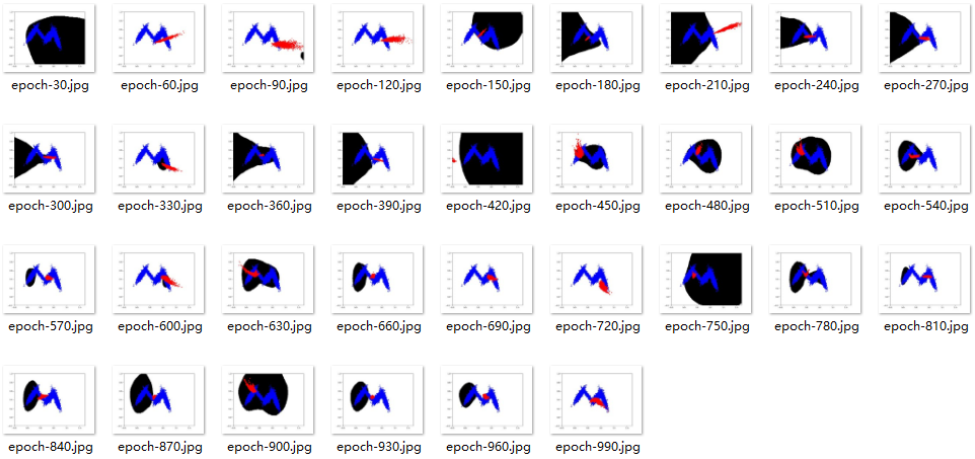
GAN + RMSprop 拟合效果

GAN + Adam：



GAN + Adam 拟合效果

GAN + SGD：



GAN + SGD 拟合效果

总结：

对比上述实验，我们不难发现，在优化器选择的对比中，选用 RMSprop 的效果最好，在 300epoch 的时候就已经拟合的非常好了，而 Adam 的话在 500 多 epoch 的时候开始收敛，相对而言慢了一点。SGD 可能是参数没有调对，在 1000epoch 中始终没有收敛。

对比不同模型来说，GAN 的性能其实已经非常不错了，在 300epoch 时已经收敛，但是在 500epoch 之后，观察黑色区域（黑色为概率高于 0.5 的点集合），却出现了过拟合的现象。WGAN 使用 weight clipping 限制了网络参数后，可能导致了梯度消失或者梯度爆炸，在本次实验中，不能收敛，效果很差。WGAN-GP 拟合的相对慢一点，在 1000 个 epoch 中并没有出现像 GAN 一样的过拟合现象。

隐空间语义方向搜索：

GAN 中的生成器通常以随机采样的潜在向量 z 作为输入，生成高保真图像。通过改变潜在向量 z，我们可以改变输出图像,例如头发颜色、面部表情、姿势、性别等，我们需要知道移动潜在向量 z 是什么。

对于 GAN 的每一层，都学习从一个空间到另一个空间的转换。对于第一层变换，有 G(z)=Az+b, 而给定 z 一个方向 n 和步长 α，G(z+αn)=G(z)+αAn，由此如果给定一个潜在码 z 和方向向量 n，则可以通过在变换后的投影码上加上 αAn 来实现对图像编辑的过程。这里可以看出 A 中包含图像变换的基本信息，所以我们只需要对 A 进行分解就可能会发现特定图像变换的方向信息。

根据论文中的方法，我们求出 A^TA 的特征向量，要找出 k 个使图像进行变换的信息，即选择特征值最大的 k 个特征向量作为方向。随后给定一定的步长，将其加到 Generator 的输入向量 z 中进行训练，最终就可以得到特定语义对于图像的变换。

最终根据生成的图像和视频，每一列对应的语义为性别，性别，视角的变化，头发和年龄。

百忙之中写完报告，

感谢助教辛苦审阅！