凸优化实验

龙振威 2022214373

1. **模型**

图像模糊可能由多种因素造成，例如图像过程中的相机或目标物的移动、光学器件的视角和大气湍流的影响等。可以将图像退化建模为一个退化算子，这个算子与一个加性噪声项共同输入图像进行运算，对生成一幅退化的图像，即：

在给定退化图像和退化算子(已知为大小的高斯核)，想要重建原始图像，使用如下损失函数：

其中是待优化的估计原始图像，第二项中为拉普拉斯算子，使用的卷积核如下：

由于使用Pytorch实现，没有使用cvx工具包，因此我没有显式地加入约束项，而是在代码实现过程中隐式地加入正则化约束。

1. **实现**
2. **退化算子的实现**

通过继承torch.nn.Module类构建前向卷积模型。首先将G.mat文件通过scio.loadmat函数导入并转化为Pytorch支持的张量类型，记作。值得注意的是Pytorch中tensor各通道的顺序为：，分别代表批大小(Batch)、通道数(Channel)、高度(Height)、宽度(Width)。

为了使得G适配Pytorch的卷积运算，需要重新组装卷积核，步骤如下：

1. 初始化一个的张量；
2. 分别为该张量的Batch=0的通道赋值，Batch=1的通道赋值，Batch=1的，其中表示数值全为0、大小的矩阵；
3. 这样操作实现的是：对于输入RGB图像，G的每一个Batch通道分别处理R、G、B通道的卷积，再拼接起来。
4. **估计原始图像的初始化**

原始图像X的初始化对于算法的训练至关重要，在实现过程中发现使用torch.randn()初始生成大小为的张量与使用torch.rand()生成，两者得到结果天差地别。

对于torch.randn()函数，从标准正态分布中抽取随机数，服从均值为0、方差为1，但经过训练后发现矩阵中很多数据超出了合理区间，因此显示后表现出异常严重的椒盐噪声(即区间之外的值被约束到0和255两个边界值)。

对于torch.rand()函数，从区间的均匀分布中抽取一组随机数，符合图像数据的分布区间。因此，一个较好的训练初试点可以产生更好的(甚至超乎寻常的)结果。

初始化后，需要使用torch.nn.Parameter将加入模型的参数集，并将优化标志require\_grad设为Ture，这样梯度反向传播才能被优化。

1. **模型前向计算**

在损失函数计算时，其卷积核的构建过程与退化核G类似，权重系数为0.5。

使用torch.optim.AdamW优化器，初试学习率为0.035。

使用torch.optim.lr\_scheduler.ReduceOnPlateau学习率调整器，衰减率(factor)为0.5，耐力(patience)为5。

每个10个epoch，学习率衰减为原来的0.8。

设置最大epoch为300；采取早停策略，当损失在10个epoch之内没有减小，则停止训练。

1. **梯度反向传播**

利用Pytorch的自动微分可以比较方便地实现梯度反传与变量优化。

1. **预处理**

对于figure2，它具有显著的盐噪声，使用自适应中值滤波器对其进行预处理。figure1和figure3预处理不对。

1. **结果与性能比较**

**表：SSIM值(Matlab版)**

|  |  |
| --- | --- |
| **Figure1** | **0.3348** |
| **Figure2** | **0.1527** |
| **Figure3** | **0.2034** |

****

****

****