## 生成对抗网络实验报告

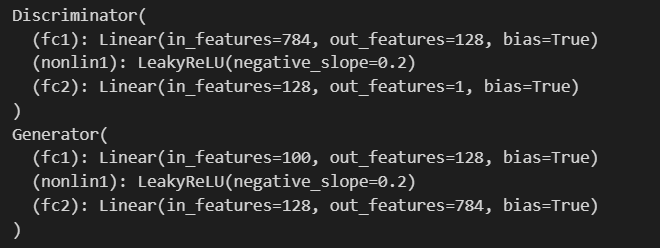
姓名：管昀玫 学号：2013750

### 实验要求

* 掌握GAN原理
* 学会使用PyTorch搭建GAN网络来训练FashionMNIST数据集

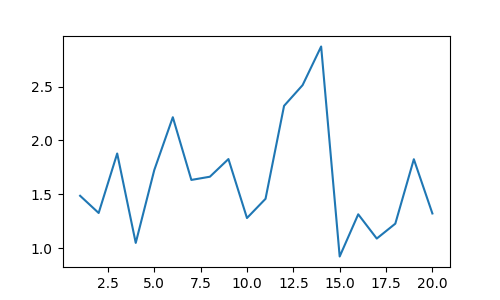
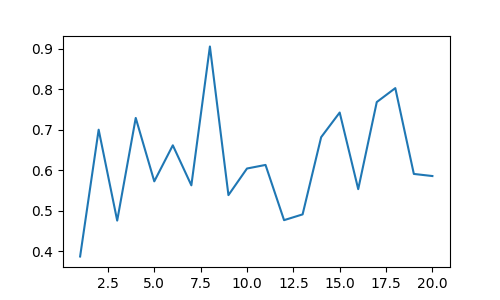
### GAN

老师提供的原始版本的GAN网络结构如下所示：



训练的loss曲线如下图所示：

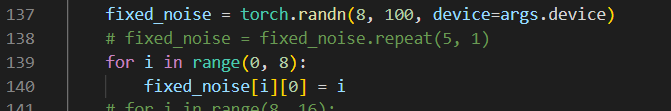
左图为生成器的loss曲线，右图为判别器的loss曲线。

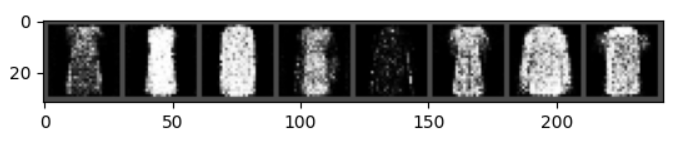


由于GAN是极大极小博弈，generator和discriminator在不断对抗中迭代升级，loss函数会呈现出不可预测的趋势，如图中所示，虽然二者的loss曲线有波动，但波动不大。GAN的loss曲线有以下规律：

1. 生成器和判别器从一开始都是非常弱的，因此一般不会在训练一开始两者损失就非常剧烈的波动。在训练一段时间达到稳定期后，生成器和判别器的损失都应该在一个小区间内波动，而不会有明显的持续上升/下降趋势。
2. 如果生成器损失持续明显上升，表明其无法学习怎么欺骗判别器，体现在结果上就是开始生成噪声。
3. 如果判别器损失持续明显上升，表示其无法学习怎么识别生成器，体现在结果上就是生成器可能会生成一致的，无意义的但是能欺骗判别器的图像。

自定义一组随机数，生成8张图



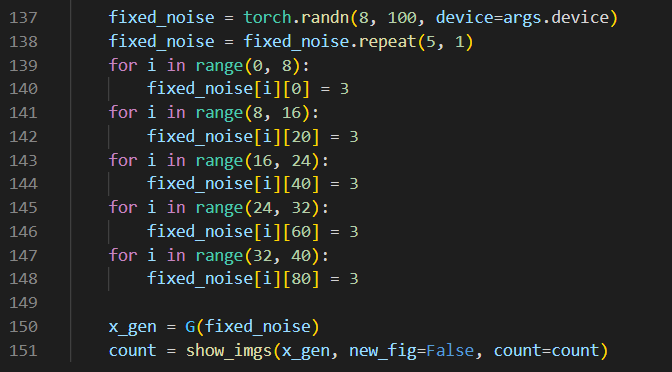


针对自定义的100个随机数，自由挑选5个随机数，查看调整每个随机数时，生成图像的变化（每个随机数调整3次，共生成15x8张图），总结调整每个随机数时，生成图像发生的变化

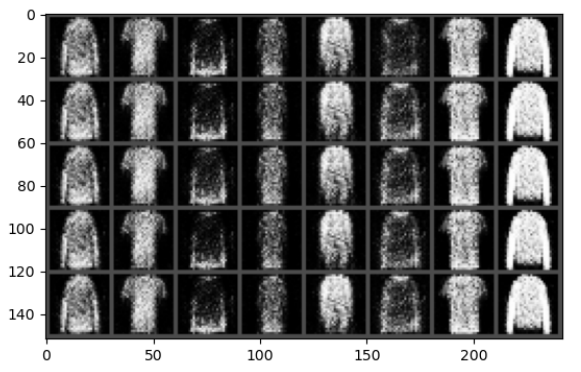
操作过程如下所示：

1. 首先，使用 torch.randn(8, 100, device=args.device) 生成了八个具有100个维度的随机数。这些随机数将被用作生成图像的输入。
2. 然后，使用 fixed\_noise = fixed\_noise.repeat(5, 1) 将这八个随机数复制五次，以便在每个五个位置上进行更改。这样做是为了确保每一行中的八个随机数图像，在每次实验中更改的位置是相同的。
3. 对于每张图像，以一行为单位，依次更改其中的八个随机数。第一行代表更改第一个位置的随机数，第二行代表更改第20个位置的随机数，依此类推。最后一行代表更改第80个位置的随机数。

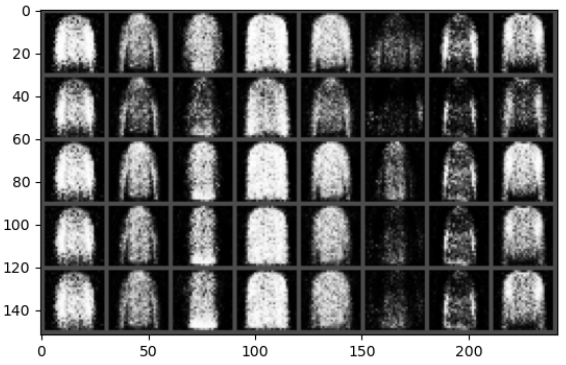
代码如下所示：



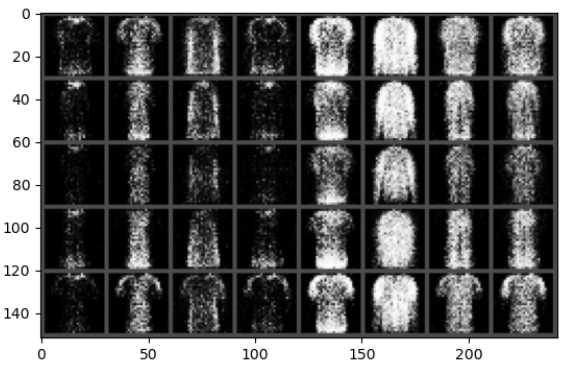
将随机数固定为0.5，生成图像如下图所示：



将随机数固定为3，生成图像如下图所示：



将随机数固定为10，生成图像如下图所示：



首先，观察结果表明，当使用较小的随机数值（如0.3）并在不同位置进行微小的变化时，对生成的图像几乎没有明显的可察觉的影响。换句话说，这种变化对图像的效果产生了微弱的影响，甚至可能无法被肉眼观察到。

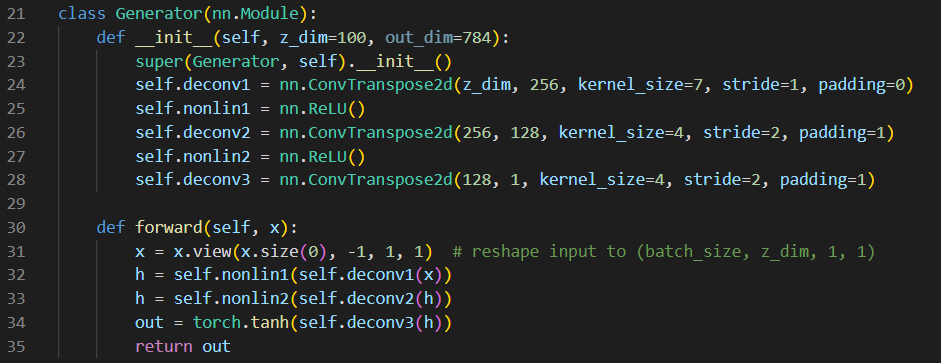
然而，当随机数值设定较大（如10）时，即使在不同位置进行微小的变化，对生成的图像的影响仍然相对明显。可以清晰地观察到，在最后一张图像中，位置60的变化将图像从裤子变为了两件衣服。这说明较大的随机数值对图像的变化产生了显著的影响。

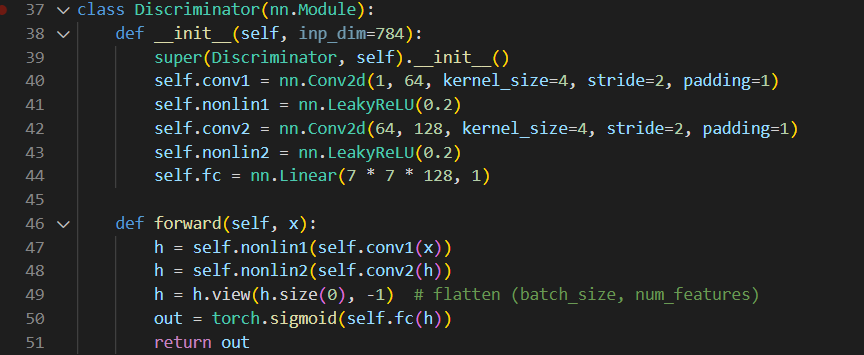
此外，当我们观察相同位置的随机数变化时，我们发现过小的随机数值（0.3）或过大的随机数值（10）都会导致生成的图像亮度较低，甚至某些图像接近全黑。然而，当我们使用适中的随机数值时，生成的图像相对较亮且质量较好。

综上所述，随机数值的选择对生成图像的质量和可观察性产生了重要影响。适当选择合适的随机数值可以改善生成图像的效果，并确保生成的图像具有良好的可视化特征。

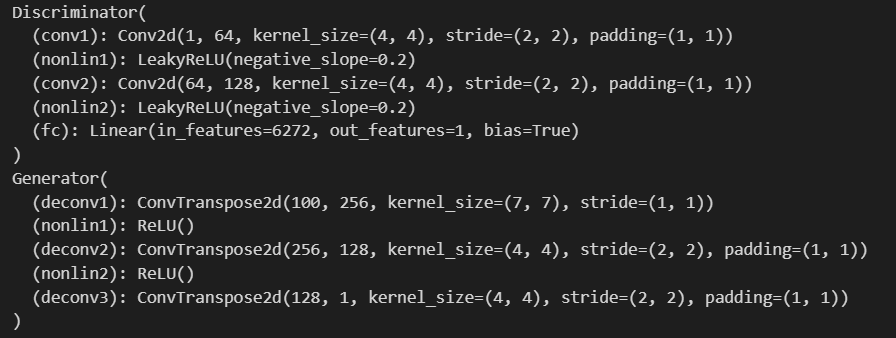
### 用卷积实现生成器和判别器

我实现的网络代码如下所示：





该网络结构打印如下所示：



该网络训练的loss曲线如下图所示：

左图为生成器的loss曲线，右图为判别器的loss曲线。

