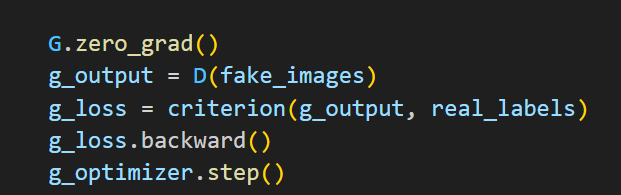
首先是判别器的训练，它将真实的图片打上标签为1，假图片打标为0，随后将所有真实图片通过判别器并记录真实图片部分的损失。随后将由随机噪声生成的假图片通过判别器并计算出假图片的损失，最后将两者加和并进行梯度回传和通过优化器。注意这里在计算假图片的损失时用了detach()，目的是防止在计算判别器对生成图像的损失时，梯度流回到生成器，这样可以防止对判别器的训练影响生成器，这两者理应独立。

生成器也是类似，但是这里需要判别器的反馈来改进生成器。生成器将判别器的反馈与真实标签作比较，以希望生成更像真实图片的图片。训练生成器时不适用detach，因为训练生成器正是需要判别器的反馈。随后进行损失函数的计算、梯度回传和优化器的优化。

在每个epoch的训练中，观察到判别器的损失相对比较低，但是随着epoch的增加，判别器的损失没有降低的趋势，而是在一个较低的值附近波动。这表明判别器能够较好地区分真实图像和生成图像，但并未完全压制生成器。而生成器的损失值始终较高，并且在训练过程中有比较大的波动，表明生成器在尝试欺骗判别器，但判别器仍然能够有效地识别生成图像的缺陷。总体来看，两者的损失函数保持（小范围的）波动和（总体上）的稳定，充分体现了对抗网络中的“对抗”。但是在每个epoch之后生成的图像中，还是可以明显地看出生成的图像越来越像真实图片（即人工手写），在第一个epoch，生成的图像还是和噪声没有什么区别，但是在最后一个epoch已经可以辨认出大部分数字，且痕迹也很像人手写的。中间的epoch也是可以体现这种连续的变化过程。这向我们揭示，生成器和判别器在不断的博弈和对抗中，整体上都有了很大的进步。

Q：生成器和鉴别器的Loss是不是越低越好？为什么

A：并非如此。如果判别器损失过低，意味着它完美区分真假图像；如果生成器损失过低，意味着它能完全欺骗判别器。这两者在目的上是矛盾的。理想状态下，生成器生成的图片和真实图片完全一致，则判别器的准确率应接近50%，生成器也一定无法得到判别器“全部为真“的判断。

Q：在本代码中，生成的数字是无法控制的（即采样一个噪声，把噪声输入到生成器中，得到的数字无法预测是什么），简单描述一下（20个字左右）如何才能生成可控的数字（即想生成哪个数字就生成哪个数字）？

A：使用条件GAN，在输入噪声时加入数字标签作为条件，将它们一起输入。