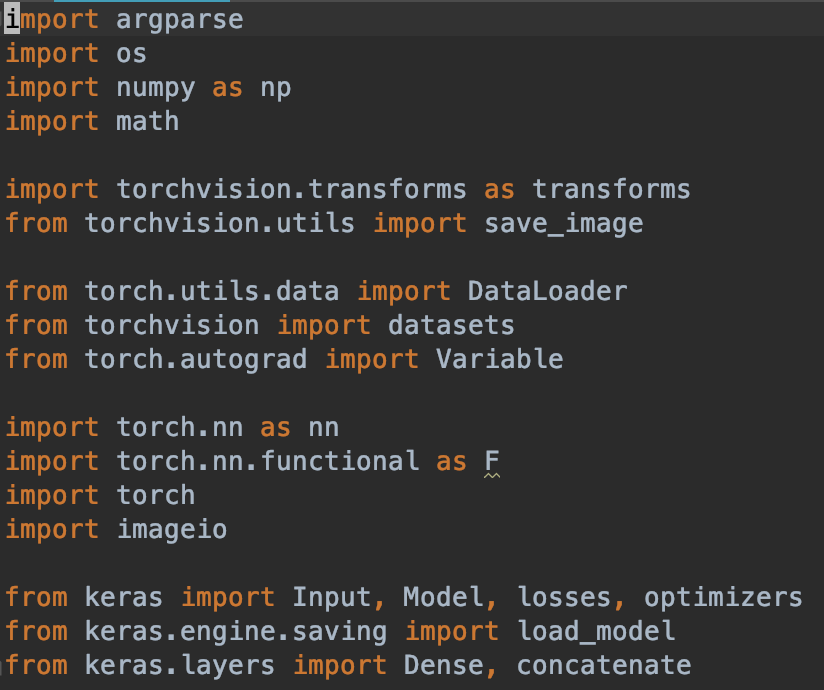
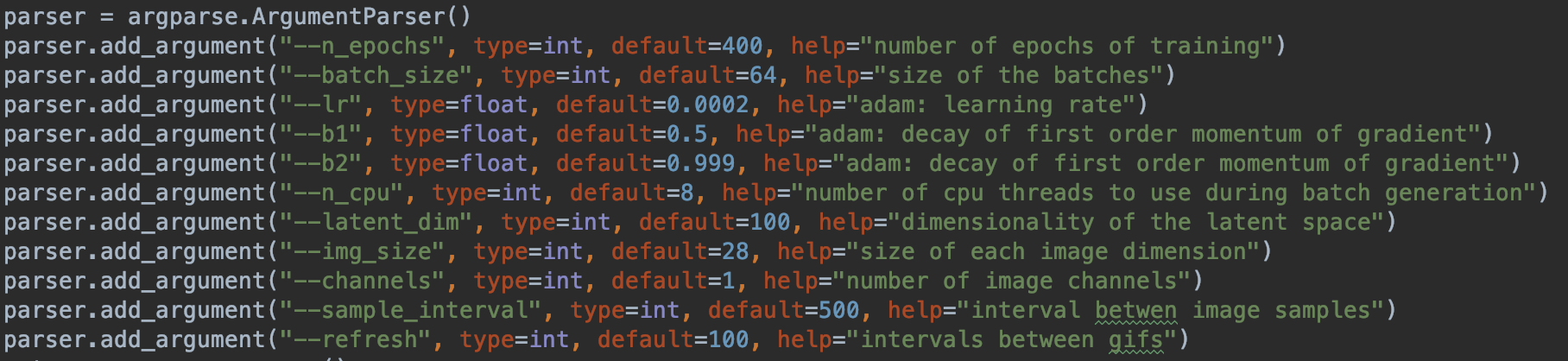
Note that the codes for GAN, WGAN, and DRAGAN are all very similar



Import the necessary libraries.



We declared all our variables using parser, which also allows the help starting, which makes it easier to understand what each individual variable is about.

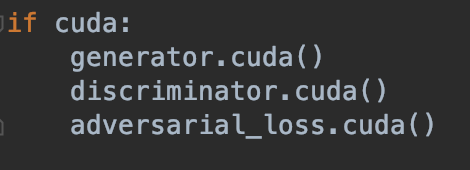


Img\_shape:

For minst, channel = 1, img\_size = 28

For cifar 100/10, channel = 3, size = 32





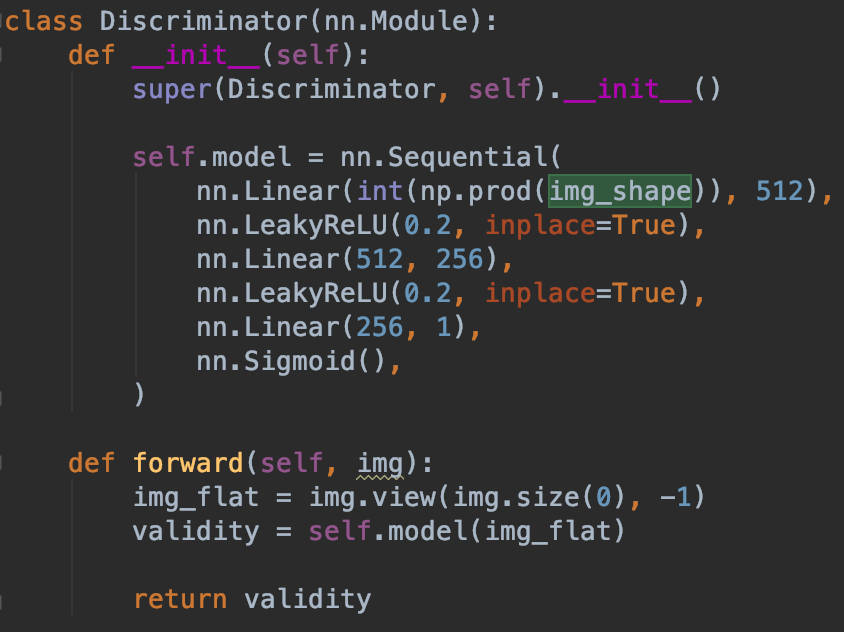
Uses GPU computation to speed up training time if available.

Generator Class

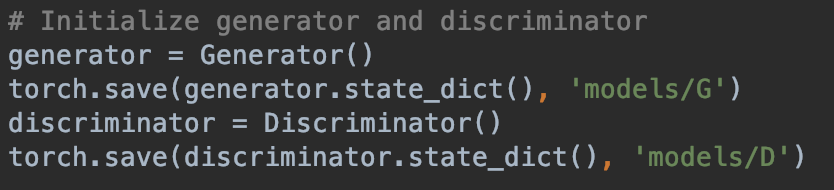


Generator made of Tanh activation function, 100\*128, 128\*256, 256\*512, 512\*1024, and 1024\*(image shape)

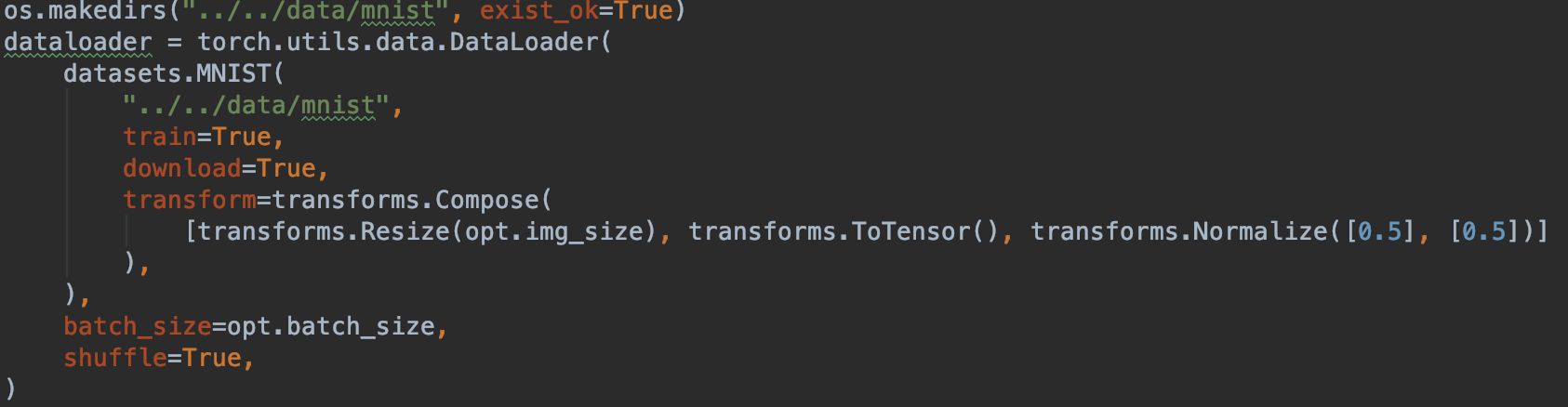
\*Image shape = (channels, img-size, img-size). For MNIST this shape is (1, 28, 28) For cifar it’s (3, 32, 32).



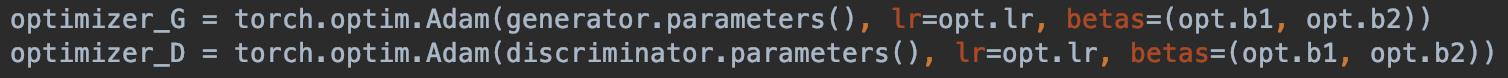
Discriminator, a normal neural network made of (image shape)\*512, leakyRelu, 512\*2356, 256\*1 and a sigmoid activation function.



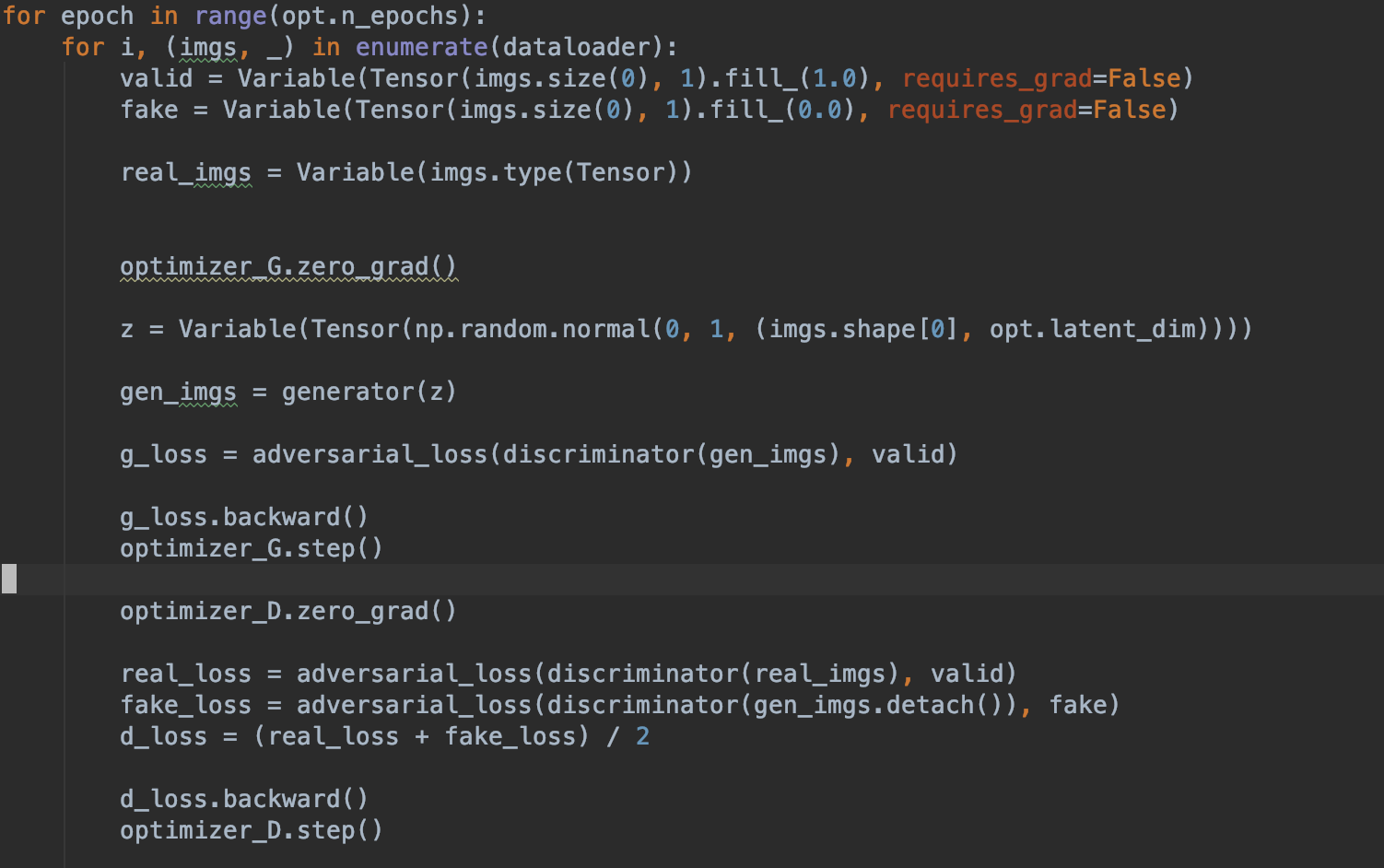
Initialize and save generator and discriminator



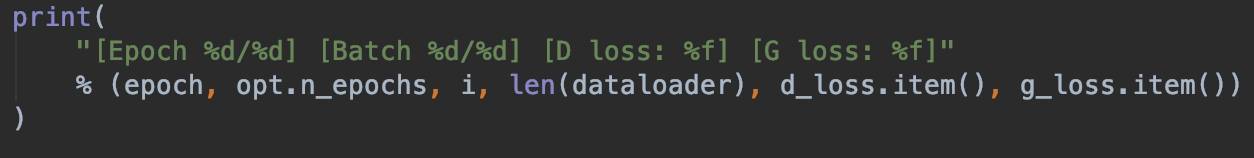
Configure data loader



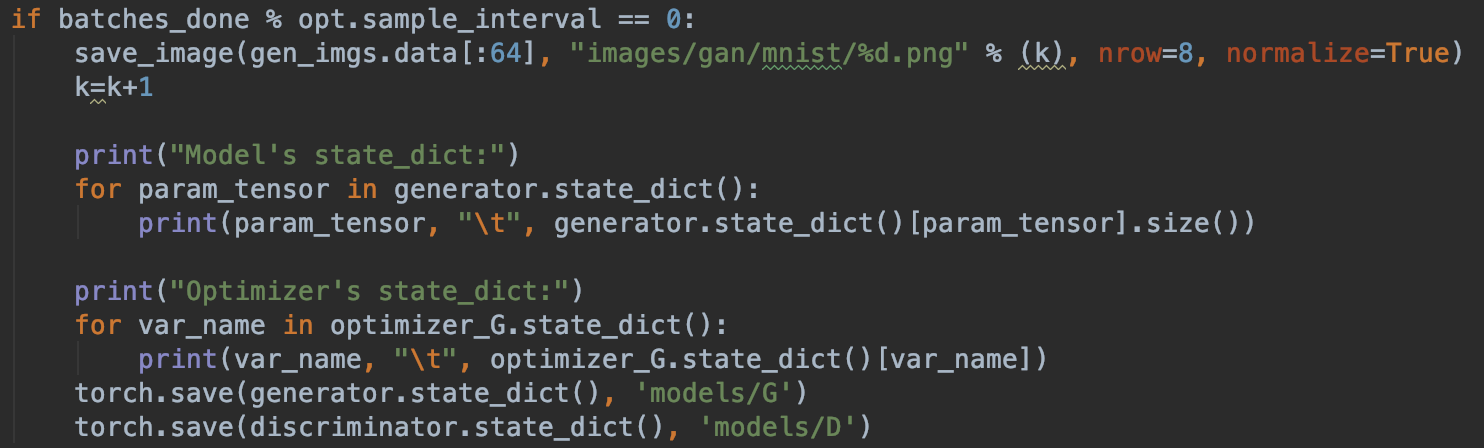
We used the Adam optimizer for our GANs



Training and updating weights



Display Loss



Save Image and model throughout training, also display current weights