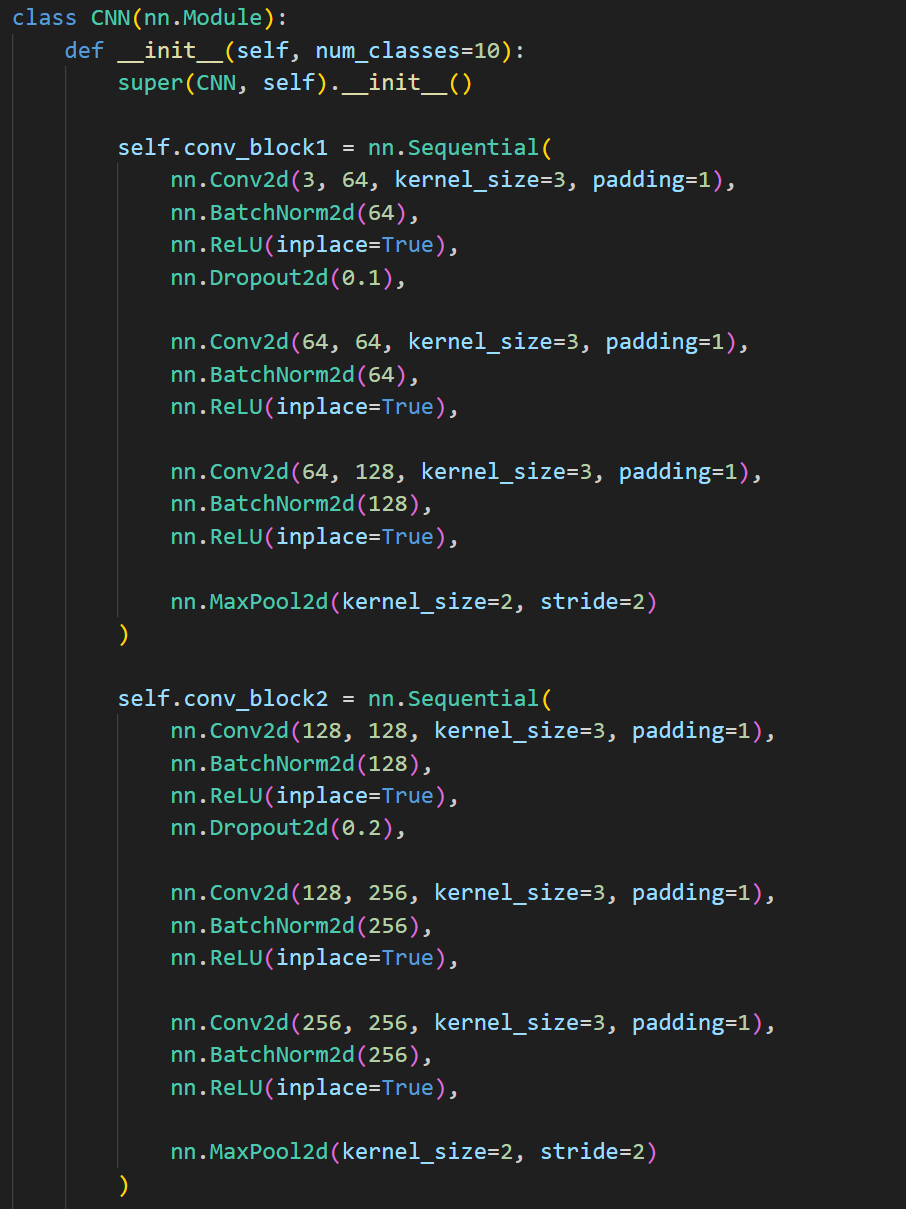
1. 网络设计

先来看网络部分：

先是定义了两个卷积块，在第一个卷积块中，第一层: 3→64通道，3×3卷积，Padding设为1，卷积完成后进行一次批归一化，随后用ReLU激活，最后在训练时设置0.1的Dropout率。

第二层: 64→64通道，3×3卷积，批归一化，ReLU激活

第三层: 64→128通道，3×3卷积，批归一化，ReLU激活

最后定义了一个池化层以增强鲁棒性，此处使用了2×2最大池化，步长为2

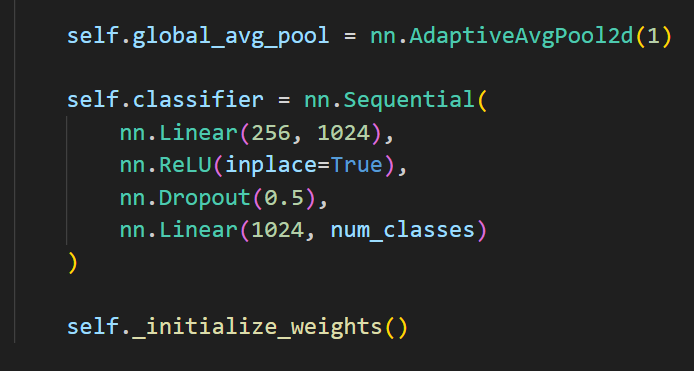
第二个卷积块也是类似。

第四层: 128→128通道，3×3卷积，批归一化，ReLU激活，Dropout(0.2)

第五层: 128→256通道，3×3卷积，批归一化，ReLU激活

第六层: 256→256通道，3×3卷积，批归一化，ReLU激活

2×2最大池化，步长为2

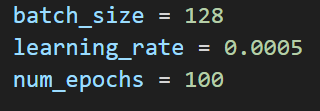
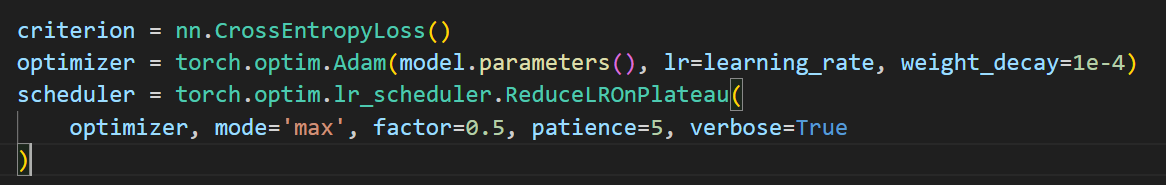


两个卷积块过后是一个全局平均池化层，最后是两个全连接层，神经元个数256->1024->10（类别数），中间使用的激活函数为ReLU，另外还用了一个Dropout(0.5)。可以看到，在训练过程中Dropout率是逐步变大的，这有利于在初期提取更多特征，而在后期更大的Dropout率有助于防止过拟合。

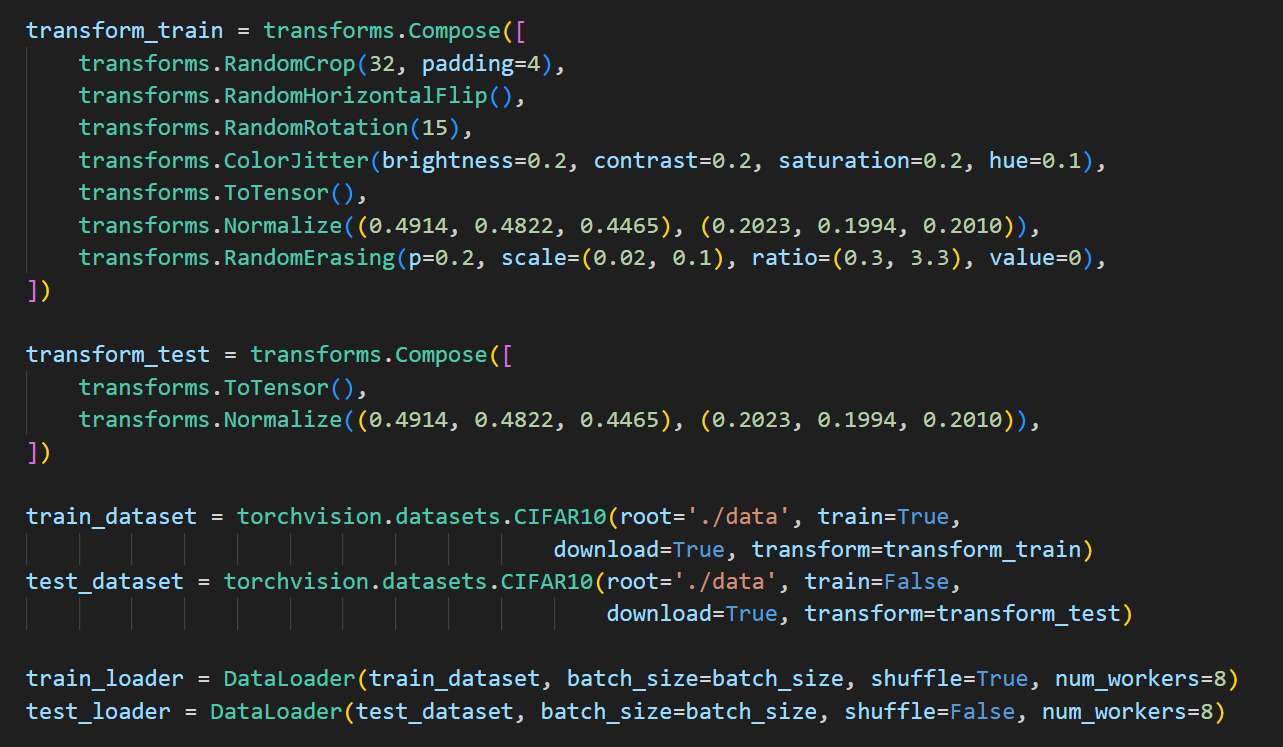
采用先卷积后全连接的结构主要是受VGG-16网络的启发，VGG网络也是先通过若干个含有池化和卷积的卷积块，随后通过一个全局池化层，最后经过若干全连接层得到结果。然而原始的VGG-16网络层数过多，对于32\*32分辨率的cifar10数据集并不适用，过小分辨率的图片在经过多层卷积层后其会变为1\*1，不利于后期处理。因而我减少了VGG-16网络的层数，采用了类似的卷积块和全连接的组合，只不过数量较少，最终形成了如上文所述的结构。

在不同的卷积层之间，可以看到通道数逐渐增加，从3→64→128→256，这有利于逐步增加特征提取能力，捕获更复杂的特征。引入批归一化主要是提高训练的稳定性，而全局平均池化主要是为了增强平移不变性，同时减少参数量简化计算。

二、训练过程中的调整及对性能的影响

损失函数选择的是多分类问题经典的交叉熵；Batch\_size选择128，大小适中；初始学习率为0.0005，考虑到EPOCH数为100，会比较多，因此可以设置较小的学习率，在多个EPOCH中慢慢调整，并且还使用了自适应学习率调度器ReduceLROnPlateau，使得它能够在连续5轮的测试集准确率下降时降低学习率，避免后期的震荡；优化器选择的是Adam优化器；同时还设置了权重衰减（即L2正则化），这样可以进一步防止过拟合。

三、数据增强技术的使用及其对模型泛化能力的影响



采用了如下方法进行数据增强：

1. 先将图像填充4个像素，然后随机裁剪出32×32大小的区域，相当于对图像进行一定程度的随机平移，保证模型具有平移不变性；
2. 以50%的概率水平翻转图像，相当于进行左右对称，增强模型对于左右对称的不敏感性；
3. 在[-15°, 15°]范围内随机旋转图像，提高模型对于角度变化的鲁棒性；
4. 随机调整图像的亮度、对比度、饱和度和色调，模拟不同光照环境下的图像变化，提高模型对于光照变化和颜色的适应能力。

随后的两行是将图片转化为pyTorch张量并进行标准化处理；最后一行时以20%的概率随即遮挡图片的一部分，也是数据增强的一种形式。若不执行前两行则会报错，因此转化为张量的操作是必要的。

如此进行的数据增强大大丰富了数据的数量和多样性，使得模型一方面可以有丰富的训练数据不至于过拟合；另一方面可以提高模型对于真实世界中各种情况的变化的鲁棒性，进而提高准确率和泛化能力。

四、训练过程中遇到的任何挑战以及你是如何解决的

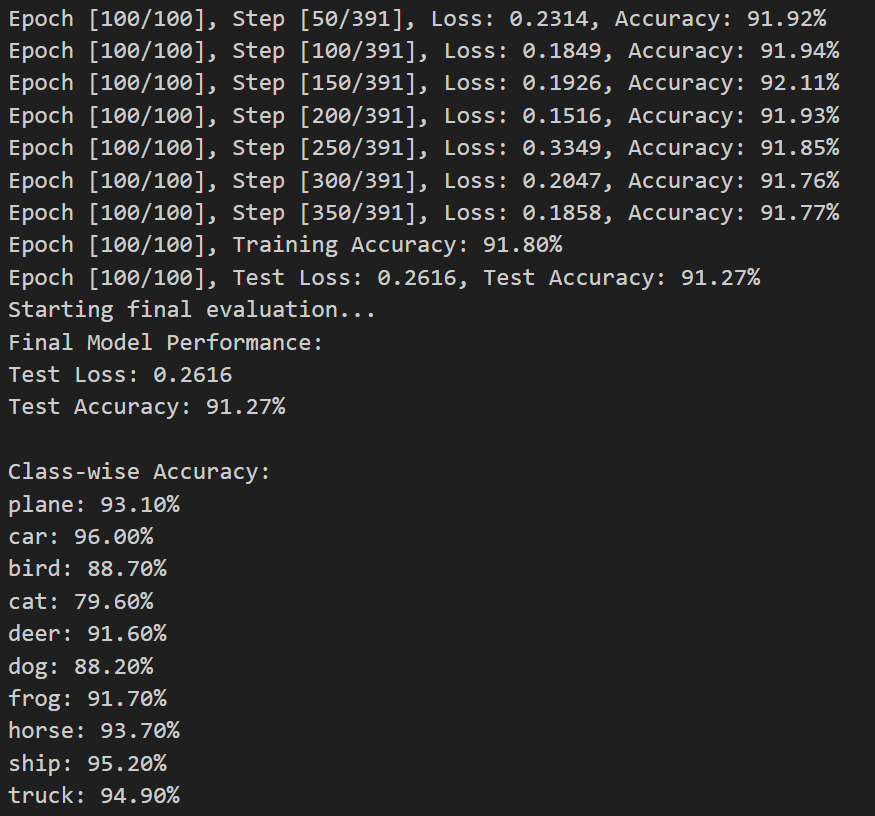
一开始我只使用了一层卷积层和一层池化层，但这一模型很容易出现过拟合现象，即经过足够多的EPOCH，模型在训练集上的准确率超过90%时在测试集上的准确率始终停在60%。正因如此，我参照了VGG-16网络的结构，设计了更贴合cifar10测试集的卷积块和全连接层结合的更复杂的网络结构。

与此同时，初始的学习率设置在0.001，这一学习率在后期的EPOCH中会表现得不甚稳定，常有测试集准确率下降的情况，说明振荡现象明显。因此我调低了学习率的初始值，并且采用了自适应学习率动态调整，有效避免了震荡，使得模型在后期能够比较稳定。

最后，观察到随着参数规模的上升，训练效率在cpu上有明显的下降，因此将训练移到了GPU上进行。加载数据时，我也通过增加使用的线程数目（num\_workers=8）来加速。不过即便有这些优化，进行100个EPOCH的训练仍需要一小时左右，这导致对模型的微调进程缓慢。硬件上的限制较难克服，只能等了……

五、最终评估

评估的具体结果放在log.txt中，此处仅做简要描述。



可以看到，经过100个EPOCH，训练集的准确率达到91.8%,而测试集准确率达到了91.27%.不过需要指出的是，这一准确率并非所有训练过程中最高的，最高的是EPOCH95的91.71%。根据学习率优化器的设置，它将在第101个EPOCH降低学习率，不过这里并没有体现出来，说明振荡现象依然存在，只不过大体上影响不大。我还增加了对不同类别图片的分别的测试，可以看到，猫、狗和鸟类别准确率较低，可能是因为它们特征相似度高。