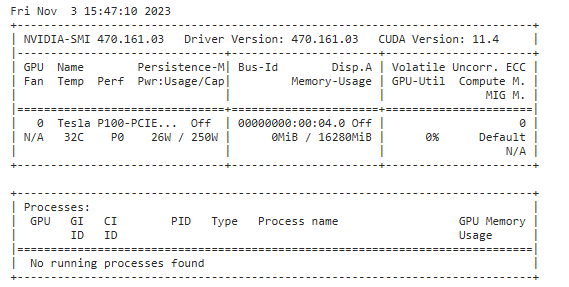
# **实验二：AlexNet测试Cifar-101**

#### 

1. **总述：**

本次实验需要通过pytorch手动搭建AlexNet网络结构并在Cifar-101数据集上进行测试。因此在写完并编译完成后，我在自己的cpu笔记本上进行了测试，并惊讶的发现一个epoch大概要花费十分钟以上的时间，同时由于最开始选择的优化器收敛较慢，大概需要80个epoch完成大概收敛，故放弃在本机上进行测试。转而想通过Colab平台白嫖T4显卡进行测试，测试到一半居然限额了，最后在Kaggle上完成本次实验。使用的显卡配置如下：

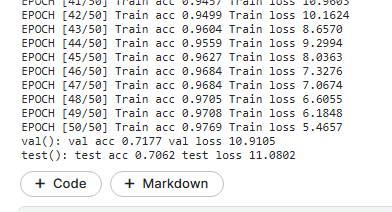


本次实验一些超参数设置如下：





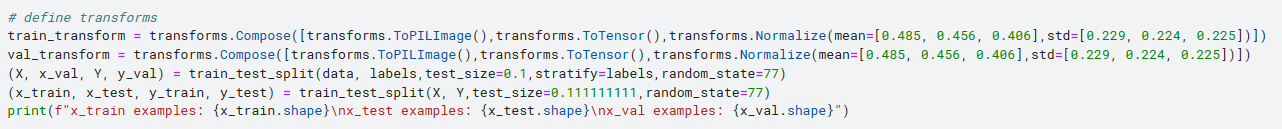
最终实现的效果如下：



1. **DataSet处理**

根据数据的的结构去进行数据的读取，依次读取每个文件夹下的图片，并将该图片与上层文件夹的名词进行匹配，并通过print进行验证是否正确处理。其中由于'BACKGROUND\_Google'不属于类别文件夹，在读取时进行跳过。本次实验最开始对图片进行65x65的resize，发现图片信息损失过大，最终识别效果不理想；然后测试了对图片的224x224的resize，但是很难保持在一个较大的batch下运行，即经常出现cuda error显存不够的情况；最后对224取一半，即112x112进行实验。



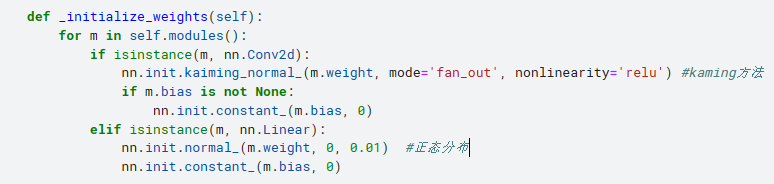
随后对图像进行变换，将图像变成张量并设置均值、标准差进行归一化处理。随后打乱图片顺序，按照8：1：1的比例分配train、val、test数据集的大小。

1. **AlexNet构建**

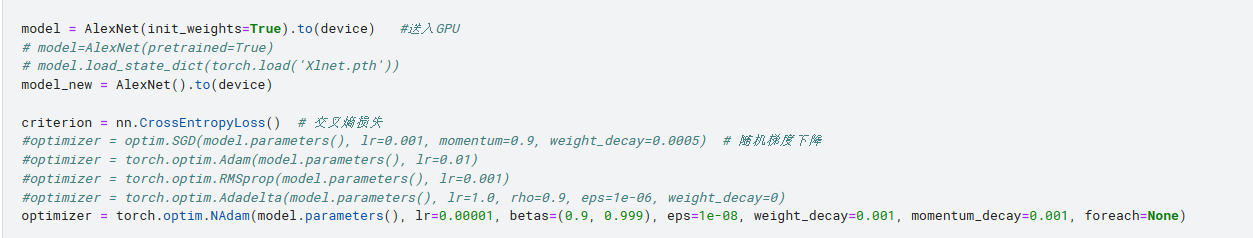
本次对于AlexNet的网络构建采用最常见的构建方式



网络内部参数的初始化：对于卷积层采用kaming初始化，对于线性层使用正态分布初始化。

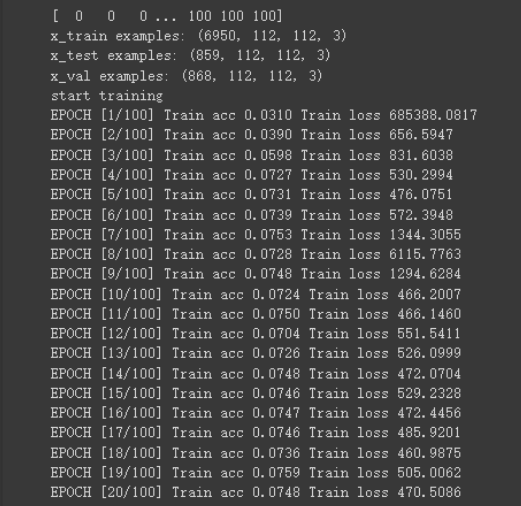


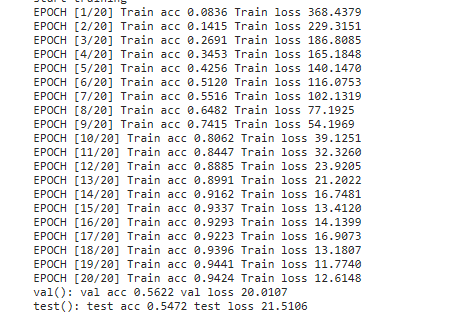
本次模型的loss采用交叉熵损失函数，优化器使用NAdam



1. **测试的一些有效的改进**
2. **优化器选择**

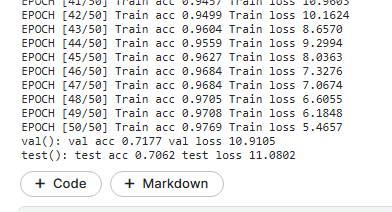
本次实验测试了多个优化器，效果最佳的是NAdam。实验最开始采用的是SGD，收敛的非常慢，大约得70~80个epoch才能大致收敛。然后我测试了RMSprop和Adadelta，RMSprop有更好的收敛速度，但是它们均在后面陷入了局部最优无法出去。于是我测试了Adam，以为能有更好的结果，结果Adam在最开始就陷入了一个局部最优并再也没出去过。



最后我找到了Adam的优化版本NAdam，在lr=0.001的时候，它只需要15个epoch就能接近收敛：

1. **Lr，weight\_decay的调整**

但是这样模型的正确率仍然较低，所以需要额外的调整。由于模型在最后的epoch中不断震荡，我将lr直接改小了两个数量级，变为0.00001，让模型缓慢下降，但是更加稳定。同时由于考虑到模型在训练集上正确率的提升空间有限，我想到可能模型会有点过拟合，所以将模型的weight\_decay从0改到0.001。最终实现了测试集上从55%到70%的提升。当然上面一张截图还没完全收敛，但是正确率大概也在60%左右。



1. **两个优化器**

我想通过先NAdam后SGD的方式来使得模型即具有快速收敛的速度优势，又具有稳定的迭代表现，初步实验过后发现效果不佳，但是理论上是一种可行的优化。

1. **实验结果**

