**《计算机视觉与机器学习》课外实践练习三**

——实现不同卷积神经网络对HelloRS32数据集分类

* 作业截止时间

请于该时间之前提交作业：2023年11月5日，5:00 pm

* 团队合作

允许自由组建团队合作完成本次实践练习（团队人数上限3人）。每个团队共同提交一份作**业文件**，团队成员共享本次实习成绩。请在Jupyter Notebook(.ipynb)文件对应位置处注明团队各个成员的姓名和学号。

* 考察方式

作业收齐后，将择期开展线上会议（具体时间另行通知），对各团队作业完成情况进行考评，考评人员（老师或助教）将随机指定程序功能或代码，要求进行演示和讲解。团队的**每位成员**都将被要求独立回答问题。

本次实习占总成绩比例为20%；团队实习成绩 = 提交作业评分(50%) + 线上考评表现(50%)

线上会议室将设等候室，考评过程中各团队将轮流进入会场，团队每位组员将被要求开启摄像头确认身份，故请各位同学开会前调试好设备，并选择适宜的场地。

* 诚信问题

严禁不同团队间代码的拷贝或抄袭，经代码查重检测确定为抄袭的作业，实习成绩将为0分。

# 实习要求

使用pytorch框架的模块完成任务：

1. **使用提供的三种卷积网络形式对HelloRS32数据集分类**
2. model1：编写并实现简单的双层卷积神经网络的训练及预测，精度**86%**左右
3. model2：编写并实现三层卷积神经网络的训练及预测，精度**88%**左右
   * 使用了relu作为激活函数
   * 在线性层中加入dropout
   * 使用局部响应归一化层lrn
   * 相比与modle1拥有更深的卷积层
4. model3：编写并实现很多层的卷积神经网络的训练及预测，精度**91%**左右
   * 更小的卷积核尺寸、卷积步长
   * 使用批量归一化batchnorm代替局部响应归一化层lrn
   * 更深的网络
5. **选做**：尝试在model3的基础上加入skip connection，进一步提升精度

**二、完成神经网络网络在HelloRS32上的训练、预测函数**

请仔细阅读以下内容，认真理解其中的提示信息，分步骤依次解决课外实践中的问题，完成代码编写和测试，最后按要求提交作业。

# PyTorch是什么?

PyTorch是一个在Tensor对象上执行动态计算图形的系统，其行为类似于numpy ndarray。它带有一个强大的自动微分引擎，使得手动编写代码进行反向传播计算不再必要。

## 为什么采用PyTorch

* 我们的代码现在将在gpu上进行速度更快的训练。
* 我们希望你准备好在你的项目中使用PyTorch，这样你就可以更高效地进行试验，而不是手动编写你想要使用的所有功能。
* 我们希望你能站在巨人的肩膀上！PyTorch是很棒的框架，会让你的试验变得更简单，现在你已经理解了它的本质，你可以自由地使用它了:)
* 我们希望你能接触到未来你在学术界或工业界可能会遇到的那种深度学习代码。

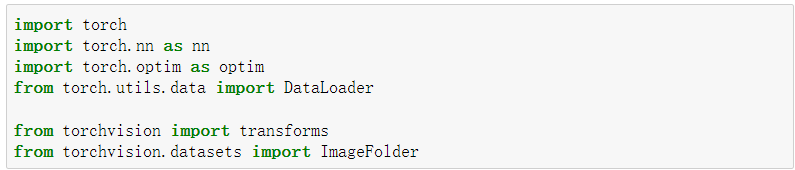
## PyTorch版本

假设你正在使用**PyTorch 1.0版本**。在之前的一些版本中(比如0.4之前)，Tensor必须被包装在变量对象中才能在autograd中使用；然而变量现在已被弃用。此外，1.0还将Tensor的数据类型从它的设备中分离出来，并使用numpy风格的接口构造Tensor，而不是直接调用Tensor构造函数。

# **PartⅠ:准备阶段**

首先，加载HelloRS32数据集，对其进行预处理，并以mini\_batch遍历它；PyTorch为我们提供了方便的工具来自动化这个过程。

## 导入模块



## 超参数设置



## HelloRS32数据集加载



## 是否启用GPU

你可以**通过设置下面的flag为True使用GPU**。本课程的任务可以不需要使用GPU，但使用GPU可以加速计算。请注意，如果你的计算机没有启用CUDA，torch.cuda.is\_available()将返回False，将回退到CPU模式。 全局变量dtype和device将控制整个赋值过程中的数据类型。



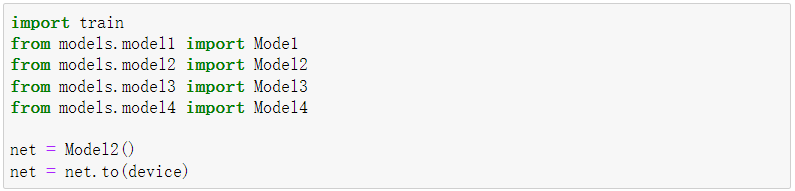
# PartⅡ：模型的设置

新建 model(\*).py,参照给定的网络构造实现不同卷积神经网络对遥感数据集HelloRS32进行分类：

你需要查看 torch.nn 和 torch.nn.functional 来了解pytorch中卷积、池化、激活函数、全连接等功能是如何实现的。

按照model.xlsx文件中的网络结构完成神经网络的定义。

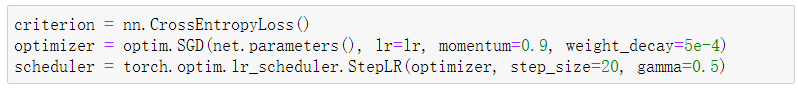
当你成功实现model之后，你可以尝试使用更多网络，如resnet、vggnet、mobilenet、efficientnet等更强大的网络来在HelloRS32上训练，以提高你的精度，当然这也意味着需要更大的算力



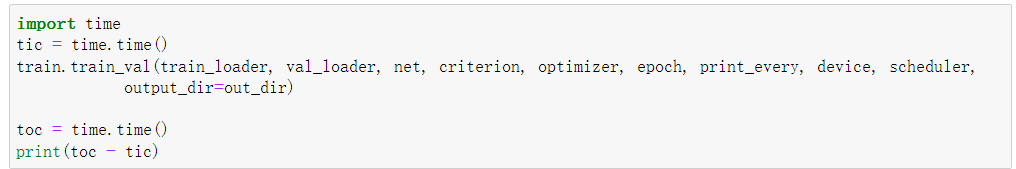
# PartⅢ：PyTorch的训练

## 训练集

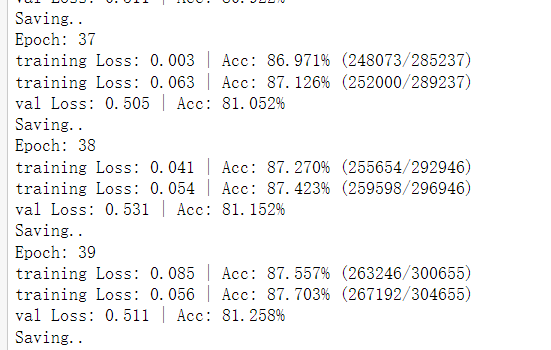
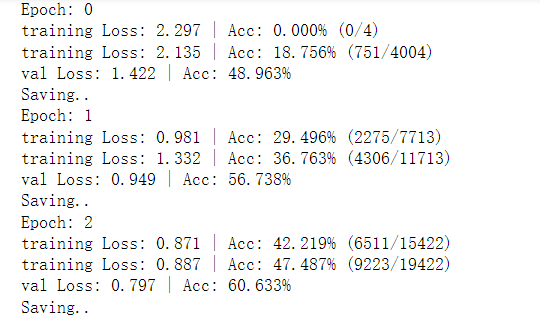
在开始网络的训练之前，我们还需要设置网络的损失函数、优化器，这些设置对于网络的训练也具有重要的影响，但本课程暂不要求对这些设置具有深刻的理解，我们会给出一些示例，你在后续的训练中可以更换不同的损失函数、优化器以及scheduler，以提高训练的精度和效率。



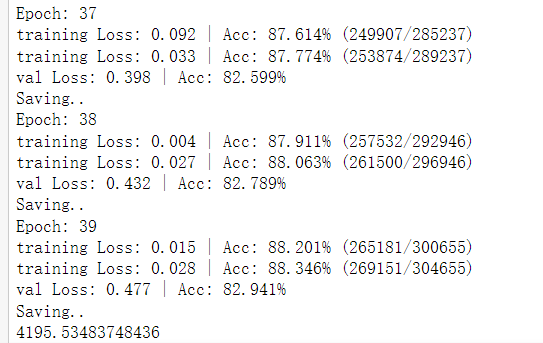
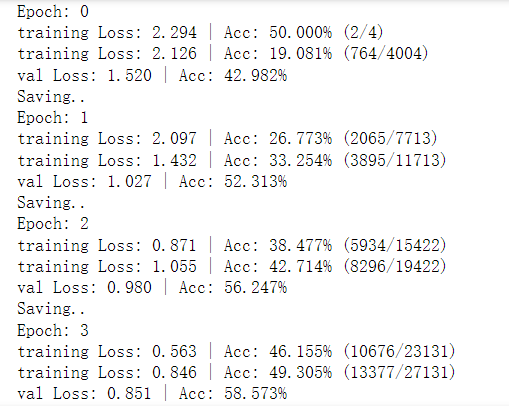
完成前期的设置之后，我们就可以开始编写训练的代码了，打开train.py，完成train和test函数，并在下面运行训练和测试的代码。



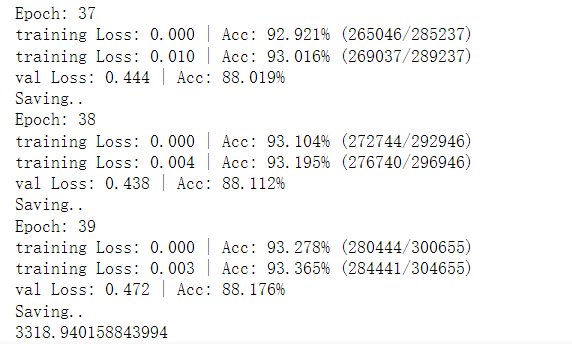
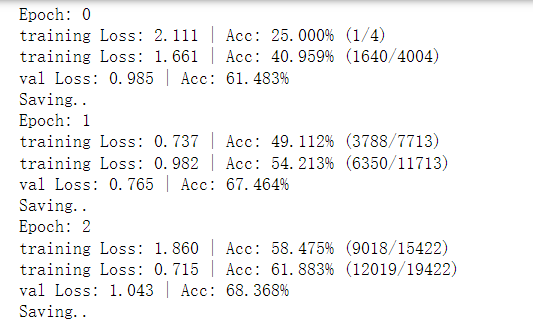
结果示例：



Model1训练过程



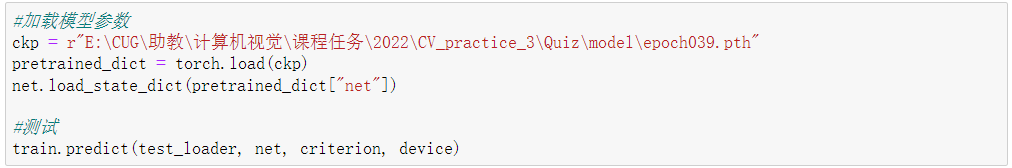
Model2训练过程



Model3训练过程

## 测试集

我们通过训练得到了精度最高的模型之后，在测试集上运行我们的模型，并得到测试精度。



结果示例：







# 作业提交要求

1. 在Jupyter Notebook下运行PyTorch.ipynb笔记本，首先需在笔记本开头处填写团队各成员的学号和姓名（如下图所示）；完成全部功能和输出相应结果后，将该笔记本内容打印为PDF文件提交，PDF需统一重命名为**队长学号\_队长姓名.pdf**。
2. 实习过程中需要不断更新train.py和model.py**或者**，实习完成后，将几个源码文件**合并保存**到一个py文件下。该源码文件需作为作业附件提交用于检查，文件需统一命名为**队长学号\_队员学号\_队员学号.py**。

