



**计算机视觉**

**上机实验报告**

|  |  |
| --- | --- |
| 专 业： | 人工智能 |
| 班 级： | 192204 |
| 学 号： | 19220432 |
| 姓 名： | 陆昊宇 |

2025年03月11日

## 一、实验目的

1. 熟悉使用pytorch完成图像分类任务。

2. 探究网络的超参数对其性能的影响。

## 二、实验内容要求

#### 1. 图像分类实验

（1）在CIFAR100数据集上选择ResNet作为分类模型，实现图像分类任务的训练和测试的完整流程，并分别针对以下问题设计实验并进行讨论：

1）**网络深度对分类性能的影响**：比较不同深度的ResNet（例如 ResNet-18、ResNet-34、ResNet-50）在CIFAR-100上的分类性能，从准确率、收敛速度等角度分析网络深度的影响；

2）**网络宽度对分类性能的影响**：比较不同宽度的ResNet34（例如每层的卷积核数量设置为原始的0.5倍、1倍、2倍等数量）在CIFAR-100上的分类性能，从准确率、收敛速度、参数量、计算负担等角度分析网络宽度的影响；

3）**残差连接的作用**：在 ResNet34 中移除残差连接（或替换为普通的卷积层），比较两种网络在相同深度和训练条件下的收敛速度和准确率，讨论残差连接在缓解梯度消失、加速收敛和提高模型性能上的作用。

**提交入口：**

https://send2me.cn/7fFNEun8/T5q5IFLRvNg36A

## 三、实验报告评分标准

1. 完成图像分类实验1）（30’）

2. 完成图像分类实验2）（25’）

3. 完成图像分类实验3）（25’）

4. 实验过程记录完整，表述逻辑清晰（10’）

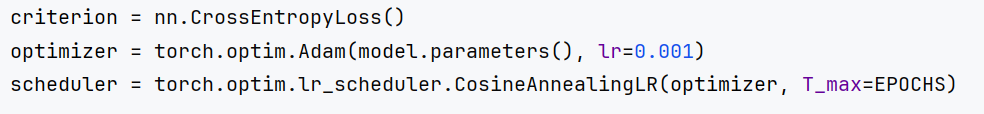
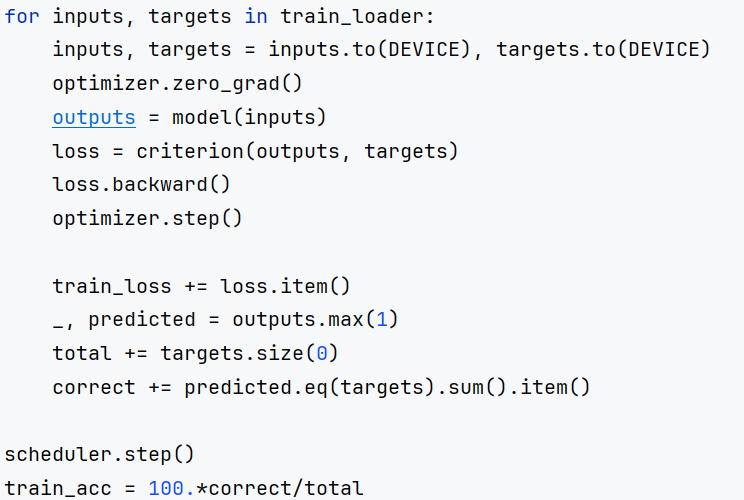
5. 排版工整自洽，图表说明完整（10’）

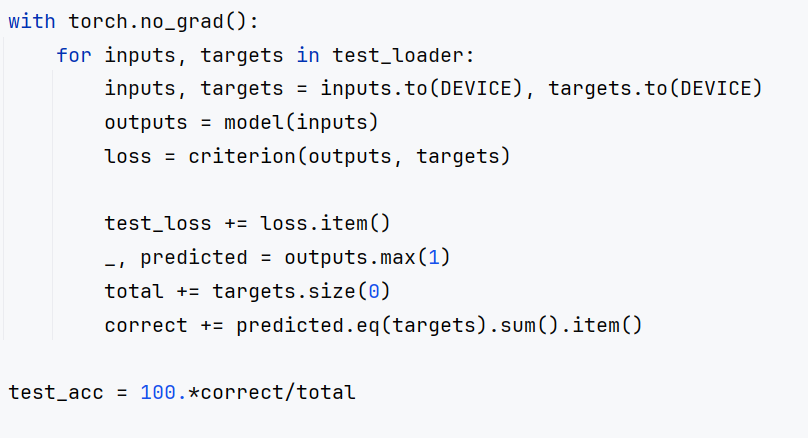
## 四、实验过程

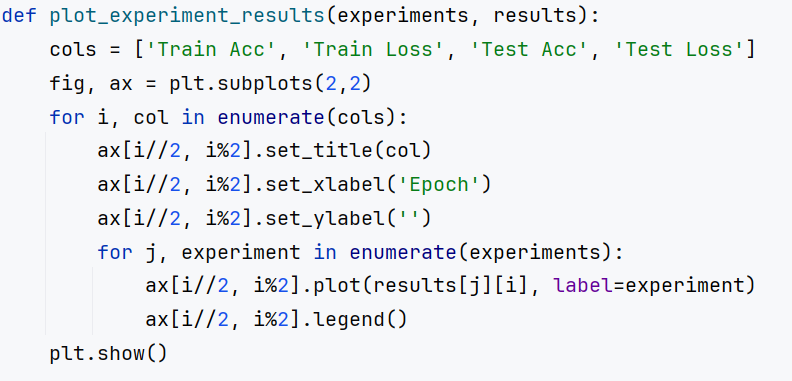
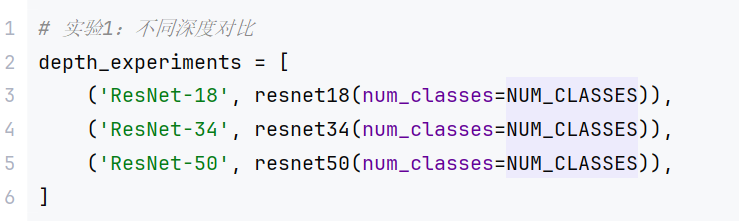
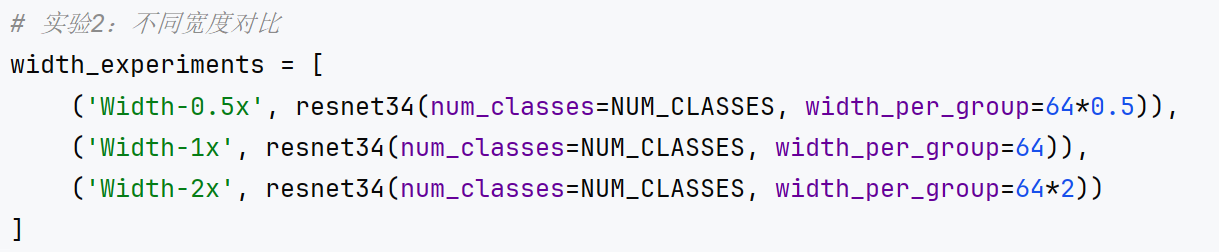
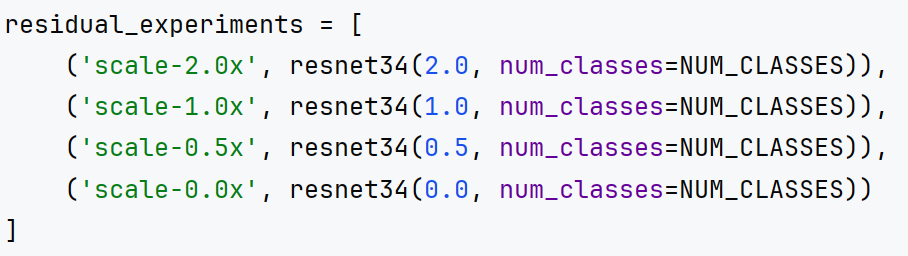
1. 读取数据集并进行预处理



1. 定义实验控制函数

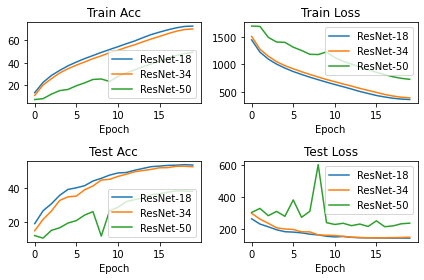
2.1. 优化器部分  
  
2.2. 训练部分（每个epoch内部）  


2.3. 测试部分（每个epoch内部）  


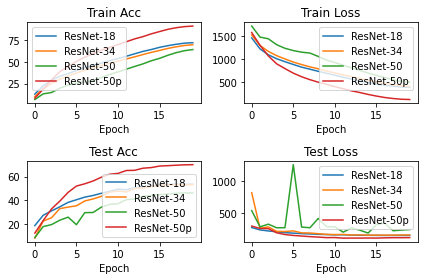
1. 可视化训练过程  
   
2. 对比不同深度的resnet网络  
   2.1. resnet 网络可以调用库函数  
    from torchvision.models.resnet import resnet18, resnet34, resnet50  
   
3. 对比不同宽度的resnet网络（以resnet34为例）  
   5.1. resnet34中有一个参数width\_per\_group，可以用于控制网络的宽度，默认值为64  
   5.2. 在block模块中，该参数的取值只能为64，解决方案将在后文给出  
   
4. 对比残差连接的作用（以resnet34为例）  
   6.1. resnet 模块中没有给出可供调用的参数，但是基于残差连接的计算方法：F(x)=H(x)-x,我们只需要加入一个缩放因子scale：F(x)=H(x)\*scale-x，具体实现在后文给出  
   6.2. 当scale=0.0时，网络即为普通CNN，当scale=1.0时，网络就是经典的resnet  
   6.3. scale可以取更多值，这代表残差模块在网络中权重  
   

## 五、实验结论

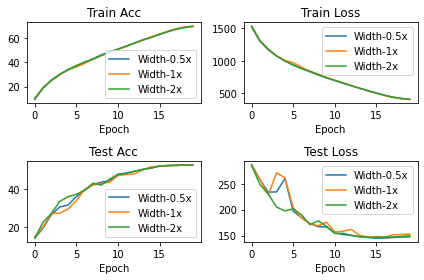
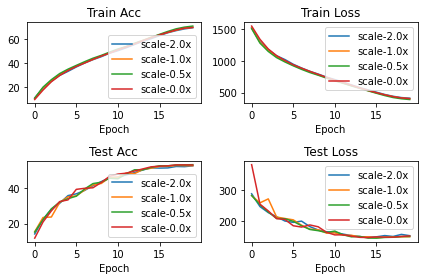
1. 不同深度的网络对比

  
这里发现resnet18的效果是更好的，而更深的resnet50反而表现很差。

经过调研发现，resnet网络出现时是针对ImageNet数据集的，而实验中使用到的cifar100数据集相对于ImageNet图片的分辨率要低得多，数据量也更少，因此resnet50可能无法充分学习其中的特征。  
基于这样的推测，我将resnet50模块进行调整，将首层卷积从7x7修改为3x3，并移除掉初始的max pooling层，得到结果如下：

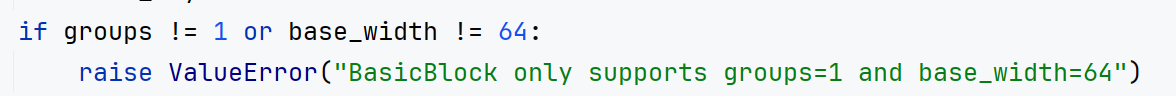


可以发现，resnet50的效果得到明显的提升，因此低质量的数据集确实容易导致resnet50性能不好，需要调整网络结构，或者增加正则项来改善。

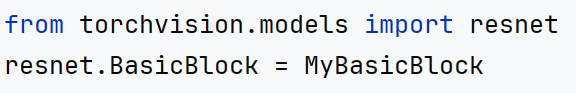
1. 不同宽度的网络对比  
     
   可能也是数据集质量低的问题，不同宽度的网络总体差异不大，更宽的网络训练似乎更加平滑。
2. 不同程度残差连接的对比  
     
   由于网络本身也不是非常深，因此残差连接的作用并不明显，经过具体的数值比对，不同scale的性能有如下关系：1.0>0.5>2.0>0.0；  
   这说明残差是有意义的，当然当scale更大时可能并不会取得更好的效果。

## 六、附录

1. 调整参数width\_per\_group  
在resnet源码中，BasicBlock模块有一个条件判断：

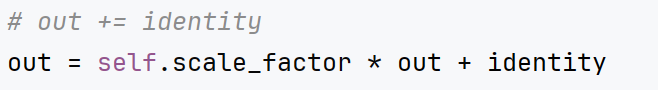


我们无法直接修改源码，因此采用曲线救国的策略。我们先将BasicBlock的代码复制到自己的文件，然后将这个条件判断注释掉。但是这时候我们的模块并不会对resnet生效，因为它是直接在resnet包中被引用的，而非通过参数的方式。利用python的语言特性，类名也可以作为对象赋值，我们通过以下方式将自己的模块注入进去：



此时，我们再导入resnet18, 34, 50模块就可以任意指定base\_width的值了。

注：这里的base\_width虽然在名字上与width\_per\_group不同，但是按照异常路径检索一下，就可以发现，base\_width的值是直接从width\_per\_group传过来的。

1. 实现缩放因子scale  
   实际上只需要对前向传播模块做一处小小的改动就可以了。（注释是无缩放因子情形）  
   

剩下的只要故技重施，按照1的办法注入一下就好了。