

## Task1

1. 我是通过在 anaconda prompt 中输入了 nvidia-smi 来查看我的 nvidia 显卡状态，从而确定使用 cuda 版本的 pytorch (这个版本运行速度基本上都要比 cpu 版本的快一些)
2. 在此过程中我其实遇到了蛮多问题的，在下载 pytorch 时，网络超时了两次，因此后面我采取了清华镜像加快了下载速度，配置环境忘记更换而也导致了下载失败，还有一次是因为依赖冲突，好在最后成功安装。而在使用 git 时，发现将 vscode 终端关联本地项目时，命令栏无法识别 git 仓库，导致花费时间较长。

## Task2

3. 学习率过大，曲线易剧烈波动，拟合图像会出现明显偏差，而学习率过小，曲线会显得过于平滑，细节丢失严重。
4. 紧紧跟着每个原始数据点波动，不平滑（隐藏层数为三层时会出现过拟合）

## Task3

### 5. 训练日志：

Epoch 1, Loss: 1.606  
Epoch 2, Loss: 1.339  
Epoch 3, Loss: 1.223  
Epoch 4, Loss: 1.138  
Epoch 5, Loss: 1.091  
Epoch 6, Loss: 1.041  
Epoch 7, Loss: 1.004  
Epoch 8, Loss: 0.980  
Epoch 9, Loss: 0.953  
Epoch 10, Loss: 0.937  
Epoch 11, Loss: 0.916  
Epoch 12, Loss: 0.896  
Epoch 13, Loss: 0.883  
Epoch 14, Loss: 0.872  
Epoch 15, Loss: 0.853  
Epoch 16, Loss: 0.844  
Epoch 17, Loss: 0.834  
Epoch 18, Loss: 0.817  
Epoch 19, Loss: 0.810  
Epoch 20, Loss: 0.800  
Test Accuracy: 73.79%

### 6. debug 复盘

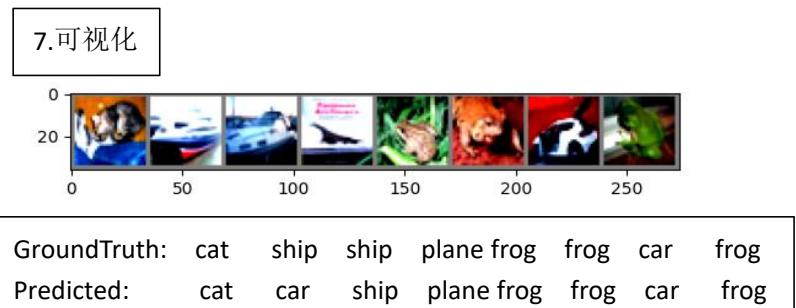
- (1) 全连接层输入维度算错：nn.Linear(32\*8\*8, 128)。当我将卷积层数变化时，这里会报错，主要是因为后续忘记用 x.view() 展平，直接丢进了全连接层。
- (2) 网络过浅，学习率设置不当，导致模型精度偏低，难以收敛。
- (3) CIFAR-10 下载经常超时，运行到 trainset 直接报错。换网络环境。

## Task4

### 7. 创新方面

#### (1) 策略优化：

引入 Warmup 学习率预热：前 5 个 epoch 线性提升学习率，之后使用余弦退火，避免训练初期梯度爆炸。



引入 Mixup 数据增强：通过随机混合两张图片及其标签，增强模型泛化能力，减少过拟合。

## (2) 结构优化

在基础 CNN 中加入 \*\*残差连接（Residual Block）\*\*，解决了加深网络带来的梯度消失问题，使模型可以训练更深的结构。

使用 \*\*BatchNorm\*\* 层，加速训练收敛，稳定分布。

## Task5

8. 分割任务（Mask）与分类任务（Label）数据处理有什么不同？

(1) 分类任务（Classification Label）：关注全局特征，给整张图贴标签（语义级）——数据量小，比较简单

(2) 分割任务（Mask）：关注空间结构，边缘，细节，给每个像素贴标签（像素级）——处理较为复杂

## 9. 常见难点

1) . 难点：Mask 与 Image 尺寸对不上

表现：训练报错、mask 错位、遮罩跑到外面

解决：1) ) 保证 image 和 mask 完全同步 resize、flip、crop

2) ) 使用同一个随机种子做数据增强

3) ) mask 用最近邻插值（nearest），不要用线性插值

2) . 难点：标注格式不统一（json、png、coco、自定义）

表现：读不出 mask、mask 全黑、类别错位

解决：1) ) 先写可视化脚本，把 mask 画出来检查

2) ) 统一转成 COCO 格式或单通道灰度 mask

3) ) 确保类别 id 从 0 开始（0=背景）

3) . 难点：正负样本极不平衡（背景像素远多于前景）

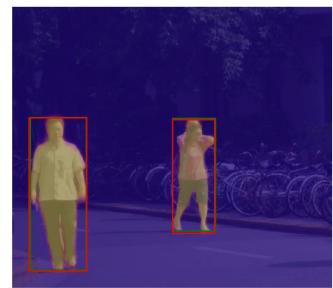
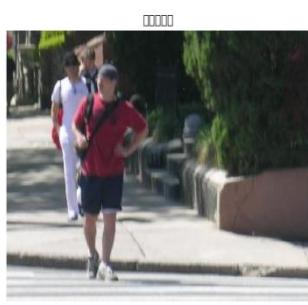
表现：模型只学背景，mask 全黑

解决：1) ) 使用 DiceLoss、FocalLoss、BCEWithLogits+权重

2) ) 数据增强，多扩增小物体

3) ) 控制训练采样，保证每批有足够前景

## 10. 原图





11. 个人认为 ai 最大的用处就是帮忙解释代码的含义，以及一些代码的编写，修改。但是还有个问题就是 ai 乱解释，对于一些问题没法正面清楚直接提出解决方案。

12. 如果没有 ai，我觉得自己可以独立完成代码的基本框架的构造，和后续的一些创新部分。