

实验报告四

1.实验准备

1.1 库的导入

- 本实验使用pytorch_grad_cam库进行可解释性分析，该库的github仓库参见[jacobgil/pytorch-grad-cam](https://github.com/jacobgil/pytorch-grad-cam)
- 导入cv2库来对图像进行处理
- 导入matplotlib库对图像进行展示
- 导入np库进行数据类型转换

1.2 模型加载与打印

- 使用torch.load加载模型
- 使用print打印模型结构，模型结构如图所示：

```
AlexNet(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(11, 11), stride=(4, 4), padding=(2, 2))
    (1): ReLU(inplace=True)
    (2): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): ReLU(inplace=True)
    (5): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(192, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU(inplace=True)
    (8): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): ReLU(inplace=True)
    (10): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(6, 6))
  (classifier): Sequential(
    (0): Dropout(p=0.5, inplace=False)
    (1): Linear(in_features=9216, out_features=4096, bias=True)
    (2): ReLU(inplace=True)
    (3): Dropout(p=0.5, inplace=False)
    (4): Linear(in_features=4096, out_features=4096, bias=True)
    (5): ReLU(inplace=True)
    (6): Linear(in_features=4096, out_features=2, bias=True)
  )
)
```

1.3 初始化

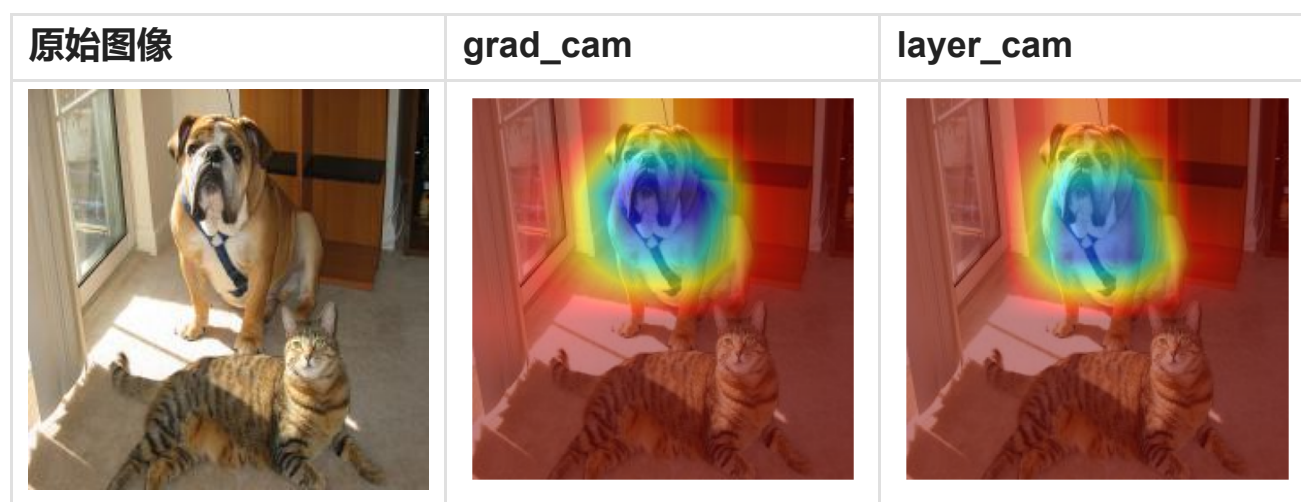
- 初始化cam：pytorch_grad_cam库提供了GradCAM和LayerCAM等类激活热力图方法，根据文档要求，我们需要先定义 `target_layers`，这里我们选择卷积层的最后一层，即`model.features[-1]`作为目标层

- 图片初始化：先使用OpenCV的 `imread` 函数来读取图像文件，再使用 `np.float32(rgb_img) / 255` 将图像的数据类型转换为 `float32`，然后对所有像素值进行归一化，使它们在0到1的范围内。
- 归一化：使用库自带的归一化函数进行进一步的归一化
- `targets`：`targets` 是另一个需要声明的变量，决定了cam图的对象类别



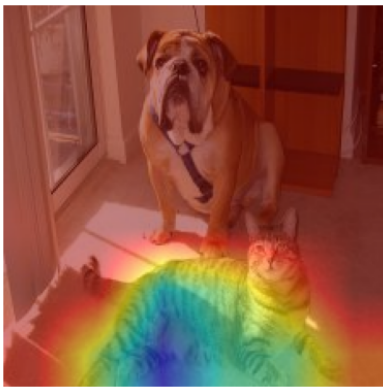
2.实验结果与实验分析

2.1 both.jpg

- 狗类别的热力图分析
 - 从图中可以看出两种可视化方法都找到了狗的位置，且主要通过狗的身体以及下巴来进行分类



- 猫类别的热力图分析
 - 在grad_cam中未能找到猫的分类，而layer_cam却能较好的找到猫的位置，可能是由于参数的问题，也又可能是layer_cam的精度要好的多，可以找到在grad_cam中被忽略的部分

原始图像	grad_cam	layer_cam
		

2.2 cat.jpg

- 狗类别的热力图分析
 - 由于原图中不存在狗的部分，cam图聚焦的是图片的边缘部分，可能是这些边缘存在一些与狗类似的部分




原始图像	grad_cam	layer_cam
		

- 猫类别的热力图分析
 - 从图中可以推测，在进行对猫的分类时，头部是一个经常被关注的区域，热力图显示的区域基本都聚焦于头部

原始图像	grad_cam	layer_cam
		

2.3 dog.jpg

- 狗类别的热力图分析
 - 可以看出这张图中对狗的分类依据是头部，热力图基本全部聚焦于一只狗的头部

原始图像	grad_cam	layer_cam
		

- 猫类别的热力图分析
 - 热力图主要聚集于脚部分，可能模型在分类时认为这一部分与猫有一定的相似程度

原始图像	grad_cam	layer_cam
