目录

[卷积神经网络可解释性分析 1](#_Toc155735554)

[一、引言 1](#_Toc155735555)

[二、实验方法 1](#_Toc155735556)

[2.1 数据集和图像处理 1](#_Toc155735557)

[2.2 可解释性分析方法 2](#_Toc155735558)

[三、实验过程 3](#_Toc155735559)

[四、实验结果 4](#_Toc155735560)

[4.1 特征图可视化结果 4](#_Toc155735561)

[4.2 Grad-CAM和LayerCAM分析 4](#_Toc155735562)

[五、实验分析 9](#_Toc155735563)

卷积神经网络可解释性分析

## 一、引言

在本次实验的目的是通过可视化技术来分析和理解一个已经训练好的卷积神经网络（CNN）的决策过程。所使用的模型是基于PyTorch的AlexNet二分类模型，能够区分猫和狗的图像。通过应用Grad-CAM和LayerCAM方法，探究模型对特定类别做出预测时的关注点。

## 二、实验方法

### 2.1 数据集和图像处理

实验使用三张提供的图像作为输入，分别是猫、狗以及同时包含猫和狗的图像。图像预处理包括调整大小为224x224，转换为张量，并进行归一化。

# 图像处理函数

*def* process\_image(image\_path):

    # 图像预处理

    preprocess = transforms.Compose([

        transforms.ToPILImage(),

        transforms.Resize((224, 224)),

        transforms.ToTensor(),

        transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),

    ])

    image = cv2.imread(image\_path)

    image = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)

    image = preprocess(image).unsqueeze(0)

    return image

### 2.2 可解释性分析方法

Grad-CAM：通过获取模型最后一个卷积层的梯度，生成热图来可视化决策重点区域。

*class* GradCAM:

*def* *\_\_init\_\_*(self, model, target\_layer):

*self*.model = model

*self*.target\_layer = target\_layer

*self*.gradients = *None*

*self*.activations = *None*

*self*.register\_hooks()

*def* register\_hooks(self):

*def* forward\_hook(module, input, output):

*self*.activations = output

*def* backward\_hook(module, grad\_in, grad\_out):

*self*.gradients = grad\_out[0]

*self*.target\_layer.register\_forward\_hook(forward\_hook)

*self*.target\_layer.register\_backward\_hook(backward\_hook)

*def* generate\_heatmap(self, input\_tensor, class\_idx):

        model\_output = *self*.model(input\_tensor)

*self*.model.zero\_grad()

        one\_hot\_output = torch.zeros\_like(model\_output).scatter\_(1, class\_idx.unsqueeze(-1), 1.0)

        model\_output.backward(gradient=one\_hot\_output, retain\_graph=*True*)

        pooled\_gradients = torch.mean(*self*.gradients, dim=[0, 2, 3])

        for i in *range*(pooled\_gradients.size()[0]):

*self*.activations[:, i, :, :] \*= pooled\_gradients[i]

        heatmap = torch.mean(*self*.activations, dim=1).squeeze()

        heatmap = F.relu(heatmap)

        heatmap /= torch.max(heatmap)

        return heatmap

LayerCAM：类似于Grad-CAM，但是在梯度计算时结合了特征图，可以提供更精细的可视化结果。

*class* LayerCAM(*GradCAM*):

*def* generate\_heatmap(self, input\_tensor, class\_idx):

        model\_output = *self*.model(input\_tensor)

*self*.model.zero\_grad()

        one\_hot\_output = torch.zeros\_like(model\_output).scatter\_(1, class\_idx.unsqueeze(-1), 1.0)

        model\_output.backward(gradient=one\_hot\_output, retain\_graph=*True*)

        guided\_gradients = *self*.activations \* *self*.gradients

        heatmap = torch.mean(guided\_gradients, dim=1).squeeze()

        heatmap = F.relu(heatmap)

        heatmap /= torch.max(heatmap)

        return heatmap

## 三、实验过程

在实验中，我们首先加载了预训练的AlexNet模型，并定义了处理图像的函数。

# 加载模型

model = torch.load('torch\_alex.pth')

model.eval()

# 图像处理函数

*def* process\_image(image\_path):

接着，我们实现了获取模型最后一个卷积层的函数，并定义了可视化特征图的函数。

# 获取最后一个卷积层

*def* get\_last\_conv\_layer(model):

    for layer in *reversed*(model.features):

        if *isinstance*(layer, torch.nn.Conv2d):

            return layer

    raise Exception("No convolutional layer found in the model")

# 可视化最后一层卷积层的特征图

*def* visualize\_feature\_maps(model, input\_tensor):

    last\_conv\_layer = get\_last\_conv\_layer(model)

    activations = *None*

*def* hook\_fn(module, input, output):

        nonlocal activations

        activations = output

    hook = last\_conv\_layer.register\_forward\_hook(hook\_fn)

    model(input\_tensor)

    hook.remove()

    num\_features = activations.shape[1]

    fig, axes = plt.subplots(1, num\_features, figsize=(20, 20))

    for i in *range*(num\_features):

        ax = axes[i]

        ax.imshow(activations[0, i].cpu().detach().numpy(), cmap='viridis')

        ax.axis('off')

    plt.show()

## 四、实验结果

### 4.1 特征图可视化结果

对于每张输入图像，我们在模型的最后一个卷积层输出了特征图的可视化结果。这些结果揭示了模型在做出分类决策时的特征激活情况。



**图4.1 both.jpg特征图可视化结果**



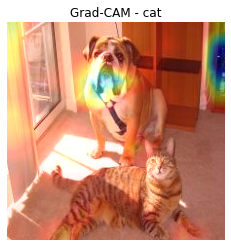
**图4.2 cat.jpg特征图可视化结果**



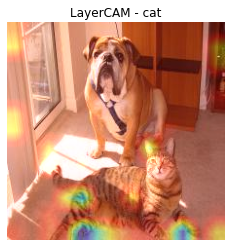
**图4.3 dog.jpg特征图可视化结果**

### 4.2 Grad-CAM和LayerCAM分析

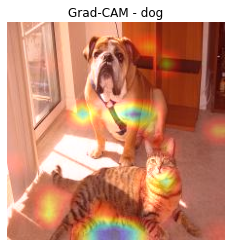
通过应用Grad-CAM和LayerCAM，我们对每张输入图片生成了针对“猫”和“狗“类别的热图，并将其叠加在原始图像上。热图清晰地显示了模型在预测“猫”类别和“狗”类别时关注的区域。



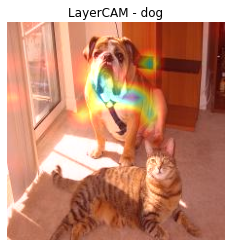
**图4.4 both.jpg针对“猫”类别的Grad-CAM分析结果**



**图4.5 both.jpg针对“猫”类别的LayerCAM分析结果**



**图4.6 both.jpg针对“狗”类别的Grad-CAM分析结果**



**图4.7 both.jpg针对“狗”类别的LayerCAM分析结果**



**图4.8 cat.jpg针对“猫”类别的Grad-CAM分析结果**



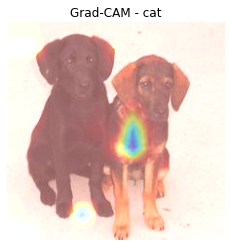
**图4.9 cat.jpg针对“猫”类别的LayerCAM分析结果**



**图4.10 cat.jpg针对“狗”类别的Grad-CAM分析结果**



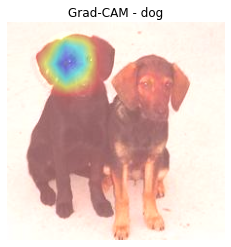
**图4.11 cat.jpg针对“狗”类别的LayerCAM分析结果**



**图4.12 dog.jpg针对“猫”类别的Grad-CAM分析结果**



**图4.13 dog.jpg针对“猫”类别的LayerCAM分析结果**



**图4.14 dog.jpg针对“狗”类别的Grad-CAM分析结果**



**图4.15 dog.jpg针对“狗”类别的LayerCAM分析结果**

## 五、实验分析

实验输出结果中的热图显示了模型在测试图像的哪些部分具有较高的激活，这些区域很可能是模型识别猫和狗的重要线索。根据实验结果，对于“猫”类别，热图通常集中在猫的脸部和身体上；而对于“狗”类别，则集中在狗的身体特征上。