华中科技大学

2023

计算机视觉实验　课程设计报告

专　　业：	计算机科学与技术
班　　级：	计卓 2101 班
学　　号：	U202112071
姓　　名：	王彬
邮　　件：	wangbin2002@hust.edu.cn
完成日期：	2024.1.12

目 录

实验四：基于卷积神经网络的可解释性分析

1.1 任务描述

针对已训练好的卷积神经网络，给定一张输入图片，生成该图片对于特定类别的可解释性分析结果。

实验将提供基于 PyTorch 和 TensorFlow 的两个不同版本的二分类模型，该模型可用于猫和狗的分类（class 0 为猫，class 1 为狗）。注：PyTorch 使用的网络架构是 AlexNet，TensorFlow 使用的是 VGG16，两者略有不同，请任选一个模型进行实验。

实验将同时提供三张输入图片，对于每张图片，分别针对猫和狗的类别，进行 Grad-CAM 和 LayerCAM 的可解释性分析。

注意事项：

1. 深度学习框架可选 PyTorch 和 TensorFlow。
2. 实验报告需包含每张输入图片在最后一层卷积层输出的的可视化结果（对输出特征图的每一个通道进行可视化），每张图片分别针对猫和狗两个类别的可解释性分析结果（Grad-CAM 及 LayerCAM），以及对应的实验分析。

1.2 实验内容

(一) 数据集获取

我们首先加载基于 PyTorch 的二分类模型。这里需要注意该模型使用的网络架构是 AlexNet，如图 1 所示。这里我们需要取出特征图的最后一层并进行分析。

```
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)
  )
)
```

图 1 AlexNet 模型

(二) 构建 Grad-CAM 和 LayerCAM 模型

我们通过构建基于类激活热力图进行可视化。Grad-CAM 模型的结构如图 2 所示，该模型对输出结果进行梯度求导，获得最后一层卷积层的梯度后，将得到的权重与最后一层卷积的特征图进行加权求和，再经过 ReLU 激活后得到类激活热力图。

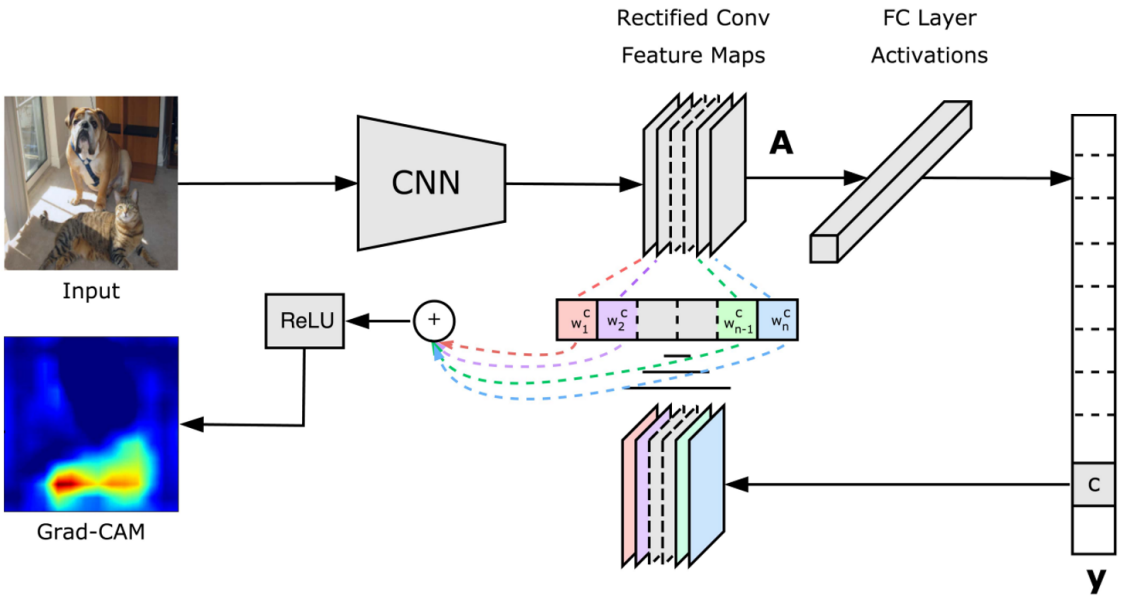

图 2 Grad-CAM 模型架构

求得分数最后一层卷积的每一个元素的偏导数，

$$g_{ij}^{kc} = \frac{\partial y^c}{\partial A_{ij}^k}$$

经过全局平均池化得到

$$w_k^c = \frac{1}{N}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$，其中N为A_{ij}^k的像素数。

类似 CAM，可以得到热力图：

$$L_{Grad-CAM}^c = ReLU(\sum_k w_k^c A_{ij}^k)$$

其中，结果量可以替换为网络中任何一个可微分的值，因此 Grad-CAM 也可以分析除了最后一层卷积以外的其它层。

然而，Grad-CAM 对于深层生成的细粒度热力图还不够精确；同时，尽管它可以用以分析中间层，但是对于浅层神经网络，其效果欠佳。因此我们对 Grad-CAM 进行一些修改得到 LayerCAM。

对于任意一层卷积神经网络，我们直接对于每个通道进行反向传播得到其权重，而不再对其每一层进行平均池化，以得到权重标量。

$$w_{ij}^{kc} = ReLU(g_{ij}^{kc})$$

因此其热力图的表示为，

$$L_{LayerCAM}^c = ReLU(\sum_k w_{ij}^{kc} A_{ij}^k)$$

这样，该方法获得的特征图的梯度可以达到元素级别，生成的类别激活图可以更加精确。

对于 Grad-CAM 方法，我们的模型计算关键代码为：

```
out = net(img)
```

```
# out = net(img.cuda())     # 前向传播
# 获取预测类别编码
cls_idx = torch.argmax(out).item()
# 获取预测类别分数
score = out[:, cls_idx].sum()
# 由预测分数反向传播得到梯度
net.zero_grad()
score.backward(retain_graph=True)

# 将各层梯度取平均池化
weights = grad[0][0].squeeze(0).mean(dim=(1, 2))

plt.subplots(16,16,figsize=(32,32),dpi=100)

# 对特征图的通道进行加权叠加
grad_cam = (weights.view(*weights.shape, 1, 1) *
feature_map[0].squeeze(0)).sum(0)
```

对于 LayerCAM 方法，我们的关键代码修改如下，

```
weights = F.relu(grad[0][0].squeeze(0))

# 对特征图的通道进行加权叠加
grad_cam = (weights * feature_map[0].squeeze(0)).sum(0)
```

在对特征图进行计算后，我们需要将计算结果进行归一化，便于图形绘制。这部分使用到的代码如下。

```
def _normalize(cams: Tensor) -> Tensor:
        """CAM normalization"""
        cams.sub_(cams.flatten(start_dim=-2).min(-1).values.unsqueeze(-
1).unsqueeze(-1))
```

```
        cams.div_(cams.flatten(start_dim=-2).max(-1).values.unsqueeze(-
1).unsqueeze(-1))
        return cams
grad_cam = _normalize(F.relu(grad_cam, inplace=True)).cpu()
mask = to_pil_image(grad_cam.detach().numpy(), mode='F')
```

对于本实验的要求，我们先对每一通道的分析结果加载至图片中，再将所有通道的计算结果进行汇总，并绘制热力图。因此我们获得的结果有两部分：一部分是每一通道的热力图分析结果，在实验提交材料./Results 文件夹下以*_each_channel.png 文件名给出。

(三) 绘制结果

我们对每一通道绘制其热力子图，并将 256 个通道绘制结果存放至一张 16*16 的图片中。我们的图像绘制代码如下。

```
# 对每一个通道进行 normalize 后，绘制每个通道的子图，汇总到 16*16 的大图中
list_channel = weights * feature_map[0].squeeze(0)
for i in range(list_channel.size(0)):
    imt = list_channel[i]
    plt.subplot(16,16,i+1)
    _grad_sub = _normalize(F.relu(imt, inplace=True)).cpu()
    mask1 = to_pil_image(_grad_sub.detach().numpy(), mode='F')
    result1 = overlay_mask(orign_img, mask1)
    plt.imshow(result1)

result = overlay_mask(orign_img, mask)
result.show()
# plt.show()
plt.savefig('./output1/both_layercam_each_channel.png')
plt.show()
result.save(save_path)
```

全篇代码详见附录中的代码文件（./test_gradcam.py & ./test_layercam.py）。

1.3 实验结果及其分析

经过实验，我们得到的结果见文件夹./Results 下。其中，文件夹 Results/Grad-CAM/下为 Grad-CAM 方法的分析结果，文件夹 Results/LayerCAM 下为 LayerCAM 方法的分析结果。

对于 Grad-CAM 方法，我们得到的分析热力图如下所示。对于每一通道，猫图像的 256 通道热力图如图 4 所示。我们可以清晰地观察到神经网络对于图片属于不同类别时，其所关注的区域影响。例如，对于猫和狗的区别，模型关注了猫的头部花纹特征，以及狗的头部的眼与耳朵特征。然而，在猫与狗同时存在的时候，则具有一定的偏差，神经网络不清楚需要激活的特征梯度信息，或者是其梯度较小，导致在归一化后，热力图效果较明显。我们在图 5 输出了每一通道的计算结果，可见有相当部分的通道的梯度计算为零，导致其图像绘制难以呈现出特征，而有特征输出的热力图则均聚焦于狗的头部，这也和加权得到的汇总特征相符。

我们在图 3 示例了对于三张图片分别对其属于猫或属于狗的可视化结果进行分析。如图 3 所示，上左图为两者都有且判定为猫的热力图，上中图为狗判定为狗的热力图，上右图为猫判定为猫的热力图。图 3 下左图为两者都有且判定为狗的热力图，下中图为狗判定为猫的热力图，下右图为猫判定为狗的热力图。从这些图像中可见，Grad-CAM 的效果相对较好，但仍然较为粗糙，这将在后面 LayerCAM 方法得到优化。

图 3 Grad-CAM 方法测试

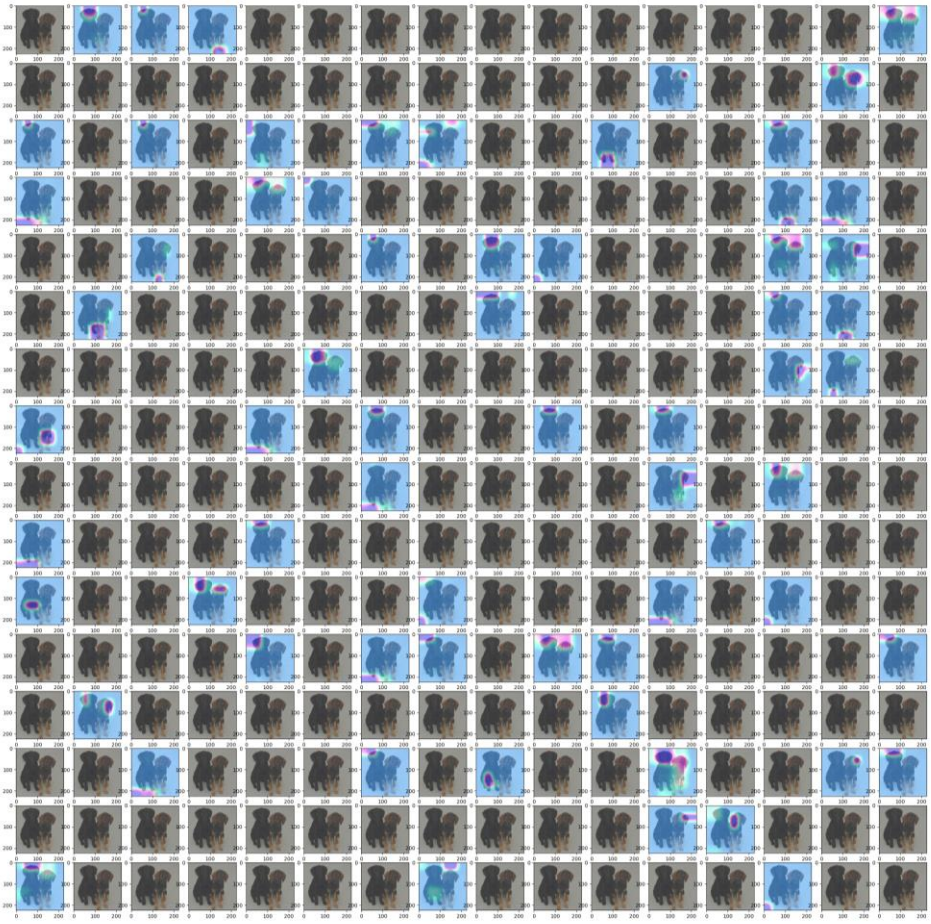

图 4 Grad-CAM 方法对于 256 通道的权重结果

对于另外两组图像的输出结果，对于猫和狗均有的多通道结果则如图 5 所示，其分析类似，因此不再赘述。更多结果详见 Results/文件夹下。

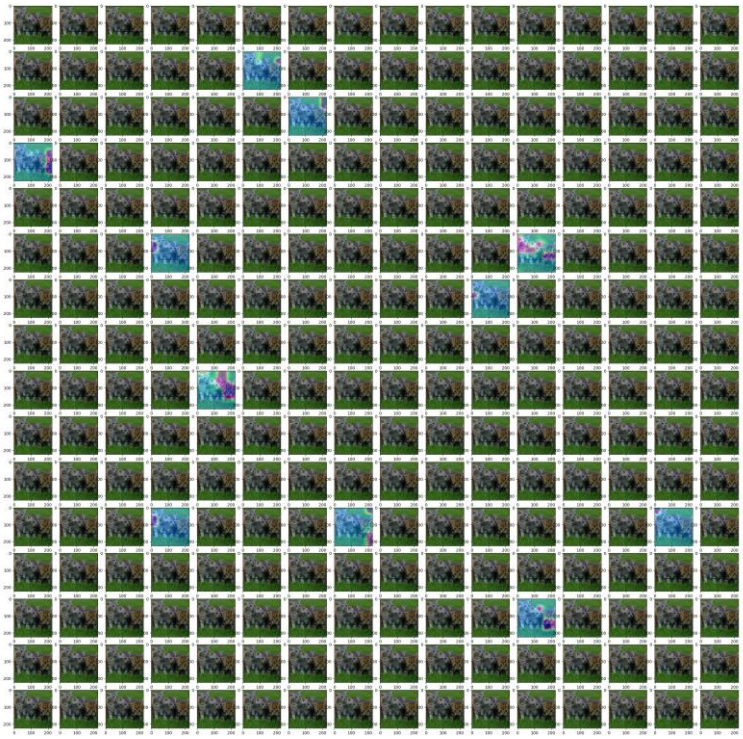

图 5 两者都有的图像多通道分类结果

对于 LayerCAM 方法，其得到的结果更加精细。如图 6 所示，其 CAM 解释性分析的粒度更细，对于猫咪的解释更侧重其面部特征，而狗的分析也关联到其头部。对于两者都有的分类图，也更侧重虎纹猫咪的手掌部分特征。这是因为 LayerCAM 不再使用梯度统一池化作为热力特征的计算输出，而是对每个元素进行激活后输出，其分类粒度更细。

我们在图 6 示例了对于三张图片分别对其属于猫或属于狗的可视化结果进行分析。如图 6 所示，上左图为狗判定为狗的热力图，上中图为猫判定为猫的热力图，上右图为两者都有且判定为猫的热力图。图 6 下左图为狗判定为猫的热力图，下中图为猫判定为狗的热力图，下右图为两者都有且判定为狗的热力图。从这些图像中可见，LayerCAM 的可视化效果相对较好，其分类粒度比 Grad-CAM 较优，例如其中的猫狗都有的场景，对于狗和猫的贡献较为精确。

图 6 LayerCAM 方法得到的热力图绘制结果

对于每个通道的结果如图 7 所示。由于篇幅问题，更多的结果详见提交材料的./Results/LayerCAM 文件夹下，此处不再赘述。

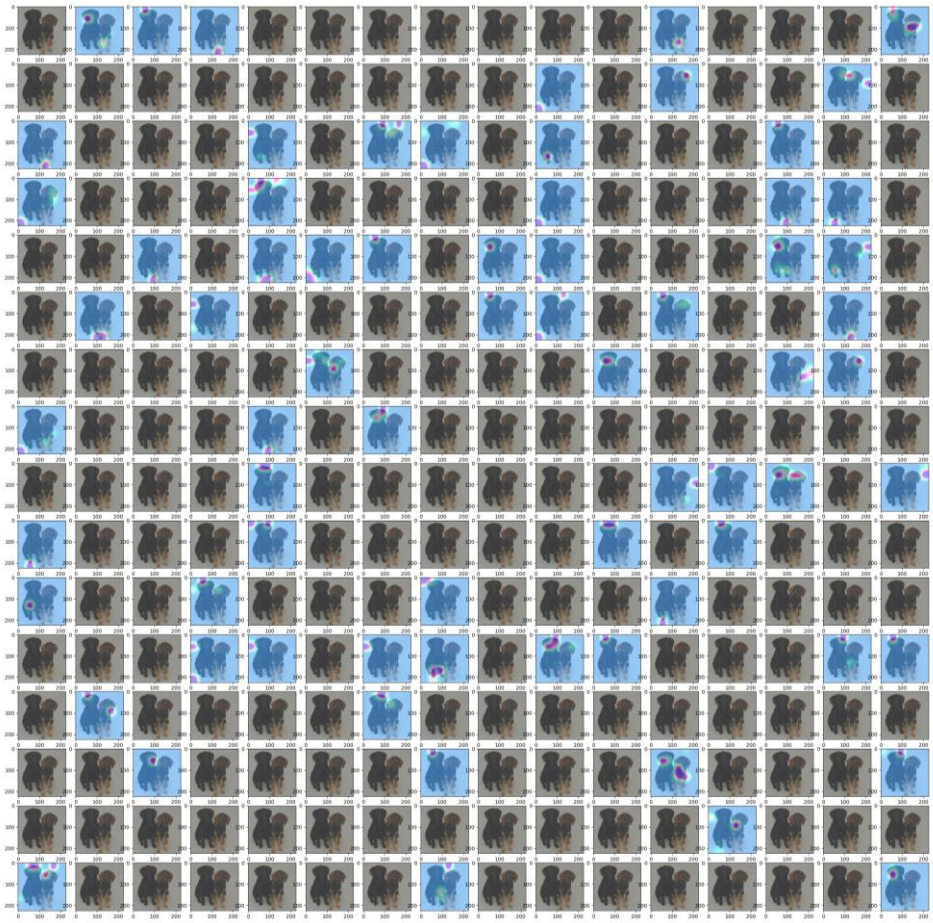

图 7 LayerCAM 方法对于狗分类的可解释性分析

1.4 讨论

我们使用 Grad-CAM 方法和 LayerCAM 方法对于基于 PyTorch 的卷积神经网络训练结果进行了可解释性分析。我们得到的可解释性类激活可视化热力图见附录./Results 文件夹下，代码文件为./test_gradcam.py 和./test_layercam.py。

我们认为这两种可解释性方法取得了较好的效果。我们对每一通道的结果进行了可视化绘制，有许多通道不具有梯度，而也有约 1/3 的通道梯度较为明显。我们对每一通道的权重进行加权求和，得到了最终的可解释性热力图。

通过对于类激活热力图，我们将黑箱的卷积神经网络可视化，可以看出对于猫或狗，神经网络具体关注哪些特征，并因为哪些特征的贡献，可能导致神经网络将其判定为不同的类别。