Grad CAM

Grad CAM stands for Gradient-Weighted Class Activation Mapping. This technique is used to produce visual explanations for decisions made by convolutional neural networks (CNNs). By calculating gradients of a target concept (e.g., a classification label) with respect to the final convolutional layer's feature maps, it generates a heat map highlighting important regions in an image.

"Which regions of the image were most important for this specific class prediction?"

Terminology

- Let y^c denote the score (logit) for class c.
- Let A^k denote the k-th feature map of the chosen convolutional layer.
- Let A^k_{ij} denote the activation at spatial location (i, j) in the k-th feature map.

Steps

Step 1: Compute gradients

Take the gradient of the class score with respect to each feature map activation:

$$\frac{\partial y^c}{\partial A^k_{ij}}$$

This measures how sensitive the class score y^c is to changes at spatial location (i, j) in feature map k.

Step 2: Global Average Pooling (importance weights)

Compute a single scalar weight for each feature map:

$$\alpha^c_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}$$

where Z is the total number of spatial locations in the feature map.

These weights α_k^c represent the importance of feature map k for class c.

Step 3: Weighted combination

Compute the class-specific localization map by linearly combining the feature maps:

$$L^c_{\text{Grad-CAM}} = \sum_k \alpha_k^c A^k$$

Step 4: ReLU

Apply a ReLU operation:

$$\text{Grad-CAM}^c = \text{ReLU}\left(L^c_{\text{Grad-CAM}}\right)$$

The ReLU retains only positive values, highlighting regions that have a positive influence on the class prediction, while discarding regions that suppress the class.