Grad-CAM: Visual Explanations using Gradient-based Localization

Contents

Explainable AI - Introduction 2

Limitations of the CAM Technique 2

Grad-CAM Overview 3

Grad-CAM Steps 4

Step 1: Compute Gradient 4

Step 2: Global Average Pool of the Gradients 5

Step 3: Compute the Final GradCAM Heatmap 6

Grad-CAM as a Generalization of CAM 7

GradCAM Visualizations 7

Further Reading 9

References 9

List of Figures

Figure 1 - GradCAM Architectural Overview 3

Figure 2- Gradient Computation w.r.t Feature Maps 5

Figure 3- Alpha Values by Averaging Gradients 6

Figure 4- GradCAM Viz 1 8

Figure 5 - GradCAM Viz 2 8

Figure 6 - GradCAM Viz 3 8

Explainable AI - Introduction

The application of AI systems in healthcare is a challenging task mainly because the factors involved in arriving at a decision by the machines are not explainable. Questions like, how did the machine arrive at this decision? or what did the machine see to predict the particular class of a condition? will always be asked to understand a machine's way of taking the decisions in healthcare.

Interpretability matters when machines take decisions on doctor's behalf. For machines to arrive at a particular medical decision, health diagnostic or the treatment course, they have to be trustable. If machine based intelligent systems have to be integrated into the healthcare systems, their decisions have to be meaningful and transparent. The transparency on the thought process of such systems will help in the following ways -

1. When AI systems are weaker in the building phase, helps in building a robust system by identifying the failure reasons.
2. When AI systems are stable and achieved a certain SOTA result, transparency then will ensure that such systems are able to establish confidence on the healthcare personnel and the patients.
3. When AI systems are powerful enough such that they are able to learn by their own experience, studying the machine's ability of decision making may help healthcare professionals make better decisions themselves.

Such transparency and explainability will only make it easier to integrate such smart intelligent systems with everyday healthcare applications and hospital workflows.

Limitations of the CAM Technique

1. CAM was bound by architectural constraints, i.e., only those architectures performing Global Average Pooling over convolutional maps immediately before final softmax layer could take advantage of the CAM visualizations.
2. The modified model needs to be retrained, which could computationally expensive when trained over the SOTA CNN architectures.

3. Since the fully connected Dense layers are replaced the performance of the model can suffer and the prediction score may the actual picture of the model's ability to classify images.

To overcome the above limitations, (Selvaraju *et al.*, 2016) came up with a more generalized approach by utilizing the gradients of the predicted class flowing into the final convolutional layer to produce a rough localization map, highlighting the important regions in the image for predicting the class of the image.

Grad-CAM Overview

One such system to provide visual explanations for the decisions AI systems and basically those involving convolutional neural networks was proposed by (Selvaraju *et al.*, 2016). The complexity involved in the neural network models makes it hard to interpret the rationality behind their decisions.

According to (Selvaraju *et al.*, 2016), a good visual explanation should be able to localize the object of interest by capturing the minutest of details in the image. The Grad-CAM technique could explain a CNN's decision making process answering, *"why they predict what they predict"*. It takes into consideration the gradients of the target object flowing into the final convolutional layer to create a rough localization mapping and highlighting the important regions of the target image the mode considered to assign the particular class. Figure 1 shows the architectural overview of GradCAM technique. Image Source - (Selvaraju *et al.*, 2016)

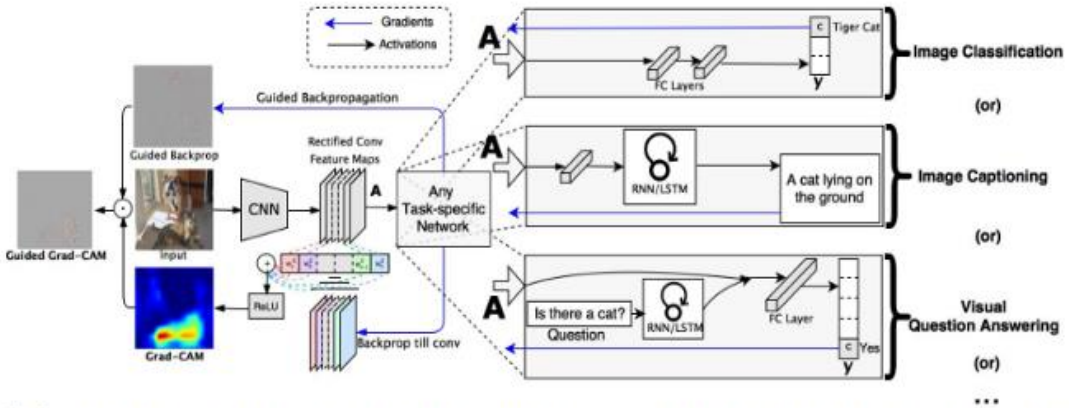

Fig. 2: Grad-CAM overview: Given an image and a class of interest (*e.g.*, 'tiger cat' or any other type of differentiable output) as input, we forward propagate the image through the CNN part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to zero for all classes except the desired class (tiger cat), which is set to 1. This signal is then backpropagated to the rectified convolutional feature maps of interest, which we combine to compute the coarse Grad-CAM localization (blue heatmap) which represents where the model has to look to make the particular decision. Finally, we pointwise multiply the heatmap with guided backpropagation to get Guided Grad-CAM visualizations which are both high-resolution and concept-specific.

Figure 1 - GradCAM Architectural Overview

Grad-CAM relies on convolutional layer as they tend to *retain the spatial information*. It utilizes the gradient activations coming into the last convolutional layer as these layer look for more class specific features. The gradient score of the concerned class is calculated with respect to the feature map activation of the convolution layer given by the formula -

$$\propto_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

Equation 1 - Gradient Activation Score

The weight $\propto_k^c$ represents a partial linearization of the deep network downstream from **A** (see Figure 1 - GradCAM Architectural Overview), and captures the *'importance' of feature map k for a target class c*. Once the gradient scores are calculated, ReLU is applied to the weighted linear combination of the forward activation maps to consider only those features that have a positive impact on the class of interest.

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \propto_k^c A^k\right)$$

Equation 2 - ReLU on Gradient Activation Score

Without ReLU, the system does not localize well enough when it could highlight more than just the required region of interest. The resultant heatmap from Equation 2 is of the same size as that of the feature map coming out of the convolutional layer in context.

Grad-CAM Steps

Viewing the gradient class activations involves 3 steps involving computing the gradients and weighing the feature maps using the alpha values computed using the Equation 1 and Equation 2 defined above. The various steps involved are -

Step 1: Compute Gradient

Calculate the gradient as mentioned and quoted from the original paper -

"compute the gradient of the score for class c, y_c (before the softmax), with respect to feature map activations A^k of a convolutional layer, i.e., $\frac{\partial y^c}{\partial A^k}$".

The gradient computed have the same shape as that of the feature maps. If the convolutional layer produces *K* feature maps each of height - '*v*' & width - '*u*', the collective shape of all the

feature maps would be [K, *v*, *u*]. Then, all the gradients calculated in this step will also be of shape [*K*, *v*, *u*] collectively. Figure 2 below shows the pictorial representation of the extracted feature maps and the gradients computed out of those feature maps w.r.t the convolutional layer. The feature map of the computed class can be represented as -

$$y^c = f(A^1, A^2, \dots, A^K)$$

The visualization of the final feature map - A^k shows the region of interest in the image.

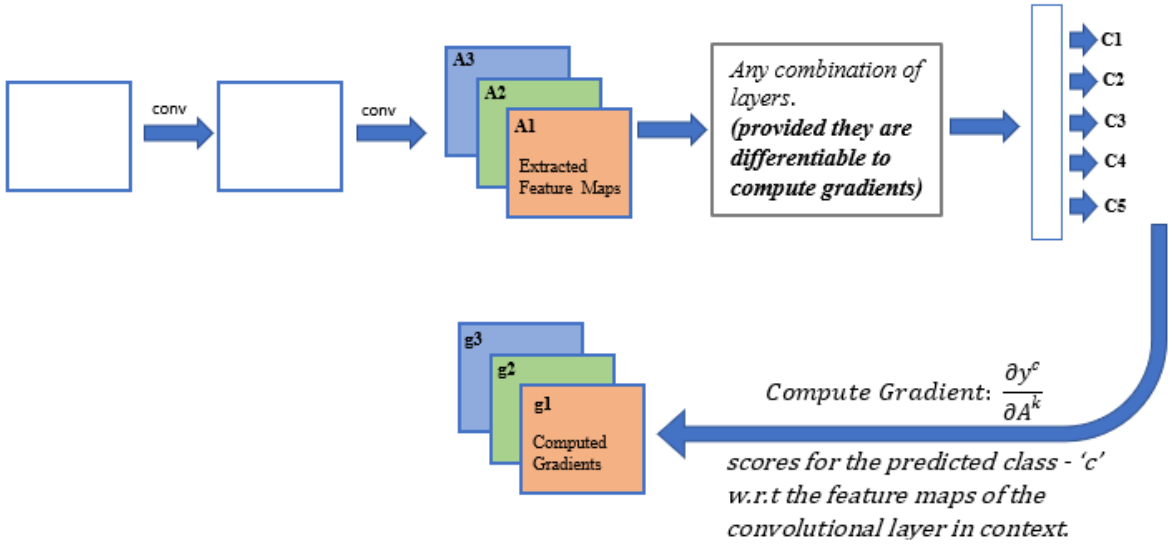

Figure 2- Gradient Computation w.r.t Feature Maps

"In general, y^c need not be the class score produced by an image classification CNN. It could be any differentiable activation ..."

Step 2: Global Average Pool of the Gradients

This step computes the alpha value for the predicted class which act as the weight applied to the feature map A^k. As seen in step 1, the gradients have a shape of [*K*, *v*, *u*]. These gradients are average pooled over the height - 'v' and width - 'u' which finally reduces to the shape [K, 1,1] which is just having K alpha values.

The gradient score of the class 'c' with respect to feature maps A^k is computed using Equation 1 - Gradient Activation Score defined above. This score measures linear effect of (i, j)th pixel in the kth feature map on the predicted class - 'c' score.

"These gradients flowing back are global-average-pooled over the width and height dimensions (indexed by i and j respectively) to obtain the neuron importance weights $\propto_k^c$ "

$$\propto_k^c = \frac{1}{Z}\sum_{i=1}^{u}\sum_{j=1}^{v}\frac{\partial y^c}{\partial A_{ij}^k}$$

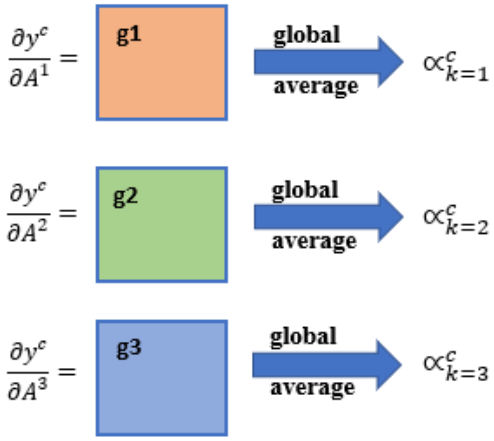

Figure 3- Alpha Values by Averaging Gradients

Step 3: Compute the Final GradCAM Heatmap

Once the feature importance are computed, a weighted combination of the activation maps are performed and followed by a ReLU operation to obtain the final heatmap.

"perform a weighted combination of forward activation maps, and follow it by a ReLU to obtain, "

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \propto_k^c A^k\right) \in \mathbb{R}^{u\,x\,v}$$

A ReLU is applied since we are only concerned only about those features that have a positive effect in deciding the predicted class of the image/object. Without the ReLU operation, the activation maps may highlight regions in the image that are irrelevant to the predicted class and hence the perform bad at localization.

Now, since the $L^c_{Grad-CAM}$ heatmap obtained would be of the dimension same as that of the feature map obtained from the convolutional layer i.e., [u x v]. For final visualization purpose, this would be too small a shape compared to that of the input image. To perfectly blend the heatmap with the input image, the heatmap is up-sampled to match the dimensions of the image.

Grad-CAM as a Generalization of CAM

Grad-CAM does not require any specific CNN architecture for interpretability. It is a generalized technique that can work with variety of architectures and problem set. Grad-CAM is a generalization of CAM - Class Activation Map, a technique that is limited to a particular architecture -

"*performing global average pooling over convolutional maps immediately prior to the final softmax layer performing predictions*.

(i.e., ***conv feature maps → global average pooling → softmax layer***)".

The CAM technique computes the final scores as per the below equation -

$$Y^c = \sum_k w_c^k \, . \, F^k$$

where,

w_c^k is the weight connecting the kth feature map with the cth class.

CAM Paper Ref - (Zhou *et al.*, 2015)

Applying a sequence of partial derivative and summation operation, it eventually turns out that the term w_c^k in CAM is identical to the term $\propto_k^c$ in GradCAM. Thus, Grad-CAM is a strict generalization of CAM.

For a detailed derivation of the generalization refer to the section 3.1 in the original paper.

GradCAM Visualizations

A few GradCAM viz from the experiment for visualizing the classification of skin cancer tumors as *benign* or *malignant* are shown in Figure 4, Figure 5 and Figure 6.

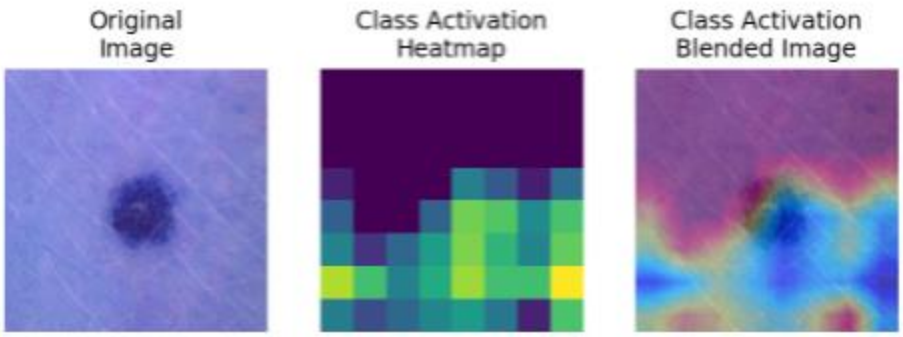

Figure 4- GradCAM Viz 1

Figure 5 - GradCAM Viz 2

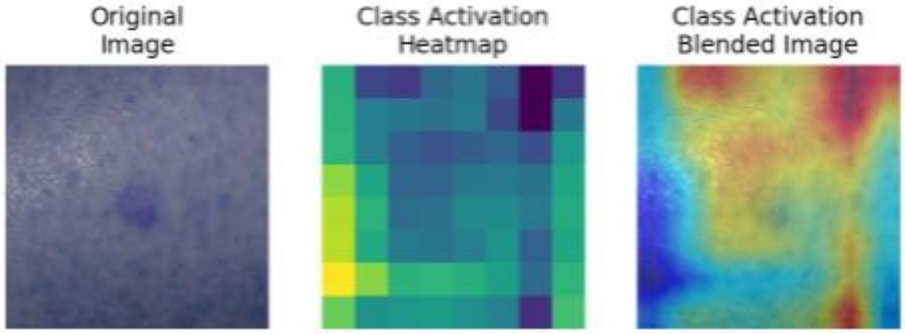

Figure 6 - GradCAM Viz 3

The visualizations seen in the images above, show the original image and the up-sampled heatmap. The blended image shows that the heatmap coincides with the location of the tumor while having good amount of intensity at the tumor location.

However, it can also be seen that the Grad-CAM heatmaps may not always .capture the object of interest in its entirety.

Further Reading

1. Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks.
   Available at - https://arxiv.org/abs/1710.11063
2. Tell Me Where to Look: Guided Attention Inference Network.
   Available at - https://arxiv.org/abs/1802.10171

References

Selvaraju, R. R. *et al.* (2016) 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization'. doi: 10.1007/s11263-019-01228-7.

Zhou, B. *et al.* (2015) 'Learning Deep Features for Discriminative Localization'. Available at: https://arxiv.org/abs/1512.04150 (Accessed: 3 June 2021).

Interpretability in Deep Learning - https://towardsdatascience.com/interpretability-in-deep-learning-with-w-b-cam-and-gradcam-45ba5296a58a