SmoothGrad: removing noise by adding noise

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, Martin Wattenberg

arXiv: https://arxiv.org/abs/1706.03825

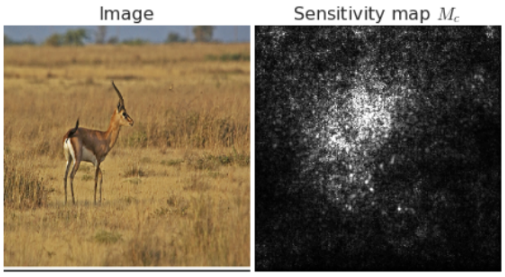
This paper describes a very simple technique, **SmoothGrad** that in practice tends to reduce visual noise, and also can be combined with other sensitivity map algorithms. The core idea is to take an image of interest, sample similar images by adding noise to the image, then take the average of the resulting sensitivity maps for each sampled image. Additionally, adding noise at training time has an additional "de-noising" effect on sensitivity maps. The two techniques (training with noise, and inferring with noise) seem to have additive effects; performing them together yields the best results.

Preliminaries:

Sensitivity maps

Given an input image x, an image classification network computes a class activation function S_c for each class $c \in C$, and the final classification $class(x)$ is determined by which class has the highest score. That is, $\text{class}(x) = \text{argmax}_{c \in C}\, S_c(x)$

If the functions S_c are piecewise differentiable, for any image x one can construct a sensitivity map $M_c(x)$ simply by differentiating M_c with respect to the input, x: $M_c(x) = \partial S_c(x)/\partial x$.



However, the sensitivity maps based on raw gradients are typically visually noisy, Moreover, the correlations with regions a human would pick out as meaningful are rough at best.

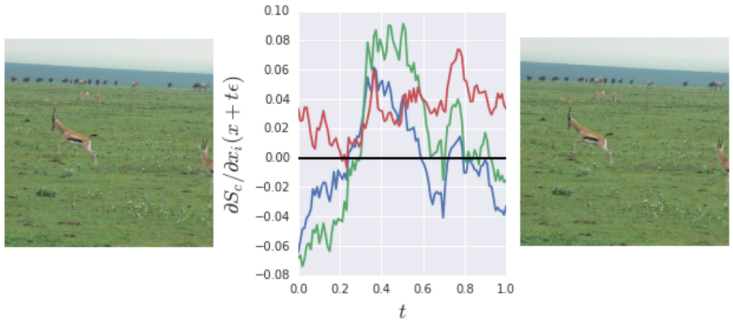
Enhancing Sensitivity maps

One issue with using the gradient as a measure of influence is that an important feature may "saturate" the function S_c. In other words, it may have a strong effect globally, but with a small derivative locally. Several approaches, Layerwise Relevance Propagation, DeepLift, and Integrated Gradients, attempt to address this potential problem by estimating the global importance of each pixel, rather than local sensitivity. Maps created with these techniques are referred to as "saliency" or "pixel attribution" maps.

Another strategy for enhancing sensitivity maps has been to change or extend the backpropagation algorithm itself,with the goal of emphasizing positive contributions to the final outcome. Two examples are the Deconvolution and Guided Backpropagation techniques, which modify the gradients of ReLU functions by discarding negative values during the backpropagation calculation.

Source of noise in sensitivity maps

There is a possible explanation for the noise in sensitivity maps: the derivative of the function S_c may fluctuate sharply at small scales. In other words, the apparent noise one sees in a sensitivity map may be due to essentially meaningless local variations in partial derivatives. To illustrate this, for particular image x, and an image pixel x_i, we plot the values of $\frac{\partial S_c}{\partial x_i}(t)$ as fraction of the maximum entry in the gradient vector, $\max i \frac{\partial Sc}{\partial x_i}(t)$, for a short-line segment $x + t\epsilon$ in the space of images parameterized by $t \in [0, 1]$.



The length of this segment is small enough that the starting image x and the final image $x + \epsilon$ looks the same to a human. Furthermore, each image along the path is correctly classified by the model. The partial derivatives with respect to the red, green, and blue components, however, change significantly.

Given these rapid fluctuations, the gradient of S_c at any given point will be less meaningful than a local average of gradient values. This suggests a new way to create improved sensitivity maps: instead of basing a visualization directly on the gradient ∂Sc, we could base it on a smoothing of ∂Sc with a Gaussian kernel. Mathematically, this means calculating

$$\hat{M}c(x) = \frac{1}{n} \sum 1^n M_c \left(x + \mathcal{N}\left(0, \sigma^2\right)\right)$$

Results:

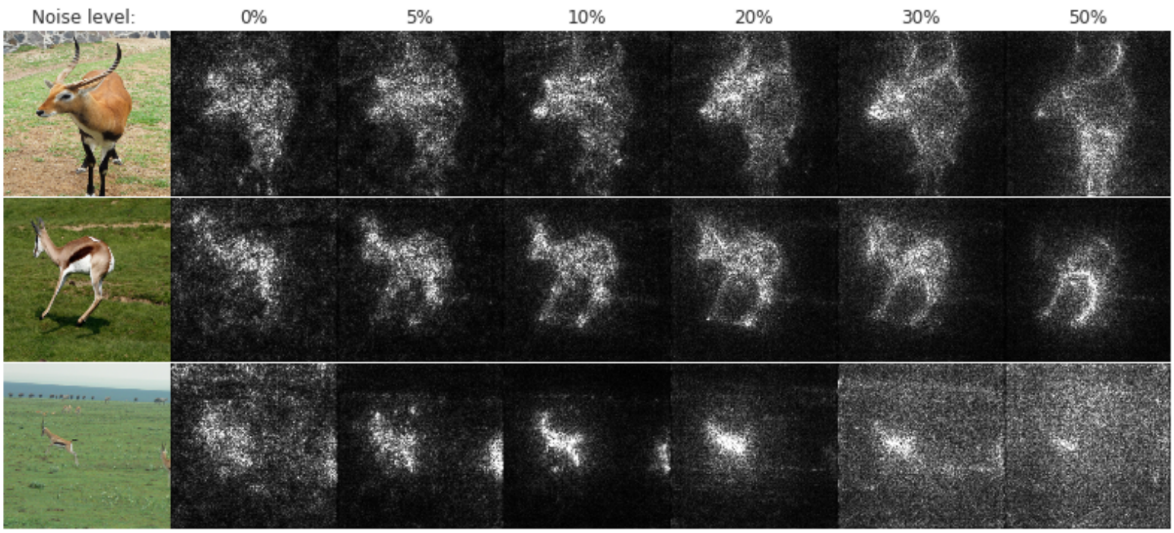


Figure 3. Effect of noise level (columns) on our method for 5 images of the gazelle class in ImageNet (rows). Each sensitivity map is obtained by applying Gaussian noise $\mathcal{N}(0,\, \sigma^2)$ to the input pixels for 50 samples, and averaging them. The noise level corresponds to $\sigma/(x_{max} - x_{min})$.

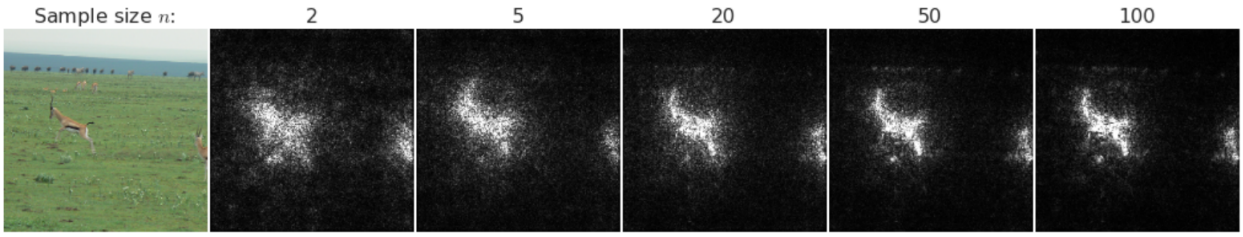
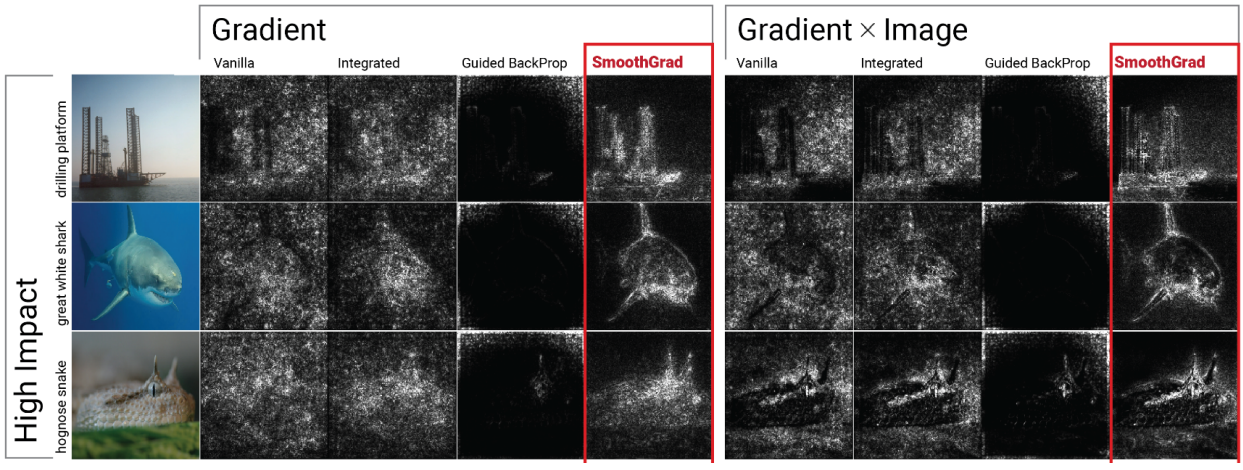


Figure 4. Effect of sample size on the estimated gradient for inception. 10% noise was applied to each image.



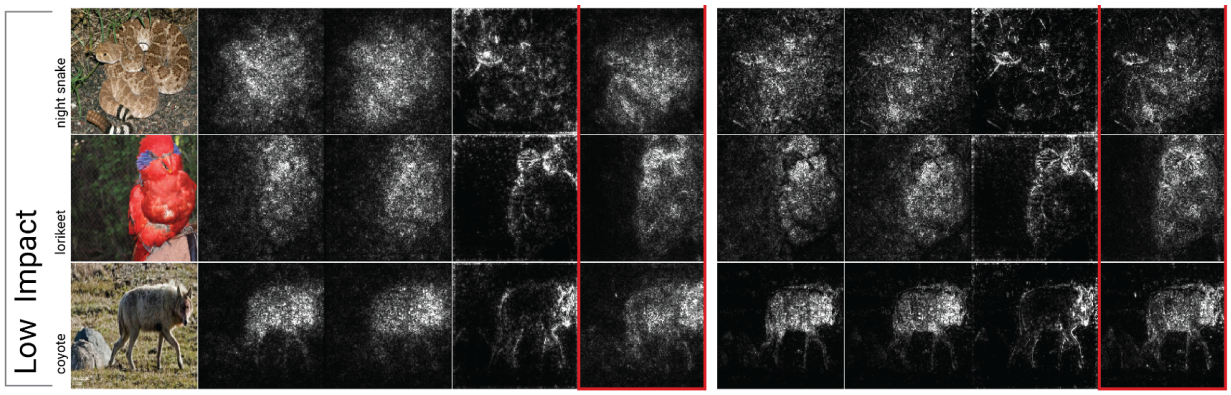


Figure 5. Qualitative evaluation of different methods. First three (last three) rows show examples where applying SMOOTHGRAD had high (low) impact on the quality of sensitivity map.

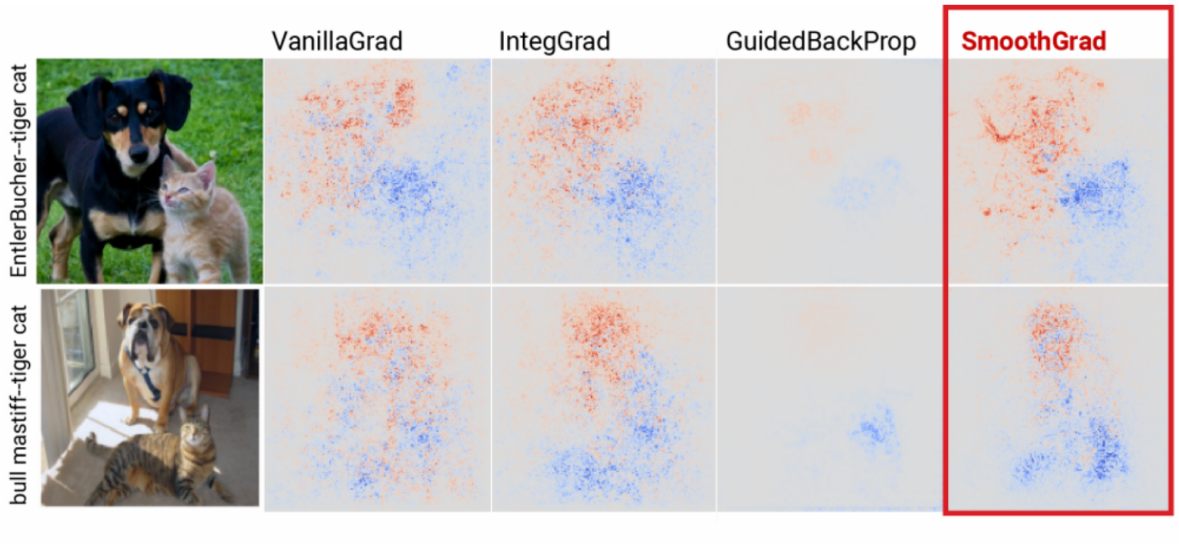


Figure 6. Discriminativity of different methods. For each image, we visualize the difference scale($\partial y_1/\partial x$) − scale($\partial y_2/\partial x$) where y_1 and y_2 are the logits for the first and the second class (i.e., cat or dog) and scale() normalizes the gradient values to be between $[0, 1]$. The values are plotted using a diverging color map $[-1, 0, 1] \mapsto$ [blue, gray, red]. Each method is represented in columns.