
SmoothGrad: removing noise by adding noise

Daniel Smilkov¹ Nikhil Thorat¹ Been Kim¹ Fernanda Viégas¹ Martin Wattenberg¹

Abstract

Explaining the output of a deep network remains a challenge. In the case of an image classifier, one type of explanation is to identify pixels that strongly influence the final decision. A starting point for this strategy is the gradient of the class score function with respect to the input image. This gradient can be interpreted as a *sensitivity map*, and there are several techniques that elaborate on this basic idea. This paper makes two contributions: it introduces SMOOTHGRAD, a simple method that can help visually sharpen gradient-based sensitivity maps, and it discusses lessons in the visualization of these maps. We publish the code for our experiments and a website with our results.

1. Introduction

Interpreting complex machine learning models, such as deep neural networks, remains a challenge. Yet an understanding of how such models function is important both for building applications and as a problem in its own right. From health care domains (Hughes et al., 2016; Doshi-Velez et al., 2014; Lou et al., 2012) to education (Kim et al., 2015), there are many domains where interpretability is important. For example, the pneumonia risk prediction case study in (Lou et al., 2012) showed that more interpretable models can reveal important but surprising patterns in the data that complex models overlooked. For reviews of interpretable models, see (Freitas, 2014; Doshi-Velez, 2017).

One case of interest is image classification systems. Finding an “explanation” for a classification decision could potentially shed light on the underlying mechanisms of such systems, as well as helping in enhancing them. For example, the technique of deconvolution helped researchers identify neurons that failed to learn any meaningful features, knowledge that was used to improve the network, as in (Zeiler & Fergus, 2014).

¹Google Inc.. Correspondence to: Daniel Smilkov <smilkov@google.com>.

A common approach to understanding the decisions of image classification systems is to find regions of an image that were particularly influential to the final classification. (Baehrens et al., 2010; Zeiler & Fergus, 2014; Springenberg et al., 2014; Zhou et al., 2016; Selvaraju et al., 2016; Sundararajan et al., 2017; Zintgraf et al., 2016). These approaches (variously called sensitivity maps, saliency maps, or pixel attribution maps; see discussion in Section 2; use occlusion techniques or calculations with gradients to assign an “importance” value to individual pixels which is meant to reflect their influence on the final classification.

In practice these techniques often do seem to highlight regions that can be meaningful to humans, such as the eyes in a face. At the same time, sensitivity maps are often visually noisy, highlighting some pixels that—to a human eye—seem randomly selected. Of course, a priori we cannot determine if this noise reflects an underlying truth about how networks perform classification, or is due to more superficial factors. Either way, it seems like a phenomenon worth investigating further.

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. We also find that the common regularization technique of adding noise at training time (Bishop, 1995) has an additional “de-noising” effect on sensitivity maps. The two techniques (training with noise, and inferring with noise) seem to have additive effect; performing them together yields the best results.

This paper compares the SMOOTHGRAD method to several gradient-based sensitivity map methods and demonstrates its effects. We provide a conjecture, backed by some empirical evidence, for why the technique works, and why it might be more reflective of how the network is doing classification. We also discuss several ways to enhance visualizations of these sensitivity maps. Finally, we also make the code used to generate all the figures in this paper available, along with 200+ examples of each compared method on the web at <https://goo.gl/EfVzEE>.

2. Gradients as sensitivity maps

Consider a system that classifies an image into one class from a set C . Given an input image x , many image classification networks (Szegedy et al., 2016; LeCun et al., 1998) compute 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,

$$class(x) = \operatorname{argmax}_{c \in C} S_c(x)$$

A mathematically clean way of locating “important” pixels in the input image has been proposed by several authors, e.g., (Baehrens et al., 2010; Simonyan et al., 2013; Erhan et al., 2009). 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 . In particular, we can define

$$M_c(x) = \partial S_c(x) / \partial x$$

Here ∂S_c represents the derivative (i.e. gradient) of S_c . Intuitively speaking, M_c represents how much difference a tiny change in each pixel of x would make to the classification score for class c . As a result, one might hope that the resulting map M_c would highlight key regions.

In practice, the sensitivity map of a label does seem to show a correlation with regions where that label is present (Baehrens et al., 2010; Simonyan et al., 2013). However, the sensitivity maps based on raw gradients are typically visually noisy, as shown in Fig. 1. Moreover, as this image shows, the correlations with regions a human would pick out as meaningful are rough at best.

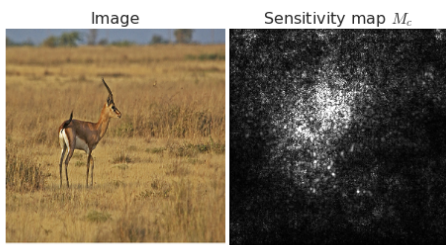


Figure 1. A noisy sensitivity map, based on the gradient of the class score for gazelle for an image classification network. Lighter pixels indicate partial derivatives with higher absolute values. See Section 3 for details on the visualization.

2.1. Previous work on enhancing sensitivity maps

There are several hypotheses for the apparent noise in raw gradient visualizations. One possibility, of course, is that

the maps are faithful descriptions of what the network is doing. Perhaps certain pixels scattered, seemingly at random, across the image are central to how the network is making a decision. On the other hand, it is also possible that using the raw gradient as a proxy for feature importance is not optimal. Seeking better explanations of network decisions, several prior works have proposed modifications to the basic technique of gradient sensitivity maps; we summarize a few key examples here.

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* (Bach et al., 2015), *DeepLift* (Shrikumar et al., 2017), and more recently *Integrated Gradients* (Sundararajan et al., 2017), 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* (Zeiler & Fergus, 2014) and *Guided Backpropagation* (Springenberg et al., 2014) techniques, which modify the gradients of ReLU functions by discarding negative values during the backpropagation calculation. The intention is to perform a type of “deconvolution” which will more clearly show features that triggered activations of high-level units. Similar ideas appear in (Selvaraju et al., 2016; Zhou et al., 2016), which suggest ways to combine gradients of units at multiple levels.

In what follows, we provide detailed comparisons of “vanilla” gradient maps with those created by integrated gradient methods and guided backpropagation. A note on terminology: although the terms “sensitivity map”, “saliency map”, and “pixel attribution map” have been used in different contexts, in this paper, we will refer to these methods collectively as “sensitivity maps.”

2.2. Smoothing noisy gradients

There is a possible explanation for the noise in sensitivity maps, which to our knowledge has not been directly addressed in the literature: 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. After all, given typical training techniques there is no reason to expect derivatives to vary smoothly. Indeed, the networks in question typically are based on ReLU activation functions, so S_c generally will not even be continu-

ously differentiable.

Fig. 2 gives example of strongly fluctuating partial derivatives. This fixes a particular image x , and an image pixel x_i , and plots the values of $\frac{\partial S_c}{\partial x_i}(t)$ as fraction of the maximum entry in the gradient vector, $\max_i \frac{\partial S_c}{\partial x_i}(t)$, for a short line segment $x + t\epsilon$ in the space of images parameterized by $t \in [0, 1]$. We show it as a fraction of the maximum entry in order to verify that the fluctuations are significant. 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.

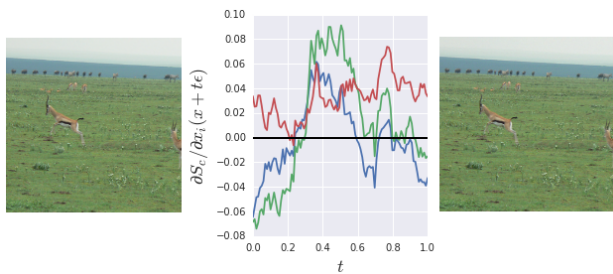


Figure 2. The partial derivative of S_c with respect to the RGB values of a single pixel as a fraction of the maximum entry in the gradient vector, $\max_i \frac{\partial S_c}{\partial x_i}(t)$, (middle plot) as one slowly moves away from a baseline image x (left plot) to a fixed location $x + \epsilon$ (right plot). ϵ is one random sample from $\mathcal{N}(0, 0.01^2)$. The final image ($x + \epsilon$) is indistinguishable to a human from the origin image x .

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 ∂S_c , we could base it on a smoothing of ∂S_c with a Gaussian kernel.

Directly computing such a local average in a high-dimensional input space is intractable, but we can compute a simple stochastic approximation. In particular, we can take random samples in a neighborhood of an input x , and average the resulting sensitivity maps. Mathematically, this means calculating

$$\hat{M}_c(x) = \frac{1}{n} \sum_1^n M_c(x + \mathcal{N}(0, \sigma^2))$$

where n is the number of samples, and $\mathcal{N}(0, \sigma^2)$ represents Gaussian noise with standard deviation σ . We refer to this method as SMOOTHGRAD throughout the paper.

3. Experiments

To assess the SMOOTHGRAD technique, we performed a series of experiments using a neural network for image classification (Szegedy et al., 2016; TensorFlow, 2017). The results suggest the estimated smoothed gradient, \hat{M}_c , leads to visually more coherent sensitivity maps than the unsmoothed gradient M_c , with the resulting visualizations aligning better—to the human eye—with meaningful features.

Our experiments were carried out using an Inception v3 model (Szegedy et al., 2016) that was trained on the ILSVRC-2013 dataset (Russakovsky et al., 2015) and a convolutional MNIST model based on the TensorFlow tutorial (TensorFlow, 2017).

3.1. Visualization methods and techniques

Sensitivity maps are typically visualized as heatmaps. Finding the right mapping from a channel values at a pixel to a particular color turns out to be surprisingly nuanced, and can have a large effect on the resulting impression of the visualization. This section summarizes some visualization techniques and lessons learned in the process of comparing various sensitivity map work. Some of these techniques may be universally useful regardless of the choice of sensitivity map methods.

Absolute value of gradients

Sensitivity map algorithms often produce signed values. There is considerable ambiguity in how to convert signed values to colors. A key choice is whether to represent positive and negative values differently, or to visualize the absolute value only. The utility of taking the absolute values of gradients or not depends on the characteristics of the dataset of interest. For example, when the object of interest has the same color across the classes (e.g., digits are always white in MNIST digits (LeCun et al., 2010)), the positive gradients indicate positive signal to the class. On the other hand, for ImageNet dataset (Russakovsky et al., 2015), we have found that taking the absolute value of the gradient produced clearer pictures. One possible explanation for this phenomenon is that the direction is context dependent: many image recognition tasks are invariant under color and illumination changes. For instance, in classifying a ball, a dark ball on a bright background would have negative gradient, while white ball on darker background would have a positive gradient.

Capping outlying values

Another property of the gradient that we observe is the presence of few pixels that have much higher gradients than the average. This is not a new discovery — this property was utilized in generating adversarial examples that are in-

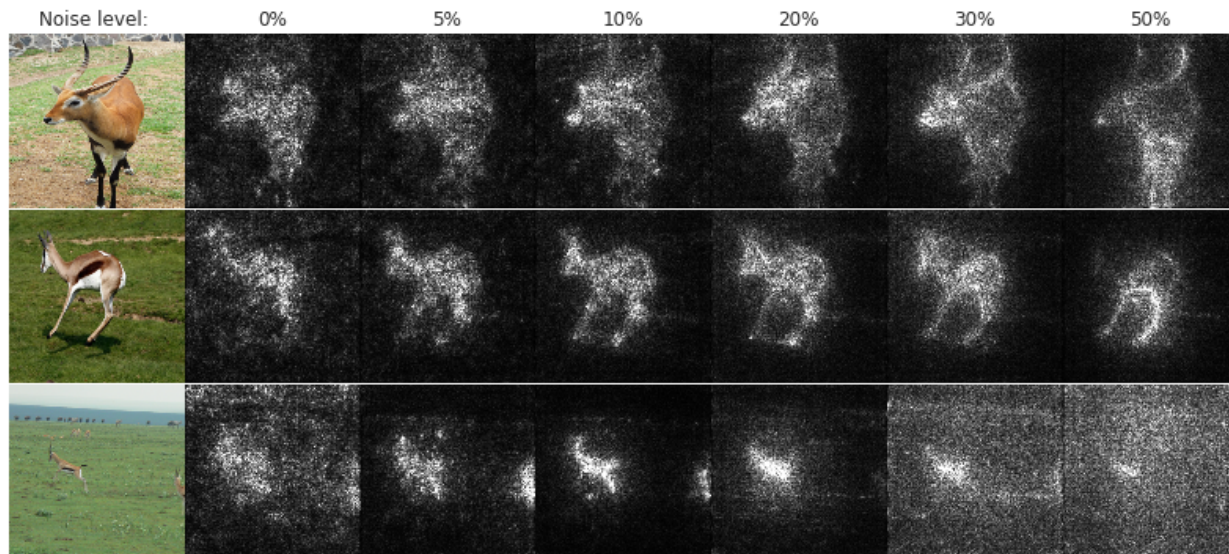


Figure 3. Effect of noise level (columns) on our method for 5 images of the gazelle class in ImageNet (rows). Each sensitivity map is obtained by applying Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the input pixels for 50 samples, and averaging them. The noise level corresponds to $\sigma/(x_{max}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

distinguishable to humans (Szegedy et al., 2013). These outlying values have the potential to throw off color scales completely. Capping those extreme values to a relatively high value (we find 99th percentile to be sufficient) leads to more visually coherent maps as in (Sundararajan et al., 2017). Without this post-processing step, maps may end up almost entirely black.

Multiplying maps with the input images

Some techniques create a final sensitivity map by multiplying gradient-based values and actual pixel values (Shrikumar et al., 2017; Sundararajan et al., 2017). This multiplication does tend to produce visually simpler and sharper images, although it can be unclear how much of this can be attributed to sharpness in the original image itself. For example, a black/white edge in the input can lead to an edge-like structure on the final visualization even if the underlying sensitivity map has no edges.

However, this may result in undesired side effect. Pixels with values of 0 will never show up on the sensitivity map. For example, if we encode black as 0, the image of a classifier that correctly predicts a black ball on a white background will never highlight the black ball in the image.

On the other hand, multiplying gradients with the input images makes sense when we view the importance of the feature as their contribution to the total score, y . For example, in a linear system $y = Wx$, it makes sense to consider $x_i w_i$

as the contribution of x_i to the final score y .

For these reasons, we show our results with and without the image multiplication in Fig. 5.

3.2. Effect of noise level and sample size

SMOOTHGRAD has two hyper-parameters: σ , the noise level or standard deviation of the Gaussian perturbations, and n , the number of samples to average over.

Noise, σ

Fig. 3 shows the effect of noise level for several example images from ImageNet (Russakovsky et al., 2015). The 2nd column corresponds to the standard gradient (0% noise), which we will refer to as the “Vanilla” method throughout the paper. Since quantitative evaluation of a map remains an unsolved problem, we again focus on qualitative evaluation. We observe that applying 10%-20% noise (middle columns) seems to balance the sharpness of sensitivity map and maintain the structure of the original image. We also observe that while this range of noise gives generally good results for Inception, the ideal noise level depends on the input. See Fig. 10 for a similar experiment on the MNIST dataset.

Sample size, n

In Fig. 4 we show the effect of sample size, n . As expected, the estimated gradient becomes smoother as the

sample size, n , increases. We empirically found a diminishing return — there was little apparent change in the visualizations for $n > 50$.

3.3. Qualitative comparison to baseline methods

Since there is no ground truth to allow for quantitative evaluation of sensitivity maps, we follow prior work (Simonyan et al., 2013; Zeiler & Fergus, 2014; Springenberg et al., 2014; Selvaraju et al., 2016; Sundararajan et al., 2017) and focus on two aspects of qualitative evaluation.

First, we inspect visual coherence (e.g., the highlights are only on the object of interest, not the background). Second, we test for discriminativity, where in an image with both a monkey and a spoon, one would expect an explanation for a monkey classification to be concentrated on the monkey rather than the spoon, and vice versa.

Regarding visual coherence, Fig. 5 shows a side-by-side comparison between our method and three gradient-based methods: *Integrated Gradients* (Sundararajan et al., 2017), *Guided BackProp* (Springenberg et al., 2014) and vanilla gradient. Among a random sample of 200 images that we inspected, we found SMOOTHGRAD to consistently provide more visually coherent maps than *Integrated Gradients* and vanilla gradient. While *Guided BackProp* provides the most sharp maps (last three rows of Fig. 5), it is prone to failure (first three rows of Fig. 5), especially for images with uniform background. On the contrary, our observation is that SMOOTHGRAD has the highest impact when the object is surrounded with uniform background color (first three rows of Fig. 5). Exploring this difference is an interesting area for investigation. It is possible that the smoothness of the class score function may be related to spatial statistics of the underlying image; noise may have a differential effect on the sensitivity to different textures.

Fig. 6 compares the discriminativity of our method to other methods. Each image has at least two objects of different classes that the network may recognize. To visually show discriminativity, we compute the sensitivity maps $M_1(x)$ and $M_2(x)$ for both classes, scale both to $[0, 1]$, and calculate the difference $M_1(x) - M_2(x)$. We then plot the values on a diverging color map $[-1, 0, 1] \mapsto [\text{blue}, \text{gray}, \text{red}]$. For these images, SMOOTHGRAD qualitatively shows better discriminativity over the other methods. It remains an open question to understand which properties affect the discriminativity of a given method – e.g. understanding why *Guided BackProp* seems to show the weakest discriminativity.

3.4. Combining SmoothGrad with other methods

One can think of SMOOTHGRAD as smoothing the vanilla gradient method using a simple procedure: averaging the

vanilla sensitivity maps of n noisy images. With that in mind, the same smoothing procedure can be used to augment any gradient-based method. In Fig. 7 we show the results of applying SMOOTHGRAD in combination with *Integrated Gradients* and *Guided BackProp*. We observe that this augmentation improves the visual coherence of sensitivity maps for both methods.

For further analysis, we point the reader to our web page at <https://goo.gl/EfVzEE> with sensitivity maps of 200+ images and four different methods.

3.5. Adding noise during training

SMOOTHGRAD as discussed so far may be applied to classification networks as-is. In situations where there is a premium on legibility, however, it is natural to ask whether there is a similar way to modify the network weights so that its sensitivity maps are sharper. One idea that is parallel in some ways to SMOOTHGRAD is the well-known regularization technique of adding noise to samples during training (Bishop, 1995). We find that the same method also improves the sharpness of the sensitivity map.

Fig. 8 and Fig. 9 show the effect of adding noise at training time and/or evaluation time for the MNIST and Inception model respectively. Interestingly, adding noise at training time seems to also provide a de-noising effect on the sensitivity map. Lastly, the two techniques (training with noise, and inferring with noise) seem to have additive effect; performing them together produces the most visually coherent map of the 4 combinations.

4. Conclusion and future work

The experiments described here suggest that gradient-based sensitivity maps can be sharpened by two forms of smoothing. First, averaging maps made from many small perturbations of a given image seems to have a significant smoothing effect. Second, that effect can be enhanced further by training on data that has been perturbed with random noise.

These results suggest several avenues for future research. First, while we have provided a plausibility argument for our conjecture that noisy sensitivity maps are due to noisy gradients, it would be worthwhile to look for further evidence and theoretical arguments that support or disconfirm this hypothesis. It is certainly possible that the sharpening effect of SMOOTHGRAD has other causes, such as a differential effect of random noise on different textures.

Second, in addition to training with noise, there may be more direct methods to create systems with smoother class score functions. For example, one could train with an explicit penalty on the size of partial derivatives. To create

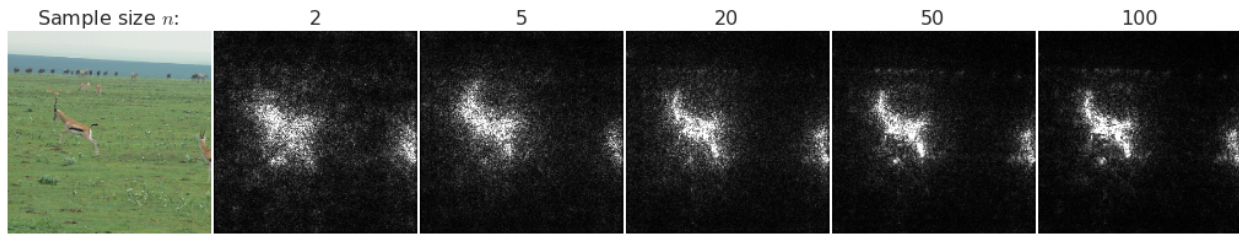


Figure 4. Effect of sample size on the estimated gradient for inception. 10% noise was applied to each image.

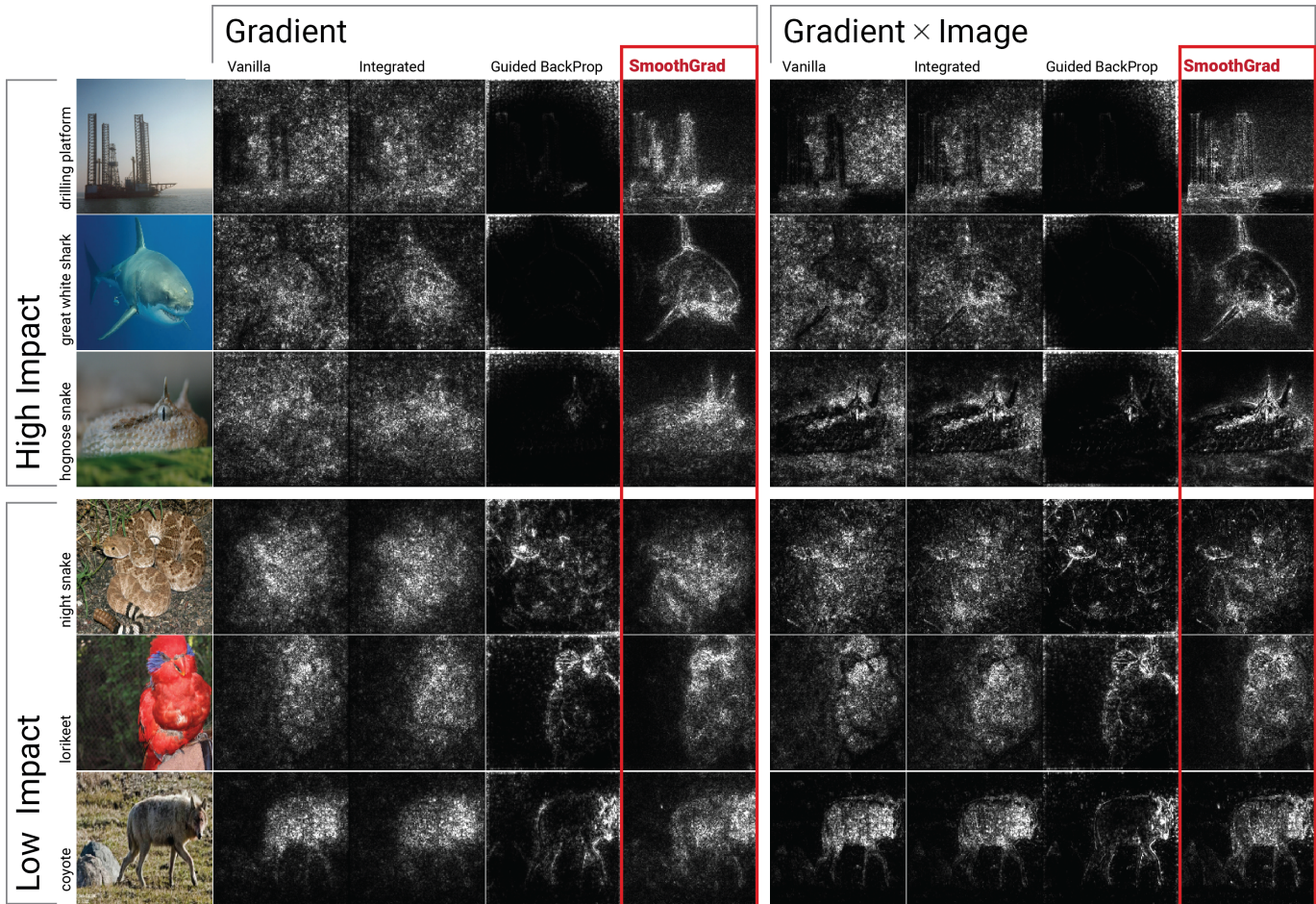


Figure 5. Qualitative evaluation of different methods. First three (last three) rows show examples where applying SMOOTHGRAD had high (low) impact on the quality of sensitivity map.

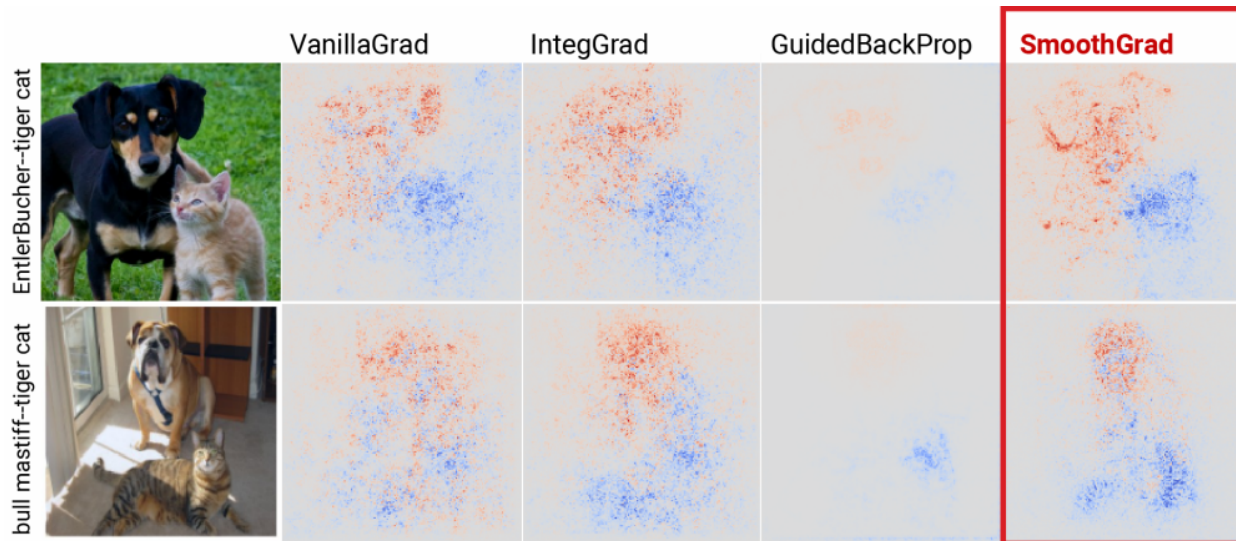


Figure 6. Discriminativity of different methods. For each image, we visualize the difference $\text{scale}(\partial y_1 / \partial x) - \text{scale}(\partial y_2 / \partial x)$ where y_1 and y_2 are the logits for the first and the second class (i.e., cat or dog) and $\text{scale}()$ normalizes the gradient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

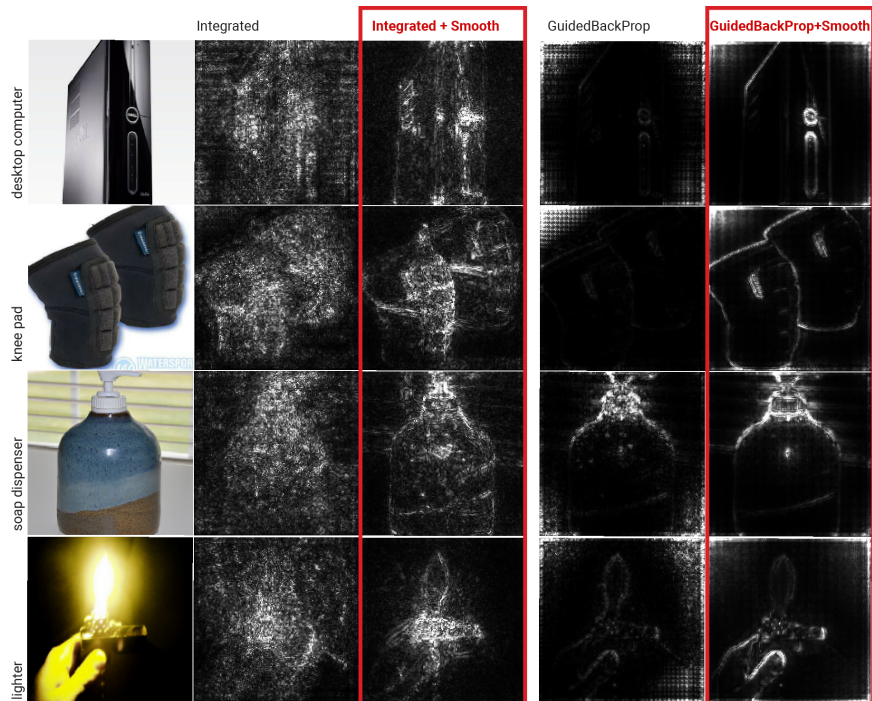


Figure 7. Using SMOOTHGRAD in addition to existing gradient-based methods: *Integrated Gradients* and *Guided BackProp*.

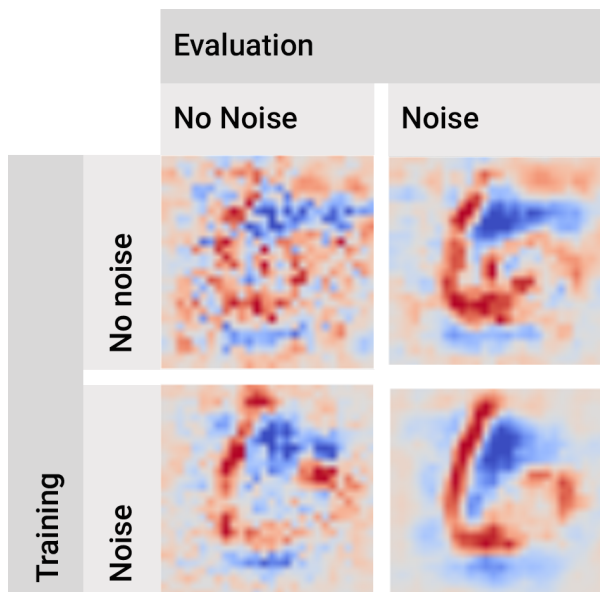


Figure 8. Effect of adding noise during training vs evaluation for MNIST.

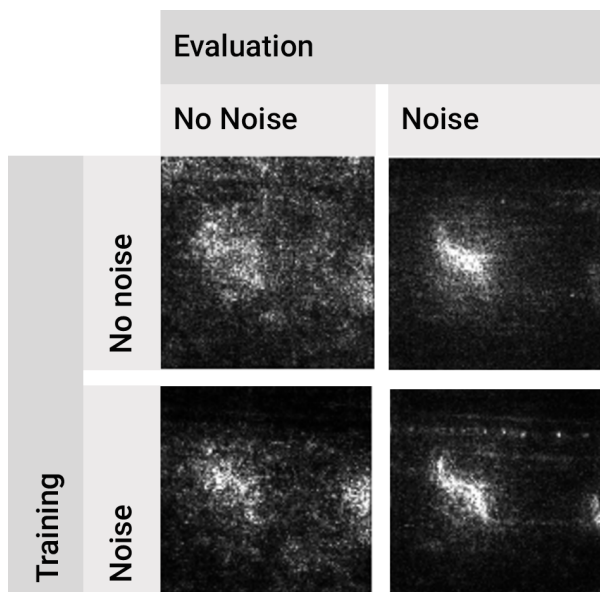


Figure 9. Effect of adding noise during training vs evaluation for Inception.

more spatial coherent maps, one could add a penalty for large differences in partial derivatives of the class score with respect to neighboring pixels. It may also be worth investigating the geometry of the class score function to understand why smoothing seems to be more effective on images with large regions of near-constant pixel values.

A further area for exploration is to find better metrics for comparing sensitivity maps. To measure spatial coherence, one might use existing databases of image segmentations, and we are already making progress (Oh et al., 2017; Selvaraju et al., 2016). Systematic measurements of discriminativity could also be valuable. Finally, a natural question is whether the de-noising techniques described here generalize to other network architectures and tasks.

Acknowledgements

We thank Chris Olah for generously sharing his code and helpful discussions, including pointing out the relation to contractive autoencoders, and Mukund Sundararajan and Qiqi Yan for useful discussions.

References

- Bach, Sebastian, Binder, Alexander, Montavon, Grégoire, Klauschen, Frederick, Müller, Klaus-Robert, and Samek, Wojciech. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS one*, 10(7):e0130140, 2015.
- Baehrens, David, Schroeter, Timon, Harmeling, Stefan, Kawanabe, Motoaki, Hansen, Katja, and Müller, Klaus-Robert. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11 (Jun):1803–1831, 2010.
- Bishop, Chris M. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.
- Doshi-Velez, Finale; Kim, Been. Towards a rigorous science of interpretable machine learning. In *eprint arXiv:1702.08608*, 2017.
- Doshi-Velez, Finale, Ge, Yaorong, and Kohane, Isaac. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. *Pediatrics*, 133 (1):e54–e63, 2014.
- Erhan, Dumitru, Bengio, Yoshua, Courville, Aaron, and Vincent, Pascal. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341:3, 2009.
- Freitas, Alex. Comprehensible classification models: a position paper. *ACM SIGKDD Explorations*, 2014.

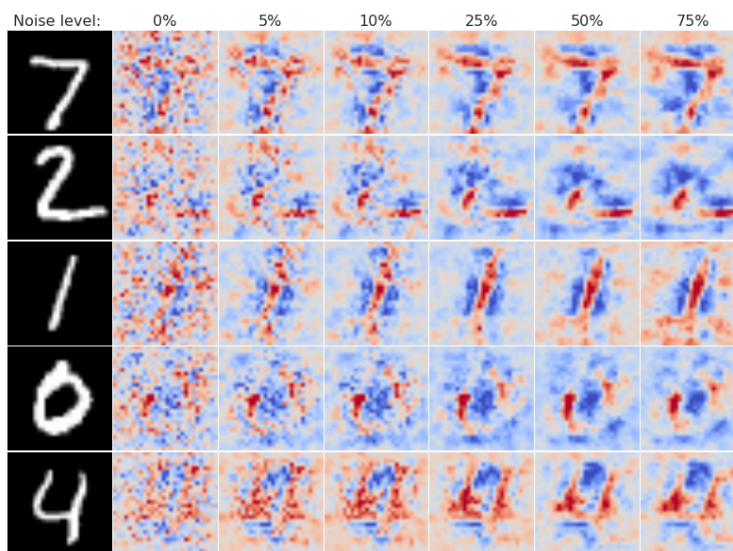


Figure 10. Effect of noise level on the estimated gradient across 5 MNIST images. Each sensitivity map is obtained by applying a Gaussian noise at inference time and averaging in the same way as in Fig. 3 over 100 samples.

Hughes, Michael C, Elibol, Huseyin Melih, McCoy, Thomas, Perlis, Roy, and Doshi-Velez, Finale. Supervised topic models for clinical interpretability. *arXiv preprint arXiv:1612.01678*, 2016.

Kim, Been, Glassman, Elena, Johnson, Brittney, and Shah, Julie. ibcm: Interactive bayesian case model empowering humans via intuitive interaction. Technical report, Massachusetts Institute of Technology, 2015.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Yann, Cortes, Corinna, and Burges, Christopher JC. Mnist handwritten digit database. *AT&T Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2, 2010.

Lou, Yin, Caruana, Rich, and Gehrke, Johannes. Intelligent models for classification and regression. In *ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

Oh, Seong Joon, Benenson, Rodrigo, Khoreva, Anna, Akata, Zeynep, Fritz, Mario, and Schiele, Bernt. Exploiting saliency for object segmentation from image level labels. *arXiv preprint arXiv:1701.08261*, 2017.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, et al.

Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Selvaraju, Ramprasaath R, Das, Abhishek, Vedantam, Ramakrishna, Cogswell, Michael, Parikh, Devi, and Batra, Dhruv. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

Shrikumar, Avanti, Greenside, Peyton, and Kundaje, Anshul. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017.

Simonyan, Karen, Vedaldi, Andrea, and Zisserman, Andrew. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Springenberg, Jost Tobias, Dosovitskiy, Alexey, Brox, Thomas, and Riedmiller, Martin. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Sundararajan, Mukund, Taly, Ankur, and Yan, Qiqi. Ax- iomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, and Fergus, Rob. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jon, and Wojna, Zbigniew. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

TensorFlow. MNIST TensorFlow tutorial. https://www.tensorflow.org/get_started/mnist/pros, 2017. [Online; accessed 9-May-2017].

Zeiler, Matthew D and Fergus, Rob. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Zhou, Bolei, Khosla, Aditya, Lapedriza, Agata, Oliva, Aude, and Torralba, Antonio. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.

Zintgraf, Luisa M., Cohen, Taco S., and Welling, Max. A new method to visualize deep neural networks. *CoRR*, abs/1603.02518, 2016. URL <http://arxiv.org/abs/1603.02518>.