A matter of attitude: Focusing on positive and active gradients to boost saliency maps

Oscar Llorente[a], Jaime Boal[b], Eugenio F. Sánchez-Úbeda[b]

[a]*BMAS SA NDO SW R&D Unit B, Ericsson, Retama Ed 1 Torre Suecia, Madrid, 28045, Madrid, Spain*
[b]*Institute for Research in Technology (IIT), ICAI School of Engineering, Comillas Pontifical University, Santa Cruz de Marcenado, 26, Madrid, 28015, Madrid, Spain*

Abstract

Saliency maps have become one of the most widely used interpretability techniques for convolutional neural networks (CNN) due to their simplicity and the quality of the insights they provide. However, there are still some doubts about whether these insights are a trustworthy representation of what CNNs use to come up with their predictions. This paper explores how rescuing the sign of the gradients from the saliency map can lead to a deeper understanding of multi-class classification problems. Using both pretrained and trained from scratch CNNs we unveil that considering the sign and the effect not only of the correct class, but also the influence of the other classes, allows to better identify the pixels of the image that the network is really focusing on. Furthermore, how occluding or altering those pixels is expected to affect the outcome also becomes clearer.

Keywords: Interpretability, convolutional neural networks, saliency maps, visualization, gradient signs

1. Introduction

The overwhelming mediatic interest that generative artificial intelligence is drawing lately is fostering the adoption of deep learning models in almost every area of our lives. Unfortunately, the outstanding advances brought by these massive models have yet to be accompanied by an equivalent effort to make them more interpretable. Blindly using complex models without worrying about how their outputs are generated entails risks that must be mitigated if we strive to adopt them in sensitive sectors, as many authorized voices and legislators are already pointing out.

There are basically two approaches to address this issue: building models that are easier to understand by design and at the same time match the performance of their black box counterparts, or developing techniques to disclose what is going on inside the black boxes. This paper concentrates on the field of computer vision, where there are indeed some attempts to construct interpretable models for object classification [1] and medical applications [2].

Email addresses: oscar.llorente.gonzalez@ericsson.com (Oscar Llorente), jaime.boal@iit.comillas.edu (Jaime Boal), eugenio.sanchez@iit.comillas.edu (Eugenio F. Sánchez-Úbeda)

However, due to their great feature extraction capabilities, far better than reknown traditional engineered features such as SIFT [3], many modern computer vision solutions still rely on regular convolutional neural networks (CNNs) [4] as part of their pipeline, whose inner workings are hard to interpret and understand.

Over the past decade, the research community has produced several techniques that seek to shed some light on how CNNs come up with their predictions. Leaving the visual inspection of the convolutional filters aside, most of the proposals consist in studying the effect of exciting the model with known stimuli and projecting the result back into the input image.

One family of techniques attempts to approximate the trained network with simpler models. Zhou et al. [5] remove information from the input images to obtain a minimal representation that preserves as little visual information as possible without significantly impacting the classification score. This is done by segmenting the image and iteratively discarding those regions that contribute the least. Ribeiro et al. [6] propose LIME, an algorithm that approximates any classifier or regressor locally with interpretable models. To deal with images they extract K superpixels and treat them as input binary features for a linear model. Similarly, Frosst and Hinton [7] suggest building a binary soft decision tree from the learned filters.

Another popular approach relies on backpropagation. Deconvolutional Networks or DeconvNets [8] invert the order of the CNN layers to discover the input pixels responsible for the information encoded in every feature map. DeconvNets allow gathering evidence about the type of features every CNN layer is able to extract, from basic geometries like corners and edges at the beginning to class-specific features as one proceeds deeper into the network.

Due to their simplicity, and perhaps for being one of the seminal methods in this category, saliency maps [9] have become one of the most popular local interpretability methods for CNNs. They compute the absolute gradient of the target output to respect to every input feature (i.e., pixels on images) and are commonly employed in multi-class classification problems:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}}\, S_c(\boldsymbol{I}) \tag{1}$$

where $\boldsymbol{I}$ is the input image, C the set of all possible classes, and S_c corresponds to the classification score for a given class c. Since images are usually encoded in the RGB color space, it is important to bear in mind that $\boldsymbol{I} \in \mathbb{R}^{\text{channels} \times \text{height} \times \text{width}}$ is a tensor. The original saliency map method is mathematically expressed as

$$\boldsymbol{M}_{i\,j} = \max_{k} \left| \frac{\partial S_{\hat{c}}}{\partial \boldsymbol{I}_{k\,i\,j}} \right| \tag{2}$$

$\boldsymbol{M}$ is a 2D map, k indicates a specific channel and, i and j, are the row and column of every pixel, respectively. According to [9], the brightest points in a saliency map (i.e., the derivatives with a larger absolute value) are the pixels that, with a minimal change, should affect the class score the most. As shown in (2), the maximum over the three channels is computed to obtain a single value per pixel. Even though this decision may seem arbitrary, it is the convention followed in almost every subsequent paper on saliency maps.

There have been several attempts to reduce the inherent noise of saliency maps like that of Shrikumar et al. [10], who suggest multiplying element-wise the input and the gradient to

produce sharper representations. However, the fact that on visual inspection the representation resembles the input image is no guarantee of its correctness as put forward in the sanity checks proposed by Adebayo et al. [11]. Apparently, the same happens to varying degrees in other similar methods such as ϵ-LRP, DeepLIFT [12], or integrated gradients [13]. The technique does not highlight what the neural network has learned to pay attention to, but rather tends to augment features from the input image such as edges that may or may not be relevant to the prediction.

This paper proposes several improvements over the regular saliency maps to increase the insights that can be extracted. The contributions can be summarized in three main points:

- Instead of taking the absolute value of the gradients, and thus neglecting their sign, we prove that preserving this information enhances interpretability in multi-class classification problems by better identifying which pixels assist or deceive the network.
- The network would want pixels with a positive gradient to have higher intensities and those with negative gradients to be dimmed towards zero. This fact makes occlusion experiments more self-explanatory, since it is easier to understand the meaning of replacing a given pixel with a brighter or darker one, ultimately with white or black.
- Typically, only the class of interest is considered when analyzing saliency maps. Based on the gradient sign, a set of metrics have been defined to quantitatively compare the impact on the prediction of a particular class caused by the rest of the classes.

The remainder of the document is structured as follows. It starts with a brief discussion of the implications of ignoring the sign of the gradients (Section 2). Using the information provided by the sign, Section 3 explores the effect that modifying a pixel value has on a multi-class classification problem. Finally, Section 4 presents the experiments conducted to support the conclusions derived in Section 5.

2. The importance of the gradient sign

To the top of our knowledge, there is little research about the meaning or impact of the sign in saliency maps. The only article that briefly discusses this topic is [14], which explains that the raw value of gradients (without taking the absolute value) is commonly used in the MNIST dataset [15], but not in other datasets like ImageNet [16]. Apparently, experimental results suggest that on MNIST raw gradients produce clearer saliency maps and, at the same time, worse representations on ImageNet. Since the latter is the *de facto* standard dataset for CNNs, in general saliency maps are implemented with the absolute value.

However, taking the absolute value of every pixel in the saliency map comes at a cost and some enlightening information is lost. In terms of explainability, the opportunity of knowing which regions of the image should be brighter or darker to improve the classification accuracy is disregarded. Moreover, if both pixels with positive and negative gradients are combined in the same image without any distinctions, the representation can become confusing. Sometimes it may seem as if the model is not able to tell apart regions that should be brighter or darker like, for instance, an object (positive gradient) on an uninformative background (negative). Therefore, two sets of pixels can be distinguished in the image:

- Pixels that improve the classification score of the predicted class if their value is *increased*, since they have a positive value (gradient) in the saliency map.

- Pixels that improve the classification score of the predicted class if their value is *decreased*, because they have a negative gradient.

The advantage of this separation with respect to focusing on the raw gradients and then normalizing their values to represent them on a single image is that in [14] zero gradients shine at medium intensity after scaling, conveying a misleading idea of importance to those pixels. Instead, we propose creating two different visualizations before taking the absolute value (or the ReLU function, which naturally provides the same result if positive and negative gradients are handled separately):

- Positive saliency maps:

$$\boldsymbol{M}_{ij} = \max_{k} \left(\text{ReLU}\left(\frac{\partial S_{\hat{c}}}{\partial \boldsymbol{I}_{kij}} \right) \right) \tag{3}$$

- Negative saliency maps:

$$\boldsymbol{M}_{ij} = \max_{k} \left(\text{ReLU}\left(-\frac{\partial S_{\hat{c}}}{\partial \boldsymbol{I}_{kij}} \right) \right) \tag{4}$$

3. Multi-class saliency maps

All saliency map techniques use the actual class to compute the derivatives. In multi-class classification problems this approach disregards the effect of the rest of the classes. Despite there can be pixel gradients computed respect to incorrect classes with a higher value, the current techniques do not draw attention to this fact. Whenever this happens, the interpretation of the saliency map changes since if the value of this pixel is increased, the classification score for the incorrect class will improve more than that of the true class, worsening the prediction.

Taking the absolute value makes things even more undecipherable. Once you lose the sign information, you can no longer determine whether increasing the intensity of a pixel is bound to increase or decrease the score of a given class. In order to extend the scope of positive and negative saliency maps to consider the effect of all the classes, the definitions put forward in the previous section can be restated:

- *Active* pixels are those that improve the most the classification score of the predicted class if their value is *increased*, more than that of any other class considered.

- Analogously, *inactive* pixels are those that improve the most the score of the predicted class if their value is *decreased*. Their gradient with respect to the actual class is therefore the lowest (the most negative) among all the classes. These pixels are the ones that cause more confusion to the classifier.

Based on these definitions, two additional saliency map visualization can be derived:

- *Active saliency maps* highlight the pixels that should be increased to improve the classification score of the true class:

$$\boldsymbol{M}_{ij} = \max_k \begin{cases} \frac{\partial S_{\hat{c}}}{\partial \boldsymbol{I}_{k\,i\,j}} & \text{if } \frac{\partial S_{\hat{c}}}{\partial \boldsymbol{I}_{k\,i\,j}} = \text{argmax}_{c \in C} \frac{\partial S_c}{\partial \boldsymbol{I}_{k\,i\,j}} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

- *Inactive saliency maps* depict the pixels that should be dimmed to enhance the classification score of the correct class:

$$\boldsymbol{M}_{ij} = \max_k \begin{cases} \frac{\partial S_{\hat{c}}}{\partial \boldsymbol{I}_{k\,i\,j}} & \text{if } \frac{\partial S_{\hat{c}}}{\partial \boldsymbol{I}_{k\,i\,j}} = \text{argmin}_{c \in C} \frac{\partial S_c}{\partial \boldsymbol{I}_{k\,i\,j}} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

In conclusion, where positive and negative saliency maps provide information about whether increasing or decreasing the value of particular pixels improves the score of the correct class, active and inactive saliency maps go a step further and identify those pixels that should be altered to increase the confidence of the model in the prediction of the true class.

4. Experiments

This section evaluates the proposed new saliency map representations both qualitatively and quantitatively on two different datasets: CIFAR-10 [17] and Imagenette [18]. The former is commonly used in image classification and interpretability papers. The latter is a subset of ten ImageNet classes that allows drawing grounded conclusions without requiring an immense computational effort.

The new saliency maps have been tested against both trained from scratch and pretrained models. Two models have been trained from scratch using CIFAR-10. The first is a basic CNN with several convolutional blocks with either max- or average-pooling and a final linear layer. The second uses the standard ResNet-18 architecture [19]. For Imagenette, in addition to the previous two models, pre-trained versions of ResNet-18 and ConvNeXt [20] have also been evaluated. In all cases, the networks were trained during 50 epochs with a learning rate of 0.001 using the AdamW optimizer. Table 1 shows the accuracies obtained. At the sight of these results, it seems that the models trained from scratch need more epochs. Stopping at this stage is an intentional decision to be able to study the performance of the proposed saliency maps during the training process.

Table 1: Test set accuracy.

	CIFAR-10	Imagenette
Basic CNN	0.6307	0.6205
ResNet-18	0.7569	0.8357
ResNet-18 pre-trained	-	0.9733
ConvNext pre-trained	-	0.9932

4.1. Qualitative evaluation

Following the same approach found in the literature of saliency maps [9, 21, 14, 13], first a visual inspection is carried out to compare the proposed visualizations with the standard saliency map. It is common practice to show only a few examples of the visualizations to perform the comparisons. However, none of the aforementioned articles explain how these examples are selected. Therefore, it could be the case that a particular visualization is better for a certain class or example. To prevent this problem, in this paper a correctly classified example has been randomly selected from the test set. To enhance readability, only two examples are shown in this section (Figures **??** and **??**). Refer to Appendix A to check the rest of the images.

It is noticeable that the proposed techniques reduce the noise and produce sharper visualization than the original saliency map. The shape of the objects that the neural network is classifying is more defined and exhibits a higher level of detail. Notwithstanding, even though this evaluation is typical in the literature, it does not prove them better. It could be the case that a noisier visualization is more faithful to what the neural network has learned to focus on. This is why a quantitative evaluation is required.

4.2. Quantitative evaluation

There have been some efforts in the literature to formulate metrics that measure the effectiveness of local interpretability techniques. While Ancona et al. [12] develop on the desirable properties of interpretability methods, [22] and [23] actually propose a metric to compare techniques. Specifically, [22] suggests using a metric called deletion that removes pixels in descending order of importance —according to the technique under evaluation— and recomputes the probability of the correct output for each fraction of deleted pixels. Deleted pixels are either replaced with a constant value (e.g., black or gray) or random noise. Hooker et al. [23] claim that it is necessary to retrain the model after deleting pixels to maintain the same distribution in the training and the test sets. However, retraining affects the network's weights and the metric no longer provides a good estimate of how the original model behaves if some pixels are occluded.

The main drawback of the deletion metric is that the value of a pixel cannot be actually deleted. No matter what value pixels are replaced with, they will still affect the internal computations of the network. The replacement value introduces unknown biases unless pixels are separated in two different sets: the ones that should be brighter to improve the classification score of the original predictions (i.e., those identified by positive or active saliency maps) and the ones that should be darker (i.e., pixels in negative or inactive saliency maps). Thanks to this distinction, the meaning of replacing a pixel with white (white-deletion) or black (black-deletion) becomes instantly clear. For positive or active pixels, using white would tend to improve the classification score of the predicted class, whereas zeroing them out should severely harm the original classification. The opposite is expected to happen with negative and inactive pixels.

Both black- (**??**) and white-deletion (**??**) measure the change in the predicted classes with respect to the original classification, which we have decided to coin *allegiance*. Using the test set —the results for the training set can be found in Appendix B—, pixels are removed in descending order of importance in blocks of 10% as suggested in [23], except for the initial interval, in which we deem it necessary to study the response in more detail.

Table 2: AUC for black deletions in saliency maps.

	CIFAR-10		Imagenette			
	CNN	ResNet-18	CNN	ResNet-18	ResNet-18 pre-trained	ConvNeXt
Original	0.23	0.23	0.24	0.26	0.28	0.49
Positive	0.14	0.11	0.12	0.14	0.21	0.37
Active	0.15	0.11	0.16	0.14	0.24	0.39

Table 3: AUC for white deletions in saliency maps.

	CIFAR-10		Imagenette			
	CNN	ResNet-18	CNN	ResNet-18	ResNet-18 pre-trained	ConvNeXt
Original	0.21	0.22	0.21	0.23	0.26	0.46
Negative	0.14	0.12	0.11	0.18	0.20	0.37
Inactive	0.15	0.12	0.14	0.16	0.23	0.42

The behavior observed in the graphs corresponds to what we expected. The decrease in allegiance for black-deletion is greater for active and positive saliency maps than for the standard implementation. Likewise, the decrease in allegiance for white-deletion is greater for inactive and negative saliency maps. Apparently, this confirms the hypothesis that the pixels identified are more important to the network when the sign of the gradients is taken into account.

It is important to note that that for active and inactive saliency maps the allegiance stops decreasing after around 50% of the pixels have been deleted. The reason is that many pixels have a value of zero because their derivative with respect to the original predicted class is not the largest (for the active saliency map) or the smallest (for the inactive). Hence, after all the non-zero pixels from the active and inactive saliency maps have been deleted there is nothing else to remove. The same happens for the positive and negative saliency maps at approximately 80%.

To provide concrete numbers, the area under the curve is shown in Table 2 for black-deletion and in Table 3 for white-deletion. The results support the hypothesis that the proposed saliency maps better identify those pixels that, when made brighter or darker as appropriate, increase the confidence of the originally predicted class. Interestingly, although the improvement over the standard saliency map is clear, it is surprising how positive and negative saliency maps sometimes work better than active and inactive. It could still be due to the use of the extremes (i.e., either black or white) as replacement values, instead of slightly darker or brighter variants of the original pixel colors. Nevertheless, the results are still more interpretable than those provided by other metrics proposed in the literature because the effect of the alteration on the image is now known.

5. Conclusion and future work

There is more information hidden in the gradients of a saliency map than is usually exploited, both in the sign of the individual pixels and in the gradients with respect to the incorrect classes. Separating pixels according to these dimensions could pave the way to

improving the quality of the insights extracted, not only from saliency maps but also from other local interpretability techniques based on gradients.

Furthermore, instead of arbitrarily choosing black to occlude pixels as it is typically done, the proposed approach allows to better understand the effect of replacing pixels with black or white, which can positively or negatively contribute to the classification score depending on the gradient sign. Analyzing the faithfulness of the different variations of the saliency map from this point of view is left as future work.

References

[1] Q. Zhang, X. Wang, Y. Wu, H. Zhou, S. Zhu, Interpretable CNNs for object classification, IEEE Transactions on Pattern Analysis & Machine Intelligence 43 (10) (2021) 3416–3431. doi:10.1109/TPAMI.2020.2982882.

[2] G. Singh, Think positive: An interpretable neural network for image recognition, Neural Networks 151 (2022) 178–189. doi:10.1016/j.neunet.2022.03.034.

[3] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110. doi:10.1023/B:VISI.0000029664.99615.94.

[4] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, MIT Press, Cambridge, MA, USA, 1998, pp. 255–258. doi:10.5555/303568.303704.

[5] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Object detectors emerge in deep scene CNNs, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations (ICLR) , May 7–9, Conference Track Proceedings, San Diego, CA, USA, 2015.

[6] M. T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier, in: Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, Association for Computational Linguistics, San Diego, CA, USA, 2016, pp. 97–101. doi:10.18653/v1/N16-3020.

[7] N. Frosst, G. Hinton, Distilling a neural network into a soft decision tree, Computing Research Repository (CoRR) (2017).

[8] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), European Conference on Computer Vision, 2014, pp. 818–833. doi:10.1007/978-3-319-10590-1_53.

[9] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, in: 2nd International Conference on Learning Representations (ICLR), April 14–16, Workshop Track Proceedings, 2014.

[10] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: Proc. of the 34th International Conference on Machine Learning (PMLR), Vol. 70, 2017, pp. 3145–3153.

[11] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, B. Kim, Sanity checks for saliency maps, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS), Curran Associates, Inc., Montréal, Canada, 2018, pp. 9525–9536.

[12] M. Ancona, E. Ceolini, C. Öztireli, M. Gross, Towards better understanding of gradient-based attribution methods for deep neural networks, in: 6th Int. Conf. Learning Representations, (ICLR), Vancouver, BC, Canada, 2018.

[13] M. Sundararajan, A. Taly, Q. Yan, Axiomatic Attribution for Deep Networks, in: Proceedings of the 34th International Conference on Machine Learning, PMLR, 2017, pp. 3319–3328.

[14] D. Smilkov, N. Thorat, B. Kim, F. Viégas, M. Wattenberg, SmoothGrad: Removing noise by adding noise, Computing Research Repository (CoRR) (2017) 10.

[15] MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges, http://yann.lecun.com/exdb/mnist/.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: IEEE Conf. Computer Vision and Pattern Recognition, 2009, pp. 248–255.
URL https://www.image-net.org/

[17] A. Krizhevsky, V. Nair, G. Hinton, CIFAR-10 and CIFAR-100 datasets.
URL https://www.cs.toronto.edu/~kriz/cifar.html

[18] J. Howard, Imagenette.
URL https://github.com/fastai/imagenette

[19] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[20] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A ConvNet for the 2020s, IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR) (2022) 11966–11976.

[21] J. T. Springenberg, A. Dosovitskiy, T. Brox, M. A. Riedmiller, Striving for simplicity: The all convolutional net, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings, 2015.
URL http://arxiv.org/abs/1412.6806

[22] V. Petsiuk, A. Das, K. Saenko, RISE: Randomized input sampling for explanation of black-box models, in: British Machine Vision Conference (BMVC), 2018.

[23] S. Hooker, D. Erhan, P.-J. Kindermans, B. Kim, A Benchmark for Interpretability Methods in Deep Neural Networks, in: Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019.

Appendix A. Signed saliency map examples

Appendix B. Black-deletion and white-deletion on the training set