Interpreting the Predictions of Complex ML Models by Layer-wise Relevance Propagation

Wojciech Samek
Fraunhofer HHI, 10587 Berlin, Germany
wojciech.samek@hhi.fraunhofer.de

Grégoire Montavon
TU Berlin, 10587 Berlin, Germany
gregoire.montavon@tu-berlin.de

Alexander Binder
SUTD, 487372 Singapore
alexander_binder@sutd.edu.sg

Sebastian Lapuschkin
Fraunhofer HHI, 10587 Berlin, Germany
sebastian.lapuschkin@hhi.fraunhofer.de

Klaus-Robert Müller
TU Berlin, 10587 Berlin, Germany
klaus-robert.mueller@tu-berlin.de

Abstract

Complex nonlinear models such as deep neural network (DNNs) have become an important tool for image classification, speech recognition, natural language processing, and many other fields of application. These models however lack transparency due to their complex nonlinear structure and to the complex data distributions to which they typically apply. As a result, it is difficult to fully characterize what makes these models reach a particular decision for a given input. This lack of transparency can be a drawback, especially in the context of sensitive applications such as medical analysis or security. In this short paper, we summarize a recent technique introduced by Bach et al. [1] that explains predictions by decomposing the classification decision of DNN models in terms of input variables.

1 Explaining Predictions by Decomposition

Nonlinear models such as neural networks are an essential component of many practical machine learning algorithms. These models have solved numerous practical problems such as visual object recognition, speech recognition, or natural language processing. Deep neural networks in particular, have been shown recently to perform extremely well on these tasks [2, 3, 4]. A typical limitation of complex machine learning models is their tendency to predict in a black-box manner, providing little information on what aspect of the input data supports the actual prediction. The problem of explaining neural network predictions [5, 6, 1, 7, 8, 9, 10, 11] has received a lot of attention recently, especially in the context of image recognition with convolutional neural networks. From the multitude of recently proposed methods for explaining predictions, we can identify *decomposition approaches* [12, 13, 1, 14], which seek to redistribute the value of the prediction function on the input variables, such that the sum of redistributed terms corresponds to the actual function value.

Let us formalize the problem of explaining a prediction from the perspective of decomposition. Let $f : \mathbb{R}^d \to \mathbb{R}$ be the prediction function, $\boldsymbol{x}$ a data point given as input to the model and $f(\boldsymbol{x})$ the prediction for this particular data point. In the context of image classification, the data point is formed by a set of pixels: $\boldsymbol{x} = (x_p)_p$. Explanation through the decomposition framework produces a vector of scores $(R_p)_p$ associated to each pixel, indicating how relevant pixels are to the prediction. These

Workshop on Interpretable Machine Learning for Complex Systems (NIPS 2016), Barcelona, Spain.

relevance scores are a decomposition if they satisfy the conservation equation $\sum_p R_p = f(\boldsymbol{x})$. In practice, other criteria are necessary to define a good decomposition, for example the ratio between positive and negative scores, and the relation between the relevance score and the effect of the corresponding pixel activation on the predicted function value.

In contrast to sensitivity analysis which assigns scores based on the effect of a small or infinitesimal pixel perturbation on the function value, e.g. $(\partial f / \partial x_p)^2$, thus producing an explanation of a local variation of f, the decomposition approach on the other hand seeks to explain the whole function value (i.e. what made the function reach a certain value and not zero). Practically, when considering, for example, an image of the class "scooter", sensitivity analysis tells us which pixels in the image, if modified, makes the image more or less belong to that class. Instead, the decomposition approach explains what pixels speak in what amount for the presence of the scooter in the image. Figure 1 (a) shows the qualitative difference between sensitivity analysis and decomposition (with the latter computed using the LRP method described below).

The difference between decomposition and sensitivity analysis can also be illustrated from the perspective of a linear classifier $f(\boldsymbol{x}) = \sum_p x_p w_p$. A possible decomposition is given by identifying relevance scores as the terms being summed: $R_p = x_p w_p$. Instead, sensitivity analysis will return $R_p = w_p^2$ (or $R_p = |w_p|$ depending on the variant), which does not involve the actual pixel activation. If in particular x_p quantifies the presence of a local feature at position p, it would be intuitive that the feature is considered relevant if not only the classifier reacts to it ($w_p > 0$), but also if that feature is actually present ($x_p > 0$). Only the decomposition approach takes these two parameters into account.

2 Layer-wise Relevance Propagation

Layer-wise relevance propagation (LRP) [1] is one such method, that operates by building for each neuron of a deep network a local redistribution rule, and applying these rules in a backward pass in order to produce the pixel-wise decomposition. LRP has been successfully applied to many different models and tasks beyond classification of images by convolutional neural networks. For instance, in [1] and [15] it has been applied to Bag-of-Words models and Fisher Vector / SVM classifiers, respectively. In [16] it was used for the identification of relevant words in text documents and in [17] for visualizing facial features related to age, happiness and attractivity. Also the authors of [18] use LRP for identifying relevant spatio-temporal EEG features in the context of Brain-Computer Interfacing.

Within the LRP framework, various rules have been proposed for redistributing the relevance assigned to a neuron onto its input neurons. Let $(x_i)_i$ be the neuron activations at layer l. Let $(R_j)_j$ be the relevance scores associated to the neurons at layer $l+1$. Let w_{ij} be the weight connecting neuron i to neuron j. The "alpha-beta" rule [1] redistributes relevance from layer $l+1$ to layer l in the following way:

$$R_i = \sum_j \Big(\alpha \cdot \frac{(x_i w_{ij})^+}{\sum_i (x_i w_{ij})^+} - \beta \cdot \frac{(x_i w_{ij})^-}{\sum_i (x_i w_{ij})^-} \Big) R_j, \tag{1}$$

where $()^+$ and $()^-$ denote the positive and negative parts, respectively. Layer-wise conservation of relevance ($\sum_i R_i = \sum_j R_j$) is enforced by choosing α, β such that $\alpha - \beta = 1$. Choosing the parameters $\alpha = 2$ and $\beta = 1$ was shown to produce nice-looking and sharp heatmaps. This set of parameters works well for a wide range of neural network models, and allows to express contradicting evidence in the input image, through negative relevance scores. Example of heatmaps produced with these parameters are shown in Figure 1. Choosing instead the parameters $\alpha = 1$ and $\beta = 0$ and assuming positive activations provides connections to other methods. In particular, it yield the simplified formula:

$$R_i = \sum_j \frac{x_i w_{ij}^+}{\sum_i x_i w_{ij}^+} R_j \tag{2}$$

which is equivalent to the z^+-rule by [14] and the redistribution rule used by [7] as part of the Excitation Backprop (EB) method[1]. Futhermore, [14] showed that this particular redistribution rule

[1] Compare Eq. (2) in [7] and the z^+ rule in [14]. Since the authors of [7] assume the response of the activation neuron to be non-negative, Eq. (2) in [7] is also equivalent to the alpha-beta LRP rule [1] for $\alpha = 1$ and $\beta = 0$.

results from a “deep Taylor decomposition” of the neural network function when the neural network is composed of ReLU neurons. The main concepts used by deep Taylor decomposition and how it leads to the propagation rule of Equation 2 are outlined in the following.

Connection to Deep Taylor Decomposition Assume that the relevance score R_j is the product of the ReLU activation x_j and a positive constant term c_j. It can then be written as:

$$\begin{aligned} R_j &= x_j c_j \\ &= \max(0, \textstyle\sum_i x_i w_{ij} + b_j) \cdot c_j \\ &= \max(0, \textstyle\sum_i x_i w_{ij} c_j + b_j c_j), \end{aligned}$$

where $w_{ij} c_j$ and $b_j c_j$ are the weights and bias parameters of a newly defined “relevance neuron”. A first-order Taylor expansion of the relevance neuron at some reference point $(\widetilde{x}_i)_i$ allows to decompose the neuron value in terms of its input neurons. The reference point is chosen to be the intersection of the neuron equation $\sum_i x_i w_{ij} c_j + b_j c_j = \varepsilon$ with ε positive and infinitesimally small, and the search line $\{(x_i)_i - t \cdot (x_i 1_{\{w_{ij} c_j > 0\}})_i : t \in \mathbb{R}^+\}$, that is, progressively deactivating positively contributing inputs until the relevance becomes almost zero. In that case, the relevance “flowing” from neuron j to neuron i is given by the identification of the corresponding first-order term of the Taylor expansion and has a closed form solution:

$$R_{i \leftarrow j} = \frac{\partial R_j}{\partial x_i}\Big|_{(x_i)_i = (\widetilde{x}_i)_i} \cdot (x_i - \widetilde{x}_i) = \frac{x_i (w_{ij} c_j)^+}{\sum_i x_i (w_{ij} c_j)^+} R_j = \frac{x_i w_{ij}^+}{\sum_i x_i w_{ij}^+} R_j.$$

When we pool the relevance messages coming from the upper-layer neurons ($R_i = \sum_j R_{i \leftarrow j}$), we recover Eq. (2). We now show that the product structure on which the decomposition relies also holds approximately for the lower-layer relevance. This can be made visible by rewriting Eq. (2) as

$$R_i = \sum_j \frac{x_i w_{ij}^+}{\sum_i x_i w_{ij}^+} R_j = x_i \cdot \underbrace{\sum_j \frac{w_{ij}^+ \cdot \max(0, \sum_i x_i w_{ij} + b_j) \cdot c_j}{\sum_i x_i w_{ij}^+}}_{c_i}$$

where c_i is positive and can indeed be considered as approximately constant due to its very weak dependence on x_i (diluted by two nested sums). Thus, if the decomposition can be performed at a certain layer, it can also be performed at the previous layer. For more details we refer the reader to the original paper [14].

3 LRP in Practice

In this section, we briefly discuss how LRP compares to sensitivity analysis, and how LRP can be used in practice for comparing machine learning models and testing what strategy these models use to predict the data.

Comparing sensitivity analysis and LRP A qualitative difference between the explanations produced by sensitivity analysis and LRP for the prediction of an image of class “scooter” is shown in Figure 1 (a). The explanations provided by sensitivity analysis are much noisier than the ones computed with LRP. Regions consisting of pure background, e.g., the empty street, have large sensitivity, although these pixels are not really indicative for this image category. However, if we put motor-bike like structures at these particular locations, then this change would certainly increase the classification score. In contrast, the explanation provided by LRP does not highlight the locations where the classifier is very sensitive, but indicates how much every pixel contributes to the prediction. Thus, it explains the given prediction and in this example only points to real scooter-like structures in the image. The explanations provided by LRP are not only better in a qualitative sense, but also quantitatively. The authors of [19] proposed an objective method for comparing visualizations based on perturbation analysis. The right plot in Figure 1 (a) displays the drop in classification score when perturbing regions identified by sensitivity analysis and LRP as being the most relevant ones (i.e., high R_p). A fast drop in classification score, i.e., a large AOPC value[2], implies a meaningful sorting of R_p and thus good explanations. The results in Figure 1 (a) clearly show that LRP provides better explanations than sensitivity analysis for images from the ILSVRC2012 dataset. More details about this experiment can be found in [19].

[2] AOPC stand for area over the perturbation curve.

Measuring importance of context in image classification The explanations provided by LRP indicate how much every pixel contributes to the prediction. By aggregating the pixel-wise scores, e.g., inside and outside of an object bounding box, we can quantify the importance of context in image classification. Figure 1 (b) displays the outside-inside relevance ratio for the 20 categories of the PASCAL VOC 2007 dataset. For classes like “airplane” or “sheep” the context does not seem to be important for the DNN prediction as most relevance lies inside the bounding box. This is different for indoor scene categories such as “chair” or “sofa”. Here the LRP explanations show that context plays a much more important role for the classification (see [15] for more discussion).

Comparing DNN architectures Finally, we demonstrate the usage of LRP for analyzing differences between DNN architectures. Figure 1 (c) displays the LRP explanations obtained from BVLC CaffeNet [20] and GoogleNet [21] when applied to an animal image. One can see that the latter network focuses more on the face of the animal and its explanation heatmap is significantly sparser than for BVLC CaffeNet. We observed this phenomenon for many animal images. This analysis suggests that GoogleNet found a better way to focus on the relevant information in the image, which is often the animal face and not the body shape or the fur.

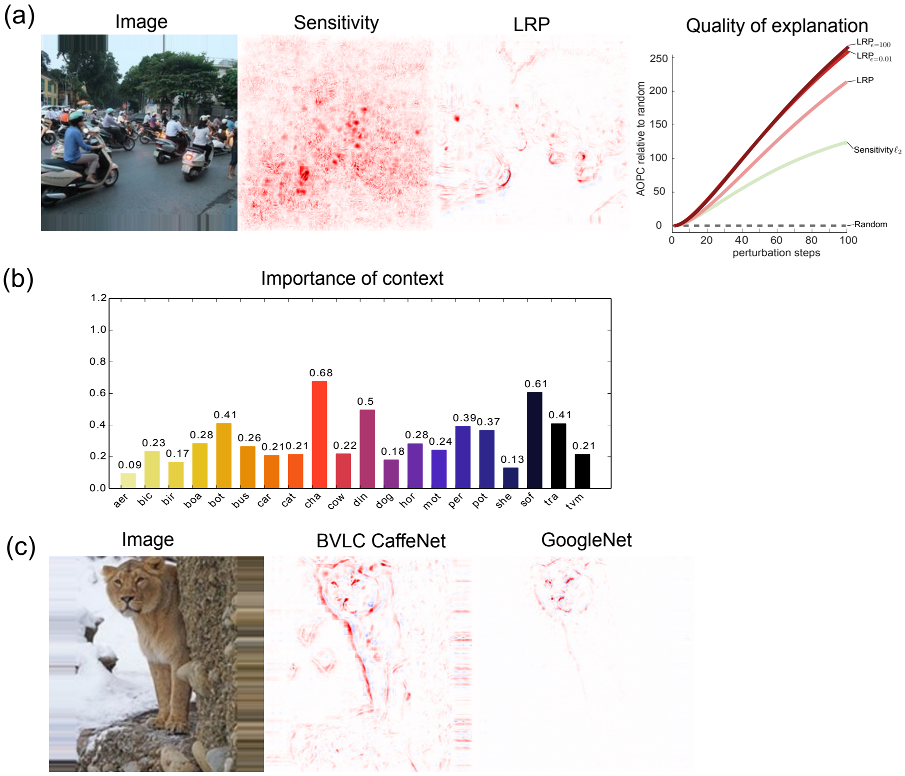

Figure 1: Experimental results. (a) Qualitative and quantitative comparison of sensitivity analysis. (b) Importance of context measured by LRP. (c) Comparison of predictions by two DNN architectures. Illustrations are taken and adapted from the papers [15] and [22].

4 Conclusion

The LRP method is a general framework for interpreting the predictions of complex ML systems such as deep neural networks, that can be used for model comparison, validation, or visualization. It was successfully applied to various tasks and machine learning models. Some redistribution rules for DNNs can be viewed as resulting from a deep Taylor decomposition of the neural network function [14]. Although the alpha-beta LRP rule was shown to work well for DNNs with ReLU activations, there is not one single LRP rule which serves all machine learning models and datasets equally well, as different models have specific layer-to-layer nonlinear mappings and input domains, that need to be considered.

References

[1] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” *PLOS ONE*, vol. 10, no. 7, p. e0130140, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] Y. Kim, “Convolutional neural networks for sentence classification,” *CoRR*, vol. abs/1408.5882, 2014.

[5] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *CoRR*, vol. abs/1312.6034, 2013.

[6] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *ECCV*, 2014, pp. 818–833.

[7] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, “Top-down neural attention by excitation backprop,” in *European Conference on Computer Vision*. Springer, 2016, pp. 543–559.

[8] A. Nguyen, J. Yosinski, and J. Clune, “Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks,” *CoRR*, vol. abs/1602.03616, 2016.

[9] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, “Not just a black box: Learning important features through propagating activation differences,” *CoRR*, vol. abs/1605.01713, 2016.

[10] G. Kasneci and T. Gottron, “Licon: A linear weighting scheme for the contribution ofinput variables in deep artificial neural networks,” in *Proc. of the 25th ACM Int. Conf. on Information and Knowledge Management*. ACM, 2016, pp. 45–54.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, “" why should i trust you?": Explaining the predictions of any classifier,” *CoRR*, vol. abs/1602.04938, 2016.

[12] B. Poulin, R. Eisner, D. Szafron, P. Lu, R. Greiner, D. S. Wishart, A. Fyshe, B. Pearcy, C. Macdonell, and J. Anvik, “Visual explanation of evidence with additive classifiers,” in *Proc. of the 21st National Conference on Artificial Intelligence*, 2006, pp. 1822–1829.

[13] W. Landecker, M. D. Thomure, L. M. A. Bettencourt, M. Mitchell, G. T. Kenyon, and S. P. Brumby, “Interpreting individual classifications of hierarchical networks,” in *IEEE Symposium on Computational Intelligence and Data Mining*, 2013, pp. 32–38.

[14] G. Montavon, S. Bach, A. Binder, W. Samek, and K.-R. Müller, “Explaining nonlinear classification decisions with deep taylor decomposition,” *CoRR*, vol. abs/1512.02479, 2015.

[15] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller, and W. Samek, “Analyzing classifiers: Fisher vectors and deep neural networks,” in *Proc. of IEEE CVPR*, 2016, pp. 2912–2920.

[16] L. Arras, F. Horn, G. Montavon, K.-R. Müller, and W. Samek, “Explaining predictions of non-linear classifiers in nlp,” in *Proc. of the 1st ACL Workshop on Representation Learning for NLP*, 2016, pp. 1–7.

[17] F. Arbabzadeh, G. Montavon, K.-R. Müller, and W. Samek, “Identifying individual facial expressions by deconstructing a neural network,” in *GCPR 2016*, ser. LNCS. Springer, 2016, vol. 9796, pp. 344–354.

[18] I. Sturm, S. Lapuschkin, W. Samek, and K.-R. Müller, “Interpretable deep neural networks for single-trial eeg classification,” *Journal of Neuroscience Methods*, vol. 274, pp. 141–145, 2016.

[19] W. Samek, A. Binder, G. Montavon, S. Bach, and K.-R. Müller, “Evaluating the visualization of what a deep neural network has learned,” *IEEE TNNLS*, 2016.

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proc. of the 22nd ACM Int. Conf. on Multimedia*, 2014, pp. 675–678.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *CoRR*, vol. abs/1409.4842, 2014.

[22] A. Binder, W. Samek, G. Montavon, S. Bach, and K.-R. Müller, “Analyzing and validating neural networks predictions,” in *Proceedings of the Workshop on Visualization for Deep Learning at International Conference on Machine Learning (ICML)*, 2016.