Explaining NonLinear Classification Decisions with Deep Taylor Decomposition

(Supplementary Material)

Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, Klaus-Robert Müller

Abstract

This supplement provides proofs, detailed derivations, pseudocode, and empirical comparisons with other relevance propagation techniques.

1. Derivations of Propagation Rules

In this section, we give the detailed derivations of propagation rules resulting from deep Taylor decomposition of the neural network of Section 4 of the paper. Each propagation rule corresponds to different choices of root point $\{\widetilde{x}_i\}^{(j)}$. For the class of networks considered here, the relevance of neurons in the detection layer is given by

$$R_j = \max(0, \textstyle\sum_i x_i w_{ij} + b_j), \qquad (1)$$

where $b_j < 0$. All rules derived in this paper are based on the search for a root in a particular search direction $\{v_i\}^{(j)}$ in the input space associated to neuron j:

$$\{\widetilde{x}_i\}^{(j)} = \{x_i\} + t\{v_i\}^{(j)} \qquad (2)$$

We need to consider two cases separately:

$$\mathcal{C}_1 = \{j \colon \textstyle\sum_i x_i w_{ij} + b_j \le 0\} = \{j \colon R_j = 0\}$$
$$\mathcal{C}_2 = \{j \colon \textstyle\sum_i x_i w_{ij} + b_j > 0\} = \{j \colon R_j > 0\}$$

In the first case ($j \in \mathcal{C}_1$), the data point itself is already the nearest root point of the function R_j. Therefore,

$$x_i - \widetilde{x}_i^{(j)} = 0. \qquad (3)$$

In the second case ($j \in \mathcal{C}_2$), the nearest root point along the defined search direction is given by the intersection of Equation 2 with the plane equation $\sum_i \widetilde{x}_i^{(j)} w_{ij} + b_j = 0$ to which the nearest root belong. In particular, resolving t by injecting (2) into that plane equation, we get

$$x_i - \widetilde{x}_i^{(j)} = \frac{\sum_i x_i w_{ij} + b_j}{\sum_i v_i^{(j)} w_{ij}} v_i^{(j)} \qquad (4)$$

Starting from the generic relevance propagation formula proposed in Section 3 of the paper, we can derive a more specific formula that involve the search directions $\{v_i\}^{(j)}$:

$$R_i = \sum_j \frac{\partial R_j}{\partial x_i}\Big|_{\{\widetilde{x}_i\}^{(j)}} \cdot (x_i - \widetilde{x}_i^{(j)}) \qquad (5)$$

$$= \sum_{j \in \mathcal{C}_1} \frac{\partial R_j}{\partial x_i} \cdot 0 + \sum_{j \in \mathcal{C}_2} w_{ij} \frac{\sum_i x_i w_{ij} + b_j}{\sum_i v_i^{(j)} w_{ij}} v_i^{(j)} \qquad (6)$$

$$= \sum_j \frac{v_i^{(j)} w_{ij}}{\sum_i v_i^{(j)} w_{ij}} R_j \qquad (7)$$

From (5) to (6) we have considered the two listed cases separately, and injected their corresponding roots found in Equations 3 and 4. From (6) to (7), we have used the fact that the relevance for the case $\mathcal{C}_1$ is always zero to recombine both terms.

The derivation of the various relevance propagation rules presented in this paper will always follow the same three steps:

1. Define for each neuron $j \in \mathcal{C}_2$ a line or segment in the input space starting from data point $\{x_i\}$ and with direction $\{v_i\}^{(j)}$.
2. Verify that the line or segment lies inside the input domain and includes at least one root of R_j.
3. Inject the search directions $\{v_i\}^{(j)}$ into Equation 7, and obtain the relevance propagation rule as a result.

An illustration of the search directions and root points selected by each rule for various relevance functions $R_j(\{x_i\})$ is given in Figure 1.

1.1. w^2-Rule

The w^2-rule is obtained by choosing the root of R_j that is nearest to $\{x_i\}$ in $\mathbb{R}^d$. Such nearest root must be searched for on the line including the point $\{x_i\}$, and with direction corresponding to the gradient of R_j (the ith component of this gradient is w_{ij}). Therefore, the components of the search vector are given by

$$v_i^{(j)} = w_{ij}$$

This line is included in the input domain $\mathbb{R}^d$, and always contains a root (the nearest of which is obtained by setting $t = -R_j / \sum_i w_{ij}^2$ in Equation 2). Injecting the defined search direction v_i into Equation 7, we get

$$R_i = \sum_j \frac{w_{ij}^2}{\sum_i w_{ij}^2} R_j.$$

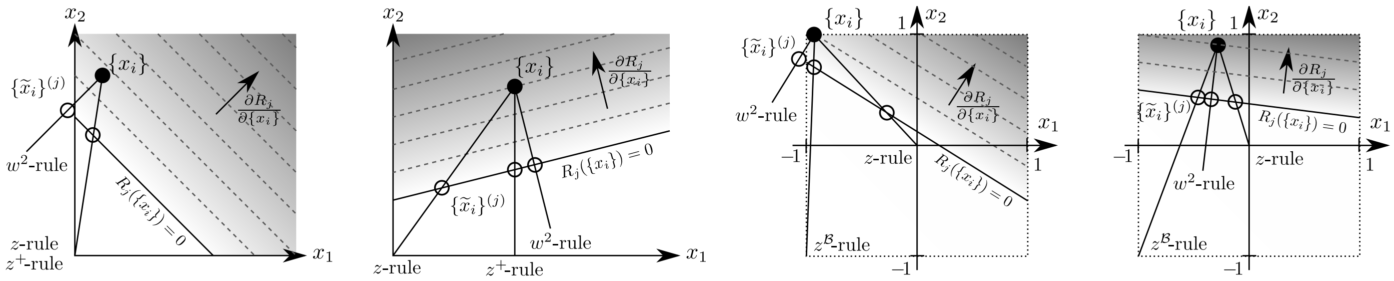

Figure 1: Illustration of root points (empty circles) found for a given data point (full circle) for various propagation rules, relevance functions, and input domains. Here, for the $z^{\mathcal{B}}$-rule, we have used the bounding box $l_1 = -1$, $h_1 = 1$, $l_2 = -1$, $h_2 = 1$.

1.2. z-Rule

The z-rule (originally proposed by [1]) is obtained by choosing the nearest root of R_j on the segment $(\mathbf{0}, \{x_i\})$. This segment is included in all domains considered in this paper ($\mathbb{R}^d, \mathbb{R}^d_+, \mathcal{B}$), provided that $\{x_i\}$ also belongs to these domains. This segment has a root at its first extremity, because $R_j(\mathbf{0}) = \max(0, \sum_i 0 \cdot w_{ij} + b_j) = \max(0, b_j) = 0$ since b_j is negative by design. The direction of this segment on which we search for the nearest root corresponds to the data point itself:

$$v_i^{(j)} = x_i.$$

Injecting this search direction into Equation 7, and defining the weighted activation $z_{ij} = x_i w_{ij}$, we get

$$R_i = \sum_j \frac{z_{ij}}{\sum_i z_{ij}} R_j.$$

1.3. z^+-Rule

The z^+-rule is obtained by choosing the nearest root on the segment $(\{x_i 1_{w_{ij}<0}\}, \{x_i\})$. If $\{x_i\}$ is in $\mathbb{R}^d_+$, then, the segment is also in the domain $\mathbb{R}^d_+$. The relevance function has a root at the first extremity of the segment:

$$\begin{aligned} R_j(\{x_i 1_{w_{ij}<0}\}) &= \max(0, \textstyle\sum_i x_i 1_{w_{ij}<0} w_{ij} + b_j) \\ &= \max(0, \textstyle\sum_i x_i w_{ij}^- + b_j) = 0, \end{aligned}$$

since $x_i \geq 0$ and $w_{ij}^- \leq 0$, and therefore $x_i w_{ij}^- \leq 0$, and since $b_j < 0$ by design. The direction of this segment on which we search for the nearest root is given by:

$$\begin{aligned} v_i^{(j)} &= x_i - x_i 1_{w_{ij}<0} \\ &= x_i 1_{w_{ij} \geq 0}. \end{aligned}$$

Injecting this search direction into Equation 7, and defining $z_{ij}^+ = x_i w_{ij}^+$ with $w_{ij}^+ = 1_{w_{ij} \geq 0} w_{ij}$, we get

$$R_i = \frac{z_{ij}^+}{\sum_i z_{ij}^+} R_j.$$

1.4. $z^{\mathcal{B}}$-Rule

The $z^{\mathcal{B}}$-rule is obtained by choosing the nearest root on the segment $(\{l_i 1_{w_{ij}>0} + h_i 1_{w_{ij}<0}\}, \{x_i\})$. Provided that $\{x_i\}$ is in $\mathcal{B}$, the segment is also in $\mathcal{B}$. The relevance function has a root at the first extremity of the segment:

$$\begin{aligned} &R_j(\{l_i 1_{w_{ij}>0} + h_i 1_{w_{ij}<0}\}) \\ &\qquad = \max(0, \textstyle\sum_i l_i 1_{w_{ij}>0} w_{ij} + h_i 1_{w_{ij}<0} w_{ij} + b_j) \\ &\qquad = \max(0, \textstyle\sum_i l_i w_{ij}^+ + h_i w_{ij}^- + b_j) = 0, \end{aligned}$$

because all summed terms are either negative or the product of a negative and positive value. The search direction for this choice of segment is given by

$$v_i^{(j)} = x_i - l_i 1_{w_{ij}>0} - h_i 1_{w_{ij}<0}$$

Injecting this search direction in to Equation 7, we get

$$R_i = \sum_j \frac{z_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-}{\sum_i z_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-} R_j.$$

2. Algorithms for Propagation Rules

We give here algorithms to implement the rules derived in Section 1 of the supplement. A useful property of these rules is that they can all be expressed in terms of matrix multiplications, thus, making them easily implementable with numerical libraries such as Matlab or Python/Numpy.

2.1. w^2-Rule

```
Input:
    Weight matrix W = $\{w_{ij}\}$
    Upper-layer relevance vector R = $\{R_j\}$

Procedure:
    V ← W ⊙ W
    N ← V ⊘ ([1] · V)
    return N · R
```

where ⊙ and ⊘ denote the element-wise multiplication and division respectively, and `[1]` is a matrix of ones. Note that for efficiency purposes, the squaring and normalization of the weight matrix can be performed once, and reused for many heatmaps computations.

2.2. z-Rule

```
Input:
    Weight matrix $\mathtt{W} = \{w_{ij}\}$
    Input activations $\mathtt{X} = \{x_i\}$
    Upper-layer relevance vector $\mathtt{R} = \{R_j\}$

Procedure:
    $\mathtt{Z} \leftarrow \mathtt{W}^\top \mathtt{X}$
    return $\mathtt{X} \odot (\mathtt{W} \cdot (\mathtt{R} \oslash \mathtt{Z}))$
```

where $\odot$ and $\oslash$ denote the element-wise multiplication and division respectively, and where the variable Z is the sum of weighted activations for each upper-layer neuron.

2.3. z^+-Rule

```
Input:
    Weight matrix $\mathtt{W} = \{w_{ij}\}$
    Input activations $\mathtt{X} = \{x_i\}$
    Upper-layer relevance vector $\mathtt{R} = \{R_j\}$

Procedure:
    $\mathtt{V} \leftarrow \mathtt{W}^+$
    $\mathtt{Z} \leftarrow \mathtt{V}^\top \mathtt{X}$
    return $\mathtt{X} \odot (\mathtt{V} \cdot (\mathtt{R} \oslash \mathtt{Z}))$
```

where $\odot$ and $\oslash$ denote the element-wise multiplication and division respectively, and where the operation $(\cdot)^+$ keeps the positive part of the input matrix. For efficiency, like for the w^2-rule, the matrix V can be precomputed and reused for multiple heatmaps computations.

2.4. $z^{\mathcal{B}}$-Rule

```
Input:
    Weight matrix $\mathtt{W} = \{w_{ij}\}$
    Input activations $\mathtt{X} = \{x_i\}$
    Upper-layer relevance vector $\mathtt{R} = \{R_j\}$
    Lower-bound $\mathtt{L} = \{l_i\}$
    Upper-bound $\mathtt{H} = \{h_i\}$

Procedure:
    $\mathtt{U} \leftarrow \mathtt{W}^-$
    $\mathtt{V} \leftarrow \mathtt{W}^+$
    $\mathtt{N} \leftarrow \mathtt{R} \oslash (\mathtt{W}^\top \mathtt{X} - \mathtt{V}^\top \mathtt{L} - \mathtt{U}^\top \mathtt{H})$
    return $\mathtt{X} \odot (\mathtt{W} \cdot \mathtt{N}) - \mathtt{L} \odot (\mathtt{V} \cdot \mathtt{N}) - \mathtt{H} \odot (\mathtt{U} \cdot \mathtt{N})$
```

where $\odot$ and $\oslash$ denote the element-wise multiplication and division respectively, and where the operations $(\cdot)^+, (\cdot)^-$ keep the positive part and the negative part of the input matrix respectively. For efficiency, like for the previous rules, the matrices U and V can be precomputed and reused for multiple heatmaps computations.

3. Proofs of Propositions

Definition 1. *A heatmapping $\boldsymbol{R}(\boldsymbol{x})$ is conservative if the sum of assigned relevances in the pixel space corresponds to the total relevance detected by the model, that is*

$$\forall \boldsymbol{x}: \; f(\boldsymbol{x}) = \sum_p R_p(\boldsymbol{x}).$$

Definition 2. *A heatmapping $\boldsymbol{R}(\boldsymbol{x})$ is positive if all values forming the heatmap are greater or equal to zero, that is:*

$$\forall \boldsymbol{x}, p: \; R_p(\boldsymbol{x}) \geq 0$$

Definition 3. *A heatmapping $\boldsymbol{R}(\boldsymbol{x})$ is consistent if it is conservative and positive. That is, it is consistent if it complies with Definitions 1 and 2.*

Proposition 1. *For all $g \in \mathcal{G}$, the deep Taylor decomposition with the w^2-rule is consistent in the sense of Definition 3.*

Proof. We first show that the heatmapping is conservative:

$$\begin{aligned} \sum_i R_i &= \sum_i \Big(\sum_j \frac{w_{ij}^2}{\sum_i w_{ij}^2} R_j \Big) \\ &= \sum_j \frac{\sum_i w_{ij}^2}{\sum_i w_{ij}^2} R_j = \sum_j R_j = \sum_j x_j = f(x). \end{aligned}$$

where we have assumed the weights to be never exactly zero. Then, we show that the heatmapping is positive:

$$R_i = \sum_j \frac{w_{ij}^2}{\sum_i w_{ij}^2} R_j = \sum_j \underbrace{w_{ij}^2}_{>0} \cdot \underbrace{\frac{1}{\sum_i w_{ij}^2}}_{>0} \cdot \underbrace{R_j}_{\geq 0} \geq 0.$$

Therefore, because the heatmapping is both conservative and positive, it is also consistent.

For the case where $\sum_i w_{ij}^2 = 0$, it implies that $w_{ij} = 0$ for all i and therefore $z_{ij} = 0$ for all i too. Because $b_j \leq 0$, then $R_j = x_j = 0$ (there is no relevance to redistribute to the lower-layer). □

Proposition 2. *For all $g \in \mathcal{G}$ and data points $\{x_i\} \in \mathbb{R}_+^d$, the deep Taylor decomposition with the z^+-rule is consistent in the sense of Definition 3.*

Proof. The proof is the same as for Proposition 1 for the case where $\sum_i z_{ij}^+ > 0$. We simply replace w_{ij}^2 by z_{ij}^+ in the proof.

For the case where $\sum_i z_{ij}^+ = 0$, it implies that $z_{ij} \leq 0$ for all i. Because $b_j \leq 0$, then $R_j = x_j = 0$ (there is no relevance to redistribute to the lower-layer). □

Proposition 3. *For all $g \in \mathcal{G}$ and data points $\{x_i\} \in \mathcal{B}$, the deep Taylor decomposition with the $z^{\mathcal{B}}$-rule is consistent in the sense of Definition 3.*

Proof. We first show that the numerator of the $z^{\mathcal{B}}$-rule $q_{ij} = z_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-$ is greater or equal than zero for $\{x_i\} \in \mathcal{B}$:

$$\begin{aligned} q_{ij} &= z_{ij} - l_i w_{ij}^+ - h_i w_{ij}^- \\ &= x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^- \\ &= x_i (w_{ij}^- + w_{ij}^+) - l_i w_{ij}^+ - h_i w_{ij}^- \\ &= \underbrace{(x_i - h_i)}_{\leq 0} \cdot \underbrace{w_{ij}^-}_{\leq 0} + \underbrace{(x_i - l_i)}_{\geq 0} \cdot \underbrace{w_{ij}^+}_{\geq 0} \geq 0 \end{aligned}$$

Then, the proof is the same as for Proposition 1 for the case where $\sum_i q_{ij} > 0$. We simply replace w_{ij}^2 by q_{ij} in the proof. For the case where $\sum_i q_{ij} = 0$, this equality implies that $\forall_i : q_{ij} = 0$, which can be satisfied by one of the three sets of conditions:

1. $x_i = h_i$ and $w_{ij}^+ = 0$. In that case, the contribution of the input to the detection neuron is $z_{ij} = h_i w_{ij}$, and because $h_i \geq 0$, then $z_{ij} \leq 0$.
2. $x_i = l_i$ and $w_{ij}^- = 0$. In that case, the contribution of the input to the detection neuron is $z_{ij} = l_i w_{ij}$, and because $l_i \leq 0$, then $z_{ij} \leq 0$.
3. $w_{ij} = 0$. In that case, the contribution is $z_{ij} = 0$.

Therefore, inputs are prevented from contributing positively to the neuron x_j. In particular, the total contribution is given by $z_j = \sum_i z_{ij} \leq 0$. Because $b_j \leq 0$, then $R_j = x_j = 0$ (there is no relevance to redistribute to the lower-layer). □

4. Empirical Comparison with LRP

In this section, we compare heatmaps produced by the rules based on deep Taylor decomposition, and the layer-wise relevance propagation (LRP) rules proposed by [1]. The LRP rules include in particular, the $\alpha\beta$-rule:

$$R_i = \sum_j \Big(\alpha \frac{z_{ij}^+}{\sum_i z_{ij}^+ + b_j^+} - \beta \frac{z_{ij}^-}{\sum_i z_{ij}^- + b_j^-} \Big) R_j,$$

where $\alpha - \beta = 1$, and the ϵ-stabilized rule:

$$R_i = \sum_j \frac{z_{ij}}{s(\sum_i z_{ij} + b_j)} R_j,$$

where $s(t) = t + \epsilon(1_{t \geq 0} - 1_{t<0})$ is a stabilizing function whose output is never zero. The respective free hyperparameters α and ϵ of these rules are typically selected such that the produced heatmaps have the desired quality.

Figure 2 compares heatmaps obtained by applying deep Taylor decomposition (min-max) and LRP (same rule in both layers) to the neural network of the MNIST experiments. Figure 3 compares heatmaps obtained by deep Taylor decomposition (training-free) and LRP (same rule in all layers) on the BVLC CaffeNet and the GoogleNet. Normalization layers are ignored in the backward pass.

It can be observed that the quality of the deep Taylor heatmaps is less influenced by the choice of model and dataset than LRP with a fixed set of parameters. Deep Taylor heatmaps look similar in all cases. LRP also produces high-quality heatmaps, but the best parameters differ in each setting. For example, the parameters $\alpha = 2, \beta = 1$ perform well for the CaffeNet, but tend to produce too sparse heatmaps for the GoogleNet, or to produce a large amount of negative relevance on MNIST. Various parameters of LRP produce various artefacts such as the presence of residual relevance on the irrelevant digit, or the presence of negative relevance in the black areas surrounding the digits. On the other hand, LRP-based heatmaps are sharper than Taylor-based heatmaps and less subject to the stride artefact that arises with convolutional neural networks. Future work will seek to identify the reason for the superiority of LRP on these particular aspects, and investigate whether the deep Taylor decomposition method and its underlying principles can be refined to incorporate these desirable properties of a heatmap while retaining stability.

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.

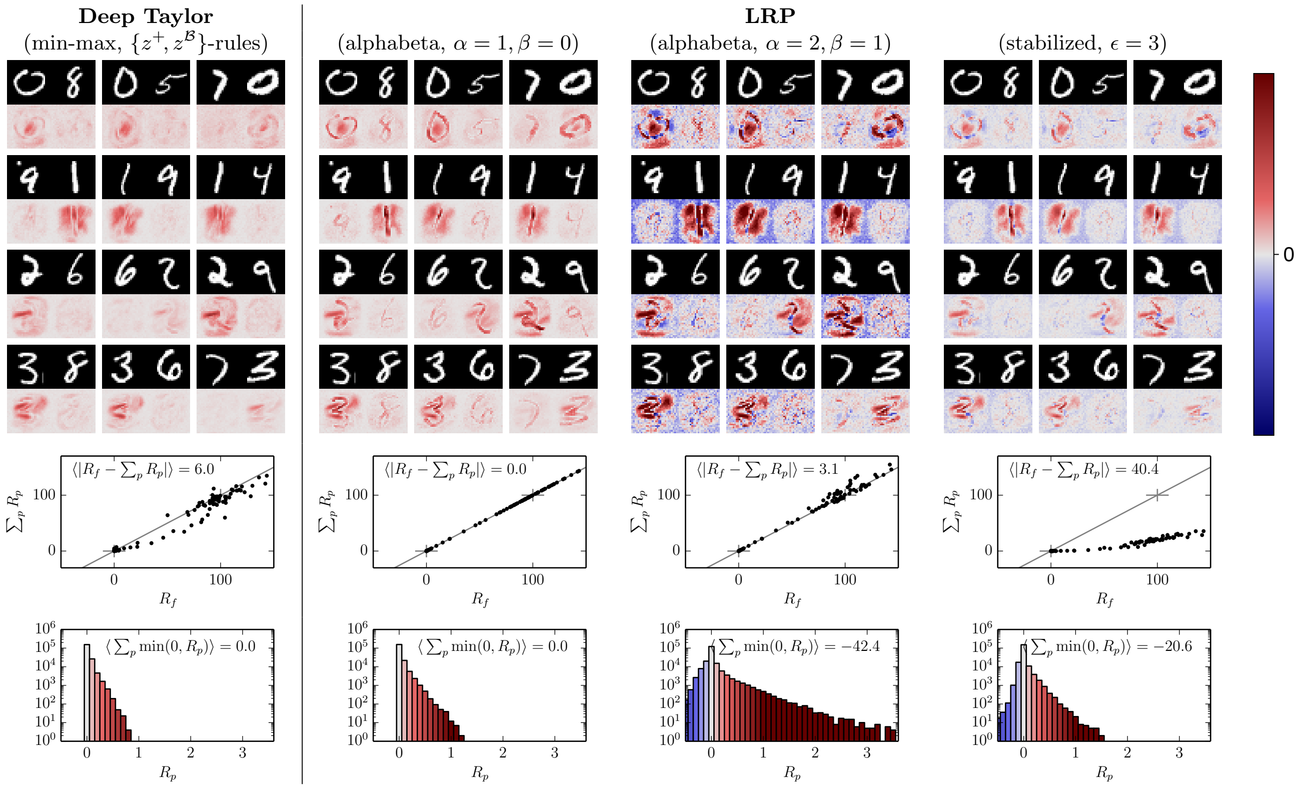

Figure 2: Heatmaps and their properties produced by various heatmapping techniques applied on the MNIST network of the paper.

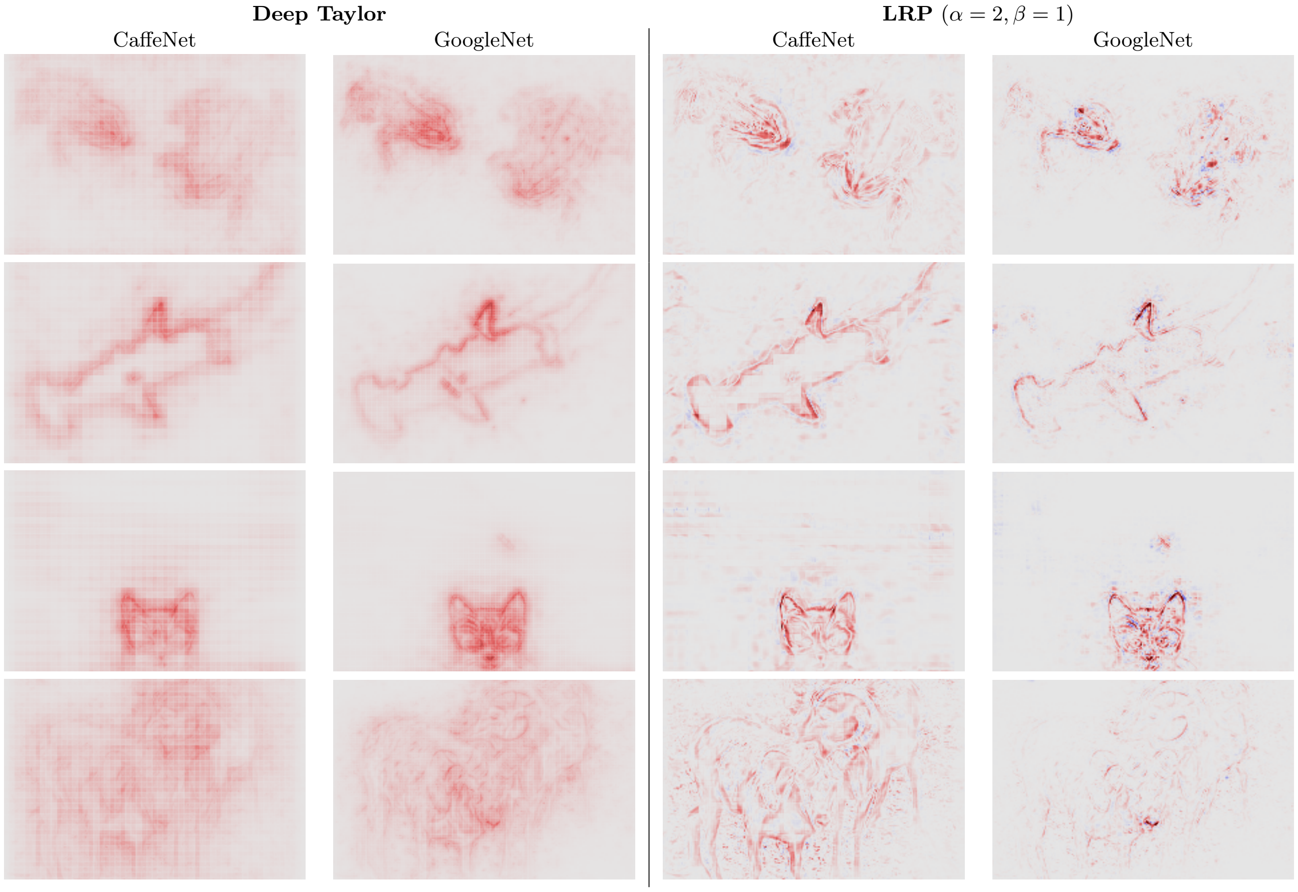

Figure 3: Heatmaps produced by deep Taylor decomposition and LRP when applied to the predictions of the CaffeNet and GoogleNet networks.