When Explanations Lie: Why Many Modified BP Attributions Fail

Leon Sixt[1] **Maximilian Granz**[1] **Tim Landgraf**[1]

Abstract

Attribution methods aim to explain a neural network's prediction by highlighting the most relevant image areas. A popular approach is to backpropagate (BP) a custom relevance score using modified rules, rather than the gradient. We analyze an extensive set of modified BP methods: Deep Taylor Decomposition, Layer-wise Relevance Propagation (LRP), Excitation BP, PatternAttribution, DeepLIFT, Deconv, RectGrad, and Guided BP. We find empirically that the explanations of all mentioned methods, except for DeepLIFT, are independent of the parameters of later layers. We provide theoretical insights for this surprising behavior and also analyze why DeepLIFT does not suffer from this limitation. Empirically, we measure how information of later layers is ignored by using our new metric, cosine similarity convergence (CSC). The paper provides a framework to assess the faithfulness of new and existing modified BP methods theoretically and empirically. [2]

1. Introduction

Explainable AI (XAI) aims to improve the interpretability of machine learning models. For deep convolutional networks, attribution methods visualize the areas relevant for the prediction with so-called saliency maps. Various attribution methods have been proposed, but do they reflect the model behavior correctly?

(Adebayo et al., 2018) proposed a sanity check: if the parameters of the model are randomized and therefore the network output changes, do the saliency maps change too? Surprisingly, the saliency maps of GuidedBP (Springenberg

[1]Dahlem Center of Machine Learning and Robotics, Freie Universität Berlin, Germany. Correspondence to: Leon Sixt <leon.sixt@fu-berlin.de>.

[2]For code see: github.com/berleon/when-explanations-lie

(a) Sanity Checks (VGG-16)

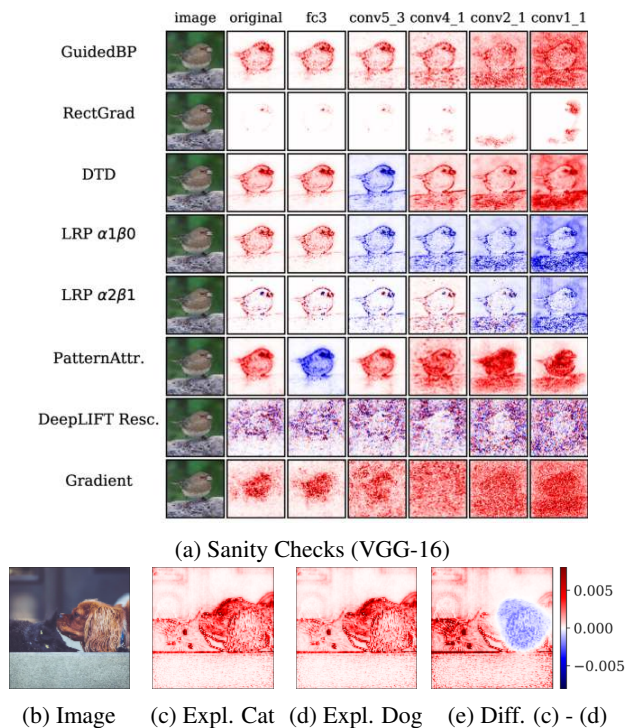(b) Image   (c) Expl. Cat   (d) Expl. Dog   (e) Diff. (c) - (d)

Figure 1: **(a)** Sanity Checks: Saliency maps should change if network parameters are randomized. Parameters are randomized from the last to the first layer. *Red* denotes positive and *blue* negative relevance. **(b-e)** Class insensitivity of LRP$_{\alpha 1 \beta 0}$ on VGG-16. Explanation for **(c)** *Persian cat (283)* and **(d)** *King Charles Spaniel (156)*. **(e)** Difference (c) - (d), both normalized to $[0, 1]$. L1-norm of (e) = 0.000371.

et al., 2014) stay identical, when the last layer (fc3) is randomized (see Figure 1a). A method ignoring the last layer can *not* explain the network's prediction faithfully.

In addition to (Adebayo et al., 2018), which only reported GuidedBP to fail, we found several modified backpropagation (BP) methods fail too: Layer-wise Relevance Propagation (LRP), Deep Taylor Decomposition (DTD), PatternAttribution, Excitation BP, Deconv, GuidedBP, and RectGrad (Bach et al., 2015; Montavon et al., 2017; Kindermans et al., 2018; Zhang et al., 2018; Zeiler & Fergus, 2014; Springenberg et al., 2014; Kim et al., 2019). The only tested modified BP method passing is DeepLIFT (Shrikumar et al., 2017).

Modified BP methods estimate relevant areas by backpropagating a custom relevance score instead of the gradient. For example, DTD only backpropagates positive relevance scores. Modified BP methods are popular with practitioners (Yang et al., 2018; Sturm et al., 2016; Eitel et al., 2019). For example, (Schiller et al., 2019) uses saliency maps to improve the classification of whale sounds or (Böhle et al., 2019) use $\text{LRP}_{\alpha 1 \beta 0}$ to localize evidence for Alzheimer's disease in brain MRIs.

Deep neural networks are composed of linear layers (dense, conv.) and non-linear activations. For each linear layer, the weight vector reflects the importance of each input directly. (Bach et al., 2015; Kindermans et al., 2018; Montavon et al., 2017) argue that aggregating explanations of each linear model can explain a a deep neural network. Why do these methods then fail the sanity check?

Theoretically, we show that the z^+-rule – used by DTD, $\text{LRP}_{\alpha 1 \beta 0}$, and Excitation BP – yields a multiplication chain of non-negative matrices. Each matrix corresponds to a layer. The saliency map is a function of this matrix chain. We show that such a non-negative matrix chain converges to a rank-1 matrix. If $C \in \mathbb{R}^{n \times m}$ is a rank-1 matrix, then it can be written as an outer product $C = c \gamma^T$, $c \in \mathbb{R}^n$, $\gamma \in \mathbb{R}^m$. Multiplying C with any vector v yields always the same the direction: $Cv = c\gamma^T v = \lambda c, \lambda \in \mathbb{R}$. The scaling is irrelevant as saliency maps are normalized. If sufficiently converged, the backpropagated vector can merely switch the sign of the saliency map. For example, in Figure 1a, the sign of the PatternAttribution saliency map switches due to the randomization of fc3. Figure 1b-1e show how the saliency maps of $\text{LRP}_{\alpha 1 \beta 0}$ become class-insensitive.

Empirically, we quantify the convergence to a rank-1 matrix using our novel cosine similarity convergence (CSC) metric. CSC allows to retrace, layer by layer, how modified BP methods lose information about previous layers. Using CSC, we observe that all analyzed modified BP methods, except for DeepLIFT, converge towards a rank-1 matrix on VGG-16 and ResNet-50. For sufficiently large values of α and β, $\text{LRP}_{\alpha \beta}$ does not converge but also produces rather noisy saliency maps.

The paper focuses on modified BP methods, as other attribution methods do not suffer from the converges problem. They either rely on the gradient directly (Smilkov et al., 2017; Sundararajan et al., 2017), which does not converge or consider the model as a black-box (Ribeiro et al., 2016; Lundberg & Lee, 2017).

Our findings show that many modified BP methods are prone to class-insensitive explanations and provide saliency maps that rather highlight low-level features. Negative relevance scores are crucial to avoid the convergence to a rank-1 matrix — a possible future research direction.

2. Theoretical Analysis

Notation   For our theoretical analysis, we consider feed-forward neural networks with a ReLU activation function $[x]^+ = \max(0, x)$. The neural network $f(x)$ contains n layers, each with weight matrices W_l. The output of the l-th layer is denoted by h_l. We use $[ij]$ to index the i, j element in W_l as in $W_{l_{[ij]}}$. To simplify notation, we absorb the bias terms into the weight matrix, and we omit the final softmax layer. We refer to the input with $h_0 = x$ and to the output with $h_n = f(x)$. The output of the l-th layer is given by:

$$h_l = [W_l h_{l-1}]^+ \tag{1}$$

All the results apply to convolutional neural networks as convolution can be expressed as matrix multiplication.

Gradient   The gradient of the k-th output of the neural network w.r.t. the input x is given by:

$$\frac{\partial f_k(x)}{\partial x} = W_1^T M_1 \frac{\partial f_k(x)}{\partial h_1} = \prod_l^n \left(W_l^T M_l \right) \cdot v_k, \tag{2}$$

where $M_l = \text{diag}(1_{h_l > 0})$ denotes the gradient mask of the ReLU operation. The last equality follows from recursive expansion. The vector v_k is a one-hot vector to select the k-th output.

The gradient of residual blocks is also a product of matrices. The gradient of $h_{l+1} = h_l + g(h_l)$ is:

$$\frac{\partial h_{l+1}}{\partial h_l} = I + G_{\partial g(h_l)/\partial h_l}, \tag{3}$$

where $G_{\partial g(h_l)/\partial h_l}$ denotes the derivation matrix of the residual block, and I is the identity matrix. For the gradient, the final saliency map is usually obtained by summing the absolute channel values of the relevance vector $r_0^\nabla(x)$ of the input layer.

The following methods modify the gradient definition and to distinguish the rules, we introduce the notation: $r_l^\nabla(x) = \frac{\partial f(x)}{\partial h_l}$ which denotes the relevance at layer l for an input x.

Interpretability of Linear Models   The relevance of the input of a linear model can be calculated directly. Let $y = w^T x$ be a linear model with a single output scalar. The relevance of the input x to the i-th output $y_{[i]}$ is :

$$r_x^{\text{Linear}}(x) = w \odot x. \tag{4}$$

2.1. z^+-Rule

The z^+-rule is used by DTD (Montavon et al., 2017), Excitation BP (Zhang et al., 2018) and also corresponds to the $\text{LRP}_{\alpha 1 \beta 0}$ rule (Bach et al., 2015). The z^+-rule backpropagates positive relevance values, which are supposed to
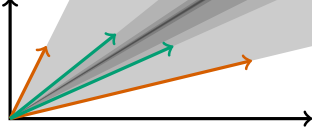
Figure 2: The positive column vectors $\boldsymbol{a}_1, \boldsymbol{a}_2$ of matrix A_1 (orange) form a cone. The resulting columns of $A_1 A_2$ (green) are contained in the cone as they are positive linear combinations of $\boldsymbol{a}_1, \boldsymbol{a}_2$. At each iteration, the cone shrinks.

correspond to the positive evidence for the prediction. Let w_{ij} be an entry in the weight matrix W_l:

$$r_l^{z^+}(\boldsymbol{x}) = Z_l^+ \cdot r_{l+1}^{z^+}(\boldsymbol{x})$$
$$\text{where } Z_l^{+^T} = \left(\frac{[w_{ij} \boldsymbol{h}_{l_{[j]}}]^+}{\sum_k [w_{ik} \boldsymbol{h}_{l_{[k]}}]^+} \right)_{[ij]} \quad (5)$$

Each entry in the derivation matrix Z_l^+ is obtained by measuring the positive contribution of the input neuron i to the output neuron j and normalizing by the total contributions to neuron j. The relevance from the previous layer $r_{l+1}^{z^+}$ is then distributed according to Z_l^+. The relevance function $r_l^{z^+} : \mathbb{R}^n \mapsto \mathbb{R}^m$ maps input $\boldsymbol{x}$ to a relevance vector of layer l. For the final layer the relevance is set to the value of the explained logit value, i.e. $r_n^{z^+}(\boldsymbol{x}) = f_k(\boldsymbol{x})$. In contrast to the vanilla backpropagation, algorithms using the z^+-rule do not apply a mask for the ReLU activation.

The relevance of multiple layers is computed by applying the z^+-rule to each of them. Similar to the gradient, we obtain a product of non-negative matrices: $C_k = \prod_l^k Z_l^+$.

Theorem 1. *Let $A_1, A_2, A_3 \ldots$ be a sequence of non-negative matrices for which $\lim_{n \to \infty} A_n$ exists. We exclude the cases where one column of $\lim_{n \to \infty} A_n$ is the zero vector or two columns are orthogonal to each other. Then the product of all terms of the sequence converges to a rank-1 matrix $\bar{C}$:*

$$\bar{C} := \prod_{i=1}^{\infty} A_i = \bar{\boldsymbol{c}} \boldsymbol{\gamma}^T . \quad (6)$$

(Hajnal, 1976; Friedland, 2006) proved a similar result for squared matrices. In appendix A, we provide a rigorous proof of the theorem using the cosine similarity.

The geometric intuition of the proof is depicted in Figure 2. The column vectors of the first matrix are all non-negative and therefore in the positive quadrant. For the matrix multiplication $A_i A_j$, observe that $A_i \boldsymbol{a}_k$ is a non-negative linear combination of the column vectors of A_i, where $\boldsymbol{a}_k$ is the k-th column vector $A_{j_{[:k]}}$. The result will remain in the convex cone of the column vectors of A_i. The conditions stated in the theorem ensure that the cone shrinks with every iteration and it converges towards a single vector. In

the appendix B, we simulate different matrix properties and find non-negative matrices to converge exponentially fast.

The column vectors of a rank-1 matrix are linearly dependent $C = \boldsymbol{c} \boldsymbol{\gamma}^T$. A rank-1 matrix C always gives the same direction of $\boldsymbol{c}$: $C Z_{k+1}^+ = \boldsymbol{c} \boldsymbol{\gamma}^T Z_{k+1}^+ = \boldsymbol{c} \boldsymbol{\lambda}^T$ and for any vector $\boldsymbol{v}$: $C Z_{k+1}^+ \boldsymbol{v} = \boldsymbol{c} \boldsymbol{\lambda}^T \boldsymbol{v} = t \boldsymbol{c}$, where $t \in \mathbb{R}$. For a finite number of matrices $C_k = \prod_l^k Z_l^+$, C_k might resemble a rank-1 matrix up to floating-point imprecision or $C_k Z_{k+1}^+$ might still be able to alter the direction. In any case, the influence of later matrices decreases.

The Z^+ matrices of dense layers fulfill the conditions of theorem 1. Convolutions can be written as matrix multiplications. For 1x1 convolutions, the kernels do not overlap and the row vectors corresponding to each location are orthogonal. In this case, the convergence happens only locally per feature map location. For convolutions with overlapping kernels, the global convergence is slower than for dense layers. In a ResNet-50 where the last convolutional stack has a size of (7x7), the overlapping of multiple (3x3) convolutions still induces a considerable global convergence (see LRP_{CMP} on ResNet-50 in section 5).

If an attribution method converges, the contributions of the layers shrink by depth. In the worst-case scenario, when converged up to floating-point imprecision, the last layer can only change the scaling of the saliency map. However, the last layer is responsible for the network's final prediction.

2.2. Modified BP algorithms

LRP$_z$ The LRP$_z$ rule of Layer-wise Relevance Propagation modifies the backpropagation rule as follows:

$$r_l^{z-\text{LRP}}(\boldsymbol{x}) = Z_l \cdot r_{l+1}^{z-\text{LRP}}(\boldsymbol{x}),$$
$$\text{where } Z_l = \left(\frac{w_{ij} \boldsymbol{h}_{l_{[j]}}}{\sum_k w_{ik} \boldsymbol{h}_{l_{[k]}}} \right)_{[ij]}^T . \quad (7)$$

If only max-pooling, linear layers, and ReLU activations are used, it was shown that LRP$_z$ corresponds to gradient⊙input, i.e. $r_0^{z-\text{LRP}}(\boldsymbol{x}) = \boldsymbol{x} \odot \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$ (Shrikumar et al., 2016; Kindermans et al., 2016; Ancona et al., 2017). LRP$_z$ can be considered a gradient-based and not a modified BP method. The gradient is not converging to a rank-1 matrix and therefore gradient⊙input is also not converging.

LRP$_{\alpha\beta}$ separates the positive and negative influences:

$$r_l^{\alpha\beta}(\boldsymbol{x}) = \left(\alpha Z_l^+ - \beta Z_l^- \right) r_{l+1}^{\alpha\beta}(\boldsymbol{x}), \quad (8)$$

where Z_l^+ and Z_l^- correspond to the positive and negative entries of the matrix Z. (Bach et al., 2015) propose to weight positives more: $\alpha \geq 1$ and $\alpha - \beta = 1$. For LRP$_{\alpha 1 \beta 0}$, this rule corresponds to the z^+-rule, which converges. For
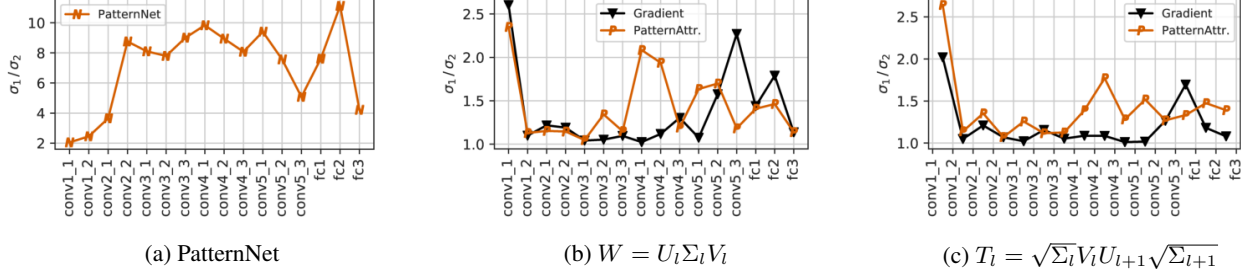
(a) PatternNet

(b) $W = U_l \Sigma_l V_l$

(c) $T_l = \sqrt{\Sigma_l} V_l U_{l+1} \sqrt{\Sigma_{l+1}}$

Figure 3: PatternNet & PatternAttr.: **(a)(b)** Ratio between the first and second singular value σ_1/σ_2 for A_l, W_l, and $A_l \odot W_l$. **(c)** σ_1/σ_2 of inter-layer derivation matrices. For (b) (c), we sliced the 3x3 convolutional kernels to 1x1 kernels.

$\alpha > 1$ and $\beta > 0$, the matrix $Z_l = \alpha Z_l^+ - \beta Z_l^-$ can contain negative entries. Our empirical results show that LRP$_{\alpha\beta}$ still converges for the most commonly used parameters $\alpha = 2, \beta = 1$ and even for a higher $\alpha = 5$ it converges considerable on the ResNet-50.

Deep Taylor Decomposition uses the z^+-rule if the input to a convolutional or dense layer is in $[0, \infty]$, i.e. if the layer follows a ReLU activation. For inputs in $\mathbb{R}$, DTD also proposed the w^2-rule and the so-call $w^\mathcal{B}$ rule for bounded inputs. Both rules were specifically designed to produce non-negative outputs. Theorem 1 applies and DTD converges to a rank-1 matrix necessarily.

PatternNet & PatternAttribution takes into account that the input h_l contains noise. If d_l corresponds to the noise and s_l to the signal, than $h_l = s_l + d_l$. To assign the relevance towards the signal direction, it is estimated using the following equation:

$$a_i = \frac{\text{cov}[h] w_i}{w_i^T \text{cov}[h] w_i}, \qquad (9)$$

where a_i is the estimated signal direction for the $i-th$ neuron with input h and weight vector $w_i = W_{[i:]}$. PatternNet is designed to recover the relevant signal in the data. Let $A_{l[i:]} = a_i$ be the corresponding signal matrix to the weight matrix W_l, the rule for PatternNet is:

$$r_l^{\text{PN}}(x) = A_l^T \cdot r_{l+1}^{\text{PN}}(x), \qquad (10)$$

PatternNet is also prone to converge to a rank-1 matrix. To recover the relevant signal, it might be even desired to converge to the a single direction – the signal direction.

The convergence of PatternNet follows from the computation of the pattern vectors a_i in equation 9. It is similar to a single step of the power iteration method $v_{k+1} = Cv_k / \|Cv_k\|$. In appendix C, we provide details on the relationship to power iteration and also derive equation 9 from the equation given in (Kindermans et al., 2018). The power iteration method converges to the eigenvector with the largest eigenvalue exponentially fast.

All column vectors in $A_{[i:]} = a_i$ underwent a single step of the power iteration and therefore tend to point towards the first eigenvector of $\text{cov}[h]$. This can also be verified empirically: the ratio of the first and second singular value $\sigma_1(A)/\sigma_2(A) > 6$ for almost all the VGG-16 patterns (see Figure 3a), indicating a strong convergence of the matrix chain towards a single direction.

The findings from PatternNet are hard to transfer to PatternAttribution. The rule for PatternAttribution uses the Hadamard product of A_l and W_l:

$$r_l^{\text{PA}}(x) = (W_l \odot A_l)^T \cdot r_{l+1}^{\text{PA}}(x), \qquad (11)$$

The Hadamard product complicates any analytic argument using the properties of A_l or W_l. The theoretical results available (Ando et al., 1987; Zhan, 1997) did not allow us to show that PatternAttribution converges to a rank-1 matrix necessarily.

We provide a mix of theoretical and empirical insights on why it converges. The conditions of convergence can be studied well on the singular value decomposition: $(W_l \odot A_l)^T = U_l \Sigma_l V_l$. Loosely speaking, the matrix chain will converge to a rank-1 matrix if the first σ_1 and second σ_2 singular values in Σ_l differ and if V_l and U_{l+1} are aligned such that higher singular values of Σ_l and Σ_{l+1} are multiplied together such that the ratio σ_1/σ_2 grows.

To see how well the per layer matrices align, we look at the inter-layer chain members: $T_l = \sqrt{\Sigma_l} V_l U_{l+1} \sqrt{\Sigma_{l+1}}$. In Figure 3, we display the ratio between the first and second singular values $\sigma_1(T_l)/\sigma_2(T_l)$. For $W \odot A$, the first singular value is considerably larger than for the plain weights W. Interestingly, the singular value ratio of inter-layer matrices shrinks for the plain W matrix. Whereas for PatternAttribution, the ratio increases for some layers indicating that the Hadamard product leads to more alignment of the matrices.

DeepLIFT is the only tested modified BP method which does not converge to a rank-1 matrix. It is an extension of

the backpropagation algorithm to finite differences:

$$\frac{f(\boldsymbol{x}) - f(\boldsymbol{x}^0)}{\boldsymbol{x} - \boldsymbol{x}^0} \tag{12}$$

For the gradient, one would take the limit $\boldsymbol{x}^0 \to \boldsymbol{x}$. DeepLIFT uses a so-called reference point for $\boldsymbol{x}^0$ instead, such as zeros or for images a blurred version of $\boldsymbol{x}$. The finite differences are backpropagated, similar to infinitesimal differences. The final relevance is the difference in the k-th logit: $r_l^{DL}(\boldsymbol{x}) = f_k(x) - f_k(x^0)$.

Additionally to the vanilla gradient, DeepLIFT separates positive and negative contributions. For ReLU activations, DeepLIFT uses either the RevealCancel or the Rescale rule. Please refer to (Shrikumar et al., 2017) for a description. The rule for linear layers is most interesting because it is the reason why DeepLIFT does not converge:

$$\begin{aligned} r_l^{DL+}(\boldsymbol{x}, \boldsymbol{x}^0) =& M_{>0}^T \odot \left(W_l^{+^T} r_{l+1}^{DL+}(\boldsymbol{x}, \boldsymbol{x}^0) \right. \\ & \left. + W_l^{-^T} r_{l+1}^{DL-}(\boldsymbol{x}, \boldsymbol{x}^0) \right) \end{aligned} \tag{13}$$

where the mask $M_{>0}$ selects the weight rows corresponding to positive deltas ($0 < \Delta \boldsymbol{h}_l = \boldsymbol{h}_l - \boldsymbol{h}_l^0$). For negative relevance r_l^{DL-}, the rule is defined analogously. An interesting property of the rule (13) is that negative and positive relevance can influence each other.

If the intermixing is removed by only considering W^+ for the positive rule and W^- for the negative rule, the two matrix chains become decoupled and converge. For the positive chain, this is clear. For the negative chain, observe that the multiplication of two non-positive matrices gives a non-negative matrix. Non-positive vectors $\boldsymbol{b}, \boldsymbol{c}$ have an angle $\leq 90°$ and $\boldsymbol{c}^T \boldsymbol{b} = \|\boldsymbol{c}\| \|\boldsymbol{b}\| \cos(\boldsymbol{c}, \boldsymbol{b}) \geq 0$. In the evaluation, we included this variant as *DeepLIFT Ablation*, and as predicted by the theory, it converges.

Guided BP & Deconv & RectGrad   apply an additional ReLU to the gradient and it was shown to be invariant to the randomization of later layers previously in (Adebayo et al., 2018) and analyzed theoretically in (Nie et al., 2018):

$$r_l^{GBP}(\boldsymbol{x}) = W_l^T \left[M_l r_{l+1}^{GBP}(\boldsymbol{x}) \right]^+ . \tag{14}$$

$M_l = \operatorname{diag}(1_{h_1 > 0})$ denotes the gradient mask of the ReLU operation. For Deconv, the mask of the forward ReLU is omitted, and the gradients are rectified directly. RectGrad (Kim et al., 2019) is related to GuidedBP and set the lowest q percentile of the gradient to zero. As recommended in the paper, we used $q = 98$.

As a ReLU operation is applied to the gradient, the backpropagation is no longer a linear function. The ReLU also results in a different failure than before. (Nie et al., 2018) provides a theoretical analysis for GuidedBP. Our results align with them.

3. Evaluation

Setup   We report results on a small network trained on CIFAR-10 (4x conv., 2x dense, see appendix D), a VGG-16 (Simonyan & Zisserman, 2014), and ResNet-50 (He et al., 2016). The last two are trained on the ImageNet dataset (Russakovsky et al., 2015), the standard dataset to evaluate attribution methods. The different networks cover different concepts: shallow vs. deep, forward vs. residual connections, multiple dense layers vs. a single one, using batch normalization. All results were computed on 200 images from the validation set. To justify the sample size, we show bootstrap confidence intervals in Figure 4b (Efron, 1979). We used the implementation from the *innvestigate* and *deeplift* package (Alber et al., 2019; Shrikumar et al., 2017) and added support for residual connections. The experiments were run on a single machine with two graphic cards and take about a day to complete.

Random Logit   We display the difference of saliency maps explaining the ground-truth and a random logit in Figure 4a. As the logit value is responsible for the predicted class, the saliency maps should change. We use the SSIM metric (Wang et al., 2004) as in (Adebayo et al., 2018).

Sanity Check   We followed (Adebayo et al., 2018) and randomized the parameters starting from the last layer to the first layer. For DTD and LRP$_{\alpha 1 \beta 0}$, randomizing the last layer flips the sign of the saliency map sometimes. We, therefore, compute the SSIM also between the inverted saliency map and report the maximum. In Figure 4b, we report the SSIM between the saliency maps (see also Figure 1a and appendix G).[1]

Cosine Similarity Convergence Metric (CSC)   Instead of randomizing the parameters, we randomize the backpropagated relevance vectors directly. We select layer k and set the corresponding relevance to $r_k(\boldsymbol{x}) := \boldsymbol{v}_1$ where $\boldsymbol{v}_1 \sim \mathcal{N}(0, I)$ and then backpropagate it as before. For example, for the gradient, we would do: $\frac{\partial h_k}{\partial h_1} \frac{\partial f(x)}{\partial h_k} := \frac{\partial h_k}{\partial h_1} \boldsymbol{v}_1$. We use the notation $r_l(\boldsymbol{x} | r_k := \boldsymbol{v}_1)$ to describe the relevance r_l at layer l when the relevance of layer k is set to $\boldsymbol{v}_1$.

Using two random relevance vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \sim \mathcal{N}(0, I)$, we measure the convergence using the cosine similarity. A rank-1 matrix $C = \boldsymbol{c}\gamma^T$ always yields the same direction: $C\boldsymbol{v} = \boldsymbol{c}\gamma^T = \lambda \boldsymbol{c}$. If the matrix chain converges, the backpropagated relevance vectors of $\boldsymbol{v}_1, \boldsymbol{v}_2$ will align more and more. We quantify their alignment using the cosine similarity $s_{\cos}(r_l(\boldsymbol{x} | r_k := \boldsymbol{v}_1), r_l(\boldsymbol{x} | r_k := \boldsymbol{v}_2)))$ where

---

[1] For GuidedBP, we report different saliency maps than shown in Figure 2 of (Adebayo et al., 2018). We were able to confirm a bug in their implementation, resulting in saliency maps of GuidedBP and Guided-GradCAM to remain identical for early layers.

(a) Random Logits
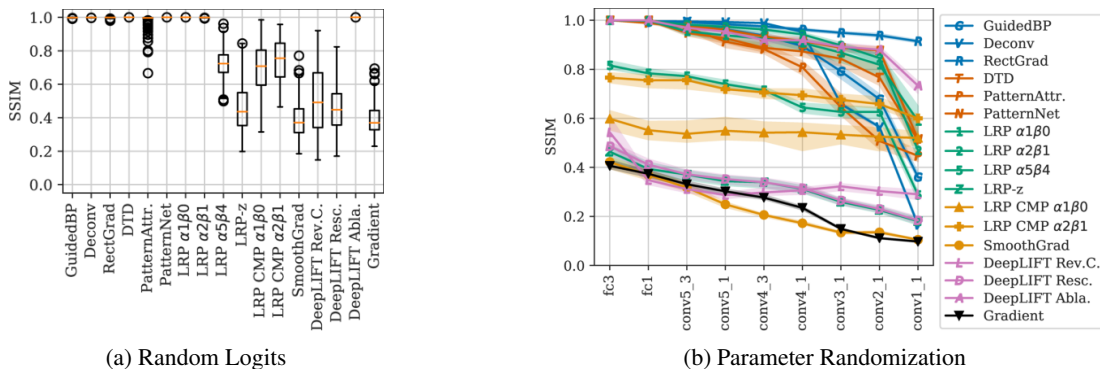
(b) Parameter Randomization

Figure 4: **(a)** SSIM between saliency maps explaining the ground-truth or a random logit. **(b)** The parameters of the VGG-16 are randomized, starting from the last to the first layer. SSIM quantifies the difference to the saliency map from the original model. Intervals show 99% bootstrap confidences.

$$s_{\cos}(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}^T \boldsymbol{b} \,/\, (\|\boldsymbol{a}\| \cdot \|\boldsymbol{b}\|).$$

Suppose the relevance matrix chain would converge to a rank-1 matrix perfectly, than we have for both $\boldsymbol{v}_1, \boldsymbol{v}_2$: $r_l(\boldsymbol{x}|r_k{=}\boldsymbol{v}_i) = C\boldsymbol{v}_i = c\boldsymbol{\gamma}^T \boldsymbol{v}_i = \lambda_i \boldsymbol{c}$ where $\lambda_i = \boldsymbol{\gamma}^T \boldsymbol{v}_i$ and their cosine similarity will be one. The opposite direction is also true. If C has shape $n \times m$ with $n \leq m$ and if for n linearly independent vectors $\boldsymbol{v}_i$, the cosine similarity $s_{\cos}(C\boldsymbol{v}_i, C\boldsymbol{v}_j) = 1$, then C is a rank-1 matrix.

An alternative way to measure convergence would have been to construct the derivation matrix $C_k = \prod_{l=1}^k Z_l$ and measure the ratio $\sigma_1(C_k)/\sigma_2(C_k)$ of the first to the second-largest singular value of C_k. Although this approach is well motivated theoretically, it has some performance downsides. C_k would be large and computing the singular values costly.

We use five different random vectors per sample – in total 1000 convergence paths. As the vectors are sampled randomly, it is unlikely to miss a region of non-convergence (Bergstra & Bengio, 2012).

For convolution layers, we compute the cosine similarity per feature map location. For a shape of (h, w, c), we obtain $h \cdot w$ values. The jump in cosine similarity for the input is a result of the input's low dimension of 3 channels. In Figure 5, we plot the median cosine similarity for different networks and attribution methods (see appendix F for additional Figures). We also report the histogram of the CSC at the first convolutional layer in Figures 5e-5g.

4. Results

Our random logit analysis reveals that converging methods produce almost identical saliency maps, independently of the output logit (SSIM very close to 1). The rest of the field (SSIM between 0.4 and 0.8) produces saliency maps different from the ground-truth logit's map (see Figure 4a).

We observe the same distribution in the sanity check results (see Figure 4b). One group of methods produces similar saliency maps even when convolutional layers are randomized (SSIM close to 1). Again, the rest of the field is sensitive to parameter randomization. The same clustering can be observed for ResNet-50 (appendix E, Figure 8).

Our CSC analysis confirms that random relevance vectors align throughout the backpropagation steps (see Figure 5). Except for LRP$_z$ and DeepLIFT, all methods show convergence up to at least 0.99 cosine similarity. LRP$_{\alpha5\beta4}$ converges less strongly for VGG-16. Among the converging methods, the rate of convergence varies. LRP$_{\alpha1\beta0}$, PatternNet, the ablation of DeepLIFT converges fastest. PatternAttribution has a slower convergence rate – still exponential. For DeepLIFT Ablation, numerical instabilities result in a cosine similarity of 0 for the first layers of the ResNet-50. Even on the small 6-layer network, the median CSC is greater than 1-1e-6 for LRP$_{\alpha1\beta0}$ (see Figure 5d).

5. Discussion

When many modified BP methods do not explain the network faithfully, why was this not widely noticed before? First, it is easy to blame the network for unreasonable explanations – no ground truth exists. Second, MNIST, CIFAR, and ImageNet contain only a single object class per image – not revealing the class insensitivity. Finally, it might not be too problematic for some applications if the saliency maps are independent of the later network's layers. For example, to explain Alzheimer's disease (Böhle et al., 2019), local low-level features are sufficient as they are predictive for the disease and the data lacks conflicting evidences (i.e. the whole brain is affected).

When noticed, different ways to address the issue were proposed and an improved class sensitivity was reported (Kohlbrenner et al., 2019; Gu et al., 2018; Zhang et al.,

(a) ResNet-50

(b) VGG-16 (logarithmic)

(c) VGG-16 (linear)

(d) CIFAR-10

(e) VGG-16
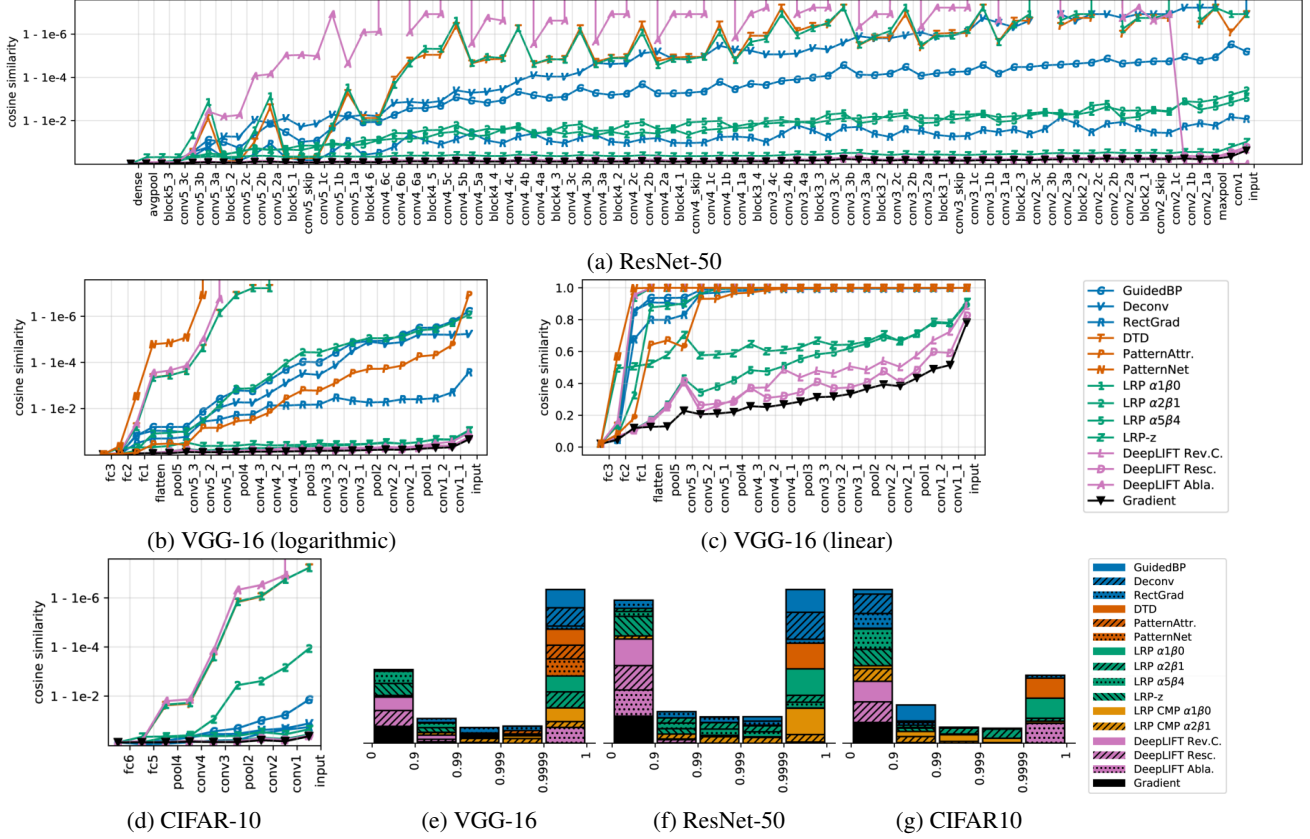
(f) ResNet-50

(g) CIFAR10

Figure 5: **(a)-(d)** Median of the cosine similarity convergence (CSC) per layer between relevance vectors obtained from randomizing the relevance vectors of the final layer. **(e)-(g)** histogram of the distribution of the CSC after the first layer.

2018). We find that the underlying convergence problem remains unchanged and discuss the methods below.

LRP$_{CMP}$ (Kohlbrenner et al., 2019; Lapuschkin et al., 2017) use LRP$_z$ for the final dense layers and LRP$_{\alpha\beta}$ for the convolutional layer. We report results for $\alpha = 1, 2$ as in (Kohlbrenner et al., 2019) in Figure 6a.

For VGG-16, the saliency maps change when the network parameters are randomized. However, structurally, the underlying image structure seems to be scaled only locally (see Figure 6a). Inspecting the CSC path of the two LRP$_{CMP}$ variants in Figure 6c, we can see why. For dense layers, both methods do not converge as LRP$_z$ is used, but the convergence start when LRP$_{\alpha\beta}$ is applied. The relevance vectors of the dense layer can change the coarse local scaling. However, they cannot alter the direction of the relevance vectors of earlier layers to highlight different details.

In the backward-pass of the ResNet-50, the global-averaging layer assigns the identical gradient vector to each location of the last convolutional layer. Furthermore, the later convolutional layers operate on (7x7), where even a few 3x3 convolutions have a dense field-of-view. LRP$_{CMP}$

does not resolve the global convergence for the ResNet-50.

Contrastive LRP   (Gu et al., 2018) noted the lack of class sensitivity and proposed to increase it by subtracting two saliency maps. The first saliency map explains only the logit $\boldsymbol{y}_k = \boldsymbol{y} \odot \boldsymbol{m}_k$, where $\boldsymbol{m}_k$ is a one-hot vector and the second explains the opposite $\boldsymbol{y}_{\neg k} = \boldsymbol{y} \odot (1 - \boldsymbol{m}_k)$:

$$\max(0, n(r_{\boldsymbol{x}}^{z^+}(\boldsymbol{x}|r_{\text{logits}} \coloneqq \boldsymbol{y}_k)) \\ - n(r_{\boldsymbol{x}}^{z^+}(\boldsymbol{x}|r_{\text{logits}} \coloneqq \boldsymbol{y}_{\neg k})) \tag{15}$$

$n(.)$ normalizes each saliency map by its sum. The results of Contrastive LRP are similar to Figre 1e, no $\max$ is applied. The underlying convergence problem is not resolved.

Contrastive Excitation BP   The lack of class sensitivity of the z^+-rule was noted in (Zhang et al., 2018) and to increase it, they proposed to change the backpropagation rule of the final fully-connected layer to:

$$r_{\text{final fc}}^{\text{cEBP}}(x) = (Z_{\text{final fc}}^+ - N_{\text{final fc}}^+)\boldsymbol{m}_k, \tag{16}$$

where $\boldsymbol{m}_k$ is a one-hot vector selecting the explained class. The added $N_{\text{final fc}}^+$ is computed as the $Z_{\text{final fc}}^+$ but on the negative weights $-W_{\text{final fc}}$. Note that the combination of the
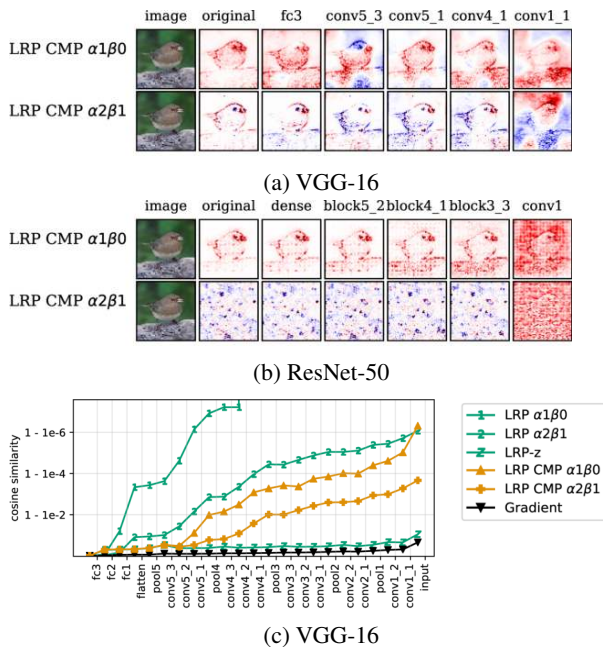
(a) VGG-16

(b) ResNet-50

(c) VGG-16

Figure 6: **(a-b)** Sanity checks and **(c)** CSC for LRP_{CMP}.

two matrices introduces negative entries. Class sensitivity is increased. It does also not resolve the underlying convergence problem. If, for example, more fully-connected layers would be used, the saliency maps would become globally class insensitive again.

Texture vs. Contours   (Geirhos et al., 2019) found that deep convolutional networks are more sensitive towards texture and not the shape of the object. For example, the shape of a cat filled with an elephant texture will be wrongly classified as an elephant. However, modified BP methods highlight the contours of objects rather.

Recurrent Neural Networks   Modified BP methods are focused on convolutional neural networks and are mostly applied on vision tasks. The *innvestigate* package does not yet support recurrent models. To our knowledge, (Arras et al., 2017) is the only work that applied modified BP rules to RNNs (LRP_z for LSTMs). Training, and applying modified backpropagation rules to RNNs, involves unrolling the network, essentially transforming it to a feed-forward architecture. Due to our theoretical results, modified BP rules that yield positive relevance matrices (e.g. z^+-rule) will converge. However, further work would be needed to measure how RNN architectures (LSTM, GRU) differ in their specific convergence behavior.

Not Converging Attribution Methods   Besides modified BP attribution methods, there also exist gradient averaging and black-box methods. *SmoothGrad* (Smilkov et al., 2017) and *Integrated Gradients* (Sundararajan et al., 2017) aver-

age the gradient. *CAM* and *Grad-CAM* (Zhou et al., 2016; Selvaraju et al., 2017) determine important areas by the activation of the last convolutional layer. Black-box attribution methods only modify the model's input but do not rely on the gradient or other model internals. The most prominent black-box methods are *Occlusion*, *LIME*, *SHAP* (Zeiler & Fergus, 2014; Ribeiro et al., 2016; Lundberg & Lee, 2017). *IBA* (Schulz et al., 2020) applies an information bottleneck to remove unimportant information. *TCAV* (Kim et al., 2018) explains models using higher-level concepts.

All here mentioned attribution methods do *not* converge, as they either rely on the gradient or treat the model as black-box. Only when the BP algorithm is modified, the convergence problem can occur. The here mentioned algorithms might still suffer from other limitations.

Limitations   Also, we tried to include most modified BP attribution methods, we left some out for our evaluation (Nam et al., 2019; Wang et al., 2019; Huber et al., 2019). In our theoretical analysis of PatternAttribution, we based our argument on why it converges on empirical observations performed on a single set of pattern matrices.

6. Related Work

Limitations of attribution   The limitations of explanation methods were studied before. (Viering et al., 2019) alter the explanations of Grad-CAM arbitrarily by modifying the model architecture only slightly. Similarly, (Slack et al., 2020) construct a biased classifier that can hide its biases from LIME and SHAP. The theoretic analysis (Nie et al., 2018) indicates that GuidedBP tends to reconstruct the input instead of explaining the network's decision. (Adebayo et al., 2018) showed GuidedBP to be independent of later layers' parameters. (Atrey et al., 2020) tested saliency methods in a reinforcement learning setting.

(Kindermans et al., 2018) show that LRP, GuidedBP, and Deconv produce incorrect explanations for linear models if the input contains noise. (Rieger, 2017; Zhang et al., 2018; Gu et al., 2018; Kohlbrenner et al., 2019; Montavon et al., 2019; Tsunakawa et al., 2019) noted the class-insensitivity of different modified BP methods, but they rather proposed ways to improve the class sensitivity than to provide correct reasons why modified BP methods are class insensitive. Other than argued in (Gu et al., 2018), the class insensitivity is not caused by missing ReLU masks and Pooling switches. To the best of our knowledge, we are the first to identify the reason why many modified BP methods do not explain the decision of deep neural networks faithfully.

Evaluation metrics for attribution   As no ground-truth data exists for feature importance, different proxy tasks were proposed to measure the performance of attribution

algorithms. One approach is to test how much relevance falls into ground-truth bounding boxes (Schulz et al., 2020; Zhang et al., 2018).

The *MoRF* and *LeRF* evaluation removes the *mo*st and *le*ast relevant input features and measures the change in model performance (Samek et al., 2016). The relevant image parts are masked usually to zero. On these modified samples, the model might not be reliable. The *ROAR* score improves it by retraining the model from scratch (Hooker et al., 2018). While computationally expensive, it ensures the model performance does not drop due to out-of-distribution samples. The ROAR performance of Int.Grad. and GuidedBP is equally bad, worse than a random baseline (see Figre 4 in (Hooker et al., 2018)). Thus, ROAR does not separate converging from non-converging methods.

Our CSC measure has some similarities with the work (Balduzzi et al., 2017), which analyzes the effect of skip connections on the gradient. They measure the convergence between the gradient vector from different samples using the effective rank (Vershynin, 2012). The CSC metric applies to modified BP methods and is an efficient tool to trace the degree of convergence.

A different approach to verify attribution methods is to measure how helpful they are for humans (Alqaraawi et al., 2020; Doshi-Velez & Kim, 2017; Lage et al., 2018).

7. Conclusion

In our paper, we analyzed modified BP methods, which aim to explain the predictions of deep neural networks. Our analysis revealed that most of these attribution methods have theoretical properties contrary to their goal. PatternAttribution and LRP cite Deep Taylor Decomposition as the theoretical motivation. In the light of our results, revisiting the theoretical derivation of Deep Taylor Decomposition may prove insightful. Our theoretical analysis stresses the importance of negative relevance values. A possible way to increase class-sensitivity and resolve the convergence problem could be to backpropagate negative relevance similar to DeepLIFT, the only method passing our test.

8. Acknowledgements

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pp. 9505–9515, 2018.

Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., Samek, W., Müller, K.-R., Dähne, S., and Kindermans, P.-J. innvestigate neural networks! *Journal of Machine Learning Research*, 20(93):1–8, 2019.

Alqaraawi, A., Schuessler, M., Weiß, P., Costanza, E., and Berthouze, N. Evaluating saliency map explanations for convolutional neural networks: A user study. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, pp. 263–274, New York, NY, USA, 2020. Association for Computing Machinery. doi: 10.1145/3377325.3377519.

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. A unified view of gradient-based attribution methods for deep neural networks. In *NIPS 2017-Workshop on Interpreting, Explaining and Visualizing Deep Learning*. ETH Zurich, 2017.

Ando, T., Horn, R. A., and Johnson, C. R. The singular values of a hadamard product: a basic inequality. *Linear and Multilinear Algebra*, 21(4):345–365, 1987. doi: 10.1080/03081088708817810.

Arras, L., Montavon, G., Müller, K.-R., and Samek, W. Explaining Recurrent Neural Network Predictions in Sentiment Analysis. In *Proceedings of the EMNLP 2017 Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 159–168. Association for Computational Linguistics, 2017.

Atrey, A., Clary, K., and Jensen, D. Exploratory Not Explanatory: Counterfactual Analysis of Saliency Maps for Deep Reinforcement Learning. In *International Conference on Learning Representations*, April 2020.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7), 2015. doi: 10.1371/journal.pone.0130140.

Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., and McWilliams, B. The shattered gradients problem: If resnets are the answer, then what is the question?

In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 342–350. JMLR.org, 2017.

Bergstra, J. and Bengio, Y. Random search for hyperparameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

Böhle, M., Eitel, F., Weygandt, M., and Ritter, K. Layerwise relevance propagation for explaining deep neural network decisions in mri-based alzheimer's disease classification. *Frontiers in Aging Neuroscience*, 11:194, 2019. ISSN 1663-4365. doi: 10.3389/fnagi.2019.00194.

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv: 1702.08608*, 2017.

Efron, B. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. ISSN 00905364.

Eitel, F., Soehler, E., Bellmann-Strobl, J., Brandt, A. U., Ruprecht, K., Giess, R. M., Kuchling, J., Asseyer, S., Weygandt, M., Haynes, J.-D., Scheel, M., Paul, F., and Ritter, K. Uncovering convolutional neural network decisions for diagnosing multiple sclerosis on conventional mri using layer-wise relevance propagation. *NeuroImage: Clinical*, 24, 2019. doi: https://doi.org/10.1016/j.nicl.2019.102003.

Friedland, S. Convergence of products of matrices in projective spaces. *Linear Algebra and its Applications*, 413 (2):247 – 263, 2006. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2004.06.021. Special Issue on the 11th Conference of the International Linear Algebra Society, Coimbra, 2004.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

Gu, J., Yang, Y., and Tresp, V. Understanding individual decisions of cnns via contrastive backpropagation. In *Asian Conference on Computer Vision*, pp. 119–134. Springer, 2018.

Hajnal, J. On products of non-negative matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 79(3):521–530, May 1976. ISSN 0305-0041, 1469-8064.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. Evaluating Feature Importance Estimates. *arXiv: 1806.10758*, 2018.

Huber, T., Schiller, D., and André, E. Enhancing explainability of deep reinforcement learning through selective layerwise relevance propagation. In *KI 2019: Advances in Artificial Intelligence*, pp. 188–202, Cham, 2019. Springer International Publishing. ISBN 978-3-030-30179-8.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pp. 2673–2682, 2018.

Kim, B., Seo, J., Jeon, S., Koo, J., Choe, J., and Jeon, T. Why are Saliency Maps Noisy? Cause of and Solution to Noisy Saliency Maps. *arXiv: 1902.04893*, 2019.

Kindermans, P.-J., Schütt, K., Müller, K.-R., and Dähne, S. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv: 1611.07270*, 2016.

Kindermans, P.-J., Schütt, K. T., Alber, M., Müller, K.-R., Erhan, D., Kim, B., and Dähne, S. Learning how to explain neural networks: Patternnet and patternattribution. In *International Conference on Learning Representations*, 2018.

Kohlbrenner, M., Bauer, A., Nakajima, S., Binder, A., Samek, W., and Lapuschkin, S. Towards best practice in explaining neural network decisions with lrp, 2019.

Lage, I., Ross, A., Gershman, S. J., Kim, B., and Doshi-Velez, F. Human-in-the-loop interpretability prior. In *Advances in Neural Information Processing Systems*, pp. 10159–10168, 2018.

Lapuschkin, S., Binder, A., Muller, K.-R., and Samek, W. Understanding and comparing deep neural networks for age and gender classification. In *The IEEE International Conference on Computer Vision Workshops (ICCVW)*, Oct 2017.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc., 2017.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K.-R. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

Montavon, G., Binder, A., Lapuschkin, S., Samek, W., and Müller, K.-R. Layer-wise relevance propagation: an

overview. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 193–209. Springer, 2019.

Nam, W.-J., Gur, S., Choi, J., Wolf, L., and Lee, S.-W. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. *arXiv:1904.00605*, Nov 2019.

Nie, W., Zhang, Y., and Patel, A. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *International Conference on Machine Learning*, pp. 3806–3815, 2018.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778.

Rieger, L. Separable explanations of neural network decisions. In *NIPS 2017 Workshop on Interpretable Machine Learning*, 2017.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

Schiller, D., Huber, T., Lingenfelser, F., Dietz, M., Seiderer, A., and André, E. Relevance-Based Feature Masking: Improving Neural Network Based Whale Classification Through Explainable Artificial Intelligence. In *Proc. Interspeech 2019*, pp. 2423–2427, 2019.

Schulz, K., Sixt, L., Tombari, F., and Landgraf, T. Restricting the flow: Information bottlenecks for attribution. In *International Conference on Learning Representations*, 2020.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. Not just a black box: Learning important features through propagating activation differences. *arXiv: 1605.01713*, 2016.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3145–3153. JMLR.org, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Slack, D., Hilgard, S., Jia, E., Singh, S., and Lakkaraju, H. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, pp. 180–186, New York, NY, USA, 2020. Association for Computing Machinery. doi: 10.1145/3375627.3375830.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. *arXiv: 1706.03825*, 2017.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for Simplicity: The All Convolutional Net. *arXiv: 1412.6806*, 2014.

Sturm, I., Lapuschkin, S., Samek, W., and Müller, K.-R. Interpretable deep neural networks for single-trial eeg classification. *Journal of neuroscience methods*, 274:141–145, 2016.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3319–3328. JMLR. org, 2017.

Tsunakawa, H., Kameya, Y., Lee, H., Shinya, Y., and Mitsumoto, N. Contrastive relevance propagation for interpreting predictions by a single-shot object detector. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, July 2019. doi: 10.1109/IJCNN.2019.8851770.

Vershynin, R. *Introduction to the non-asymptotic analysis of random matrices*, pp. 210–268. Cambridge University Press, 2012. doi: 10.1017/CBO9780511794308.006.

Viering, T., Wang, Z., Loog, M., and Eisemann, E. How to manipulate cnns to make them lie: the gradcam case, 2019.

Wang, S., Zhou, T., and Bilmes, J. Bias also matters: Bias attribution for deep neural network explanation. In *International Conference on Machine Learning*, pp. 6659–6667, 2019.

Wang, Z., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P., et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Yang, Y., Tresp, V., Wunderle, M., and Fasching, P. A. Explaining therapy predictions with layer-wise relevance propagation in neural networks. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 152–162, June 2018.

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

Zhan, X. Inequalities for the Singular Values of Hadamard Products. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1093–1095, October 1997. ISSN 0895-4798, 1095-7162. doi: 10.1137/S0895479896309645.

Zhang, J., Bargal, S. A., Lin, Z., Brandt, J., Shen, X., and Sclaroff, S. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10): 1084–1102, 2018.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2016.