

Understanding Individual Decisions of CNNs via Contrastive Backpropagation

Jindong Gu^{1,2}, Yinchong Yang², Volker Tresp^{1,2}

¹ the University of Munich, Germany

²Siemens AG, Corporate Technology, Munich, Germany

Abstract. A number of backpropagation-based approaches such as DeConvNets, vanilla Gradient Visualization and Guided Backpropagation have been proposed to better understand individual decisions of deep convolutional neural networks. The saliency maps produced by them are proven to be non-discriminative. Recently, the Layer-wise Relevance Propagation (LRP) approach was proposed to explain the classification decisions of rectifier neural networks. In this work, we evaluate the discriminativeness of the generated explanations and analyze the theoretical foundation of LRP, i.e. Deep Taylor Decomposition. The experiments and analysis conclude that the explanations generated by LRP are not class-discriminative. Based on LRP, we propose Contrastive Layer-wise Relevance Propagation (CLRP), which is capable of producing instance-specific, class-discriminative, pixel-wise explanations. In the experiments, we use the CLRP to explain the decisions and understand the difference between neurons in individual classification decisions. We also evaluate the explanations quantitatively with a Pointing Game and an ablation study. Both qualitative and quantitative evaluations show that the CLRP generates better explanations than the LRP.

1 Introduction

Deep convolutional neural networks (DCNNs) achieve start-of-the-art performance on many tasks, such as visual object recognition[1,2,3], and object detection[4,5]. However, since they lack transparency, they are considered as "black box" solutions. Recently, research on explainable deep learning has received increased attention: Many approaches have been proposed to crack the "black box". Some of them aim to interpret the components of a deep-architecture model and understand the image representations extracted from deep convolutional architectures [6,7,8]. Examples are Activation Maximization [9,10], DeConvNets Visualization [11]. Others focus on explaining the individual classification decisions. Examples are Prediction Difference Analysis [12,13], Guided Backpropagation [10,14], Layer-wise Relevance Propagation (LRP) [15,16], Class Activation Mapping [17,18] and Local Interpretable Model-agnostic Explanations [19,20].

More concretely, the models in [21,17] were originally proposed to detect object only using category labels. They work by producing saliency maps of objects corresponding to the category labels. Their produced saliency maps can

also explain the classification decisions to some degree. However, the approaches can only work on the model with a specific architecture. For instance, they might require a fully convolutional layer followed by a max-pooling layer, a global average pooling layer or an aggregation layer, before a final softmax output layer. The requirement is not held in most off-the-shelf models e.g., in [1,2]. The perturbation methods [19,20,12] require no specific architecture. For a single input image, however, they require many instances of forward inference to find the corresponding classification explanation, which is computationally expensive.

The backpropagation-based approaches [10,14,15] propagate a signal from the output neuron backward through the layers to the input space in a single pass, which is computationally efficient compared to the perturbation methods. They can also be applied to the off-the-shelf models. In this paper, we focus on the backpropagation approaches. The outputs of the backpropagation approaches are instance-specific because these approaches leverage the instance-specific structure information (ISSInfo). The ISSInfo, equivalent to *bottleneck* information in [22], consist of selected information extracted by the forward inference, i.e., the Pooling switches and ReLU masks. With the ISSInfo, the backpropagation approaches can generate instance-specific explanations. A note on terminology: although the terms "sensitivity map", "saliency map", "pixel attribution map" and "explanation heatmap" may have different meanings in different contexts, in this paper, we do not distinguish them and use the term "saliency map" and "explanation" interchangeably.

The primal backpropagation-based approaches, e.g., the vanilla Gradient Visualization [10] and the Guided Backpropagation [14] are proven to be inappropriate to study the neurons of networks because they produce non-discriminative saliency maps [22]. The saliency maps generated by them mainly depend on ISSInfo instead of the neuron-specific information. In other words, the generated saliency maps are not class-discriminative with respect to class-specific neurons in output layer. The saliency maps are selective of any recognizable foreground object in the image [22]. Furthermore, the approaches cannot be applied to understand neurons in intermediate layers of DCNNs, either. In [11,23], the differences between neurons of an intermediate layer are demonstrated by a large dataset. The neurons are often activated by certain specific patterns. However, the difference between single neurons in an individual classification decision has not been explored yet. In this paper, we will also shed new light on this topic.

The recently proposed Layer-wise Relevance Propagation (LRP) approach is proven to outperform the gradient-based approaches [16]. Apart from explaining image classifications[24,16], the LRP is also applied to explain the classifications and predictions in other tasks [25,26]. However, the explanations generated by the approach has not been fully verified. We summarise our three-fold contributions as follows:

- 1 We first evaluate the explanations generated by LRP for individual classification decisions. Then, we analyze the theoretical foundation of LRP, i.e., Deep Taylor Decomposition and shed new insight on LRP.

- 2 We propose Contrastive Layer-wise Relevance Propagation (CLRP). To generate class-discriminative explanations, we propose two ways to model the contrastive signal (i.e., an opposite visual concept). For individual classification decisions, we illustrate explanations of the decisions and the difference between neuron activations using the proposed approach.
- 3 We build a GPU implementation of LRP and CLRP using Pytorch Framework, which alleviates the inefficiency problem addressed in [27,13].

Related work is reviewed in the next section. Section 3 analyzes LRP theoretically and experimentally. In Section 4, the proposed approach CLRP is introduced. Section 5 shows experimental results to evaluate the CLRP qualitatively and quantitatively on two tasks, namely, explaining the image classification decisions and understanding the difference of neuron activations in single forward inference. The last section contains conclusions and discusses future work.

2 Related Work

The DeConvNets were originally proposed for unsupervised feature learning tasks [28]. Later they were applied to visualize units in convolutional networks [11]. The DeConvNets maps the feature activity to input space using ISSInfo and the weight parameters of the forward pass. [10] proposed identifying the vanilla gradients of the output with respect to input variables as their relevance. The work also showed its relation to the DeConvNets. They use the ISSInfo in the same way except for the handling of rectified linear units (ReLUs) activation function. The Guided Backpropagation [14] combine the two approaches to visualize the units in higher layers.

The paper [15] propose LRP to generate the explanations for classification decisions. The LRP propagates the class-specific score layer by layer until to input space. The different propagation rules are applied according to the domain of the activation values. [16] proved that the Taylor Expansions of the function at the different points result in the different propagation rules. Recently, one of the propagation rules in LRP, z -rule, has been proven to be equivalent to the vanilla gradients (saliency map in [10]) multiplied elementwise with the input [29]. The vanilla Gradient Visualization and the Guided Backpropagation are shown to be not class-discriminative in [22]. This paper rethinks the LRP and evaluates the explanations generated by the approach.

Existing work that is based on discriminative and pixel-wise explanations are [30,27,18]. The work Guided-CAM [18] combines the low-resolution map of CAM and the pixel-wise map of Guided Backpropagation to generate a pixel-wise and class-discriminative explanation. To localize the most relevant neurons in the network, a biologically inspired attention model is proposed in [31]. The work uses a top-down (from the output layer to the intermediate layers) Winner-Take-All process to generate binary attention maps. The work [27] formulate the top-down attention of a CNN classifier as a probabilistic Winner-Take-All process. The work also uses a contrastive top-down attention formulation to enhance the discriminativeness of the attention maps. Based on their work and the LRP,

we propose Contrastive Layer-wise Relevance Propagation (CLRP) to produce class-discriminative and pixel-wise explanations. Another publication related to our approach is [30], which is able to produce class-discriminative attention maps. While the work [30] requires modifying the traditional CNNs by adding extra feedback layers and optimizing the layers during the backpropagation, our proposed methods can be applied to all exiting CNNs without any modification and further optimization.

3 Rethinking Layer-wise Relevance Propagation

Each neuron in DCNNs represents a nonlinear function $X_i^{L+1} = \phi(\mathbf{X}^L \mathbf{W}_i^L + \mathbf{b}_i^L)$, where ϕ is an activation function and \mathbf{b}_i^L is a bias for the neuron X_i^{L+1} . The inputs of the nonlinear function corresponding to a neuron are the activation values of the previous layer \mathbf{X}_i or the raw input of the network. The output of the function are the activation values of the neuron X_i^{L+1} . The whole network are composed of the nested nonlinear functions.

To identify the relevance of each input variables, the LRP propagates the activation value from a single class-specific neuron back into the input space, layer by layer. The logit before softmax normalization is taken, as explained in [10,15]. In each layer of the backward pass, given the relevance score \mathbf{R}^{L+1} of the neurons \mathbf{X}^{L+1} , the relevance R_i^L of the neuron X_i^L are computed by redistributing the relevance score using local redistribution rules. The most often used rules are the z^+ -rule and the z^β -rule, which are defined as follows:

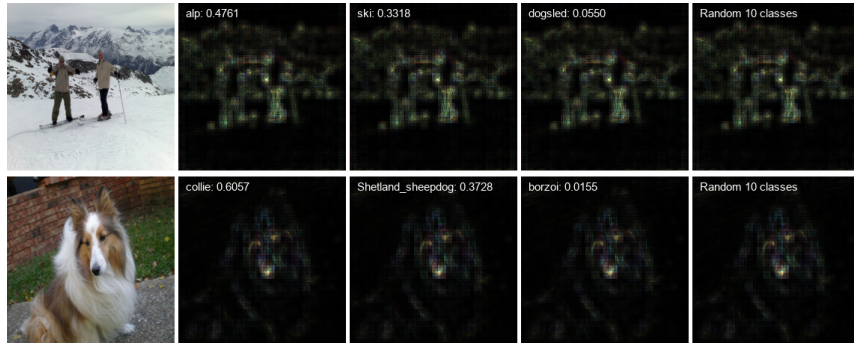
$$\begin{aligned} z^+ \text{-rule: } R_i^L &= \sum_j \frac{x_i w_{ij}^+}{\sum_{i'} x_{i'} w_{i'j}^+} R_j^{L+1} \\ z^\beta \text{-rule: } R_i^L &= \sum_j \frac{x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-}{\sum_{i'} x_{i'} w_{i'j} - l_{i'} w_{i'j}^+ - h_{i'} w_{i'j}^-} R_j^{L+1} \end{aligned} \quad (1)$$

where w_{ij} connecting X_i^L and X_j^{L+1} is a parameter in L -th layer, $w_{ij}^+ = w_{ij} * 1_{w_{ij}>0}$ and $w_{ij}^- = w_{ij} * 1_{w_{ij}<0}$, and the interval $[l, h]$ is the domain of the activation value x_i .

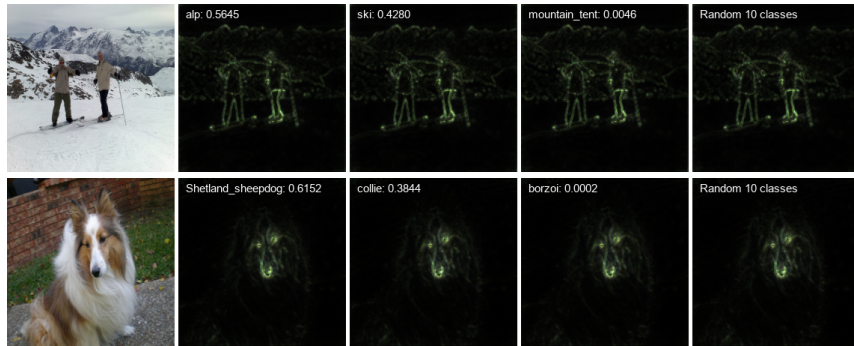
3.1 Evaluation of the Explanations Generated by the LRP

The explanations generated by LRP are known to be instance-specific. However, the discriminativeness of the explanations has not been evaluated yet. Ideally, the visualized objects in the explanation should correspond to the class the class-specific neuron represents. We evaluate the explanations generated by LRP on the off-the-shelf models from *torchvision*, specifically, AlexNet [1], VGG16 [2] and GoogLeNet [3] pre-trained on the ImageNet dataset [32].

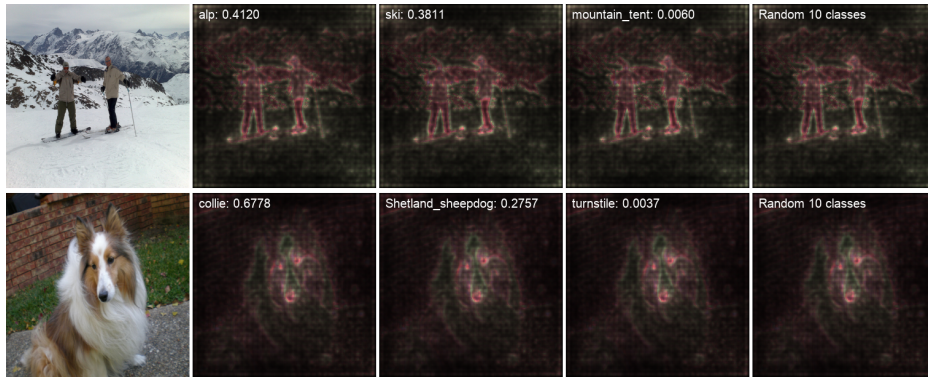
The experiment settings are similar to [16]. The z^β -rule is applied to the first convolution layer. For all higher convolutional layers and fully-connected layers, the z^+ -rule is applied. In the MaxPooling layers, the relevance is only



(a) The explanations generated by LRP on AlexNet.



(b) The explanations generated by LRP on VGG16 Network.



(c) The explanations generated by LRP on GoogLeNet.

Fig. 1: The images from validation datasets of ImageNet are classified using the off-the-shelf models pre-trained on the ImageNet. The classifications of the images are explained by the LRP approach. For each image, we generate four explanations that correspond to the top-3 predicted classes and a randomly chosen multiple-classes.

redistributed to the neuron with the maximal value inside the pooling region, while it is redistributed evenly to the corresponding neurons in the Average Pooling layers. The biases and normalization layers are bypassed in the relevance propagation pass.

The results are shown in figure 1. For each test image, we create four saliency maps as explanations. The first three explanation maps are generated for top-3 predictions, respectively. The fourth one is created for randomly chosen 10 classes from the top-100 predicted classes (which ensure that the score to be propagated is positive). The white text in each explanation map indicates the class the output neuron represents and the corresponding classification probability. The explanations generated by AlexNet are blurry due to incomplete learning (due to the limited expressive power). The explanations of VGG16 classifications are sharper than the ones created on GoogLeNet. The reason is that VGG16 contains only MaxPooling layers and GoogLeNet, by contrast, contains a few average pooling layers.

The generated explanations are instance-specific, but not class-discriminative. In other words, they are independent of class information. The explanations for different target classes, even randomly chosen classes, are almost identical. The conclusion is consistent with the one summarised in the paper [7,33], namely, almost all information about input image is contained in the pattern of non-zero pattern activations, not their precise values. The high similarity of those explanations resulted from the leverage of the same ISSInfo (see section 3.2). In summary, the explanations are not class-discriminative. The generated maps recognize the same foreground objects instead of a class-discriminative one.

3.2 Theoretical Foundation: Deep Taylor Decomposition

Motivated by the divide-and-conquer paradigm, Deep Taylor Decomposition decomposes a deep neural network (i.e. the nested nonlinear functions) iteratively [16]. The propagation rules of LRP are derived from Deep Taylor Decomposition of rectifier neuron network. The function represented by a single neuron is $X_j^{L+1} = \max(0, \mathbf{X}^L \mathbf{W}_j^L + b_j^{L+1})$. The relevance R_j^{L+1} of the neurons X_j^{L+1} is given. The Deep Taylor Decomposition assumes $R_j^{L+1} = \max(0, \mathbf{X}^L \mathbf{W}_j^L + b_j^{L+1})$. The function is expanded with Taylor Series at a point \mathbf{X}_i^r subjective to $\max(0, \mathbf{X}^r \mathbf{W}_j^L + b_j^{L+1}) = 0$. The LRP propagation rules are resulted from the first degree terms of the expansion.

One may hypothesize that the non-discriminateness of LRP is caused by the first-order approximation error in Deep Taylor Decomposition. We proved that, under the given assumption, the same propagation rules are derived, even though all higher-order terms are taken into consideration (see the proof in the supplementary material). Furthermore, we found that the theoretical foundation provided by the Deep Taylor Decomposition is inappropriate. The assumption $R_j^{L+1} = \max(0, \mathbf{X}^L \mathbf{W}_j^L + b_j^{L+1}) = X_j^{L+1}$ is not held at all the layers except for the last layer. The assumption indicates that the relevance value is equal to the activation value for all the neurons, which, we argue, is not true.

In our opinion, the explanations generated by the LRP result from the ISSInfo (ReLU masks and Pooling Switches). The activation values of neurons are required to create explanations using LRP. In the forward pass, the network output a vector (y_1, y_2, \dots, y_m) . In the backward pass, the activation value of the class y_1 is layer-wise backpropagated into input space. In fully connected layers, only the activated neurons can receive the relevance according to any LRP propagation rule. In the Maxpooling layers, the backpropagation conducts an unpooling process, where only the neuron with maximal activations inside the corresponding pooling region can receive relevance. In the convolutional layer, only specific part of neurons R_{conv1} in feature map have non-zero relevance in the backward pass. The part of input pixels P_{input} live in the convolutional regions of those neurons (R_{conv1}). Only the pixels P_{input} will receive the propagated relevance. The pattern of the P_{input} is the explanation generated by LRP.

The backward pass for the class y_2 is similar to that of y_1 . The neurons that receive non-zero relevance are the same as in case of y_1 , even though their absolute values may be slightly different. Regardless of the class chosen for the backpropagation, the neurons of each layer that receive non-zero relevance stay always the same. In other words, the explanations generated by LRP are independent of the class category information, i.e., not class-discriminative.

In summary, in deep convolutional rectifier neuron network, the ReLU masks and Pooling Switches decide the pattern visualized in the explanation, which is independent of class information. That is the reason why the explanations generated by LRP on DCNNs are not class-discriminative. The analysis also explains the non-discriminative explanations generated by other backpropagation approaches, such as the DeConvNets Visualization [11], The vanilla Gradient Visualization [10] and the Guided Backpropagation [14].

4 Contrastive Layer-wise Relevance Propagation

Before introducing our CLRP, we first discuss the conservative property in the LRP. In a DNN, given the input $\mathbf{X} = \{x_1, x_2, x_3, \dots, x_n\}$, the output $\mathbf{Y} = \{y_1, y_2, y_3, \dots, y_m\}$, the score S_{y_j} (activation value) of the neuron y_j before softmax layer, the LRP generate an explanation for the class y_j by redistributing the score S_{y_j} layer-wise back to the input space. The assigned relevance values of the input neurons are $\mathbf{R} = \{r_1, r_2, r_3, \dots, r_n\}$. The conservative property is defined as follows:

Definition 1. *The generated saliency map is conservative if the sum of assigned relevance values of the input neurons is equal to the score of the class-specific neuron, $\sum_{i=1}^n r_i = S_{y_j}$.*

In this section, we consider redistributing the same score from different class-specific neurons respectively. The assigned relevance \mathbf{R} are different due to different weight connections. However, the non-zero patterns of those relevance vectors are almost identical, which is why LRP generate almost the same explanations for different classes. The sum of each relevance vector is

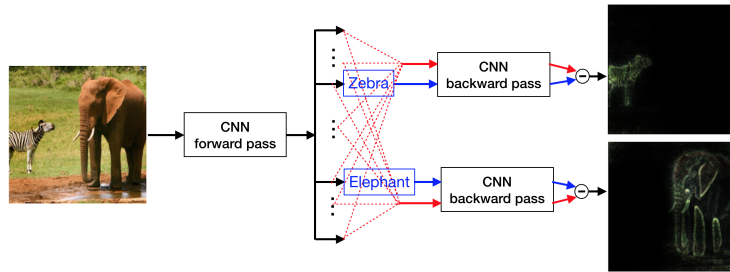


Fig. 2: The figure shows an overview of our CLRP. For each predicted class, the approach generates a class-discriminative explanation by comparing two signals. The blue line means the signal that the predicted class represents. The red line models a dual concept opposite to the predicted class. The final explanation is the difference between the two saliency maps that the two signal generate.

equal to the redistributed score according to the conservative property. The input variables that are discriminative to each target class are a subset of input neurons, i.e., $\mathbf{X}_{dis} \subset \mathbf{X}$. The challenge of producing the explanation is to identify the discriminative pixels \mathbf{X}_{dis} for the corresponding class.

In the explanations of image classification, the pixels on salient edges always receive higher relevance value than other pixels including all or part of \mathbf{X}_{dis} . Those pixels with high relevance values are not necessary discriminative to the corresponding target class. We observe that \mathbf{X}_{dis} receive higher relevance values than that of the same pixels in explanations for other classes. In other words, we can identify \mathbf{X}_{dis} by comparing two explanations of two classes. One of the classes is the target class to be explained. The other class is selected as an auxiliary to identify \mathbf{X}_{dis} of the target class. To identify \mathbf{X}_{dis} more accurately, we construct a virtual class instead of selecting another class from the output layer. We propose two ways to construct the virtual class.

The overview of the CLRP are shown in figure 2. We describe the CLRP formally as follows. The j -th class-specific neuron y_j is connected to input variables by the weights $\mathbf{W} = \{\mathbf{W}^1, \mathbf{W}^2, \dots, \mathbf{W}^{L-1}, \mathbf{W}_j^L\}$ of layers between them, where \mathbf{W}^L means the weights connecting the $(L-1)$ -th layer and the L -th layer, and \mathbf{W}_j^L means the weights connecting the $(L-1)$ -th layer and the j -th neuron in the L -th layer. The neuron y_j models a visual concept O . For an input example \mathbf{X} , the LRP maps the score S_{y_j} of the neuron back into the input space to get relevance vector $\mathbf{R} = f_{LRP}(\mathbf{X}, \mathbf{W}, S_{y_j})$.

We construct a dual virtual concept \bar{O} , which models the opposite visual concept to the concept O . For instance, the concept O models the **zebra**, and the constructed dual concept \bar{O} models the **non-zebra**. One way to model the \bar{O} is to select all classes except for the target class representing O . The concept \bar{O} is represented by the selected classes with weights $\bar{\mathbf{W}} = \{\mathbf{W}^1, \mathbf{W}^2, \dots, \mathbf{W}^{L-1}, \mathbf{W}_{\{-j\}}^L\}$, where $\mathbf{W}_{\{-j\}}^L$ means the weights connected to the output layer excluding the j -th neuron. E.g. the dashed red lines in figure 2 are connected to all classes

except for the target class **zebra**. Next, the score S_{y_j} of target class is uniformly redistributed to other classes. Given the same input example \mathbf{X} , the LRP generates an explanation $\mathbf{R}_{dual} = f_{LRP}(\mathbf{X}, \overline{\mathbf{W}}, S_{y_j})$ for the dual concept. The Contrastive Layer-wise Relevance Propagation is defined as follows:

$$\mathbf{R}_{CLRP} = \max(\mathbf{0}, (\mathbf{R} - \mathbf{R}_{dual})) \quad (2)$$

where the function $\max(\mathbf{0}, \mathbf{X})$ means replacing the negative elements of \mathbf{X} with zeros. The difference between the two saliency maps cancels the common parts. Without the dominant common parts, the non-zero elements in \mathbf{R}_{CLRP} are the most relevant pixels \mathbf{X}_{dis} . If the neuron y_j lives in an intermediate layer of a neural network, the constructed \mathbf{R}_{CLRP} can be used to understand the role of the neuron.

Similar to [27], the other way to model the concept \overline{O} is to negate the weights W_{ij} . The concept \overline{O} can be represented by the weights $\overline{\mathbf{W}} = \{\mathbf{W}^1, \mathbf{W}^2, \dots, \mathbf{W}^{L-1}, -1 * \mathbf{W}_j^L\}$. All the weights are same as in the concept O except that the weights of the last layer W_{ij} are negated. In the experiments section, we call the first modeling method CLRP1 and the second one CLRP2. The contrastive formulation in [27] can be applied to other backpropagation approaches by normalizing and subtracting two generated saliency maps. However, the normalization strongly depends on the maximal value that could be caused by a noisy pixel. Based on the conservative property of LRP, the normalization is avoided in the proposed CLRP.

5 Experiments and Analysis

In this section, we conduct experiments to evaluate our proposed approach. The first experiment aims to generate class-discriminative explanations for individual classification decisions. The second experiment evaluates the generated explanations quantitatively on the ILSVRC2012 validation dataset. The discriminativeness of the generated explanations is evaluated via a Pointing Game and an ablation study. The last experiment aims to understand the difference between neurons in a single classification forward pass.

5.1 Explaining Classification Decisions of DNNs

In this experiment, the LRP, the CLRP1 and the CLRP2 are applied to generate explanations for different classes. The experiments are conducted on a pre-trained VGG16 Network [2]. The propagation rules used in each layer are the same as in the section 3.1. We classify the images of multiple objects. The explanations are generated for the two most relevant predicted classes, respectively. The figure 3 shows the explanations for the two classes (i.e., *Zebra* and *African_elephant*). The explanations generated by the LRP are same for the two classes. Each generated explanation visualizes both *Zebra* and *African_elephant*, which is not class-discriminative. By contrast, both CLRP1 and CLRP2 only

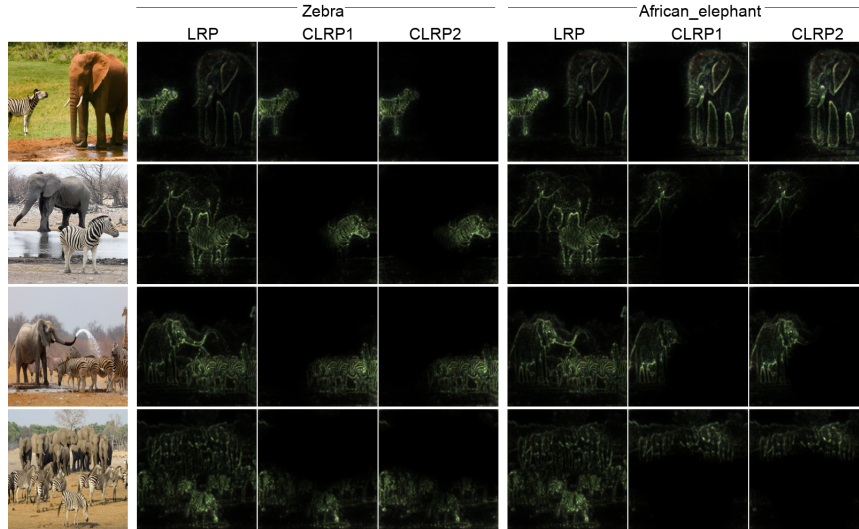


Fig. 3: The images of multiple objects are classified using VGG16 network pre-trained on ImageNet. The explanations for the two relevant classes are generated by LRP and CLR. The CLR generates class-discriminative explanations, while LRP generates almost same explanations.

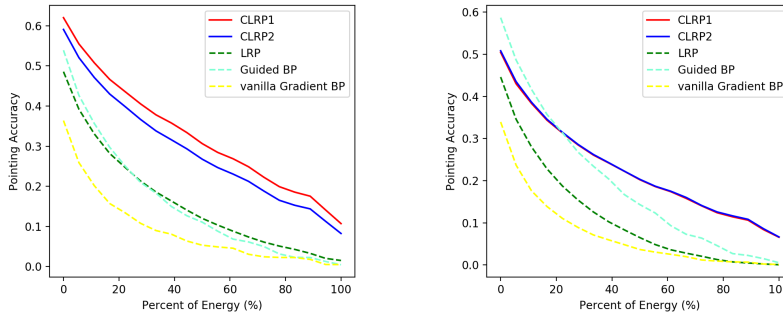
identify the discriminative pixels related to the corresponding class. For the target class *Zebra*, only the pixels on the zebra object are visualized. Even for the complicated images where a zebra herd and an elephant herd co-exist, the CLR methods are still able to find the class-discriminative pixels.

We evaluate the approach with a large number of images with multiple objects. The explanations generated by CLR are always class-discriminative, but not necessarily semantically meaningful for every class. One of the reasons is that the VGG16 Network is not trained for multi-label classification. Other reasons could be the incomplete learning and bias in the training dataset [34].

The implementation of the LRP is not trivial. The one provided by their authors only supports CPU computation. For the VGG16 network, it takes the 30s to generate one explanation on an Intel Xeon 2.90GHz×6 machine. The computational expense makes the evaluation of LRP impossible on a large dataset [27]. We implement a GPU version of the LRP approach, which reduces the 30s to 0.1824s to generate one explanation on a single NVIDIA Telsa K80 GPU. The implementation alleviates the inefficiency problem addressed in [27,13] and makes the quantitative evaluation of LRP on a target dataset possible.

5.2 Evaluating the explanations

In this experiments, we quantitatively evaluate the generated explanations on the ILSVRC2012 validation dataset containing 50, 000 images. A Pointing Game and an ablation study are used to evaluate the proposed approach.



(a) Pointing Accuracy On the AlexNet (b) Pointing Accuracy On the VGG16

Fig. 4: The figure shows the localization ability of the saliency maps generated by the LRP, the CLRP1, the CLRP2, the vanilla Gradient Visualization and the Guided Backpropagation. On the pre-trained models, AlexNet and VGG16, the localization ability is evaluated at different thresholds. The x-axis corresponds to the threshold that keeps a certain percentage of energy left, and the y-axis corresponds to the pointing accuracy.

Pointing Game: To evaluate the discriminativeness of saliency maps, the paper [27] proposes a pointing game. The maximum point on the saliency map is extracted and evaluated. In case of images with a single object, a hit is counted if the maximum point lies in the bounding box of the target object, otherwise a miss is counted. The localization accuracy is measured by $Acc = \frac{\#Hits}{\#Hits + \#Misses}$. In case of ILSVRC2012 dataset, the naive pointing at the center of the image shows surprisingly high accuracy. Based on the reason, we extend the pointing game into a difficult setting. In the new setting, the first step is to preprocess the saliency map by simply thresholding so that the foreground area covers p percent energy out of the whole saliency map (where the energy is the sum of all pixel values in saliency map). A hit is counted if the remaining foreground area lies in the bounding box of the target object, otherwise a miss is counted.

The figure 4 show that the localization accuracy of different approaches in case of different thresholds. With more energy kept, the remained pixels are less likely to fall into the ground-truth bounding box, and the localization accuracy is low correspondingly. The CLRP1 and the CLRP2 show constantly much better pointing accuracy than that of the LRP. The positive results indicate that the pixels that the contrastive backpropagation cancels are on the cluttered background or non-target objects. The CLRP can focus on the class-discriminative part, which improves the LRP. The CLRP is also better than other primal backpropagation-based approaches. One exception is that the Guided Backpropagation shows a better localization accuracy in VGG16 network in case of high thresholds. In addition, the localization accuracy of the CLRP1 and the CLRP2 is similar in the deep VGG16 network, which indicates the equivalence of the two methods to model the opposite visual concept.

Table 1: Ablation study on ImageNet Validation dataset. The dropped activation values after the corresponding ablation are shown in the table.

	Random	vanilla Gradient[10]	Guided BP[14]	LRP[15]	CLRP1	CLRP2
AlexNet	0.0766	0.1716	0.1843	0.1624	0.2093	0.2030
VGG16	0.0809	0.3760	0.4480	0.3713	0.3844	0.3913

Ablation Study: In the Pointing Game above, we evaluate the discriminativeness of the explanations according to the localization ability. In this ablation study, we evaluate the discriminativeness from another perspective. We observe the changes of activation in case of ablating the found discriminative pixels. The activation value of the class-specific neuron will drop if the ablated pixels are discriminative to the corresponding class.

For an individual image classification decision, we first generate a saliency map for the ground-truth class. We identify the maximum point in the generated saliency map as the most discriminative position. Then, we ablate the pixel of the input image at the identified position with a 9×9 image patch. The pixel values of the image patch are the mean value of all the pixel values at the same position across the whole dataset. We classify the perturbed image and observe the activation value of the neuron corresponding to the ground-truth class. The dropped activation value is computed as the difference between the activations of the neuron before and after the perturbation. The dropped score is averaged on all the images in the dataset.

The experimental results of different approaches are shown in the table 1. For the comparison, we also ablate the image with a randomly chosen position. The random ablation has hardly impact on the output. The saliency maps corresponding to all other approaches find the relevant pixel because the activations of the class-specific neurons dropped a lot after the corresponding ablation. In both networks, CLRP1 and CLRP2 show the better scores, which means the discriminativeness of explanations generated by CLRP is better than that of the LRP. Again, the Guided Backpropagation shows better score than CLRP. This ablation study only considers the discriminative of the pixel with maximal relevance value, which corresponds to a special case in the Pointing Game, namely, only one pixel with maximal relevance is left after the thresholding. The two experiments show the consistent result that the Guided Backpropagation is better than LRP in the special case. We do not report the performance of the GoogLeNet in the experiments. Our approach shows that the zero-padding operations of convolutional layers have a big impact on the output of the GoogLeNet model in *torchvision* module of Pytorch. The impact leads to a problematic saliency map (see supplementary material).

5.3 Understanding the Difference between Neurons

The neurons of DNNs have been studying with their activation values. The DeConvNets [11] visualize the patterns and collect the images that maximally activate the neurons, given an image set. The activation maximization method [9,35] aims to generate an image in input space that maximally activates a single neuron or a group of neurons. Furthermore, the work [36,23] understand the semantic concepts of the neurons with an annotated dataset. In this experiment, we aim to study the difference among neurons in a single classification decision.

The neurons of low layers may have different local receptive fields. The difference between them could be caused by the different input stimuli. We visualize high-level concepts learned by the neurons that have the same receptive fields, e.g., a single neuron in a fully connected layer. For a single test image, the LRP and the CLRP2 are applied to visualize the stimuli that activate a specific neuron. We do not use CLRP1 because the opposite visual concept cannot be modeled by the remaining neurons in the same layer.

In the VGG16 network, we visualize the 8 activated neurons x_{1-8} from the *fc1* layer. The visualized maps are shown in figure 5. The image is classified as a *toyshop* by the VGG16 network. The receptive field (the input image) is shown in the center, and the 8 explanation maps are shown around it. While the LRP produces almost identical saliency map for the 8 neurons (in figure 5a), the CLRP2 gains a meaningful insight about their difference, which shows that different neurons focus on different parts of images. By comparison (see figure 5b), the neurons x_1, x_2, x_3 in the first row are activated more by the *lion*, the *gorilla*, and the *monkey* respectively. The neurons x_4, x_5 in the second row by the eye of the *elephant* and the *bird* respectively. The right-down one x_6 by the *panda*. The last two neurons x_7 and x_8 focus on the similar patterns (i.e., the *tiger*).

To our knowledge, there is no known work on the difference between neurons in an individual classification decision and also no evaluation metric. We evaluate the found difference by an ablation study. More concretely, we first find the discriminative patch for each neuron (e.g., x_{1-8}) using CLRP2. Then, we ablate the patch and observe the changes of neuron activations in the forward pass. The discriminative patch of a neuron is identified by the point with maximal value in its explanation map created by CLRP2. The 9×9 neighboring pixels around the maximum point are replaced with the values that are mean of pixel values in the same positions across the whole dataset.

The ablation study results are shown in the figure 5c. The positive value in the grid of the figure means the decreased activation value, and the negative ones mean the activations increase after the corresponding ablation. In case of the ablation corresponding to neuron x_i , we see that the activation of x_i is significantly dropped (could become not-activated). The maximal dropped values of each row often occur on the diagonal axis. We also try with other ablation sizes and other neurons, which shows the similar results. The ablations for the last two neurons x_7 and x_8 are same because their explanation maps are similar. The changes of activations of all other neurons are also the same for the same

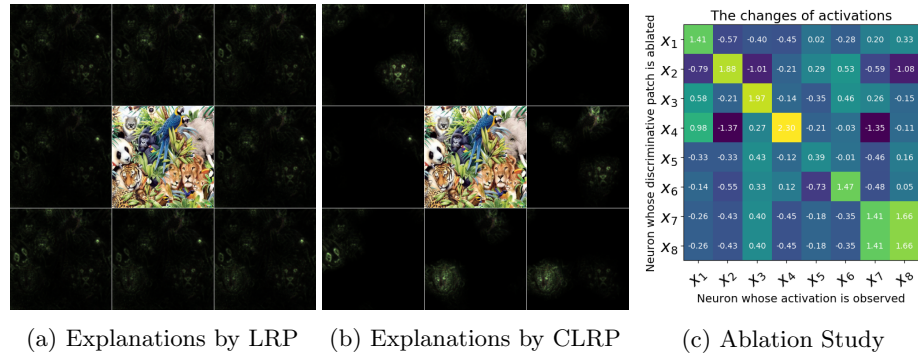


Fig. 5: The figures show explanation maps of neurons in $fc1$ layers. The explanations generated by LRP are not discriminative. By contrast, the ones generated by CLRP explain the difference between the neurons.

ablation. We found that many activated neurons correspond to same explanation maps.

6 Conclusion

The explanations generated by LRP are evaluated. We find that the explanations are not class-discriminative. We discuss the theoretical foundation and provide our justification for the non-discriminativeness. To improve discriminativeness of the generated explanations, we propose the Contrastive Layer-wise Relevance Propagation. The qualitative and quantitative evaluations confirm that the CLRP is better than the LRP. We also use the CLRP to shed light on the role of neurons in DCNNs.

We propose two ways to model the opposite visual concept the class-specific neuron represents. However, there could be other more appropriate modeling methods. Even though our approach produces a pixel-wise explanation for the individual classification decisions, the explanations for similar classes are similar. The fine-grained discriminativeness are needed to explain the classifications of the intra-classes. We leave the further exploration in future work.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
2. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
3. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: 2015 IEEE

- Conference on Computer Vision and Pattern Recognition (CVPR). Volume 00. (2015) 1–9
4. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 580–587
 5. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 779–788
 6. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. *CoRR* **abs/1412.0035** (2014)
 7. Dosovitskiy, A., Brox, T.: Inverting visual representations with convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4829–4837
 8. Mahendran, A., Vedaldi, A.: Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision* **120** (2016) 233–255
 9. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. *University of Montreal* **1341** (2009) 1
 10. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013)
 11. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: *European conference on computer vision*, Springer (2014) 818–833
 12. Robnik-Sikonja, M., Kononenko, I.: Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering* **20** (2008) 589–600
 13. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685* (2017)
 14. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806* (2014)
 15. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* **10** (2015) e0130140
 16. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition* **65** (2017) 211–222
 17. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, IEEE (2016) 2921–2929
 18. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. See <https://arxiv.org/abs/1610.02391> v3 **7** (2016)
 19. Ribeiro, M.T., Singh, S., 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*, ACM (2016) 1135–1144
 20. Ribeiro, M.T., Singh, S., Guestrin, C.: Nothing else matters: model-agnostic explanations by identifying prediction invariance. *arXiv preprint arXiv:1611.05817* (2016)
 21. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2015) 685–694

22. Mahendran, A., Vedaldi, A.: Salient deconvolutional networks. In: European Conference on Computer Vision, Springer (2016) 120–135
23. Gonzalez-Garcia, A., Modolo, D., Ferrari, V.: Do semantic parts emerge in convolutional neural networks? *International Journal of Computer Vision* **126** (2018) 476–494
24. Lapuschkin, S., Binder, A., Müller, K.R., Samek, W.: Understanding and comparing deep neural networks for age and gender classification. *arXiv preprint arXiv:1708.07689* (2017)
25. Srinivasan, V., Lapuschkin, S., Hellge, C., Müller, K.R., Samek, W.: Interpretable human action recognition in compressed domain. In: Acoustics, Speech and Signal Processing, 2017 IEEE International Conference on, IEEE (2017) 1692–1696
26. Arras, L., Montavon, G., Müller, K.R., Samek, W.: Explaining recurrent neural network predictions in sentiment analysis. *arXiv preprint arXiv:1706.07206* (2017)
27. Zhang, J., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. In: European Conference on Computer Vision, Springer (2016) 543–559
28. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 2528–2535
29. Kindermans, P.J., Schütt, K., Müller, K.R., Dähne, S.: Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270* (2016)
30. Cao, C., Liu, X., Yang, Y., Yu, Y., Wang, J., Wang, Z., Huang, Y., Wang, L., Huang, C., Xu, W., et al.: Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2956–2964
31. Tsotsos, J.K., Culhane, S.M., Wai, W.Y.K., Lai, Y., Davis, N., Nufflo, F.: Modeling visual attention via selective tuning. *Artificial intelligence* **78** (1995) 507–545
32. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. *IJCV* **115** (2015) 211–252
33. Agrawal, P., Girshick, R., Malik, J.: Analyzing the performance of multilayer neural networks for object recognition. In: European conference on computer vision, Springer (2014) 329–344
34. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1521–1528
35. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., Clune, J.: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In: Advances in Neural Information Processing Systems. (2016) 3387–3395
36. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856* (2014)

Understanding Individual Decisions of CNNs via Contrastive Backpropagation

SUPPLEMENTARY MATERIAL

Jindong Gu, Yinchong Yang, Volker Tresp

Abstract. This is a supplementary material (appendix) for the paper . The appendix A provides proofs, detailed derivations for decomposing rectifier neural networks with higher-order Taylor Decomposition. The appendix B gives more details about GoogLeNet model from *torchvision* module of *Pytorch*.

A Decomposition of Rectifier Neural Network with Higher-order Taylor Expansion

First, we give the detailed derivations of propagation rules resulting from deep higher-order Taylor decomposition of the neural network. The rules can be applied to any activation function where a root point exists. In DNNs, the function of each neuron is the activation of the linear combination of activation values of the previous layer or input variables. The input of the function $f(\mathbf{x})$ of the neuron k is $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$. Given a root point $\mathbf{x}^r = (x_1^r, x_2^r, x_3^r, \dots, x_n^r)$ where $f(\mathbf{x}_r) = 0$ of the function, the relevance $R_k = f(\mathbf{x})$ of the neuron k is decomposed at the root point with higher-order Taylor expansion as follows

$$\begin{aligned}
 R_k = & \underbrace{\sum_i \left(\frac{\partial f}{\partial x_i} \Big|_{\mathbf{x}=\mathbf{x}_r} \right) * (x_i - x_i^r)}_{\text{first-order term}} + \underbrace{\sum_i \frac{1}{2!} \sum_j \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \Big|_{\mathbf{x}=\mathbf{x}_r} \right) * (x_i - x_i^r) * (x_j - x_j^r)}_{\text{second-order term}} + \\
 & \underbrace{\sum_i \frac{1}{3!} \sum_j \sum_m \left(\frac{\partial^3 f}{\partial x_i \partial x_j \partial x_m} \Big|_{\mathbf{x}=\mathbf{x}_r} \right) * (x_i - x_i^r) * (x_j - x_j^r) * (x_m - x_m^r)}_{\text{third-order term}} + \\
 & \underbrace{\sum_{n=4}^{\infty} \sum_i \frac{1}{n!} \underbrace{\sum \dots \sum}_{n-1} \left(\frac{\partial^n f}{\partial x_i \partial x_{l_1} \dots \partial x_{l_{n-1}}} \Big|_{\mathbf{x}=\mathbf{x}_r} \right) * (x_i - x_i^r) * \prod_{p=1}^{n-1} (x_{l_p} - x_{l_p}^r)}_{\text{higher-order terms}}
 \end{aligned}$$

The neuron i , which lives in the last layer of the neuron k , receives relevance R_{ik} from the neuron k . The decomposition of the relevance R_k can be reformulated as below

$$R_k = \sum_i R_{ik1} + \sum_i R_{ik2} + \sum_i R_{ik3} + \sum_{n=4}^{\infty} \sum_i R_{ikn}$$

where $R_{ik} = \sum_{n=1}^{\infty} R_{ikn}$. The relevance R_i of the neuron i is the accumulative sum of the received relevance from the neurons of the next layer, namely, $R_i = \sum_{k=1}^m R_{ik}$, where m is the number of neurons of the next layer. The relevance R_i of the neuron i is

$$\begin{aligned}
R_i &= \sum_{k=1}^m R_{ik} = \sum_{k=1}^m (R_{ik1} + R_{ik2} + R_{ik3} + \sum_n^{\infty} R_{ikn}) \\
&= \sum_{k=1}^m \left(\left(\frac{\partial f}{\partial x_i} \Big|_{\mathbf{x}=\mathbf{x}^r} \right) * (x_i - x_i^r) + \frac{1}{2!} \sum_j \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \Big|_{\mathbf{x}=\mathbf{x}^r} \right) * (x_i - x_i^r) * (x_j - x_j^r) + \right. \\
&\quad \left. \frac{1}{3!} \sum_j \sum_m \left(\frac{\partial^3 f}{\partial x_i \partial x_j \partial x_m} \Big|_{\mathbf{x}=\mathbf{x}^r} \right) * (x_i - x_i^r) * (x_j - x_j^r) * (x_m - x_m^r) + \right. \\
&\quad \left. \sum_{n=4}^{\infty} \frac{1}{n!} \underbrace{\sum_{n-1} \dots \sum}_{n-1} \left(\frac{\partial^n f}{\partial x_i \partial x_{l_1} \dots \partial x_{l_{n-1}}} \Big|_{\mathbf{x}=\mathbf{x}^r} \right) * (x_i - x_i^r) * \prod_{p=1}^{n-1} (x_{l_p} - x_{l_p}^r) \right)
\end{aligned}$$

Next, we give the detailed derivations of propagation rules resulting from deep higher-order Taylor decomposition of the rectifier neural network. The relevance of neurons in the detection layer is given as $R_k = \max(0, \sum_i x_i w_{ik} + b_k)$, where $b_j < 0$. All the rules correspond to different root points where the function is expanded with higher-order Taylor decomposition. If the root point lives in the search direction \mathbf{v} from the input point \mathbf{x} , the root point is written as $\mathbf{x}^r = \mathbf{x} + t\mathbf{v}$. We consider two cases separately:

$$\begin{aligned}
\chi_1 &= \{k : \sum_i x_i w_{ik} + b_k \leq 0\} = \{k : R_k = 0\} \\
\chi_2 &= \{k : \sum_i x_i w_{ik} + b_k > 0\} = \{k : R_k > 0\}
\end{aligned}$$

In the first case ($k \in \chi_1$), the data point itself is already the nearest root point of the function R_k , i.e., $\mathbf{x}^r = \mathbf{x}$. In the second case ($k \in \chi_2$), the i th element of the root point is subject to following equation

$$x_i^r = x_i - \frac{\sum_i x_i w_{i'k} + b_k}{\sum_{i'} v_{i'} w_{i'k}} v_i$$

In case of the ReLU activation function, the propagation rule resulting from higher-order Taylor decomposition is as below

$$\begin{aligned}
R_i &= \sum_{k \in \chi_1} \left(\sum_{n=1}^{\infty} \frac{1}{n!} \underbrace{\sum \dots \sum}_{n-1} \left(\frac{\partial^n f}{\partial x_i \partial x_{l_1} \dots \partial x_{l_{n-1}}} \Big|_{\mathbf{x}=\mathbf{x}_r} \right) * (x_i - x_i^r) * \prod_{p=1}^{n-1} (x_{l_p} - x_{l_p}^r) \right) + \\
&\quad \sum_{k \in \chi_2} \left(\left(\frac{\partial f}{\partial x_i} \Big|_{\mathbf{x}=\mathbf{x}_r} \right) * (x_i - x_i^r) + \frac{1}{2!} \sum_j \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \Big|_{\mathbf{x}=\mathbf{x}_r} \right) * (x_i - x_i^r) * (x_j - x_j^r) \right) + \\
&\quad \frac{1}{3!} \sum_j \sum_m \left(\frac{\partial^3 f}{\partial x_i \partial x_j \partial x_m} \Big|_{\mathbf{x}=\mathbf{x}_r} \right) * (x_i - x_i^r) * (x_j - x_j^r) * (x_m - x_m^r) + \\
&\quad \sum_{n=4}^{\infty} \frac{1}{n!} \underbrace{\sum \dots \sum}_{n-1} \left(\frac{\partial^n f}{\partial x_i \partial x_{l_1} \dots \partial x_{l_{n-1}}} \Big|_{\mathbf{x}=\mathbf{x}_r} \right) * (x_i - x_i^r) * \prod_{p=1}^{n-1} (x_{l_p} - x_{l_p}^r) \\
&= 0 + \sum_{k \in \chi_2} \left(\left(\frac{\partial f}{\partial x_i} \Big|_{\mathbf{x}=\mathbf{x}_r} \right) * (x_i - x_i^r) + 0 \right) \\
&= \sum_{k \in \chi_2} \frac{\sum_i x_{i'} w_{i'k} + b_k}{\sum_{i'} v_{i'} w_{i'k}} v_i w_{ik} = \sum_{k \in \chi_2} \frac{v_i w_{ik}}{\sum_{i'} v_{i'} w_{i'k}} R_k
\end{aligned}$$

The rule derived from higher-order Taylor decomposition is the same as the one derived from first-order Taylor approximation. The reason is that the higher-order terms are zeros in two cases of ReLU function. The expansions at different root points (i.e. different search directions v_i) correspond to the different propagation rules in LRP and Deep Taylor Decomposition, such as w^2 -rule, z -rule, z^+ -rule and z^β -rule.

B The MaxPooling Indices of GoogLeNet (inception v3)

We do not report the performance on the GoogLeNet in both the Pointing Game task and the ablation study. The saliency maps generated by CLRP and LRP on the GoogLeNet [?] show that the network works in an unexpected way. The model we use is from torchvision module in Pytorch. After further exploring the model inception v3, we find that the low performance of GoogLeNet on both tasks is caused by the first two MaxPooling layers before Inception module. In the two pooling layers, the pooling indices occur often on the boundary in the outermost pooling regions. In the backpropagation through the MaxPooling layer, the relevance score is only redistributed to the pixels which achieve maximal values within a pooling region. Because the pooling indices often occur on the boundary, the pixels on the boundary receive much relevance score. The effect can also be seen in the figure 1 by zooming in. Therefore, the saliency maps from GoogLeNet shows pool localization ability.

We now show more details about the issue. First, we show that explanations generated by the LRP, and CLRP1 and CLRP2. The GoogLeNet Model used in

this paper is the inception v3 from torchvision. In the figure 1, the explanation from the CLR P1 and the CLR P2 focus on the boundary pixels of images. By zooming in the explanation generated by the LRP, you can also see that the boundary pixels receive more relevance inside a pooling region. A clear boundary edge can be seen in the explanations.

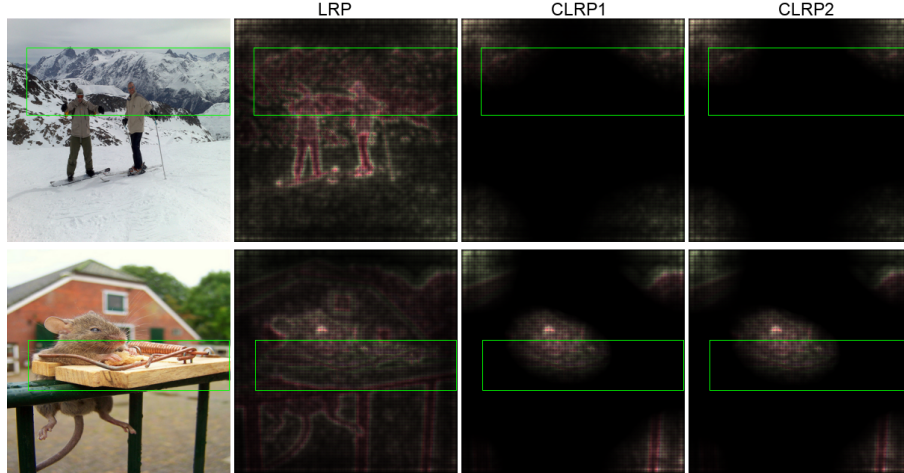


Fig. 1: The figure shows two examples generated on the GoogLeNet model. The three approaches correspond to the LRP, the CLR P1 and the CLR P2. The classes to be explained in each example are "alp" and "mousetrap" respectively.

We investigate the backpropagation process and find that the reason is the MaxPooling layers in first several building blocks. In the MaxPooling Layers, the pixels on the boundary are more often to be the maximal value inside the corresponding pooling region. In the backpropagation unpooling process, those pixels receives more relevance, which leads to the obvious boundary in the explanations, in both the LRP and the CLR P.

To show the effect of the MaxPooling layer of the GoogLeNet model in torchvision, we visualize the pooling indices of randomly chosen feature maps. In the figure 2, the indices of the first two MaxPooling layers are visualized. Only the indices corresponding to the boundary of pooling result feature are visualized. In case of the pooling size (3×3), the indices of maximal values of pooling input should distribute uniformly on the outmost three rows or columns. However, it is not the case. In most feature maps, the indices lie more often on the outmost row or column.

We further study the reason for the unexpected effect of pool indices in MaxPooling layers. The effect that the pooling indices lie more often on the boundary is caused by the convolutional result from the last layer. We conjecture that the convolutional result with higher activation value on the boundary is

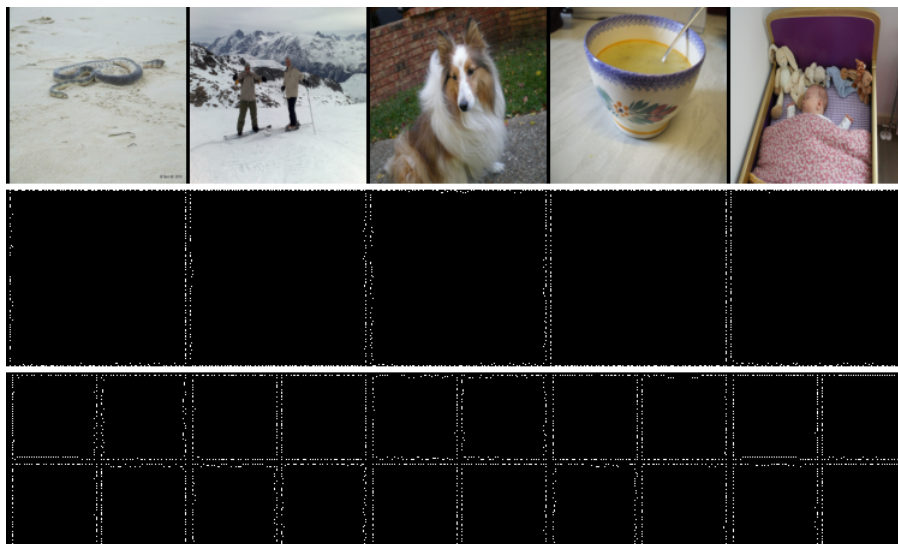


Fig. 2: The figure visualizes the pooling indices of the MaxPooling Layers from the GoogLeNet model. The feature maps are chosen from a random channel.

caused by the Zero Padding of the convolutional layer. It is only the case in inception v3 model in torchvision.