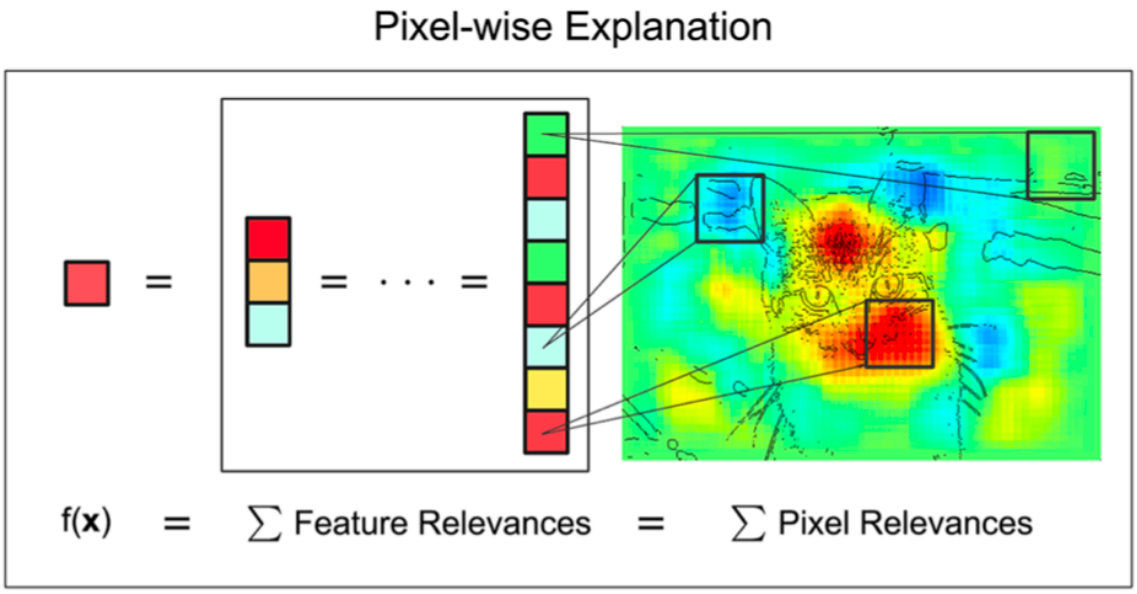
On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation

Pixel-wise Decomposition

- The overall idea of pixel-wise decomposition is to understand the contribution of a single pixel of an image x to the prediction f(x) made by a classifier f in an image classification task
- We would like to find out, separately for each image x, which pixels contribute to what extent to a positive or negative classification result



- In the classification step the image is converted to a feature vector representation and a classifier is applied to assign the image to a given category, e.g., “cat” or “no cat”. Note that the computation of the feature vector usually involves the usage of several intermediate representations. Our method decomposes the classification output f(x) into sums of feature and pixel relevance scores. The final relevances visualize the contributions of single pixels to the prediction.
- **In this paper we propose a novel concept we denote as layer-wise relevance propagation as a general concept for the purpose of achieving a pixel-wise decomposition**
- Layer wise relevance propagation in its general form assumes that the classifier can be decomposed into several layers of computation. Such layers can be parts of the feature extraction from the image or parts of a classification algorithm run on the computed features.
- len zaujimavost - this is possible for Bag of Words features with non-linear SVMs as well as for neural networks.

Ako funguje - celkom pekne opisane:

The first layer are the inputs, the pixels of the image, the last layer is the real-valued prediction output of the classifier f. The l-th layer is modeled as a vector $z = (z_d^{(l)})_{d=1}^{V(l)}$ with dimensionality $V(l)$. Layer-wise relevance propagation assumes that we have a Relevance score $R_d^{(l+1)}$ for each dimension $z_d^{(l+1)}$ of the vector z at layer $l+1$. The idea is to find a Relevance score $R_d^{(l)}$ for each dimension $z_d^{(l)}$ of the vector z at the next layer l which is closer to the input layer such that the following equation holds.

$$f(x) = \cdots = \sum_{d \in l+1} R_d^{(l+1)} = \sum_{d \in l} R_d^{(l)} = \cdots = \sum_d R_d^{(1)} \tag{2}$$

Iterating Eq (2) from the last layer which is the classifier output f(x) down to the input layer x consisting of image pixels then yields the desired Eq (1). The Relevance for the input layer will serve as the desired sum decomposition in Eq (1)

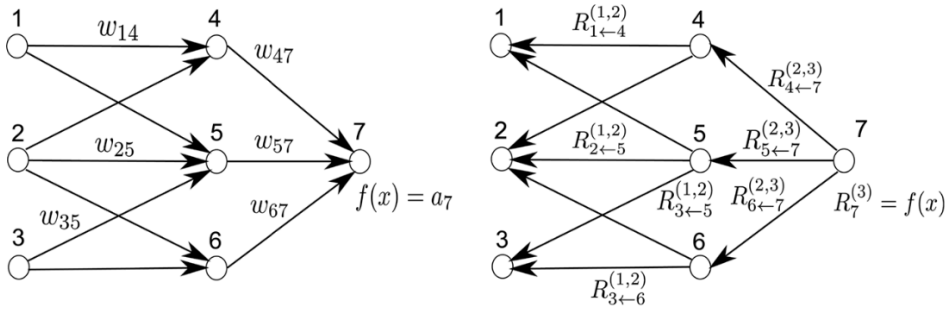


Fig 2. Left: A neural network-shaped classifier during prediction time. w_{ij} are connection weights. a_i is the activation of neuron i. Right: The neural network-shaped classifier during layer-wise relevance computation time. $R_i^{(l)}$ is the relevance of neuron i which is to be computed. In order to facilitate the computation of $R_i^{(l)}$ we introduce messages $R_{i\leftarrow j}^{(l,l+1)}$. $R_{i\leftarrow j}^{(l,l+1)}$ are messages which need to be computed such that the layer-wise relevance in Eq (2) is conserved. The messages are sent from a neuron i to its input neurons j via the connections used for classification, e.g. 2 is an input neuron for neurons 4, 5, 6. Neuron 3 is an input neuron for 5, 6. Neurons 4, 5, 6 are the input for neuron 7.

$$R_7^{(3)} = R_4^{(2)} + R_5^{(2)} + R_6^{(2)}$$

$$R_4^{(2)} + R_5^{(2)} + R_6^{(2)} = R_1^{(1)} + R_2^{(1)} + R_3^{(1)}$$

The layer-wise relevance propagation should reflect the mes- sages passed during classification time. We know that during classification time, a neuron i inputs aiwik to neuron k, provided that i has a forward connection to k.

$$R_7^{(3)} = R_7^{(3)} \frac{a_4 w_{47}}{\sum_{i=4,5,6} a_i w_{i7}} + R_7^{(3)} \frac{a_5 w_{57}}{\sum_{i=4,5,6} a_i w_{i7}} + R_7^{(3)} \frac{a_6 w_{67}}{\sum_{i=4,5,6} a_i w_{i7}}$$

$$R_4^{(2)} = R_4^{(2)} \frac{a_1 w_{14}}{\sum_{i=1,2} a_i w_{i4}} + R_4^{(2)} \frac{a_2 w_{24}}{\sum_{i=1,2} a_i w_{i4}}$$

Vseobecny vzorec pre relevanciu je potom (16)

$$R_{i \leftarrow k}^{(l,l+1)} = R_k^{(l+1)} \frac{a_i w_{ik}}{\sum_h a_h w_{hk}}$$

Dalsia zaujimava vlastnost

The Formula (16) has a second property: The sign of the relevance sent by message $R_{i \leftarrow k}^{(l,l+1)}$ becomes inverted if the contribution of a neuron $a_i\, w_{ik}$ has different sign then the sum of the contributions from all input neurons, i.e. if the neuron fires against the overall trend for the top neuron from which it inherits a portion of the relevance. Same as for the example with the linear mapping in Eq (5), an input neuron can inherit positive or negative relevance depending on its input sign. This is a difference to the Eq (4). While this sign switching property can be defined analogously for a range of architectures, we do not add it as a constraint for layer-wise relevance propagation.

Alternativny pristup k dekompozicii je **taylorov rozklad prveho stupna - first order taylor approximation**

$$\begin{aligned} f(x) &\approx f(x_0) + Df(x_0)[x - x_0] \\ &= f(x_0) + \sum_{d=1}^{V} \frac{\partial f}{\partial x_{(d)}}(x_0)(x_{(d)} - x_{0(d)}) \end{aligned}$$

Kvôli presnosti Taylorovej aproximácie predikcie by sa x0 malo zvoliť tak, aby sa priblížilo k x pod euklidovskou normou, aby sa minimalizovali zvyšky Taylora podľa Taylorových aproximácií vyšších rádov. V prípade viacerých jestvujúcich koreňov x0 s minimálnou normou sa môžu spriemerovať alebo integrovať, aby

sa získal priemer za všetky tieto riešenia.

$$f(x) \approx \sum_{d=1}^{V} \frac{\partial f}{\partial x_{(d)}}(x_0)(x_{(d)} - x_{0(d)}) \qquad \text{such that } f(x_0) = 0$$

LRP multilayer NN

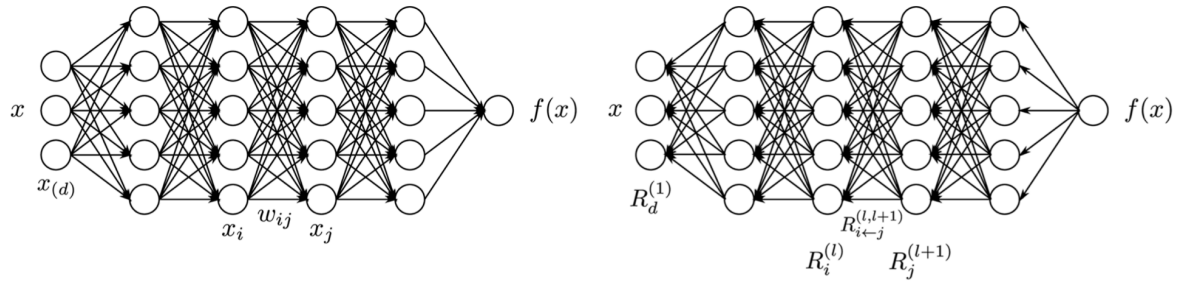


Fig 5. Multilayer neural network annotated with the different variables and indices describing neurons and weight connections. Left: forward pass. Right: backward pass.

The method works as follows: Knowing the relevance of a certain neuron Rj (l+1) for the classification decision f(x), one would like to obtain a decomposition of such relevance in terms of messages sent to neurons of the previous layers. We call these messages Ri<-j

$$\sum_i R_{i\leftarrow j}^{(l,l+1)} = R_j^{(l+1)}$$

In the case of a linear network f(x) = ∑i zij where the relevance Rj = f(x), such decomposition is immediately given by Ri j = zij. However, in the general case, the neuron activation xj is a non-linear function of zj.

A first possible choice of relevance decomposition is based on the ratio of local and global pre- activations and is given by:

$$R_{i\leftarrow j}^{(l,l+1)} = \frac{z_{ij}}{z_j} \cdot R_j^{(l+1)}$$

$$f(x) = \cdots = \sum_{d \in l+1} R_d^{(l+1)} = \sum_{d \in l} R_d^{(l)} = \cdots = \sum_{d} R_d^{(1)}$$

conservation law!!!