EMBC Tutorial on Interpretable and Transparent Deep Learning

Wojciech Samek
(Fraunhofer HHI)

Grégoire Montavon
(TU Berlin)

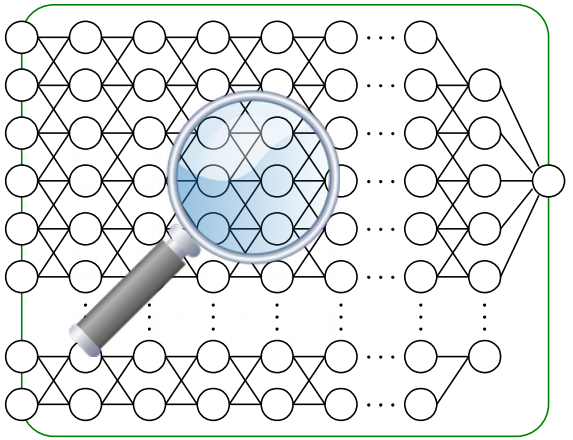
Klaus-Robert Müller
(TU Berlin)

13:30 - 14:00	Introduction KRM
14:00 - 15:00	Techniques for Interpretability GM
15:00 - 15:30	Coffee Break ALL
15:30 - 16:15	Evaluating Interpretability & Applications WS
16:15 - 17:15	Applications in BME & the Sciences and Wrap-Up KRM

Narrowing the Concept of Explanation

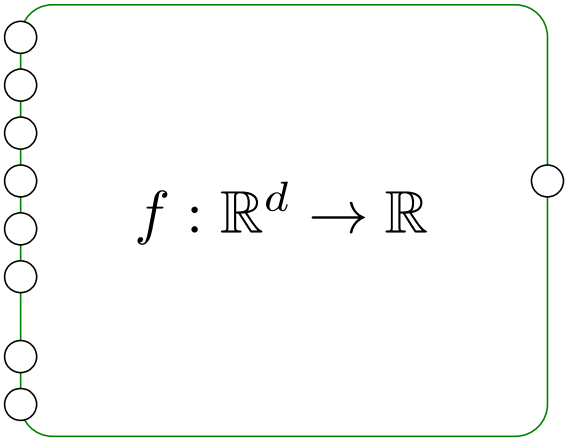
Explaining ML Models: Two Views

mechanistic understanding



Understanding what mechanism the network uses to solve a problem or implement a function.

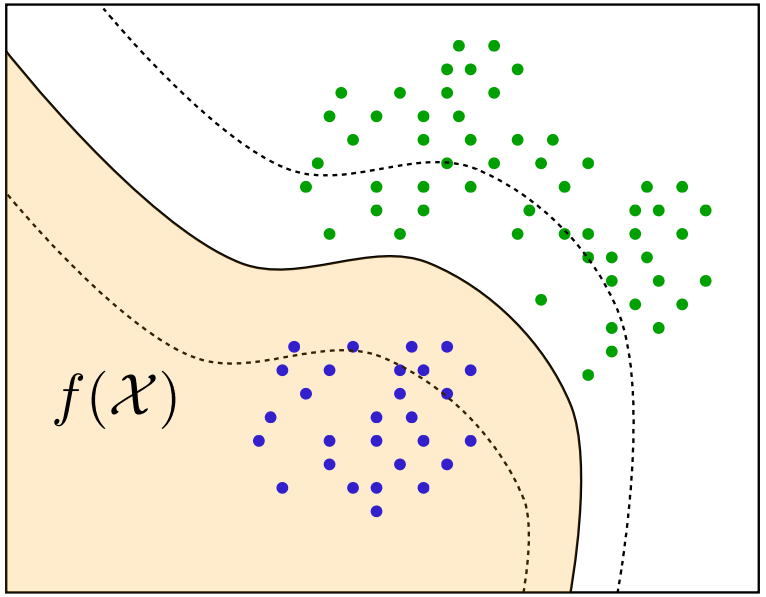
functional understanding



Understanding how the networks relates the input to the output variables.

Explaining ML Models: Two Problems

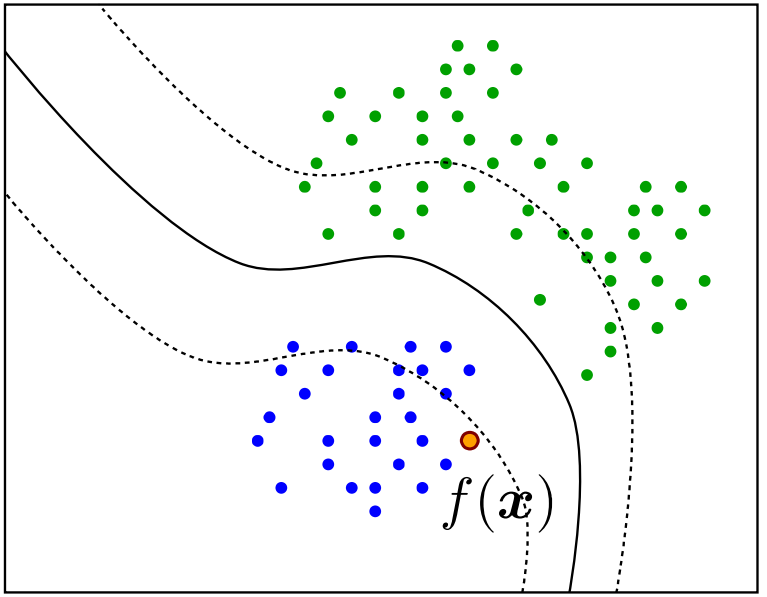
model analysis



possible approach

- build prototypes of "typical" examples of a certain class.

decision analysis



possible approach

- identify which input variables contribute to the prediction.

Explaining ML Models: Two Problems

Model Analysis

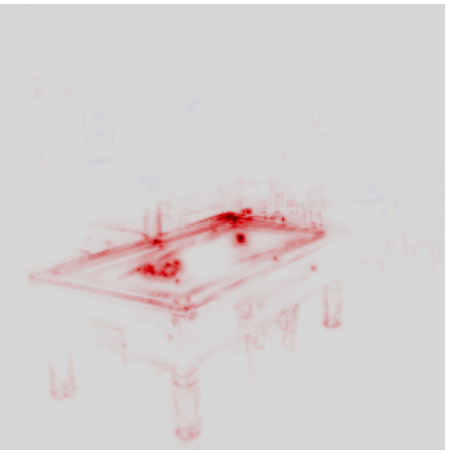
"what does something predicted as a pool table typically look like."

model's prototypical pool table

Decision Analysis

"why a given image is classified as a pool table"

some pool table



why it is classified as a pool table

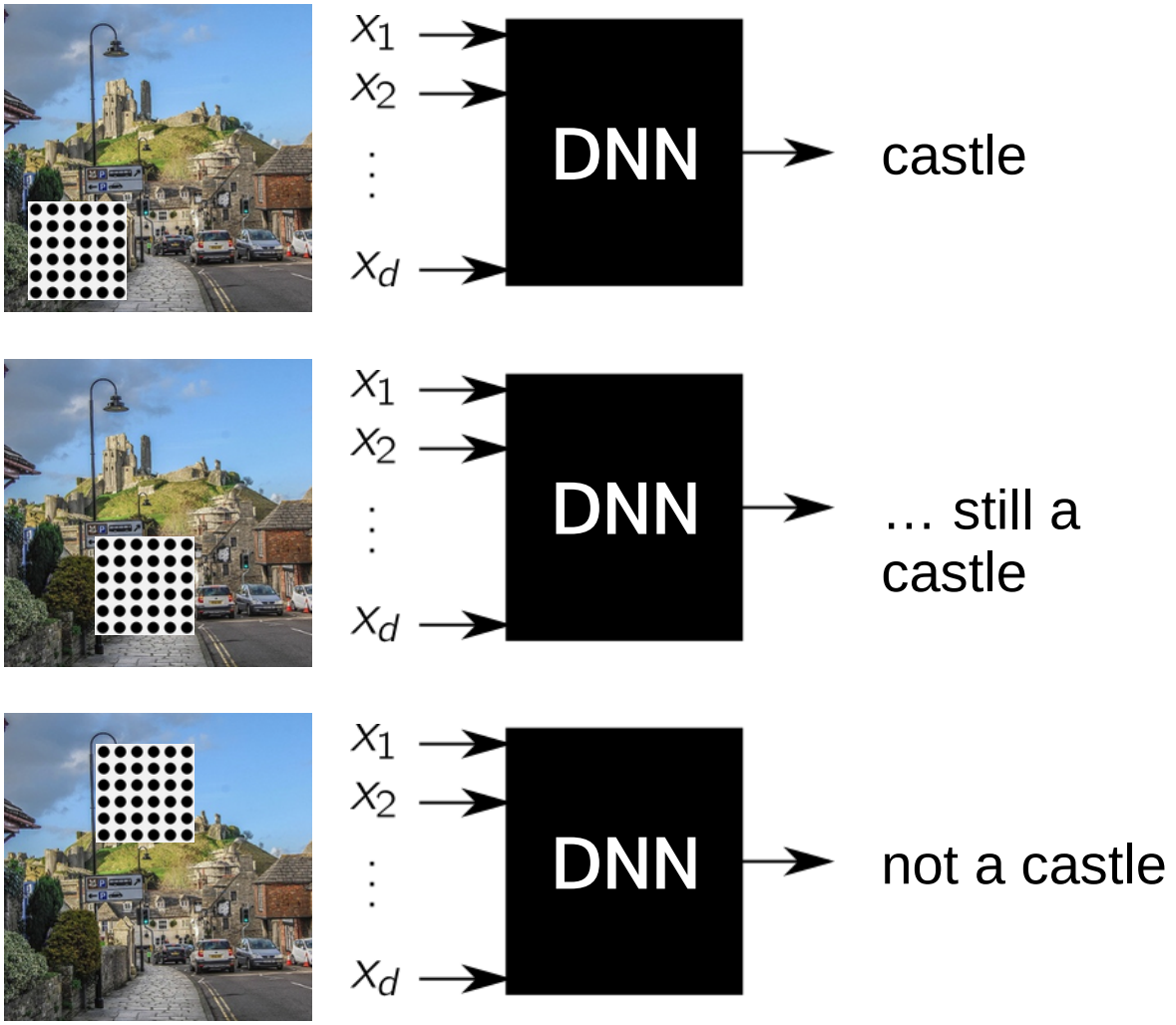
A Survey of Explanation Techniques

Overview of Explanation Methods

1. Perturbation-Based Methods
2. Meaningful Perturbations
3. Simple Taylor Expansion
4. Gradient x Input
4. Layer-Wise Relevance Propagation (LRP)

Approach 1: Perturbation

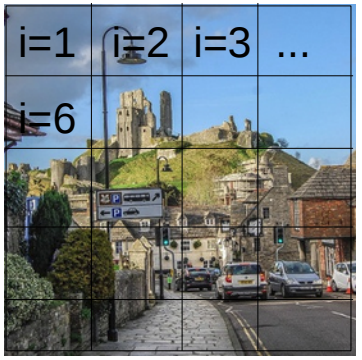
Idea: Assess features relevance by testing the model response to their removal or perturbation.



Approach 1: Perturbation

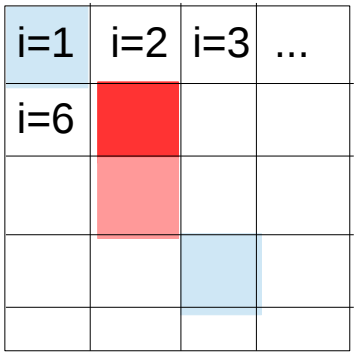
Building an explanation

input



$$\forall i: \; R_i = f(\boldsymbol{x}) - f(\boldsymbol{x} - \{x_i\})$$

heatmap



Advantages

- Simple.

- Applicable to *any* ML model.

Disadvantages

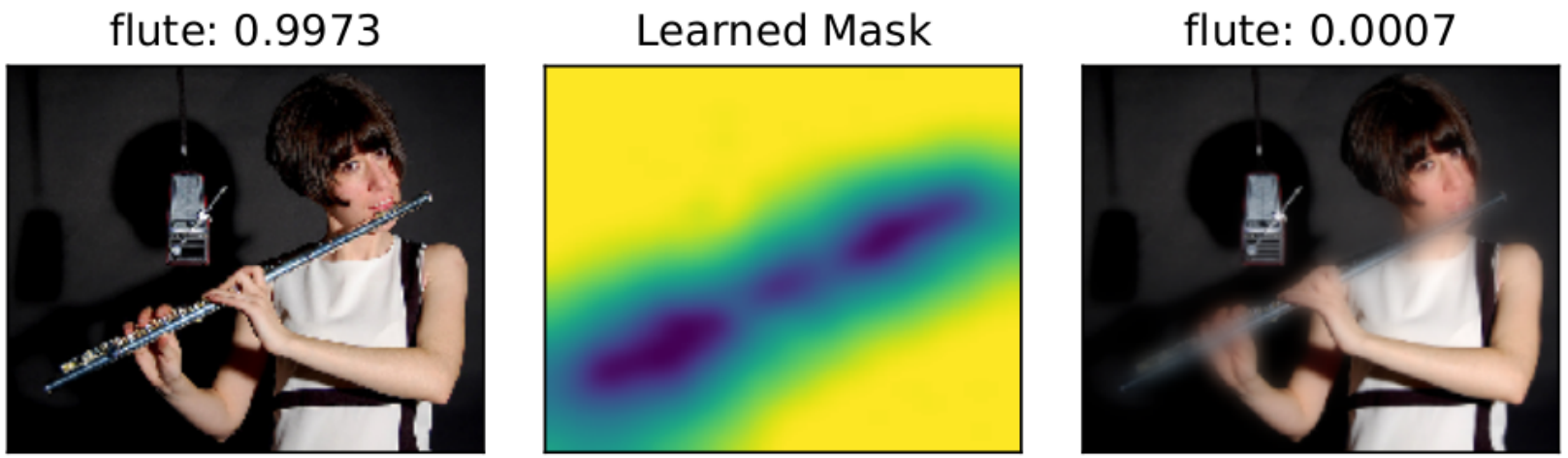
- Need to reevaluate the function for many perturbations $\rightarrow$ slow

- Perturbation process may introduce artefacts in the image $\rightarrow$ unreliable

Approach 2: Meaningful Perturbations

Idea: Don't iterate over all possible perturbation, search locally for the best perturbation m* (or mask).

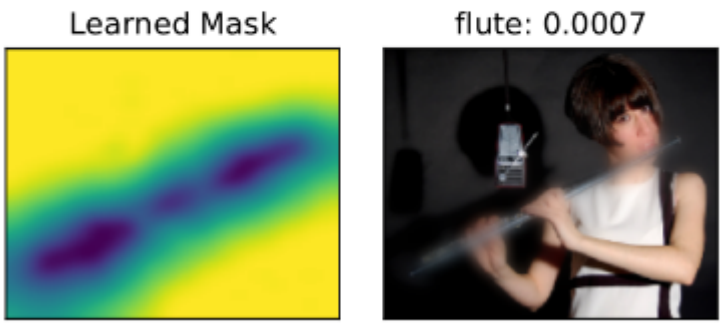
$$m^* = \underset{m \in [0,1]^\Lambda}{\operatorname{argmin}} \lambda \|\mathbf{1} - m\|_1 + f_c(\Phi(x_0; m))$$



Fong and Vedaldi 2017, Interpretable Explanations of Black Boxes by Meaningful Perturbation

Approach 2: Meaningful Perturbations

$$m^* = \underset{m \in [0,1]^\Lambda}{\operatorname{argmin}} \lambda \|\mathbf{1} - m\|_1 + f_c(\Phi(x_0; m))$$



Advantages

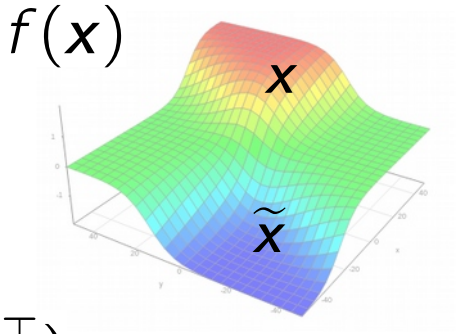
- Can be applied to *any* (differentiable) ML model.

Limitations

- Need to run an optimization procedure

Approach 3: (Simple) Taylor Expansions

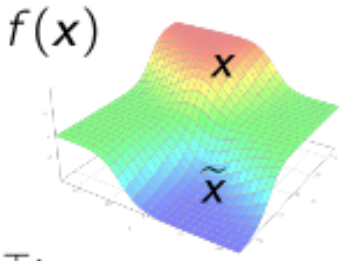
Idea: identify the contribution of input features as the first-order terms of a Taylor expansion



Taylor Expansion

$$f(\boldsymbol{x}) = f(\widetilde{\boldsymbol{x}}) + \sum_{i=1}^{d} \underbrace{[\nabla f(\widetilde{\boldsymbol{x}})]_i \cdot (x_i - \widetilde{x}_i)}_{R_i} + \mathcal{O}(\boldsymbol{x}\boldsymbol{x}^\top)$$

Approach 3: (Simple) Taylor Expansions



Taylor Expansion

$$f(\boldsymbol{x}) = f(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{d} \underbrace{[\nabla f(\tilde{\boldsymbol{x}})]_i \cdot (x_i - \tilde{x}_i)}_{R_i} + \mathcal{O}(\boldsymbol{x}\boldsymbol{x}^\top)$$

Advantages

- Can be applied to *any* (differentiable and mildly nonlinear) ML model.

Limitations

- Need to find a meaningful root point where to perform the expansion.

  ($\rightarrow$ optimization, or heuristics)

Approach 4: Gradient x Input

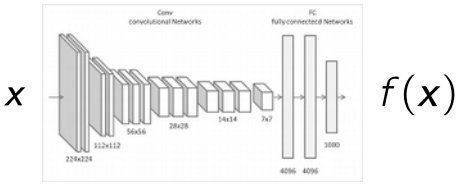
Motivation

- Compute an explanation in a single pass without having to optimize or search for a root point.

Gradient x Input

$$\forall_i : \; R_i = [\nabla f(\boldsymbol{x})]_i \cdot x_i$$

$$\boldsymbol{R} = \nabla f(\boldsymbol{x}) \odot \boldsymbol{x}$$



Approach 4: Gradient x Input

Observation: Complex analyses reduce to gradient x input for simple cases.

Perturbation Analysis

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} x_i w_i + b$$

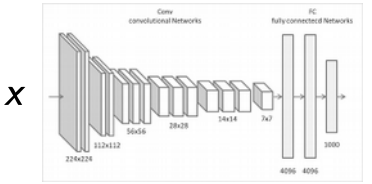
Taylor Expansions

$$\forall_{\boldsymbol{x}, t \geq 0} : \; f(t\boldsymbol{x}) = t f(\boldsymbol{x})$$

Gradient x Input

$$\forall_i : \; R_i = [\nabla f(\boldsymbol{x})]_i \cdot x_i$$

$$\boldsymbol{R} = \nabla f(\boldsymbol{x}) \odot \boldsymbol{x}$$



Question: Does it work in practice?

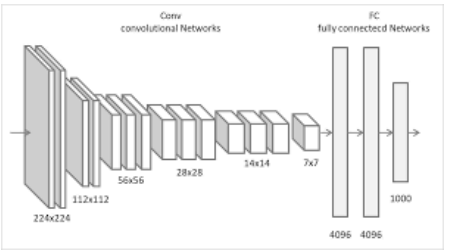
Approach 4: Gradient x Input

Input

x

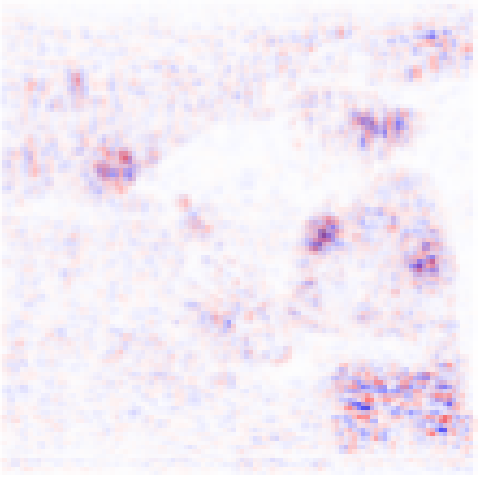
Prediction

(class: baseball)

x 

$f(x)$

Explanation



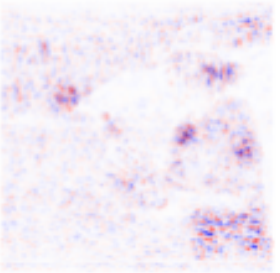
$$R = \nabla f(x) \odot x$$

Alber et al. iNNvestigate Neural Networks, JMLR Software, 2019

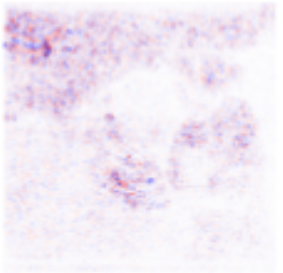
Approach 4: Gradient x Input

Input **Model** **Explanation**

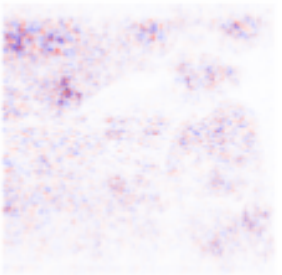
VGG-16



Inception V3



ResNet 50



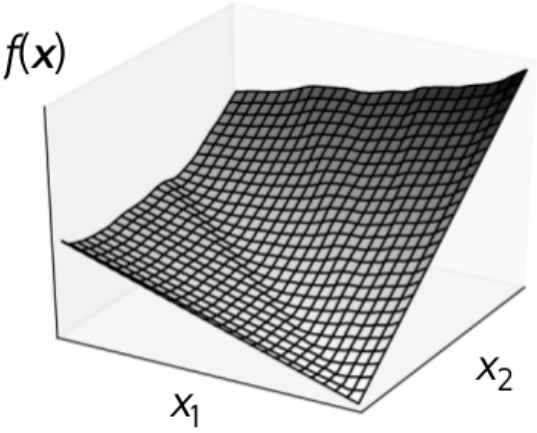
Observation:
Explanations are noisy.

Alber et al. iNNvestigate Neural Networks, JMLR Software, 2019

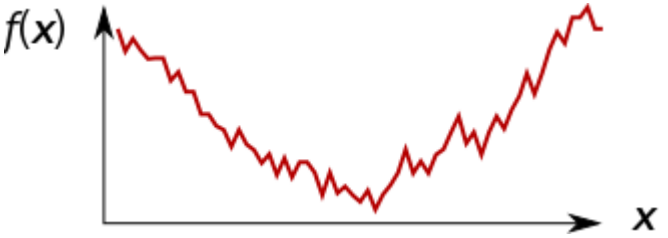
Approach 4: Gradient x Input

Two reasons why explanations are noisy:

Not local enough. Too much context introduced when multiplying by the input.

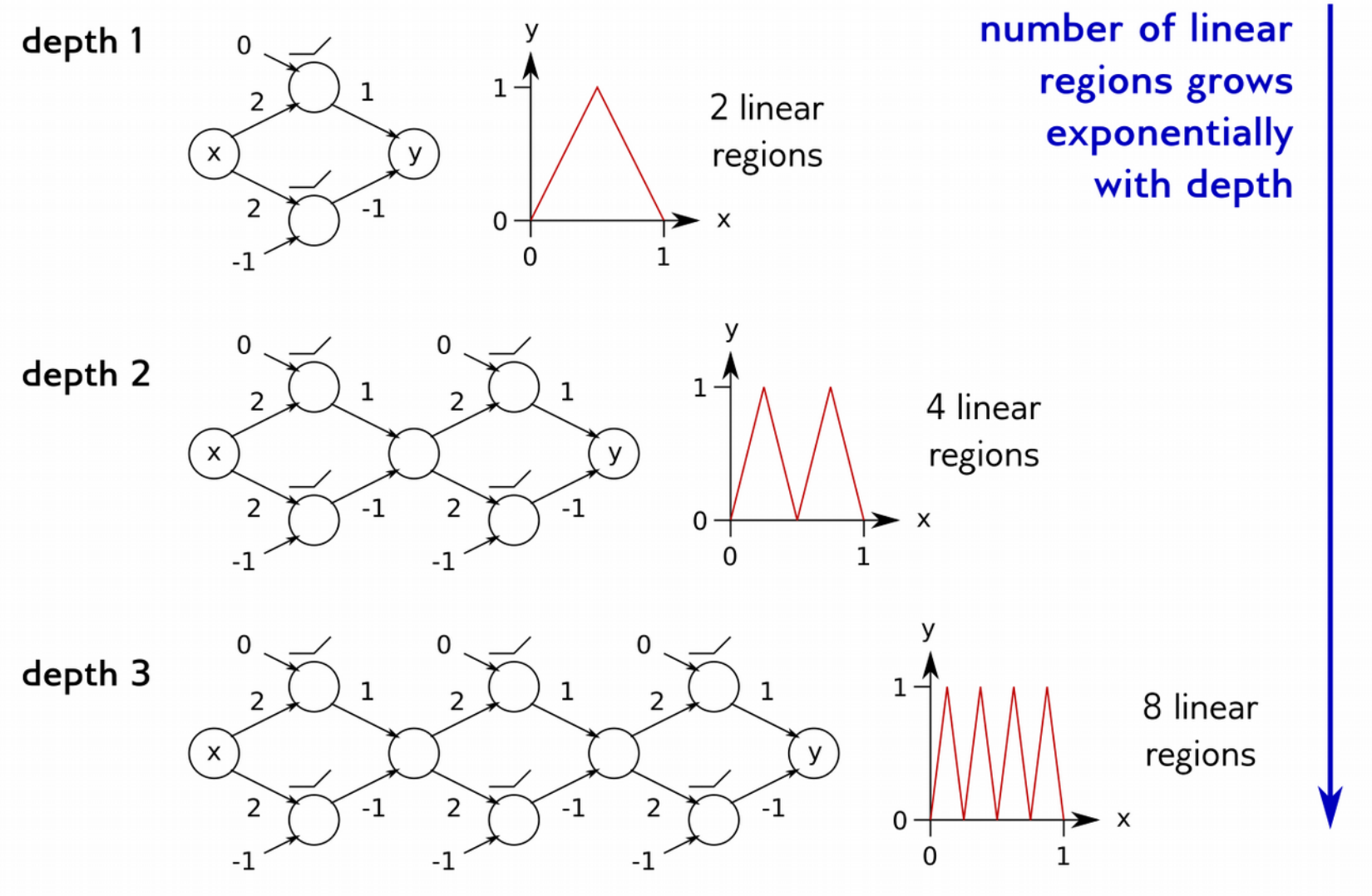


Shattered gradient problem $\rightarrow$ gradient of deep nets has low informative value



Approach 4: Gradient x Input

The Shattered gradients problem [Montufar'14, Balduzzi'17]



Overview of Explanation Methods - Recap

1. Perturbation-Based Methods
   $\rightarrow$ universally applicable but slow

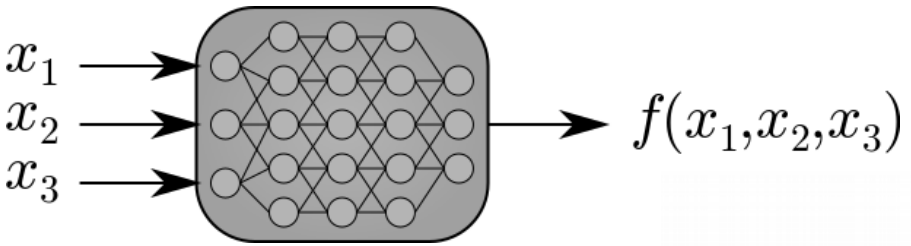
2. Meaningful Perturbations
   $\rightarrow$ widely applicable but requires optimization

3. Taylor Expansions
   $\rightarrow$ quite widely applicable but requires to find a root point

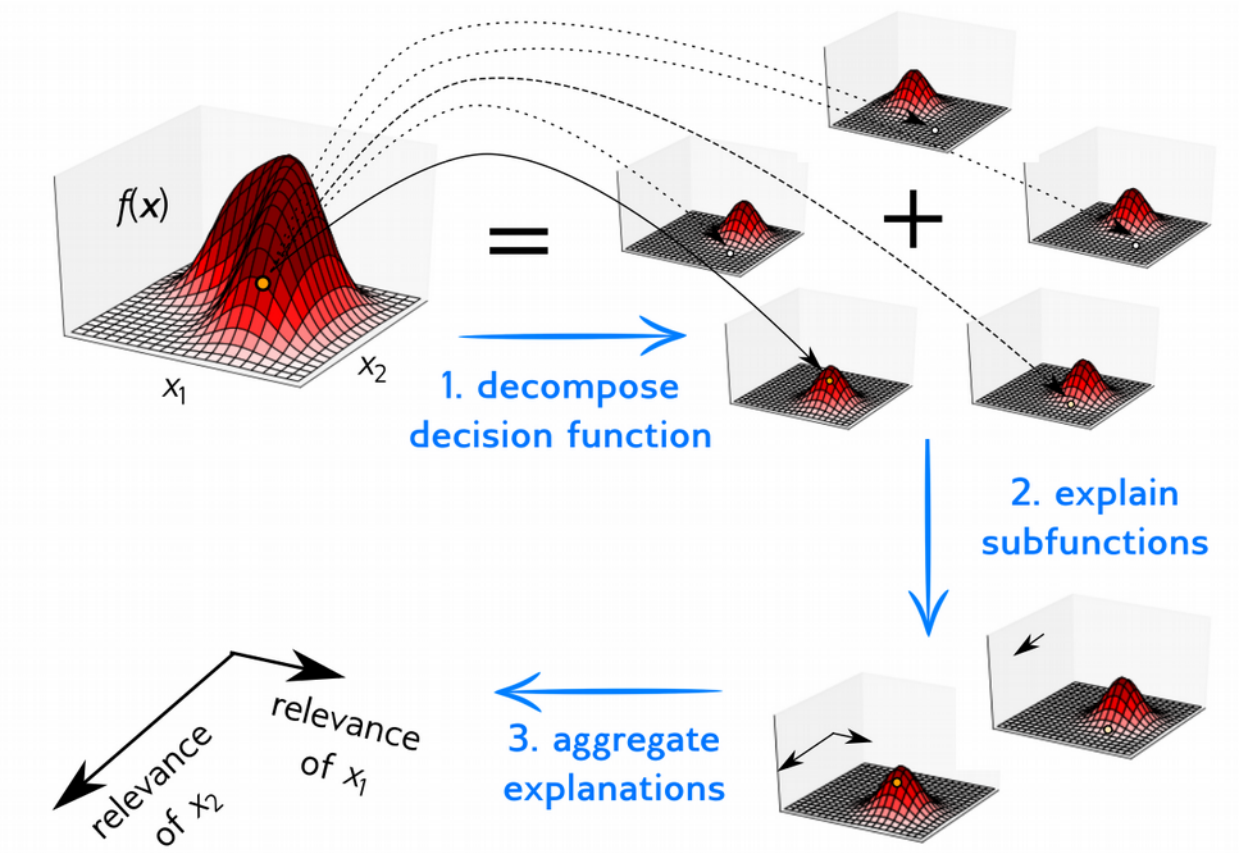
4. Gradient x Input
   $\rightarrow$ applicable with some restrictions
   $\rightarrow$ fast, O(forward pass)
   $\rightarrow$ does not work well on highly nonlinear functions (e.g. DNNs)

Layer-Wise Relevance Propagation

Idea: Reusing Model Structure

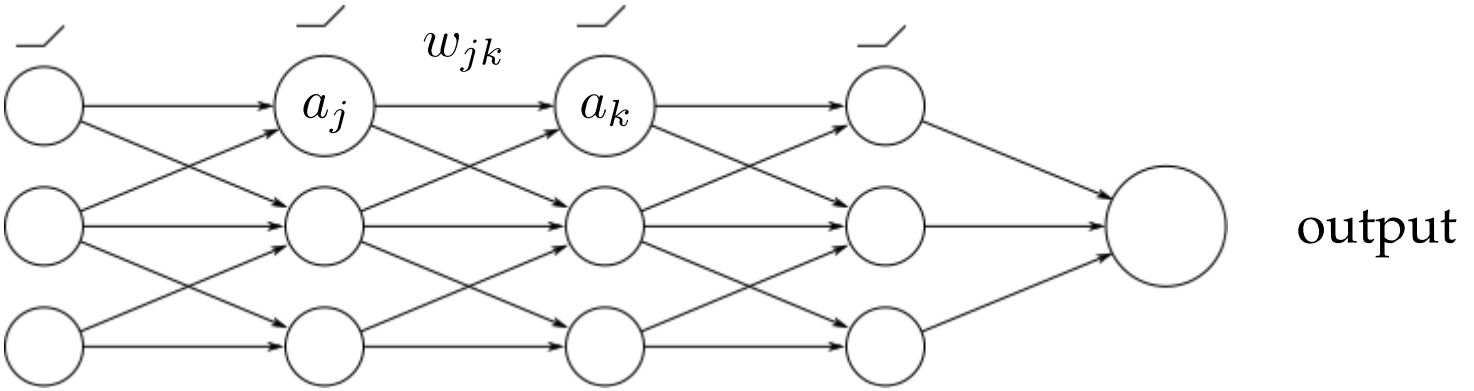


model is a composition of neurons. This can be exploited to make explanation easier.



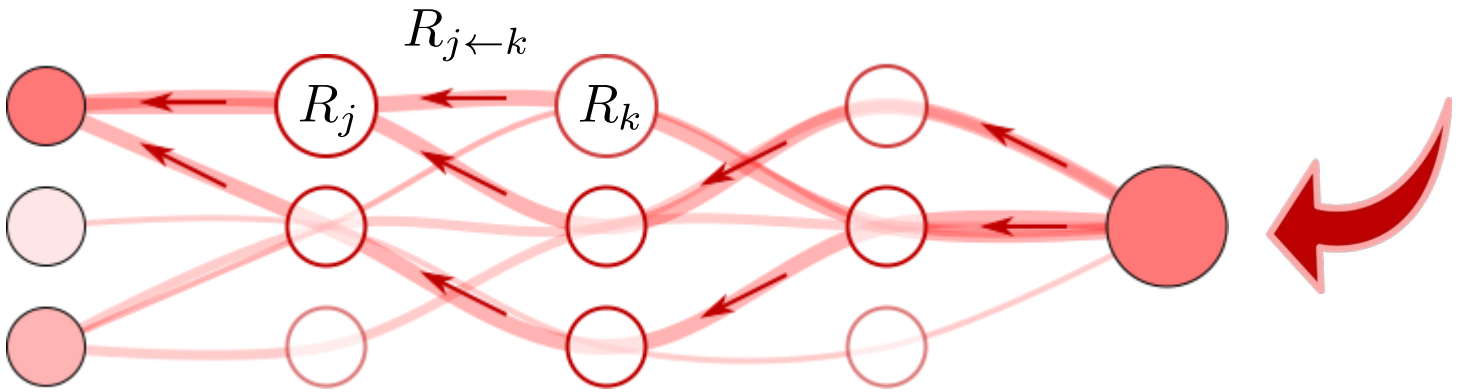
Layer-wise Relevance Propagation (LRP)

1. forward pass



2. conservative propagation

explanation

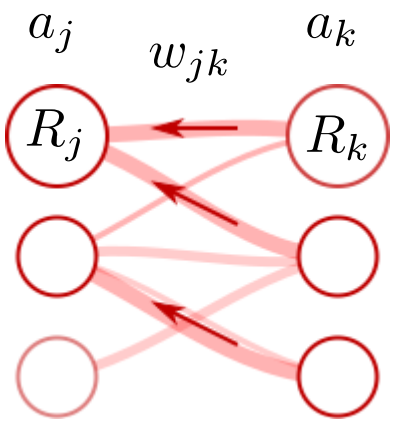


$$R_j = \sum_k R_{j\leftarrow k}$$

$$\sum_j R_{j\leftarrow k} = R_k$$

S Bach, A Binder, G Montavon, F Klauschen, KR Müller, W Samek. On Pixel-wise Explanations [...] by Layer-wise Relevance Propagation, PLOS ONE, 10(7):e0130140, 2015

Various LRP Propagation Rules



LRP-0

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_j a_j w_{jk}} R_k$$

LRP-ϵ

$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_j a_j w_{jk}} R_k$$

LRP-γ

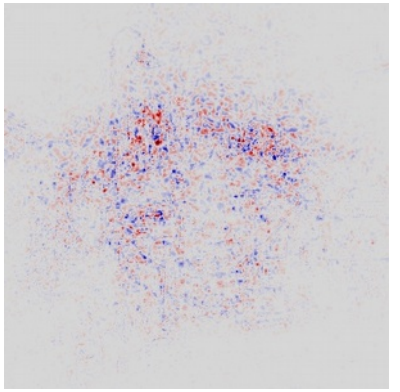
$$R_j = \sum_k \frac{a_j (w_{jk} + \gamma w_{jk}^+)}{\sum_j a_j (w_{jk} + \gamma w_{jk}^+)} R_k$$

Various LRP Propagation Rules

LRP-0

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_j a_j w_{jk}} R_k$$

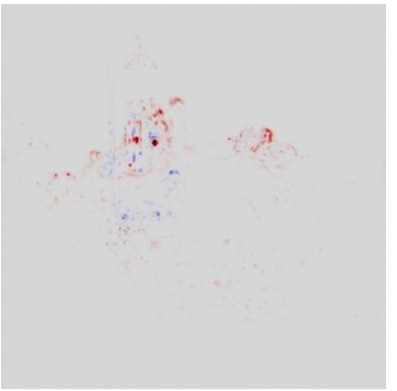
Equivalent to gradient x input, noisy



LRP-ϵ

$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_j a_j w_{jk}} R_k$$

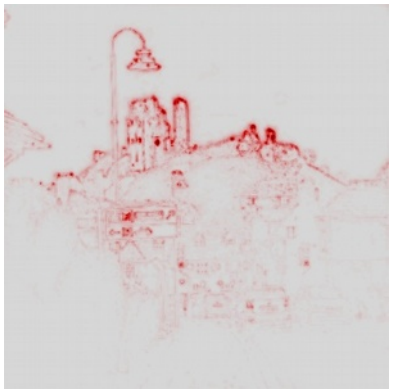
Reduces noise, increases sparsity



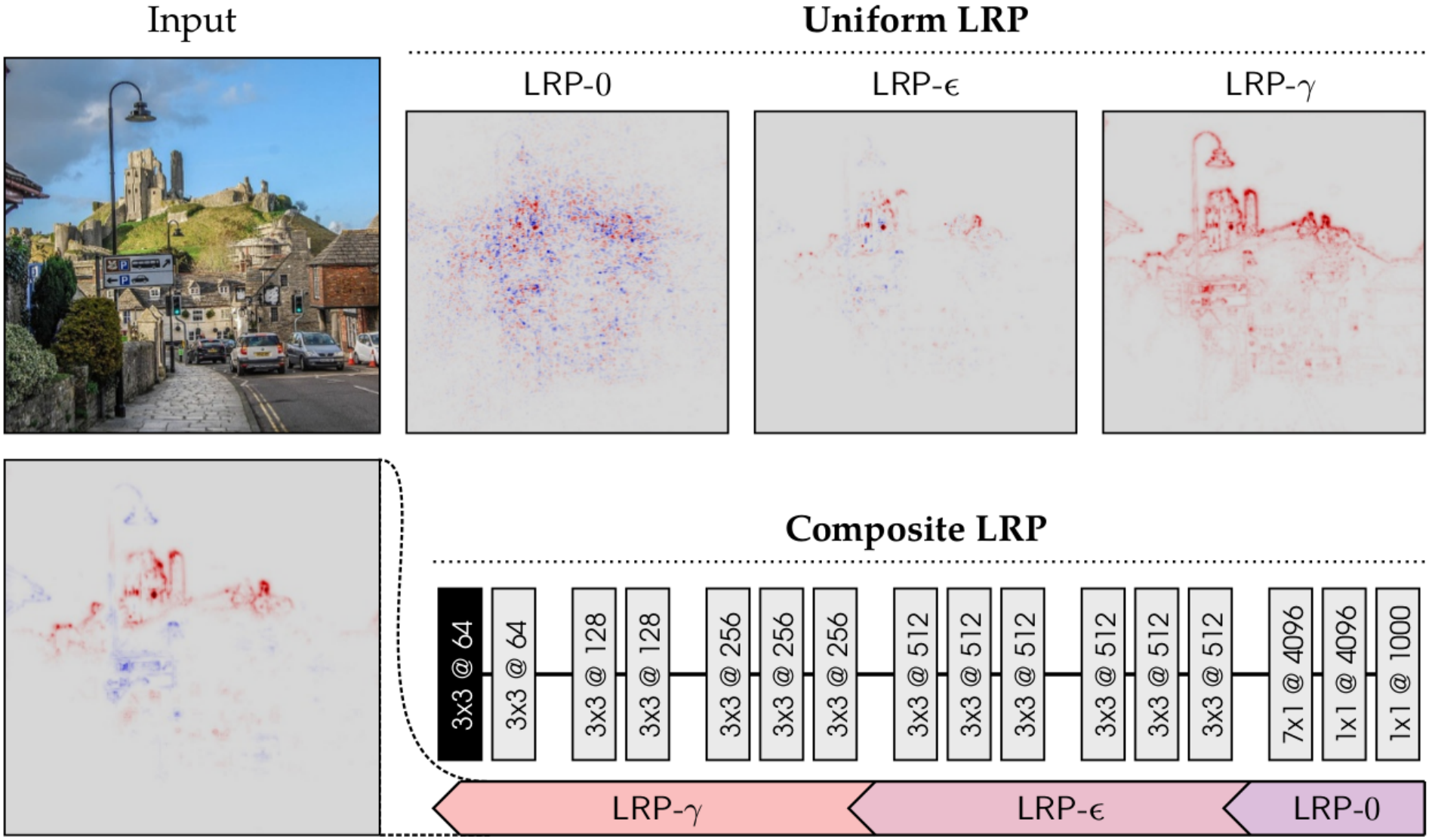
LRP-γ

$$R_j = \sum_k \frac{a_j (w_{jk} + \gamma w_{jk}^+)}{\sum_j a_j (w_{jk} + \gamma w_{jk}^+)} R_k$$

Reduces noise, reduces sparsity



Trick: Use a Different Rule at each Layer



Implementing LRP Efficiently

LRP-0/ϵ/γ

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_j a_j w_{jk}} R_k$$

$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_j a_j w_{jk}} R_k$$

$$R_j = \sum_k \frac{a_j (w_{jk} + \gamma w^+_{jk})}{\sum_j a_j (w_{jk} + \gamma w^+_{jk})} R_k$$

$\subset$

More general LRP rule

$$R_j = \sum_k \frac{a_j \cdot \rho(w_{jk})}{\epsilon + \sum_{0,j} a_j \cdot \rho(w_{jk})} R_k$$

$$\forall_k : \; z_k = \epsilon + \textstyle\sum_{0,j} a_j \cdot \rho(w_{jk}) \qquad \text{(forward pass)}$$

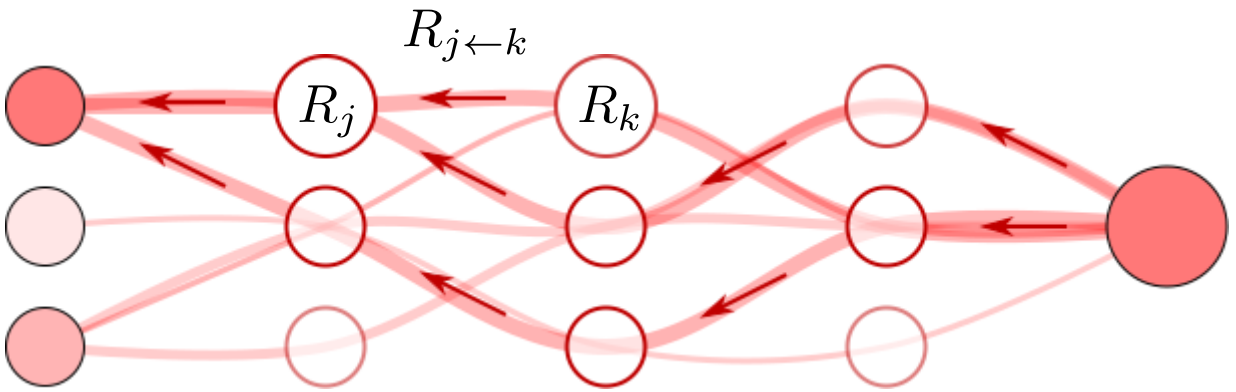
$$\forall_k : \; s_k = R_k / z_k \qquad \text{(element-wise division)}$$

$$\forall_j : \; c_j = \textstyle\sum_k \rho(w_{jk}) \cdot s_k \qquad \text{(backward pass)}$$

$$\forall_j : \; R_j = a_j \, c_j \qquad \text{(element-wise product)}$$

$$c_j = \left[\nabla\left(\textstyle\sum_k z_k(\boldsymbol{a}) \cdot s_k\right)\right]_j$$

Implementing LRP in PyTorch




```
def relprop(a,layer,R):
    z = epsilon + rho(layer).forward(a)
    s = R/(z+1e-9)
    (z*s.data).sum().backward()
    c = a.grad
    R = a*c
    return R
```

Various LRP Rules Used in Practice

Name	Formula	Usage	DTD
LRP-0 [7]	$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$	upper layers	✓
LRP-ϵ [7]	$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} R_k$	middle layers	✓
LRP-γ	$R_j = \sum_k \frac{a_j (w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j (w_{jk} + \gamma w_{jk}^+)} R_k$	lower layers	✓
LRP-$\alpha\beta$ [7]	$R_j = \sum_k \left(\alpha \frac{(a_j w_{jk})^+}{\sum_{0,j} (a_j w_{jk})^+} - \beta \frac{(a_j w_{jk})^-}{\sum_{0,j} (a_j w_{jk})^-} \right) R_k$	lower layers	$\times^\star$
flat [30]	$R_j = \sum_k \frac{1}{\sum_j 1} R_k$	lower layers	$\times$
w^2-rule [36]	$R_i = \sum_j \frac{w_{ij}^2}{\sum_i w_{ij}^2} R_j$	first layer ($\mathbb{R}^d$)	✓
$z^\mathcal{B}$-rule [36]	$R_i = \sum_j \frac{x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-}{\sum_i x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-} R_j$	first layer (pixels)	✓

(* DTD interpretation only for the case $\alpha = 1, \beta = 0$.)

Justifying LRP as a 'Deep Taylor Decomposition'

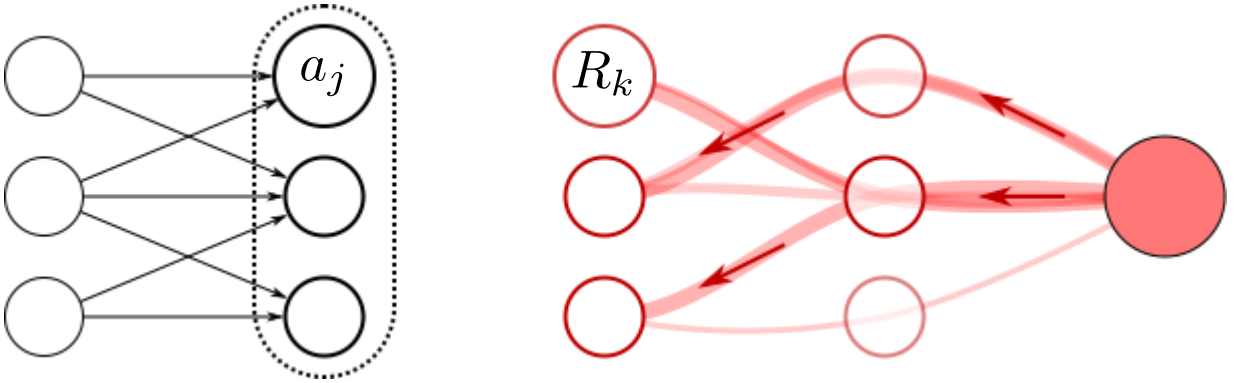
Simple Taylor Decomposition

$$\boldsymbol{x} \mapsto f(\boldsymbol{x})$$

$$f(\boldsymbol{x}) = f(\widetilde{\boldsymbol{x}}) + \sum_{i=1}^{d} [\nabla f(\widetilde{\boldsymbol{x}})]_i \cdot (x_i - \widetilde{x}_i) + \mathcal{O}(\boldsymbol{x}\boldsymbol{x}^\top)$$

Deep Taylor Decomposition

$$\boldsymbol{a} \mapsto R_k(\boldsymbol{a})$$

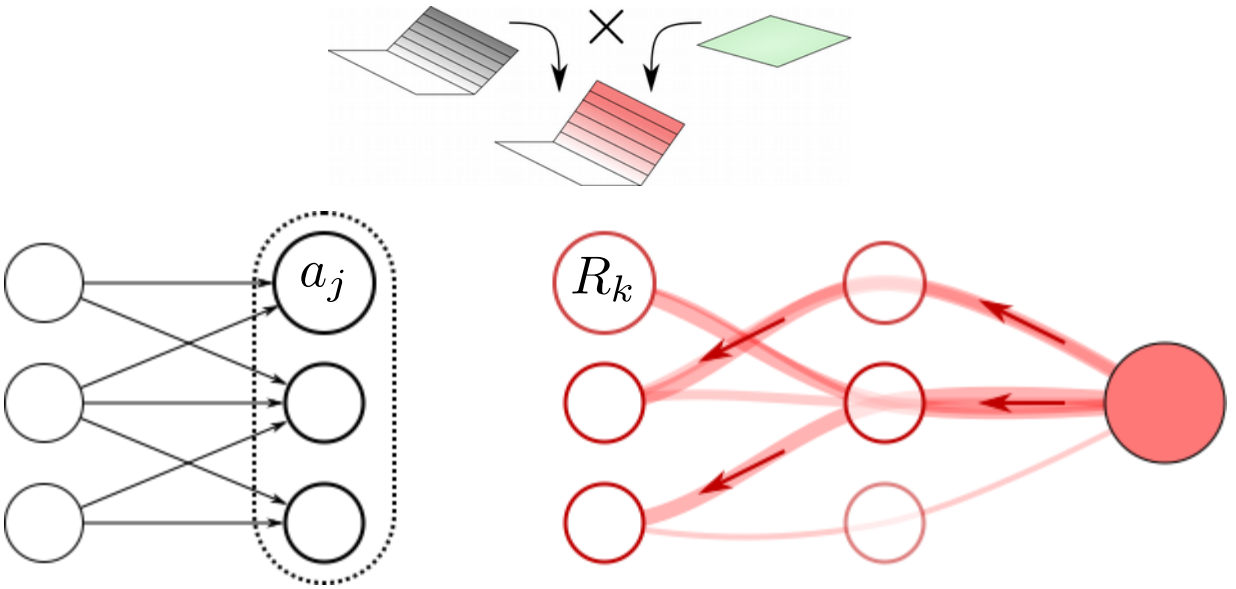


$$R_k(\boldsymbol{a}) = R_k(\widetilde{\boldsymbol{a}}) + \sum_j [\nabla R_k(\widetilde{\boldsymbol{a}})]_j \cdot (a_j - \widetilde{a}_j) + \mathcal{O}(\boldsymbol{a}\boldsymbol{a}^\top)$$

hard to analyze

G Montavon, S Lapuschkin, A Binder, W Samek, KR Müller. Explaining NonLinear Classification Decisions with Deep Taylor Decomposition, Pattern Recognition, 65:211–222, 2017

Deep Taylor Decomposition



Key Idea: Use a "relevance model" that is easy to analyze

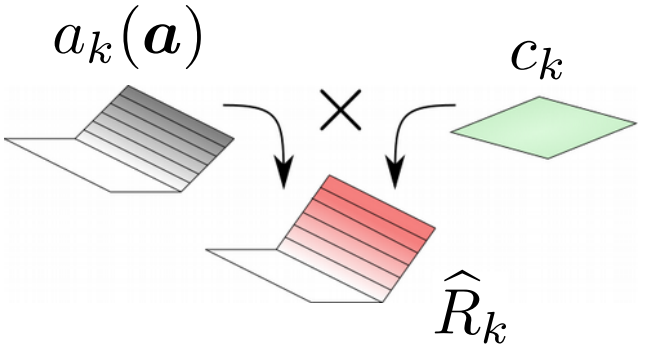
$$\widehat{R}_k(\boldsymbol{a}) = \max(0, \textstyle\sum_j a_j\, w_{jk})\, c_k$$

G Montavon, S Lapuschkin, A Binder, W Samek, KR Müller. Explaining NonLinear Classification Decisions with Deep Taylor Decomposition, Pattern Recognition, 65:211–222, 2017

Deep Taylor Decomposition

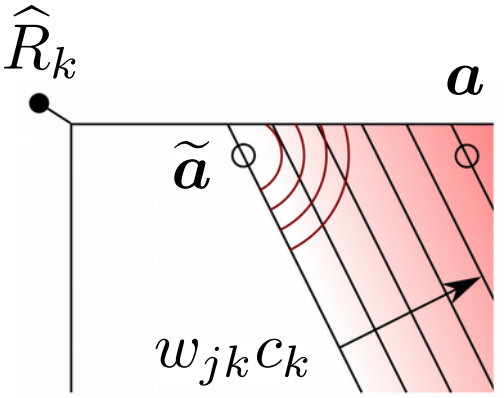
1. Relevance model

$$\widehat{R}_k(\boldsymbol{a}) = \max(0, \textstyle\sum_j a_j\, w_{jk})\, c_k$$



2. Taylor expansion

$$\widehat{R}_k(\boldsymbol{a}) = \widehat{R}_k(\widetilde{\boldsymbol{a}}) + \textstyle\sum_j \underbrace{(a_j - \widetilde{a}_j) \cdot w_{jk}\, c_k}_{R_{j \leftarrow k}} + 0$$



3. Choosing the reference point

$$\widetilde{\boldsymbol{a}}^{(k)} = \mathbf{0} \quad \Longleftrightarrow \quad \rho = (\cdot), \epsilon = 0 \qquad \text{(LRP-0)}$$

$$\widetilde{\boldsymbol{a}}^{(k)} = \boldsymbol{a} - t \cdot \boldsymbol{a} \quad \Longleftrightarrow \quad \rho = (\cdot), \epsilon = (t^{-1} - 1) \cdot a_k \qquad \text{(LRP-}\epsilon\text{)}$$

$$\widetilde{\boldsymbol{a}}^{(k)} = \boldsymbol{a} - t \cdot \boldsymbol{a} \odot \mathbf{1}_{\boldsymbol{w}_k \succ \mathbf{0}} \quad \Longleftrightarrow \quad \rho = \max(0, \cdot) \qquad \text{(LRP-}\gamma\text{)}$$

LRP What's New

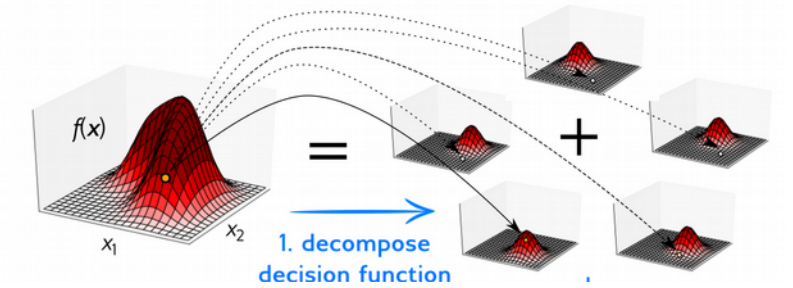
LRP What's New

1. Neuralization Propagation (NEON)

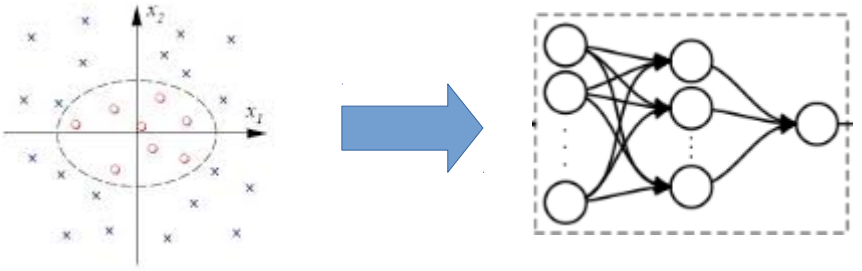
2. Dataset-Wide Analysis with SpRAy

NEON (Neuralization-Propagation)

LRP's idea: To robustly explain a model, leverage the neural network structure of the decision function.



NEON's idea: When the ML model is not a neural network (e.g. a kernel machine), convert it into a neural network first ('neuralize' it).



Neuralizing the One-Class SVM

Original one-class SVM structuration:

$$g(\boldsymbol{x}) = \sum_{j=1}^{m} \alpha_j \, \mathbb{k}(\|\boldsymbol{x} - \mathbf{u}_j\|).$$

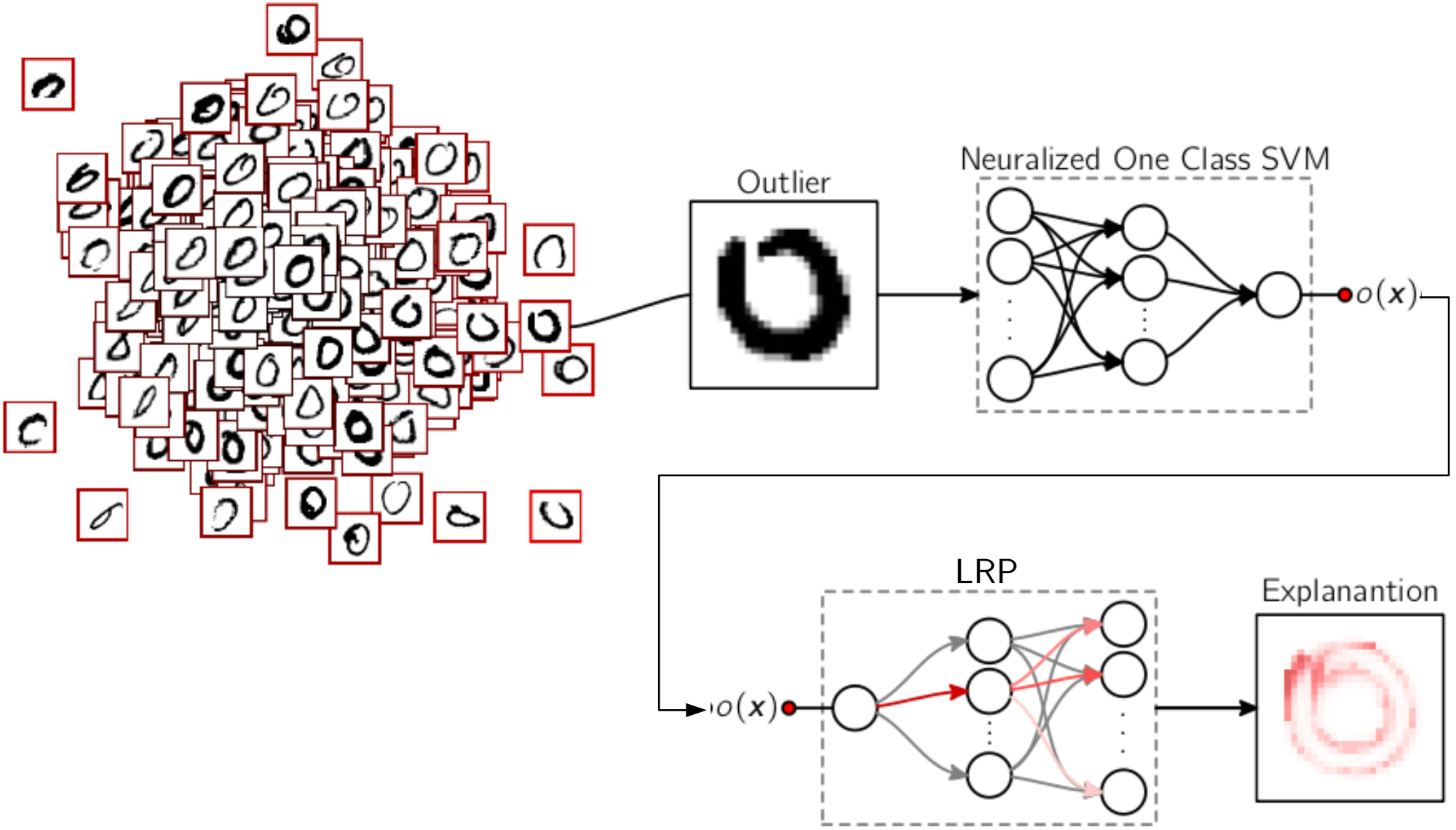
Neuralized one-class SVM:

outlierness (exponential kernel)

$$\text{layer 1:} \quad h_j = \frac{\|\boldsymbol{x} - \mathbf{u}_j\|^q}{q \cdot \sigma^q} - \log \alpha_j \qquad \text{(distance)}$$

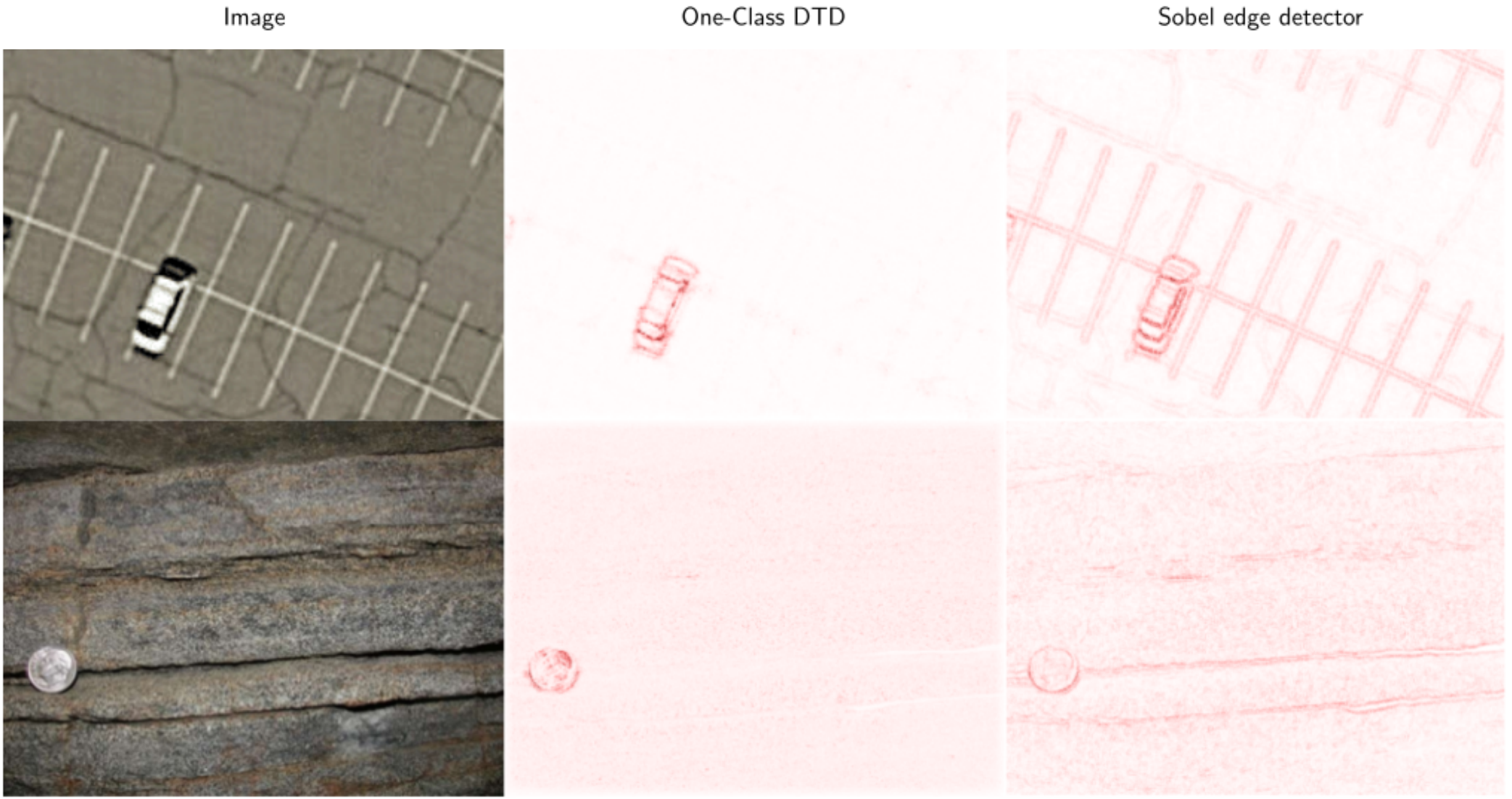
$$\text{layer 2:} \quad o = -\text{LSE}(-(h_j)_j) \qquad \text{(min-pooling)}$$

Neuralizing the One-Class SVM



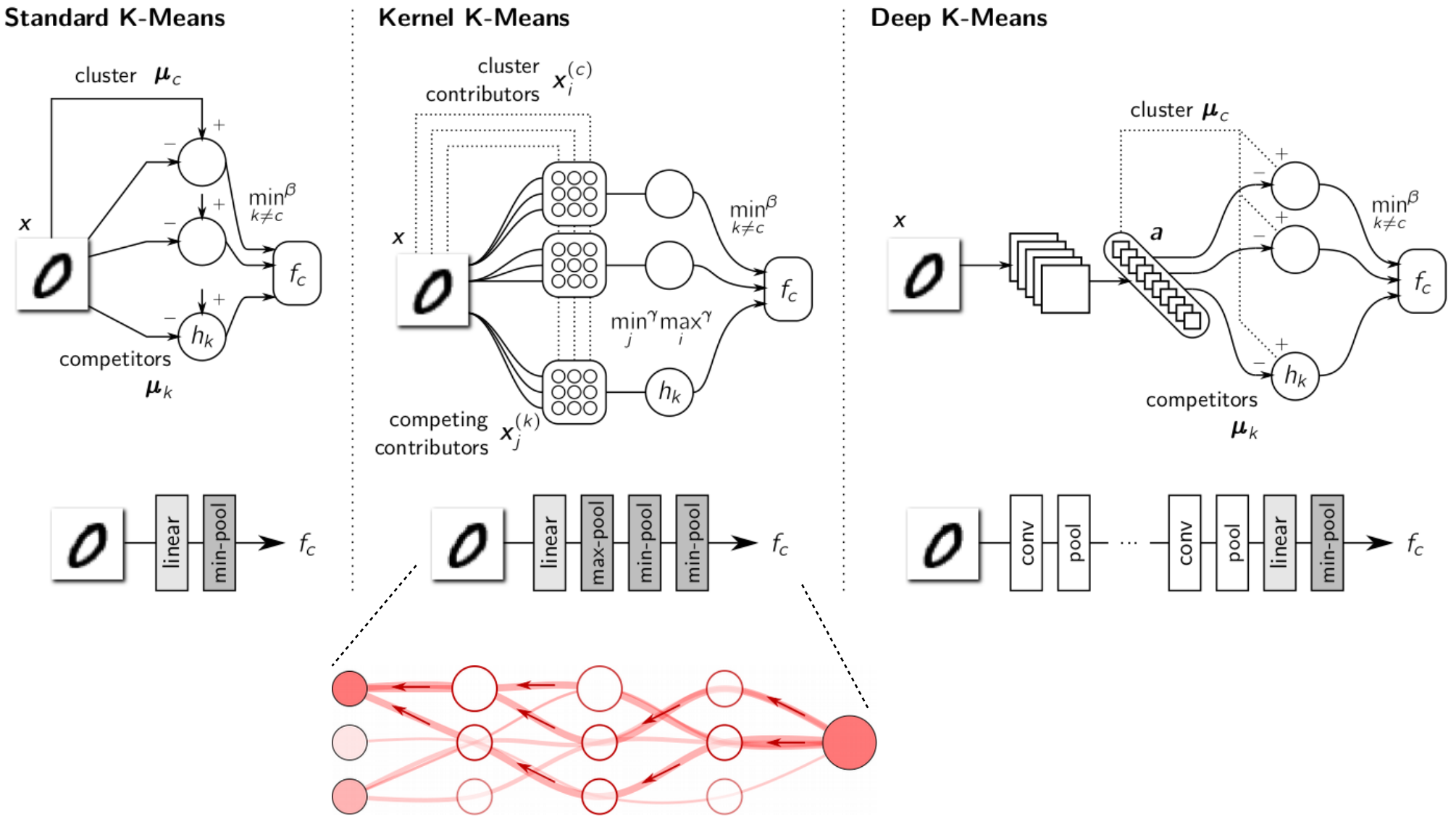
J Kauffmann, KR Müller, G Montavon. Towards Explaining Anomalies: A Deep Taylor Decomposition of One-Class Models, arXiv:1805.06230, 2018

Neuralized One-Class SVM



J Kauffmann, KR Müller, G Montavon. Towards Explaining Anomalies: A Deep Taylor Decomposition of One-Class Models, arXiv:1805.06230, 2018

Neuralizing K-means



J Kauffmann, M Esders, G Montavon, W Samek, KR Müller. From Clustering to Cluster Explanations via Neural Networks, arXiv:1906.07633, 2019

LRP What's New

1. Neuralization Propagation (NEON)

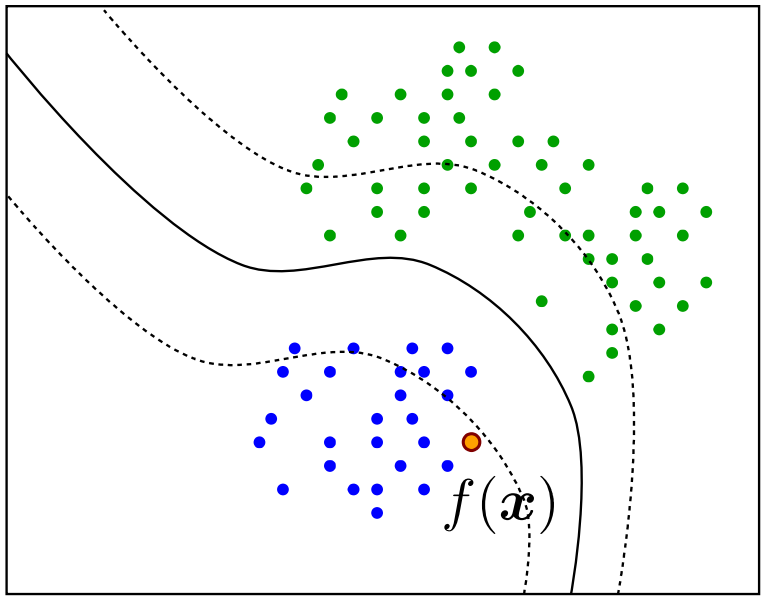
2. Dataset-Wise Analysis with SpRAy

Dataset-Wide Analysis

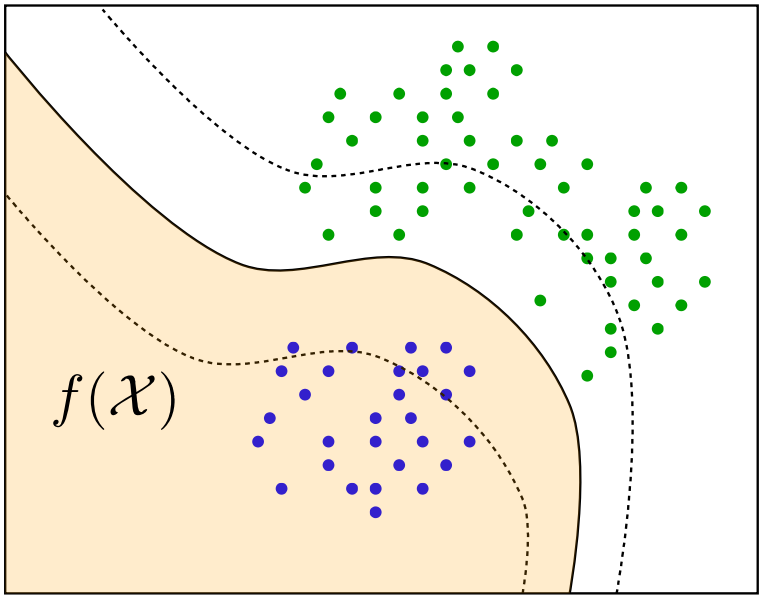
LRP's idea: Explain *individual* decisions of a ML model in a way that is reliable and interpretable for a human.

SpRAy's idea: Explain *whole dataset* decisions of a ML model by systematically analyzing distributions of LRP heatmaps.

LRP



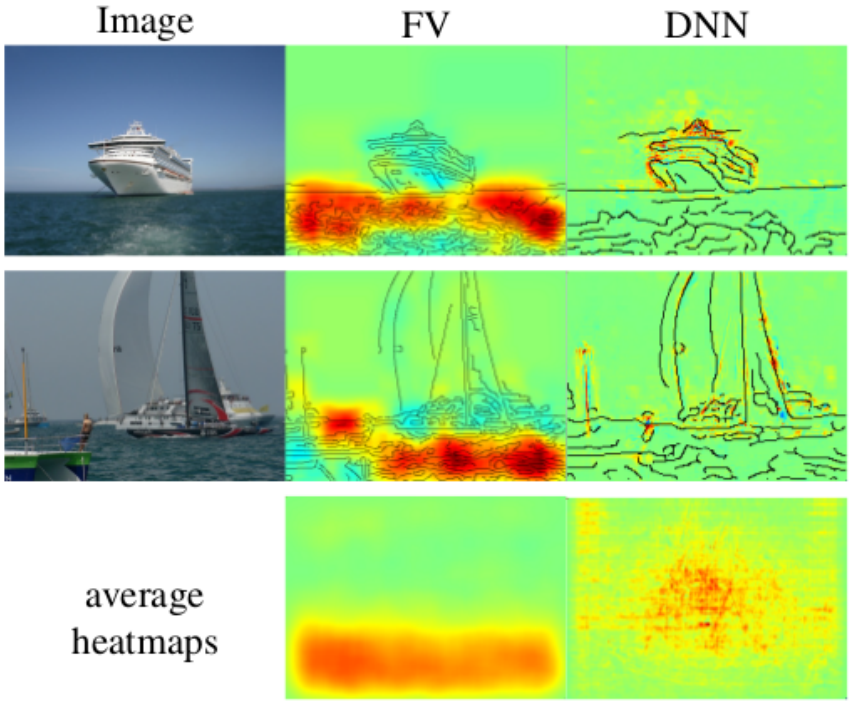
SpRAy

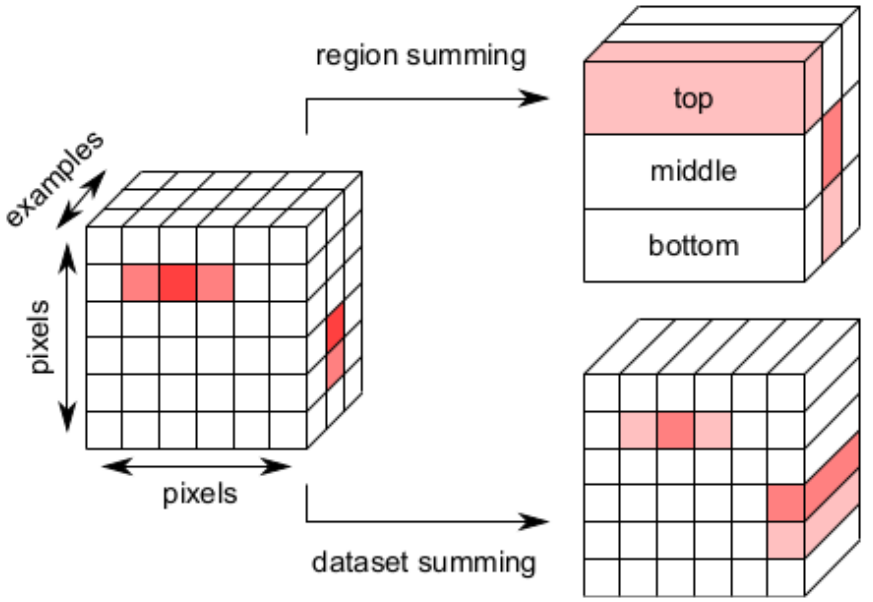


Dataset-Wide Analysis

Idea: detect different strategies of classifiers on dataset-wide basis.

This analysis is possible due to the conservation property of LRP.



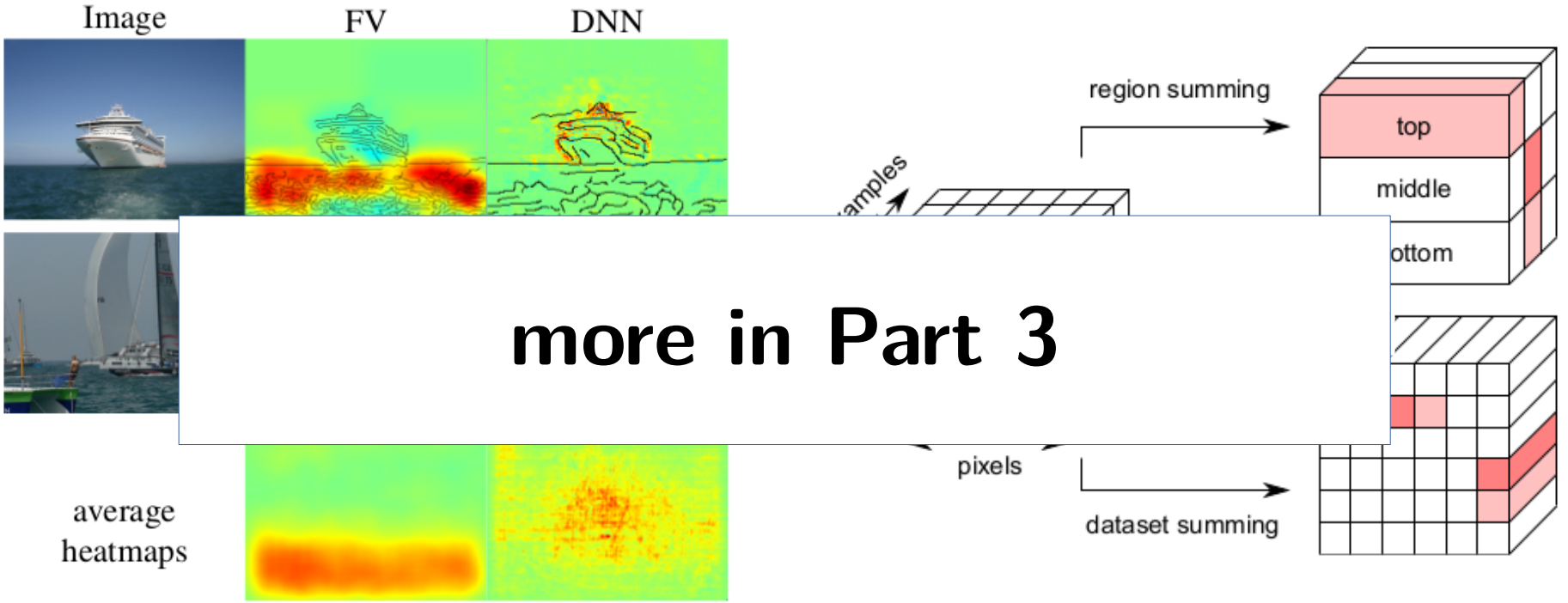


S Lapuschkin, A Binder, G Montavon, KR Müller, W Samek. Analyzing Classifiers: Fisher Vectors and Deep Neural Networks, IEEE CVPR, 2912-2920, 2016

Dataset-Wide Analysis

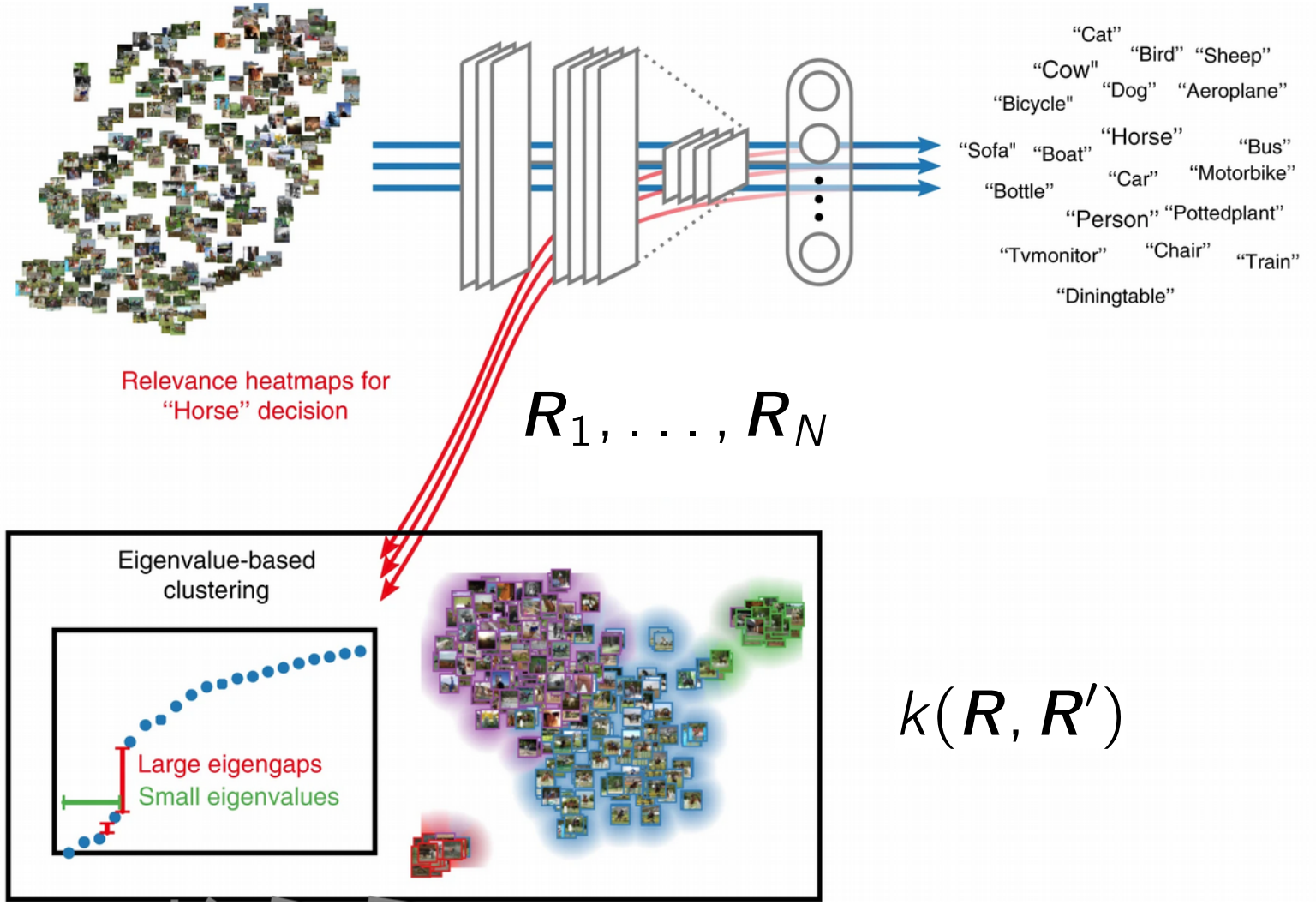
Idea: detect different strategies of classifiers on dataset-wide basis.

This analysis is possible due to the conservation property of LRP.



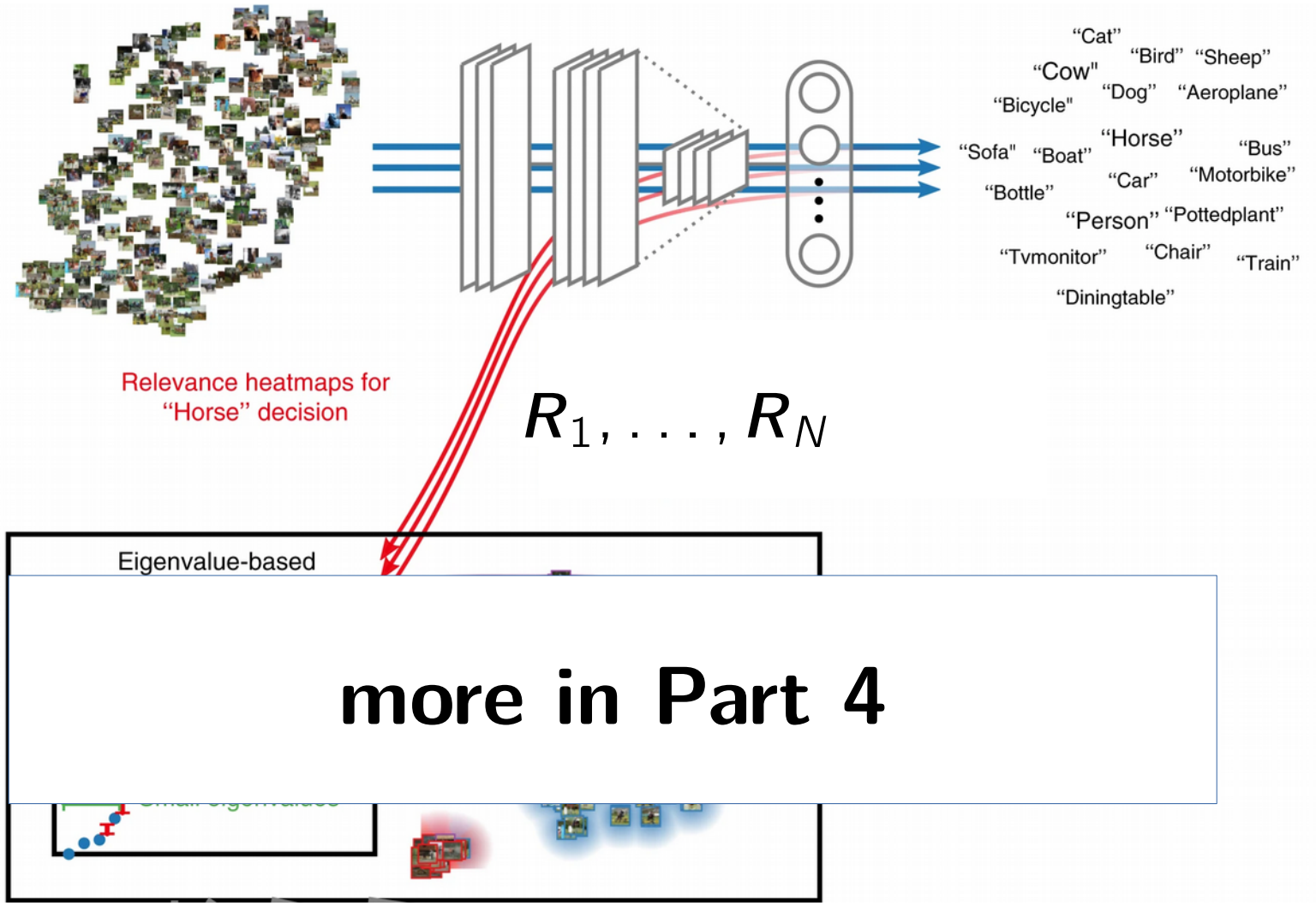
more in Part 3

SpRAy (Spectral Relevance Analysis)



S Lapuschkin, S Wäldchen, A Binder, G Montavon, W Samek, KR Müller. Unmasking Clever Hans Predictors and Assessing What Machines Really Learn, Nature Communications, 10:1096, 2019

SpRAy (Spectral Relevance Analysis)



Lapuschkin et al. Unmasking Clever Hans predictors and assessing what machines really learn (2019)

Open Challenges

Open Challenges: Systematic Application

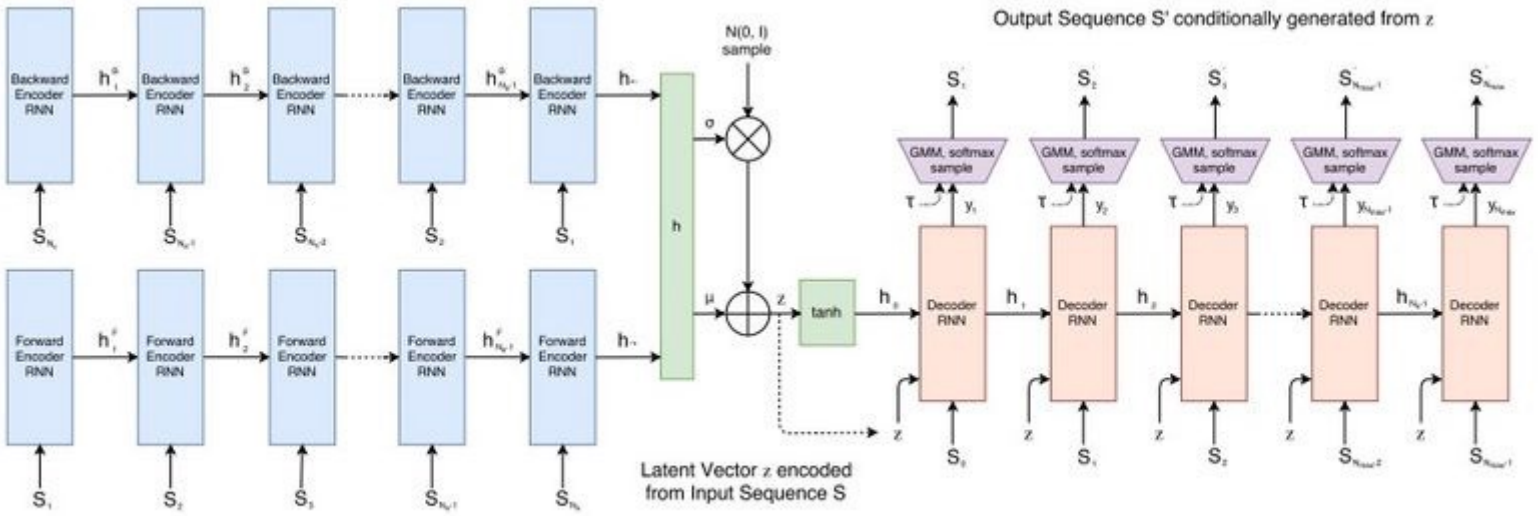
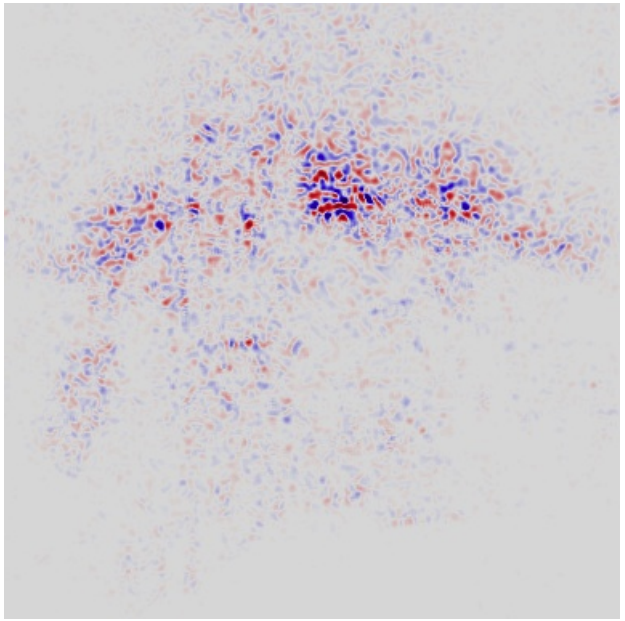
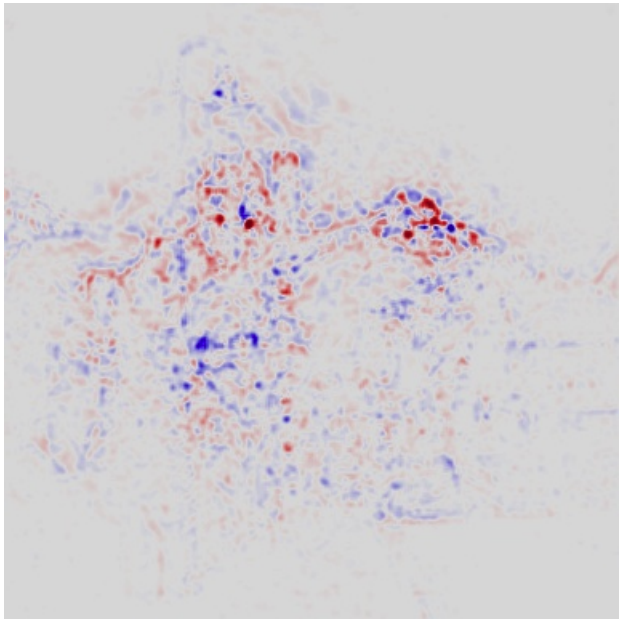
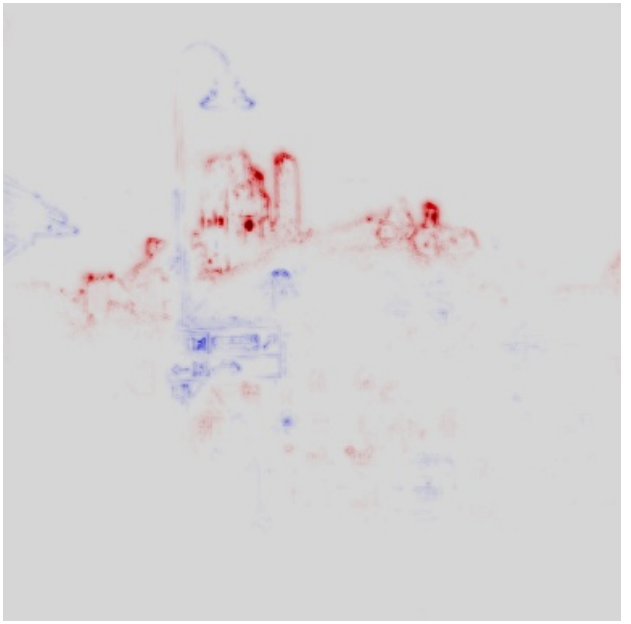


Figure 2: Schematic diagram of `sketch-rnn`.

- How much manual tuning is needed to **adapt** LRP to new architectures?
- Can explanation techniques be implemented in a **modular** way?
- Can explanation be made **differentiable** and learned?

Open Challenges: Systematic Evaluation

- How to evaluate the quality of an explanation?
- Is there a tradeoff between explanation faithfulness and interpretabilty?
- What are the limits of explanations.

Open Challenges: Systematic Evaluation

more in Part 3

- How to evaluate the quality of an explanation?
- Is there a tradeoff between explanation faithfulness and interpretabilty?
- What are the limits of explanations?

Summary

- Before explaining a ML model, it is important to ask whether a given explanation techniques provides the desired type of explanation (e.g. local vs. global explanation).
- Many methods have been proposed explaining individual predictions. LRP requires to carefully tune propagation rules. After this initial step, LRP works quickly and reliably.
- LRP is not simply heuristics, LRP rules can be derived form the deep Taylor decomposition framework.
- Explanation methods such as LRP can be combined with other techniques to extend their scope of application (e.g. NEON for use with kernels, SpRAy for dataset-wide analysis).

Check our website

Online demos, tutorials, code examples, etc.

and tutorial papers

G Montavon, W Samek, KR Müller: Methods for Interpreting and Understanding Deep Neural Networks Digital Signal Processing, 73:1-15, 2018

G Montavon, A Binder, S Lapuschkin, W Samek, KR Müller: Layer-wise Relevance Propagation: An Overview, Springer LNCS 11700, 2019 (to appear)

References

- S Bach, A Binder, G Montavon, F Klauschen, KR Müller, W Samek. On Pixel-wise Explanations [...] by Layer-wise Relevance Propagation, PLOS ONE, 10(7):e0130140, 2015
- J Kauffmann, KR Müller, G Montavon. Towards Explaining Anomalies: A Deep Taylor Decomposition of One-Class Models, arXiv:1805.06230, 2018
- J Kauffmann, M Esders, G Montavon, W Samek, KR Müller. From Clustering to Cluster Explanations via Neural Networks, arXiv:1906.07633, 2019
- S Lapuschkin, S Wäldchen, A Binder, G Montavon, W Samek, KR Müller. Unmasking Clever Hans Predictors and Assessing What Machines Really Learn, Nature Communications, 10:1096, 2019
- S Lapuschkin, A Binder, G Montavon, KR Müller, W Samek. Analyzing Classifiers: Fisher Vectors and Deep Neural Networks, IEEE CVPR, 2912-2920, 2016
- G Montavon, S Lapuschkin, A Binder, W Samek, KR Müller. Explaining NonLinear Classification Decisions with Deep Taylor Decomposition, Pattern Recognition, 65:211–222, 2017
- G Montavon, W Samek, KR Müller: Methods for Interpreting and Understanding Deep Neural Networks Digital Signal Processing, 73:1-15, 2018
- G Montavon, A Binder, S Lapuschkin, W Samek, KR Müller: Layer-wise Relevance Propagation: An Overview, Springer LNCS 11700, 2019 (to appear)