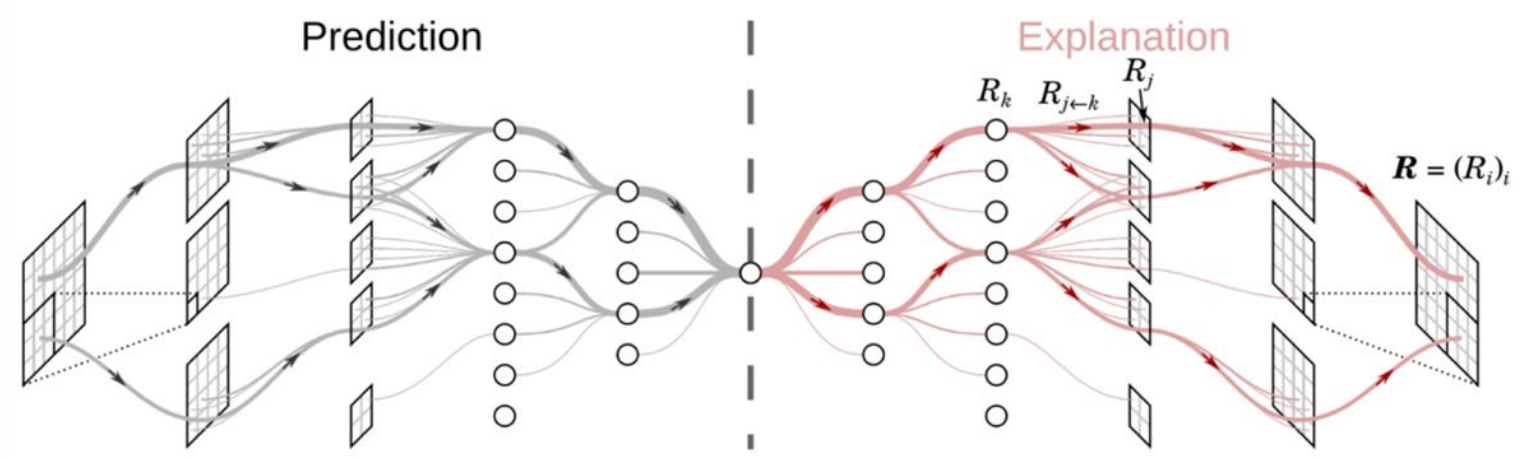
From Feature Attributions to Next-Generation Explainable AI

Wojciech Samek

TU Berlin & Fraunhofer HHI



Syllabus

Part I: First Generation XAI

- What to explain
- Explaining by attribution
- DTD Framework
- Evaluating explanations

Part II: New Developments

- Concepts and prototypes
- XAI for LLMs
- Non-Interpretable domains
- Beyond classification

Part III: Beyond Explaining

- XAI-Based model surgery
- Reveal and revise
- Explanatory interactive ML
- Future of XAI

First Wave of Explainable AI Research

Zintgraf'17
Pred Diff

Ribeiro'16
LIME

Bach'15
LRP

Zhou'16
GAP

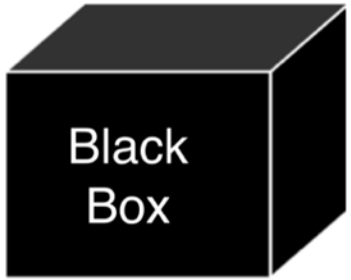
Zeiler'14
Occlusion

Fong'17
M Perturb

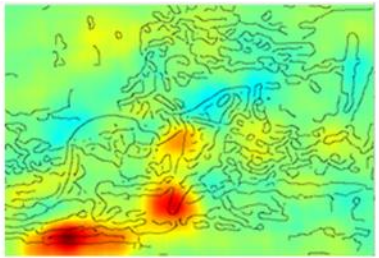
Zeiler'14
Deconv

Kindermanns'17
PatternNet

classify:
"horse"



explain:
"heatmap"



Zhang'16
Excitation BP

Landecker'13
Contrib Prop

Lungberg'17
SHAP

Shrikumar'17
DeepLIFT

Springenberg'14
Guided BP

Selvaraju'17
Grad-CAM

Symonian'13
Gradient

Sundarajan'17
Int Grad

Montavon'17
Deep Taylor

First Wave of Explainable AI Research

Zintgraf'17
Pred Diff

Ribeiro'16
LIME

Fong'17

Kindermanns'17
PatternNet

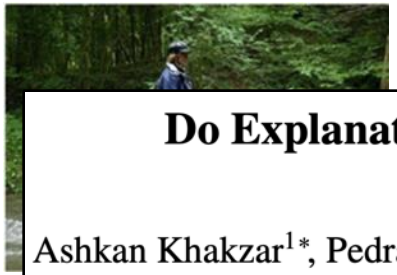
When Explanations Lie: Why Many Modified BP Attributions Fail

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

class

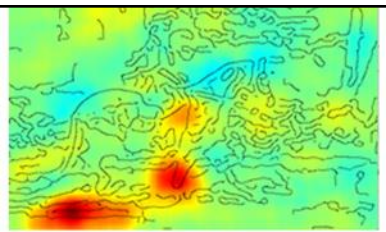
"horse"

heatmap



Do Explanations Explain? Model Knows Best

Ashkan Khakzar[1*], Pedram Khorsandi[1*], Rozhin Nobahari[2*], Nassir Navab[1]



Zhang'16
Excitation BP

Sanity Checks for Saliency Maps

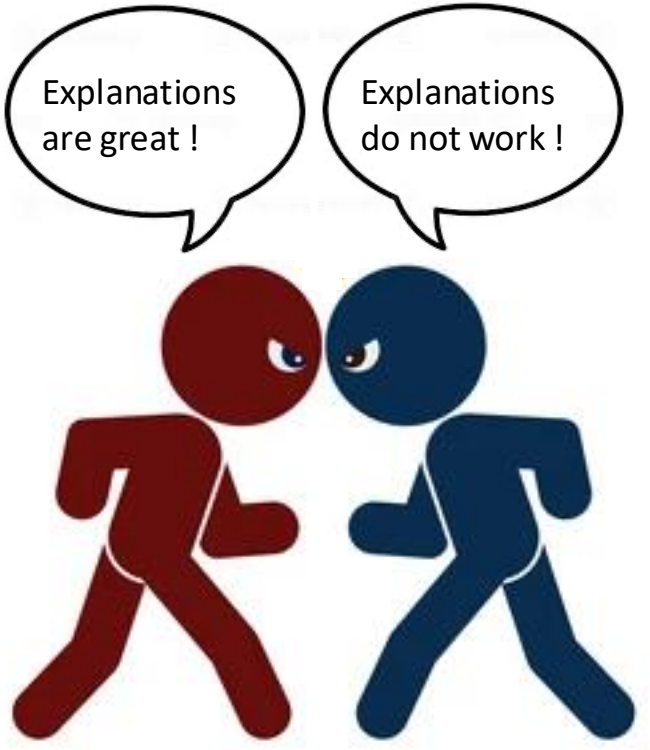
Julius Adebayo[*], Justin Gilmer[♯], Michael Muelly[♯], Ian Goodfellow[♯], Moritz Hardt[♯†], Been Kim[♯]

Springenberg'14
Guided BP

Selvaraju'17
Grad-CAM

"Interpretability may be one of the most confused topics in all of machine learning, fraught with confusion and conflict. Read an interpretability paper selected at random and you'll find representations (or insinuations) that the work is addressing "trust", "insights", "fairness", "causality". Then look at what the authors actually do and you'll be hard-pressed to tie back the method to any of these underlying motivations. Half the papers produce a set of feature important scores."
– Zachary Lipton in Goldblum et al. (2023)

The Two Cultures



The Two Cultures

Leo Breiman

Statistical Science
2001, Vol. 16, No. 3, 199–231

Statistical Modeling: The Two Cultures

Leo Breiman

Abstract. There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools.

The Two XAI Cultures

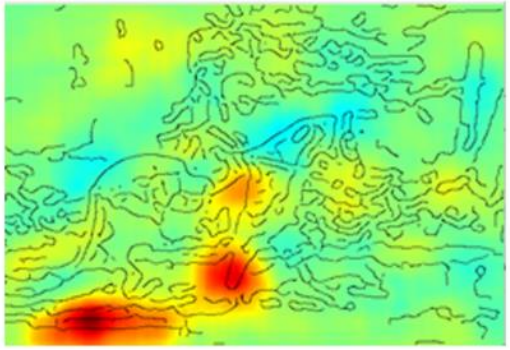
	Model-validation oriented RED XAI	Human-values oriented BLUE XAI
Why explanations are produced?	Research on data, Explore models, Debug models	responsiBle models, Legal issues, trUst in predictions, Ethical issues
When explanations are read and used?	Empower model developer, mostly during training	Empower user, mostly during model inference
Who is the direct audience of the explanations?	Power user, Model developers, AI researchers	Lay user, Customer, Patient
What are desired characteristics of explanations	Faithful to model and data, Actionable	Simple and easy to understand

(Biecek & Samek, ICML, 2024)

Why Explaining?

Verify & debug

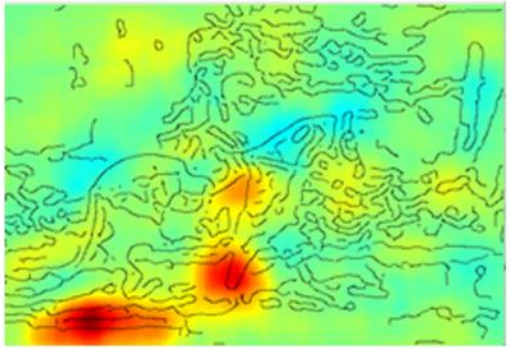
(Lapuschkin et al. Nat Comm, 2019)



Why Explaining?

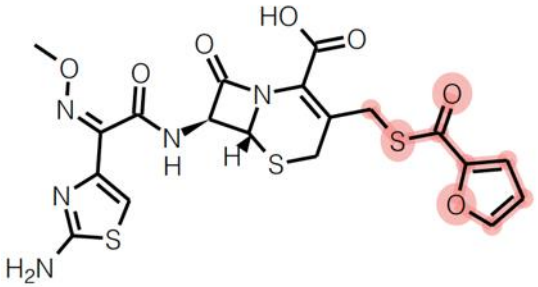
Verify & debug

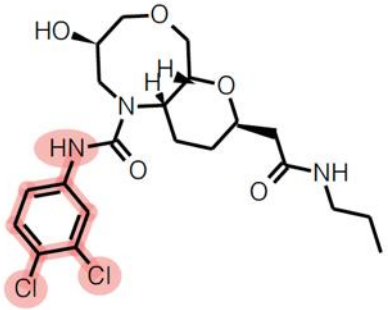
(Lapuschkin et al. Nat Comm, 2019)



New insights

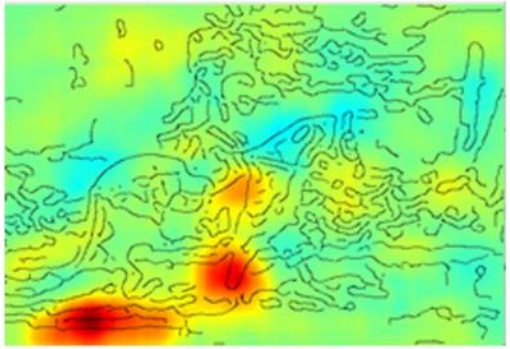
(Wong et al. Nature, 2023)



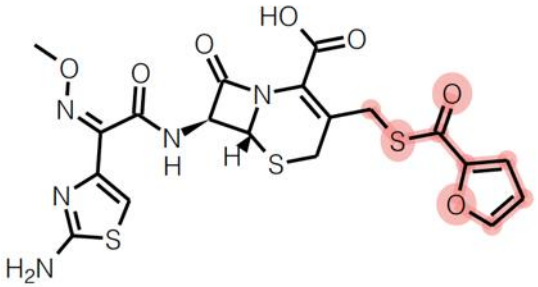


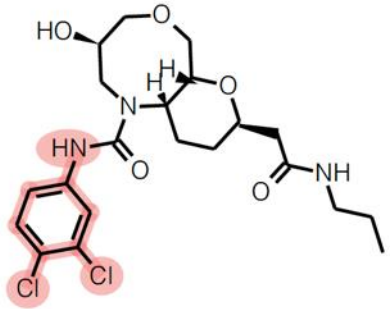
Why Explaining?

Verify & debug
(Lapuschkin et al. Nat Comm, 2019)



New insights
(Wong et al. Nature, 2023)





"BLUE XAI"
(Biecek & Samek, ICML, 2024)

Human-values oriented
- Responsible models
- Legal issues
- Trust in predictions
- Ethical issues

EU AI Act

Proposal for a
Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts

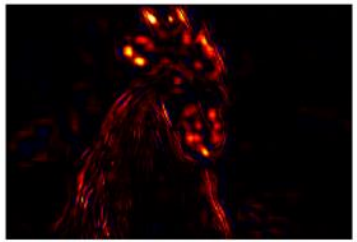
2021/0106 (COD)

Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence

What to explain?

What to Explain?

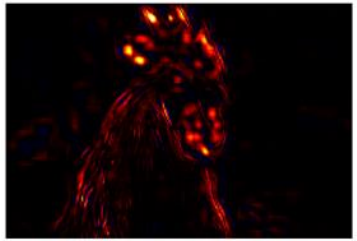
What: Individual prediction
How: Input attribution



Visualize how much each pixel contributes to the prediction "rooster".

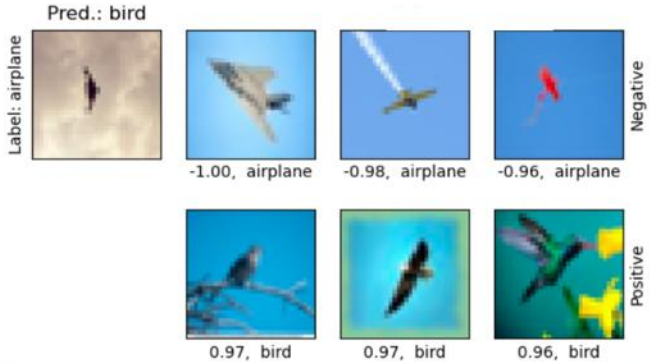
What to Explain?

What: Individual prediction
How: Input attribution



Visualize how much each pixel contributes to the prediction "rooster".

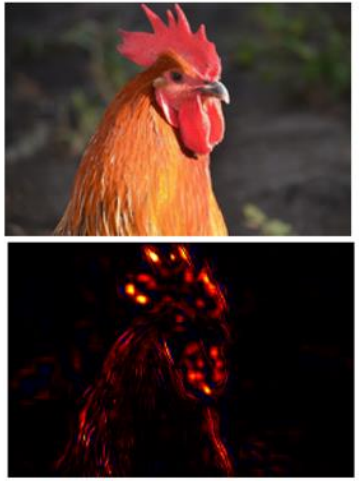
What: Individual prediction
How: By example



Find training samples, which explain the wrong model prediction.

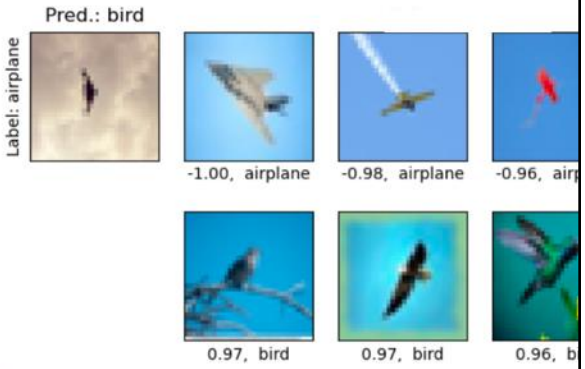
What to Explain?

What: Individual prediction
How: Input attribution



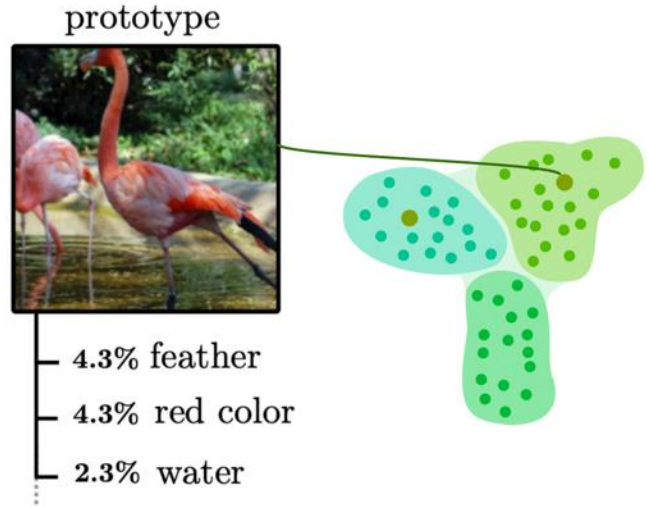
Visualize how much each pixel contributes to the prediction "rooster".

What: Individual prediction
How: By example



Find training samples, which expla
the wrong model prediction.

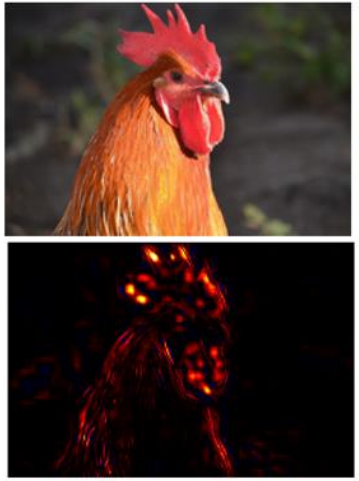
What: Prototypical behaviour
How: Concepts



List the concepts, which are present in a prototypical "flamingo" image.

What to Explain?

What: Individual prediction
How: Input attribution

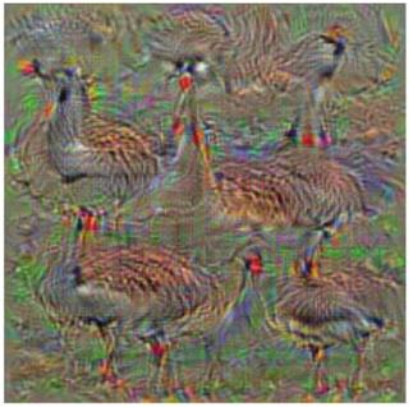


Visualize how much each pixel contributes to the prediction "ro

What: Individual prediction
How: By example

Pred.: bird

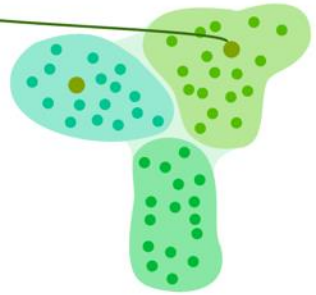
What: Neuron representation
How: Synthetic input



Synthetize an input which maximally activates a particular neuron.

What: Prototypical behaviour
How: Concepts

prototype



lor

ts, which are present
l "flamingo" image.

What to Explain?

What: Individual prediction
How: Input attribution

Visualize how much each pixel contributes to the prediction "r

What: Individual prediction
How: By example

Pred.: bird

What: Prototypical behaviour
How: Concepts

prototype

What: Neuron representation

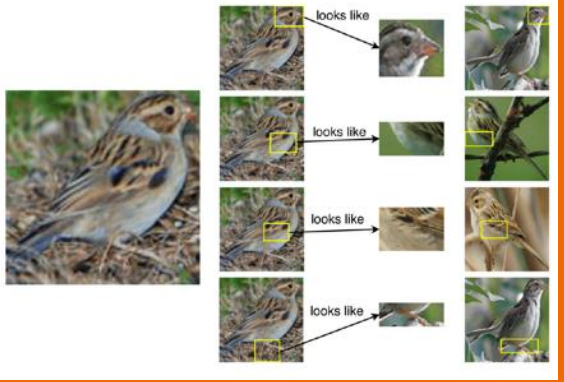
Synthetize an input which maximally activates a particular neuron.

"flamingo" image.

Many more

- Counterfactual explanations
- Prototypical explanations ("this looks like that")

....



1st Take Home Message

"Explanation is an ambigious concept"

1st Take Home Message

What
(individual prediction, internal representation, processing steps, general behaviour, data importance)

How
(attributions, concepts, interactions, rules, prototypes, examples)

"Explanation is an ambigious concept"

Relative to what
(dog vs. cat, dog vs. car)

Under what assumptions
(linear approximation, sparsity, zero noise, feature independence)

Don't worry about the ambiguity

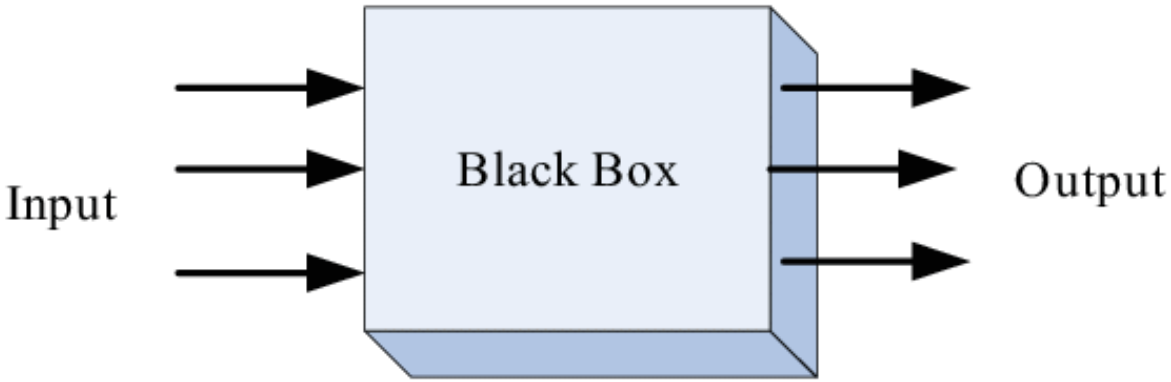
What
(individual prediction, in
steps, general behaviour

How
s, interactions, rules,
)

Relative to what
(dog vs. cat, dog vs. car)

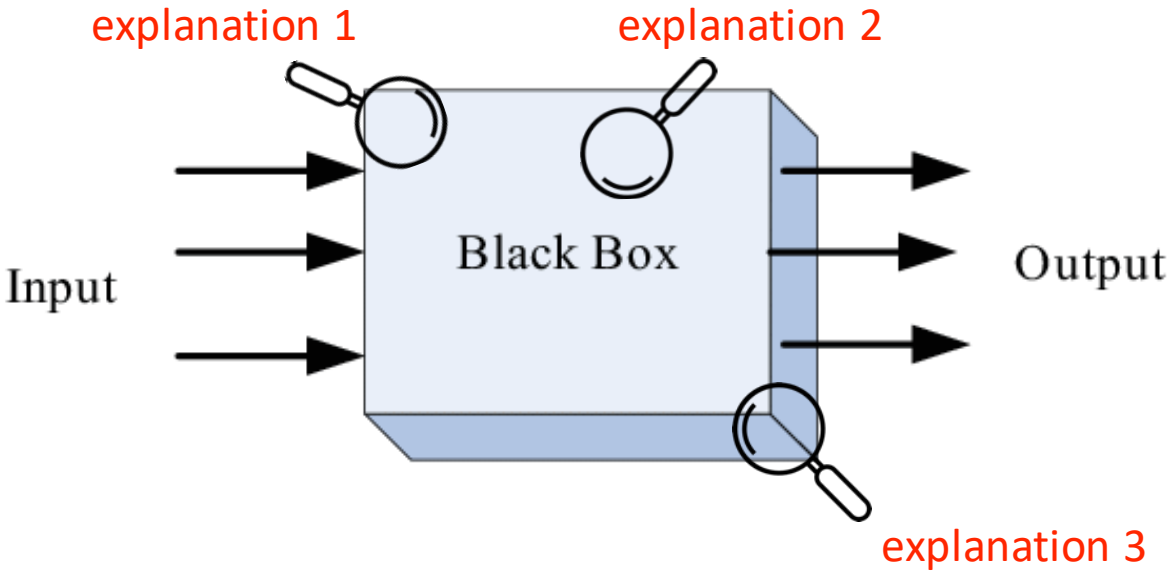
t assumptions
nation, sparsity, zero noise,
ndence)

Epistemological perspective



Full explanation of model behavior is the model itself.

Epistemological perspective



Full explanation of model behavior is the model itself.

Explanations capture only aspects of model behaviour (still useful).

Utilitarian perspective

Explanations are good if they provide some additional (measurable) advantage.

Full explanation of model behavior is the model itself.

Explanations capture only aspects of model behaviour (still useful).

Explaining by Attribution

Explanation Methods

Perturbation-Based

Occlusion-Based (Zeiler & Fergus 14)

Meaningful Perturbations (Fong & Vedaldi 17)

…

Gradient-Based

Sensitivity Analysis (Simonyan et al. 14)

(Simple) Taylor Expansions

Gradient x Input (Shrikumar et al. 16)

…

Surrogate- / Sampling-Based

LIME (Ribeiro et al. 16)

SmoothGrad (Smilkov et al. 16)

…

Propagation-Based

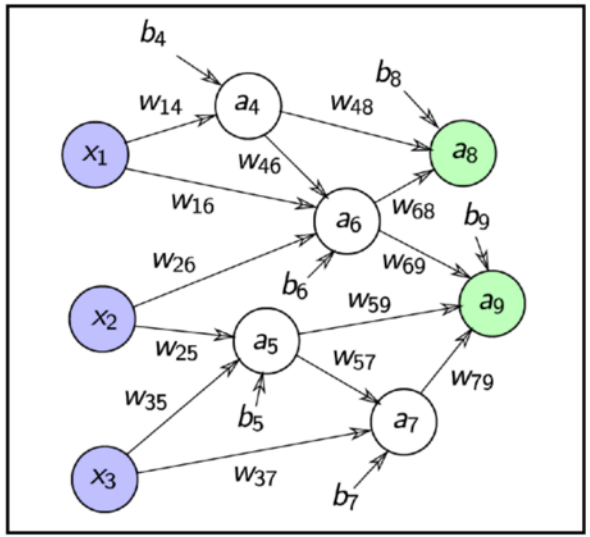
LRP (Bach et al. 15)

Deep Taylor Decomposition (Montavon et al. 17)

Excitation Backprop (Zhang et al. 16)

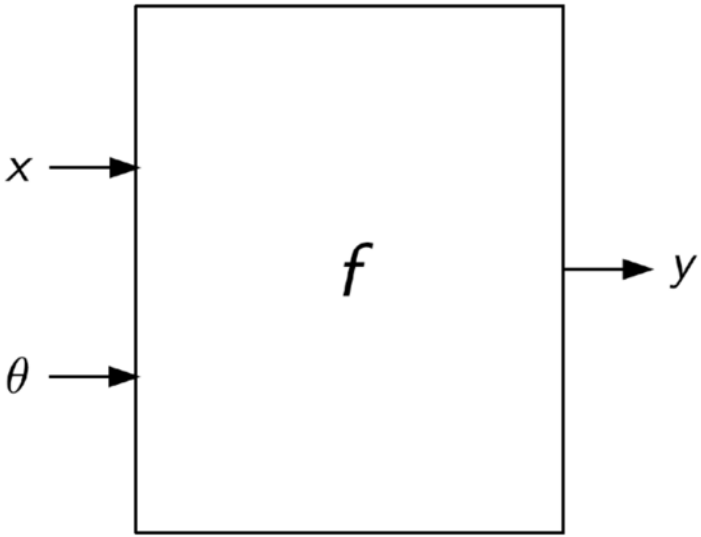
Two Views on the Model

graph view



$x = [x_1, x_2, x_3] \quad y = [a_8, a_9]$

function view



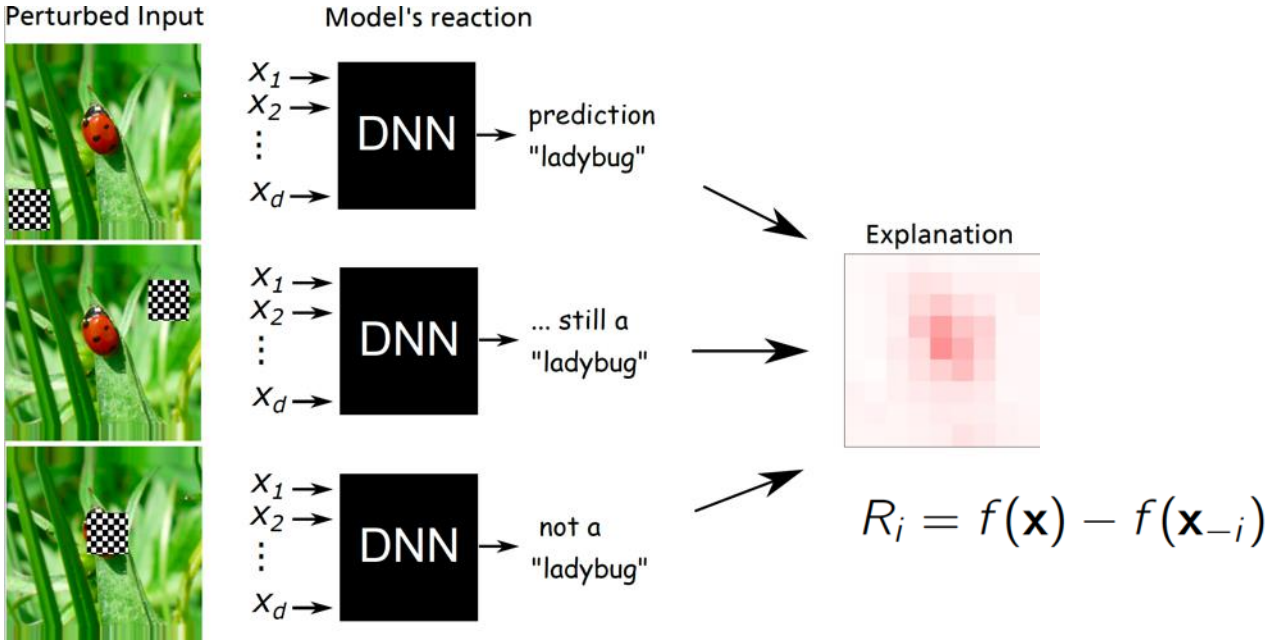
$\theta = (w_{ij})_{ij}, (b_j)_j$

model aware XAI

model agnostic XAI

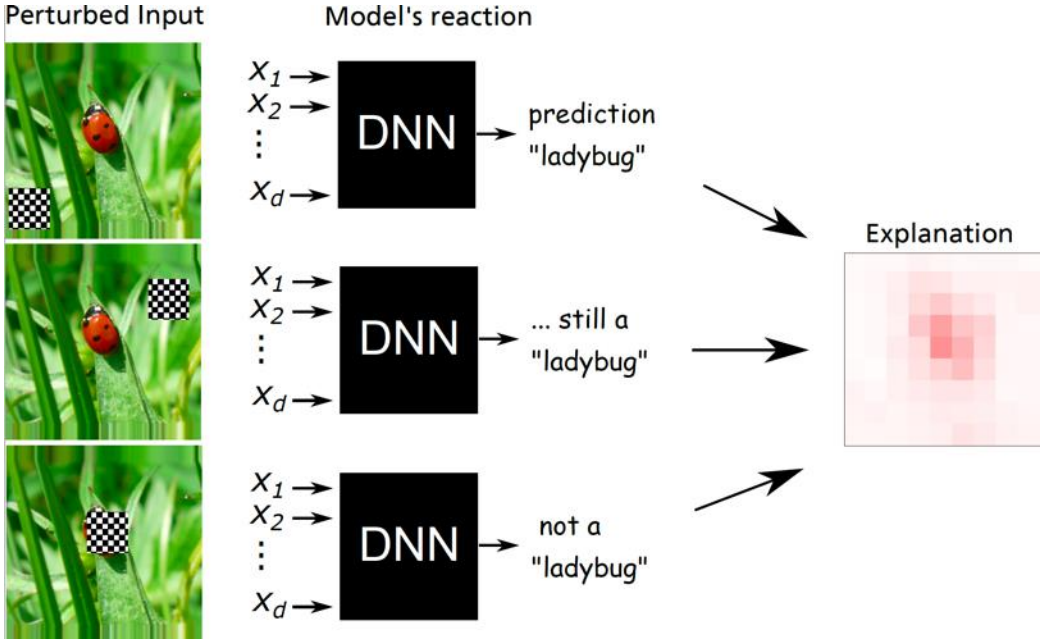
Perturbation Framework

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



Perturbation Framework

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



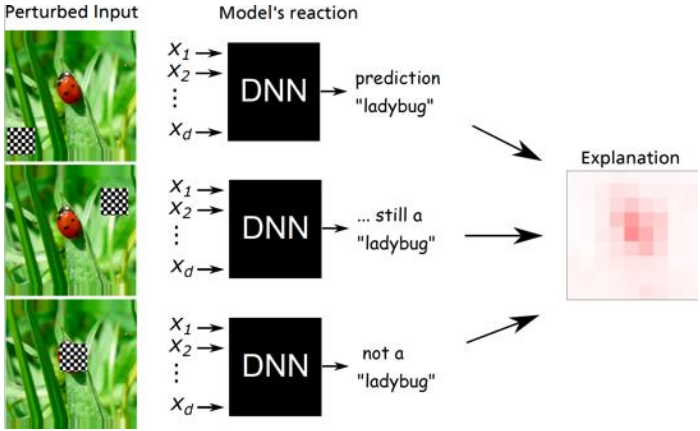
Advantages

- Model agnostic
- Easy to implement
- Flexible
- Can use with continuous or categorical features

Example: Occlusion

Perturbation Framework:

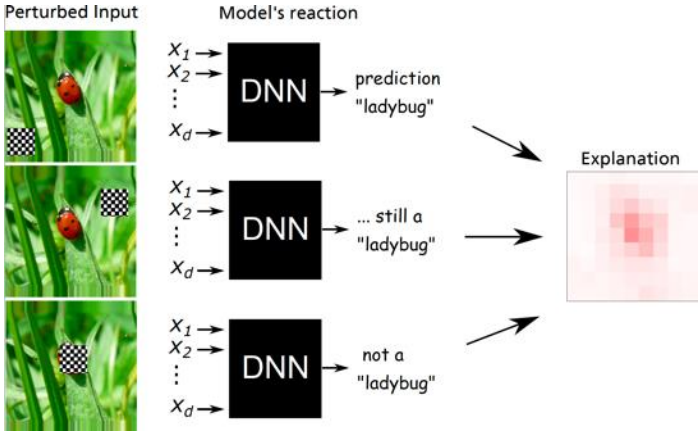
1. How to perturb? → Replace patch with black-white pattern
2. How to measure change ? → Change in prediction probability $f(\mathbf{x}) - f(\mathbf{x}_{-i})$
3. Which measurements to aggregate ? → All patches individually



Example: Occlusion

Perturbation Framework:

1. How to perturb? → Replace patch with black-white pattern
2. How to measure change ? → Change in prediction probability $f(\mathbf{x}) - f(\mathbf{x}_{-i})$
3. Which measurements to aggregate ? → All patches individually



Disadvantages

- slow
- assumes locality (no interactions)
- perturbation may introduce artefacts

Example: Meaningful Perturbations

Iden: Learn (minimal) blurring mask, which changes the prediction (lowers confidence) when applied to the image.

original image,
high confidence for correct classs

blurred image,
very low confidence for correct classs

Example: Meaningful Perturbations

Let $x \in \mathbb{R}^{w \times h}$ be the image

Let $m \in [0,1]^{w \times h}$ be a mask

Let $\Phi(x, m) \in \mathbb{R}^{w \times h}$ be the masked image

blurring operation

$f_y(\Phi(x, \mathbf{1})) \approx 1$

Goal: find m

$f_y(\Phi(x, m)) \approx 0$

Example: Meaningful Perturbations

Mask should be

1) minimal

2) smooth

3) not a result of adversarial perturbation

Loss function

$$\mathcal{L}(m) = \mathbb{E}_{\tau}\left[f_y\left(\Phi(x(\cdot - \tau), m)\right)\right] + \lambda_1 \|1 - m\|_1 + \lambda_2 \|\nabla m\|_{\beta}^{\beta}$$

Optimization can be performed using SGD.

Example: Meaningful Perturbations

Perturbation Framework:

1. How to perturb? → Blur
2. How to measure change ? → Change in prediction probability
3. Which measurements to aggregate ? → Minimal feature subset (smooth)

$f_y(\Phi(x, \mathbf{1})) \approx 1$

Goal: find m

$f_y(\Phi(x, m)) \approx 0$

Explanation Methods

Perturbation-Based

Occlusion-Based (Zeiler & Fergus 14)

Meaningful Perturbations (Fong & Vedaldi 17)

…

Gradient-Based

Sensitivity Analysis (Simonyan et al. 14)

(Simple) Taylor Expansions

Gradient x Input (Shrikumar et al. 16)

…

Surrogate- / Sampling-Based

LIME (Ribeiro et al. 16)

SmoothGrad (Smilkov et al. 16)

…

Propagation-Based

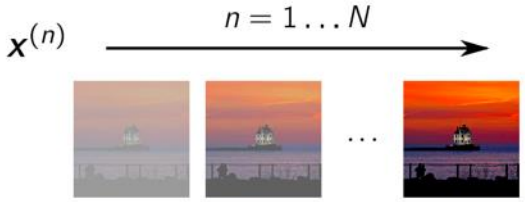
LRP (Bach et al. 15)

Deep Taylor Decomposition (Montavon et al. 17)

Excitation Backprop (Zhang et al. 16)

From Perturbations to Gradients

Consider a sequence of inputs $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ interpolating between $\mathbf{x}^{(0)} = \mathbf{0}$ and $\mathbf{x}^{(N)} = \mathbf{x}$.



From Perturbations to Gradients

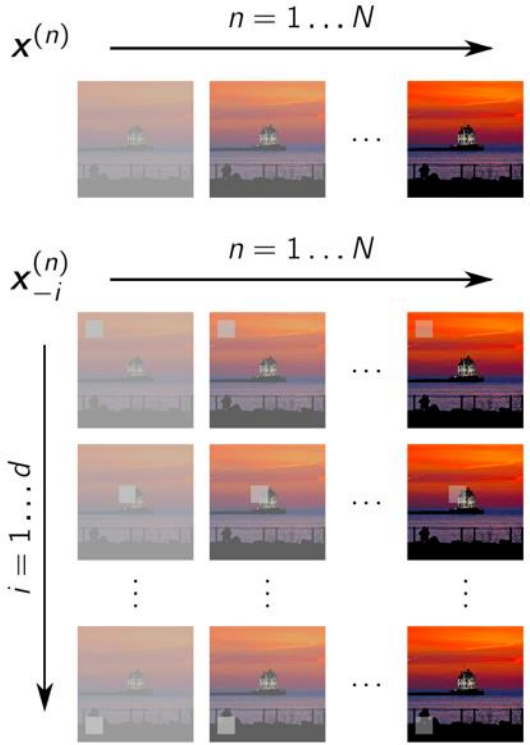
Consider a sequence of inputs $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ interpolating between $\mathbf{x}^{(0)} = \mathbf{0}$ and $\mathbf{x}^{(N)} = \mathbf{x}$.

Perform for each n the perturbation analysis

$$R_i^{(n)} = f(\mathbf{x}^{(n)}) - f(\mathbf{x}_{-i}^{(n)})$$

where

$$\mathbf{x}_{-i}^{(n)} = (x_1^{(n)}, \ldots, x_{i-1}^{(n)}, x_i^{(n-1)}, x_{i+1}^{(n)}, \ldots, x_d^{(n)})$$



From Perturbations to Gradients

Consider a sequence of inputs $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ interpolating between $\mathbf{x}^{(0)} = \mathbf{0}$ and $\mathbf{x}^{(N)} = \mathbf{x}$.

Perform for each n the perturbation analysis

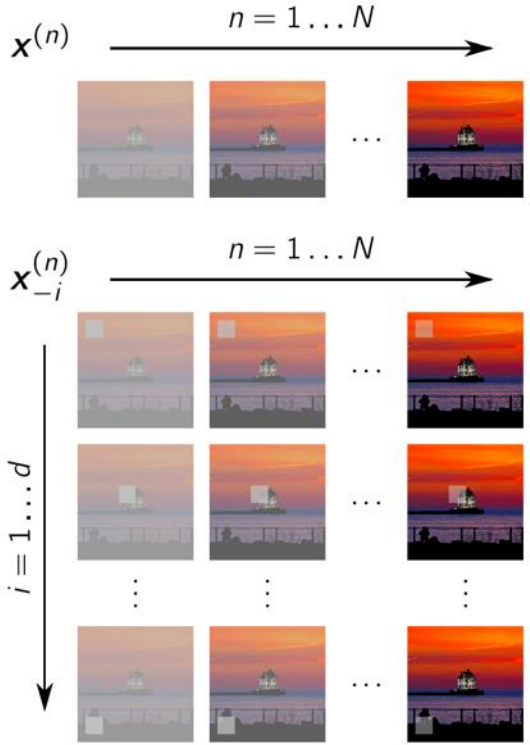
$$R_i^{(n)} = f(\mathbf{x}^{(n)}) - f(\mathbf{x}_{-i}^{(n)})$$

where

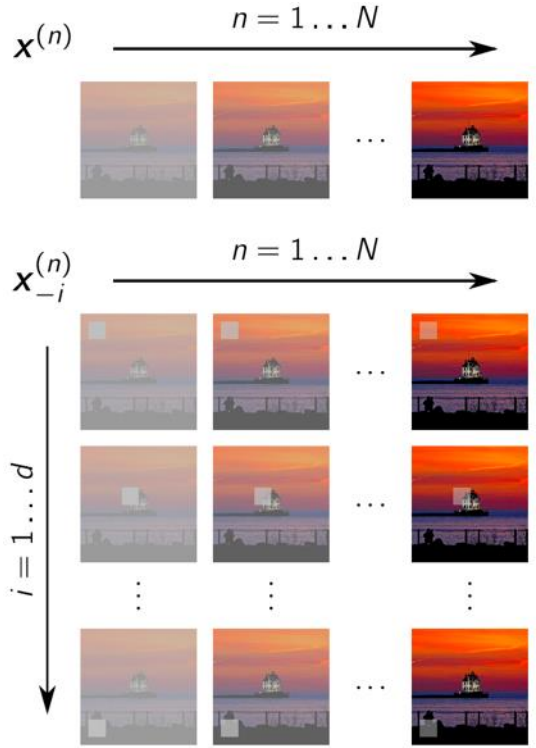
$$\mathbf{x}_{-i}^{(n)} = (x_1^{(n)}, \ldots, x_{i-1}^{(n)}, x_i^{(n-1)}, x_{i+1}^{(n)}, \ldots, x_d^{(n)})$$

Sum them up:

$$R_i = \sum_{n=1}^{N} R_i^{(n)}$$



Integrated Gradients

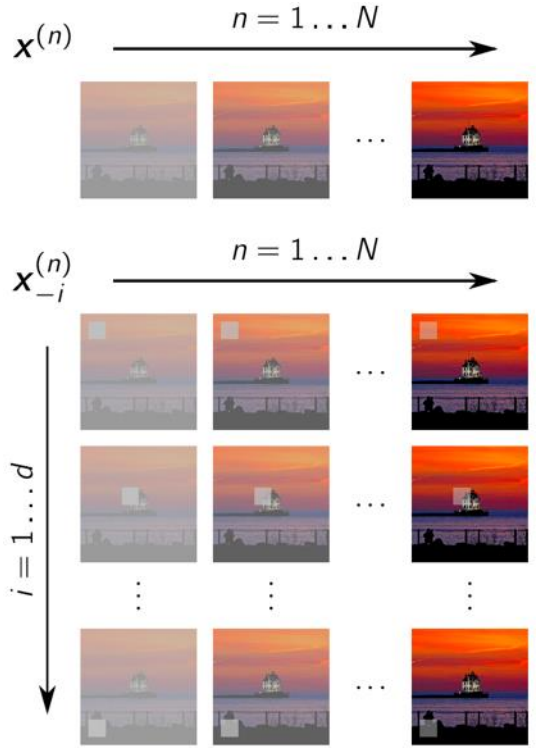


- **Observation:** When the interpolation steps are small enough and when f is differentiable,

$$R_i^{(n)} \approx [\nabla f(\mathbf{x}^{(n)})]_i \cdot (\mathbf{x}_i^{(n)} - \mathbf{x}_i^{(n-1)})$$

where the function's gradient appears.

Integrated Gradients



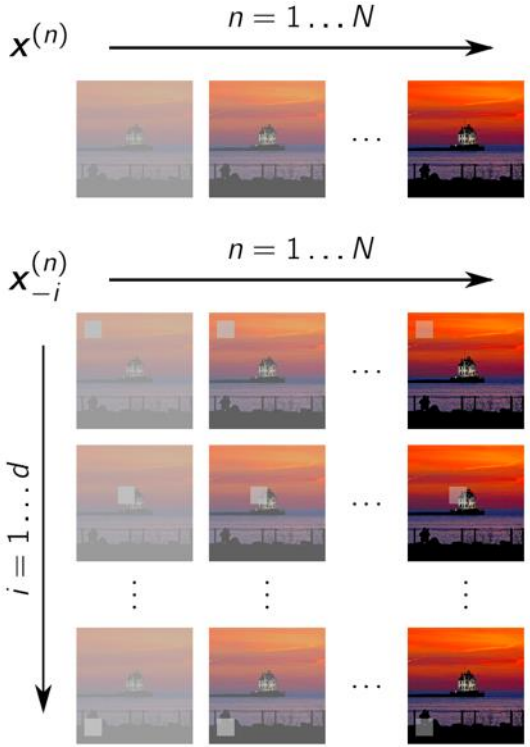
- **Observation:** When the interpolation steps are small enough and when f is differentiable,

$$R_i^{(n)} \approx [\nabla f(\mathbf{x}^{(n)})]_i \cdot (\mathbf{x}_i^{(n)} - \mathbf{x}_i^{(n-1)})$$

where the function's gradient appears.

- At each step, the perturbation for *all* dimensions can be computed using only one gradient evaluation.

Integrated Gradients



- **Observation:** When the interpolation steps are small enough and when f is differentiable,

$$R_i^{(n)} \approx [\nabla f(\mathbf{x}^{(n)})]_i \cdot (\mathbf{x}_i^{(n)} - \mathbf{x}_i^{(n-1)})$$

  where the function's gradient appears.

- At each step, the perturbation for *all* dimensions can be computed using only one gradient evaluation.

- This is the integrated gradients method (in discretized form) (Sundararajan et al. 2017)

Integrated Gradients

- Integrated Gradients (IG) (Sundararajan et al. 2017):

$$R_i = \sum_{n=1}^{N} [\nabla f(\mathbf{x}^{(n)})]_i \cdot (x_i^{(n)} - x_i^{(n-1)})$$

- Gradient $\times$ Input (GI) (Shrikumar et al. 2016)

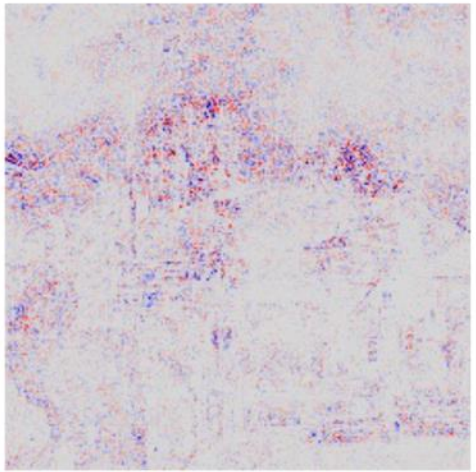
$$R_i = [\nabla f(\mathbf{x})]_i \cdot x_i$$

i.e. an input feature i contributes if it is present in the data ($x_i > 0$) and if the model reacts to it ($[\nabla f(\mathbf{x})]_i > 0$).

Problem: Gradients are 'Shattered'

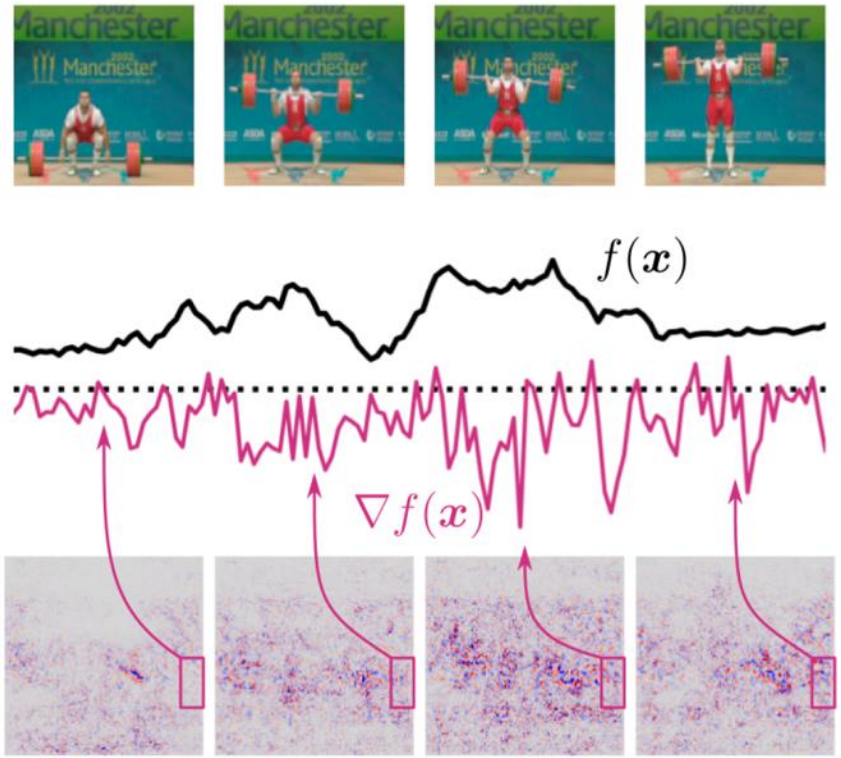
Gradient × Input (GI) **(Shrikumar et al. 2016)**

$$R_i = [\nabla f(\mathbf{x})]_i \cdot x_i$$



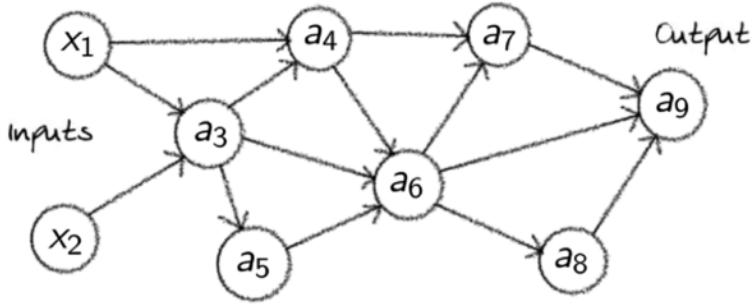
Gradient Explanation: Very noisy, not really informative.

Problem: Gradients are 'Shattered'

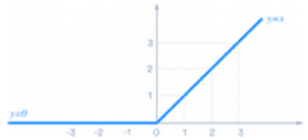


- We look at the DNN output (and its gradient) along some trajectory in the input space, e.g. an athlete lifting a barebell.
- The function is relatively stable, but the gradient strongly oscillates and appears noisy (cf. Balduzzis et al. 2017)

Backward Propagation



$$z_6 = a_3 \cdot w_{36} + a_4 \cdot w_{46} + a_5 \cdot w_{56} + b_6$$



$$g(z_6) = \max(0, z_6)$$

$$g'(z_6) = 1_{a_6 > 0}$$

Gradient only propagate when activated.

Recap from previous slide (derivative of neuron 6):

$$\delta_6 = [\delta_9 \cdot w_{69} + \delta_8 \cdot w_{68} + \delta_7 \cdot w_{67}] \cdot g'(z_6)$$

Error derivatives for parameters of neuron 6:

$$\partial \mathcal{E} / \partial b_6 = \delta_6 \cdot \partial z_6 / \partial b_6 = \delta_6 \cdot 1$$

$$\partial \mathcal{E} / \partial w_{56} = \delta_6 \cdot \partial z_6 / \partial w_{56} = \delta_6 \cdot a_5$$

$$\partial \mathcal{E} / \partial w_{46} = \delta_6 \cdot \partial z_6 / \partial w_{46} = \delta_6 \cdot a_4$$

$$\partial \mathcal{E} / \partial w_{36} = \delta_6 \cdot \partial z_6 / \partial w_{36} = \delta_6 \cdot a_3$$

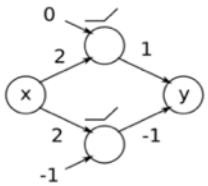
Problem: Gradients are 'Shattered'

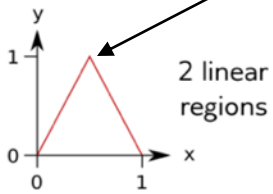
Input gradient (on which sensitivity analysis is based), becomes increasingly highly varying and unreliable with neural network depth.

Example in [0,1]:

$$g(x) = 2 \cdot \text{ReLU}(x) - 4 \cdot \text{ReLU}(x - 0.5)$$

depth 1





gradient flips here

2 linear regions

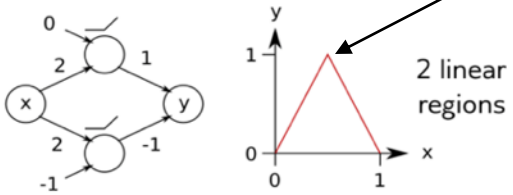
Problem: Gradients are 'Shattered'

Input gradient (on which sensitivity analysis is based), becomes increasingly highly varying and unreliable with neural network depth.

Example in [0,1]:

$$g(x) = 2 \cdot \text{ReLU}(x) - 4 \cdot \text{ReLU}(x - 0.5)$$

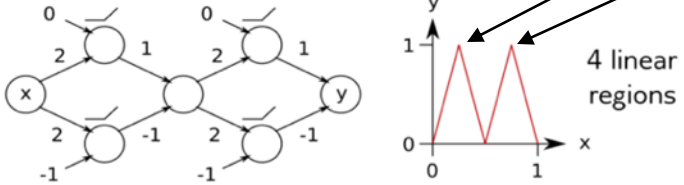
depth 1



gradient flips here

2 linear regions

depth 2



gradient flips here

4 linear regions

Problem: Gradients are 'Shattered'

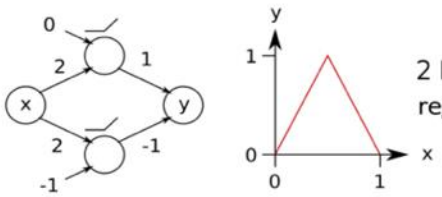
Input gradient (on which sensitivity analysis is based), becomes increasingly highly varying and unreliable with neural network depth.

Example in [0,1]:

$$g(x) = 2 \cdot \mathrm{ReLU}(x) - 4 \cdot \mathrm{ReLU}(x - 0.5)$$

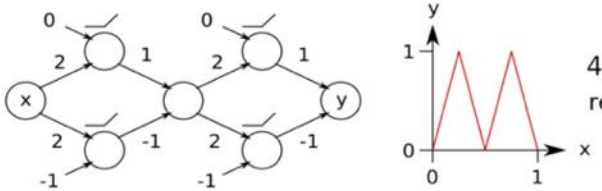
function	# linear pieces
$g(x)$	2
$g \circ g(x)$	4
$g \circ g \circ g(x)$	8
$g \circ g \circ g \circ g(x)$	16

depth 1



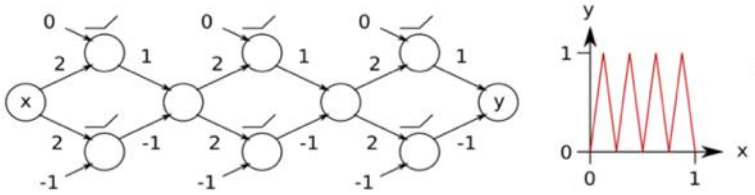
2 linear regions

depth 2



4 linear regions

depth 3



8 linear regions

number of linear regions grows exponentially with depth

2nd Take Home Message

"Gradient-Based XAI methods are affected by shattering effect, specially in deep models"

We will also see more results for transformer-based models later.

Problem: High Noise Due to 'Gradient Shattering'

Stabilize by:

- Integrating gradients over path (IntGrad)
- Averaging in neigborhood (SmoothGrad)

...

- Explaing model in terms of last conv layer instead of input layer (GradCAM)

....

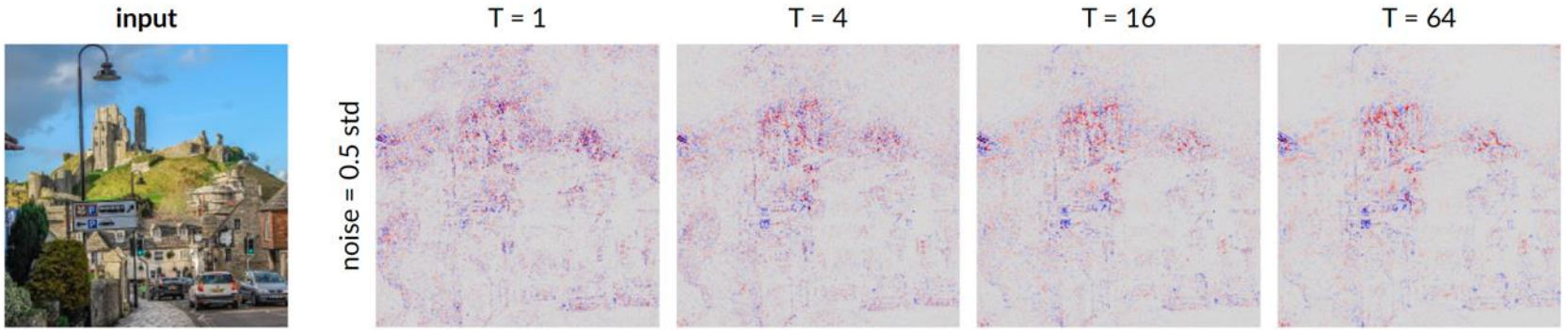
- Propagate relevance instead of gradients (LRP)

SmoothGrad: "Removing Noise by Adding Noise"

Idea: Perform the gradient-based analysis with multiple random perturbations $\epsilon_1, \ldots, \epsilon_T$ of the input, and average the explanations. (Smilkov et al. 2017)

Example: Smooth Gradient × Input

$$R_i = \frac{1}{T} \sum_{t=1}^{T} [\nabla f(\mathbf{x} + \epsilon_t)]_i [\mathbf{x} + \epsilon_t]_i$$



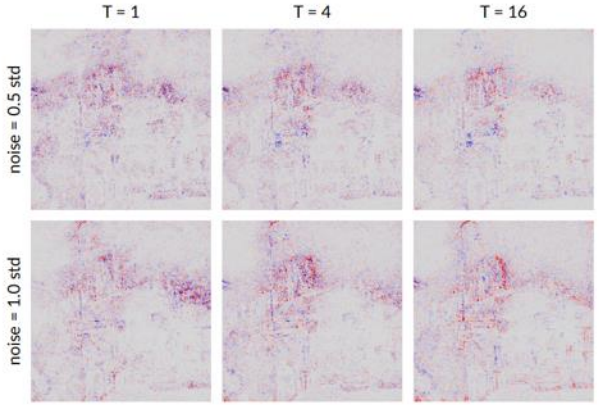
SmoothGrad

Advantages

- Reduces explanation noise.
- Simple to implement (just call the same code multiple time)
- Widely applicable (can be applied on top of any explanation technique).

Limitations

- Computation cost increases by a factor T while explanation noise is in the best case only reduced by a factor $\sqrt{T}$.
- Adding noise to the input implies that we explain a slightly different quantity than the input (this may add a bias to the explanation).



Explanation Methods

Perturbation-Based

Occlusion-Based (Zeiler & Fergus 14)

Meaningful Perturbations (Fong & Vedaldi 17)

…

Gradient-Based

Sensitivity Analysis (Simonyan et al. 14)

(Simple) Taylor Expansions

Gradient x Input (Shrikumar et al. 16)

…

Surrogate- / Sampling-Based

LIME (Ribeiro et al. 16)

SmoothGrad (Smilkov et al. 16)

…

Propagation-Based

LRP (Bach et al. 15)

Deep Taylor Decomposition (Montavon et al. 17)

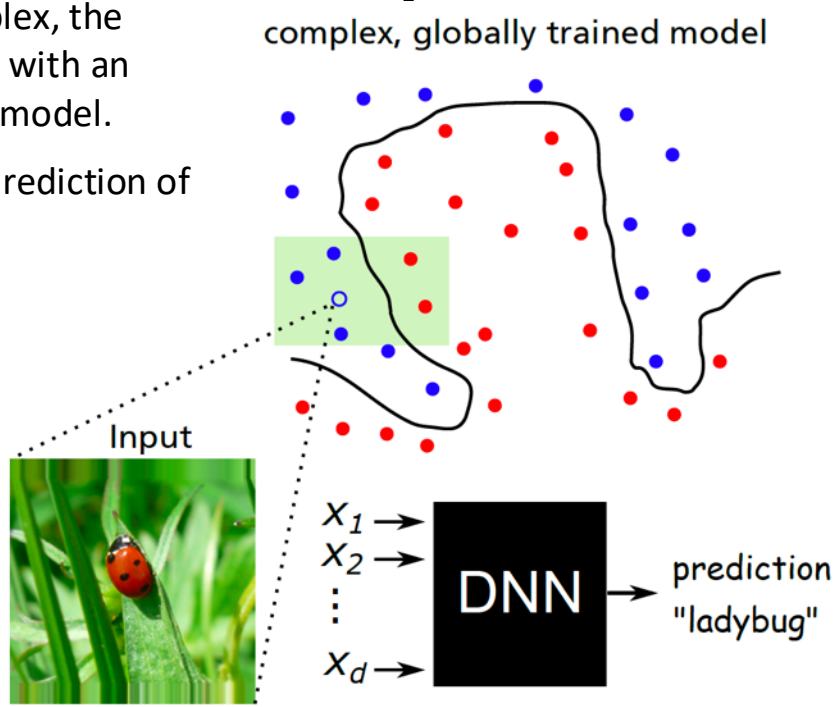
Excitation Backprop (Zhang et al. 16)

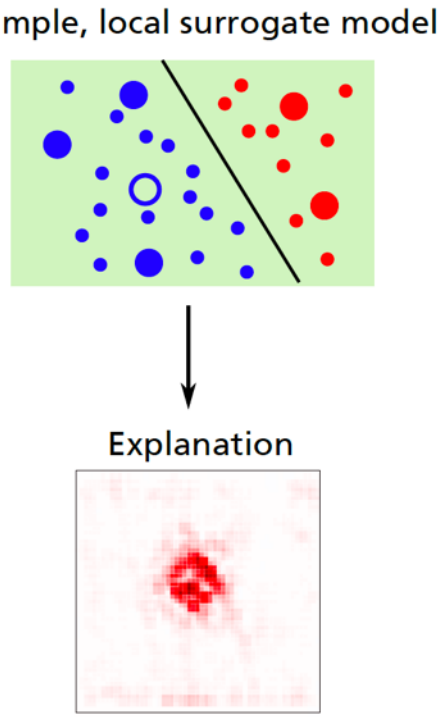
Local Interpretable Model-Agnostic (LIME)

Explanations (LIME

Idea: If the overall decision boundary is too complex, the approximate it locally with an interpretable (linear) model.

--> Then explain the prediction of the surrogate.





Local Interpretable Model-Agnostic (LIME)

Explanations (LIME

LIME Pseudocode

Explanations can be calculated with a following instructions.

Let $x' = h(x)$ be a version of x in the interpretable data space

```
for i in 1...N {
    z'[i] = sample_around(x')
    y'[i] = f(z'[i])
    w'[i] = similarity(x', z'[i])
}
return K-LASSO(y', x', w')
```

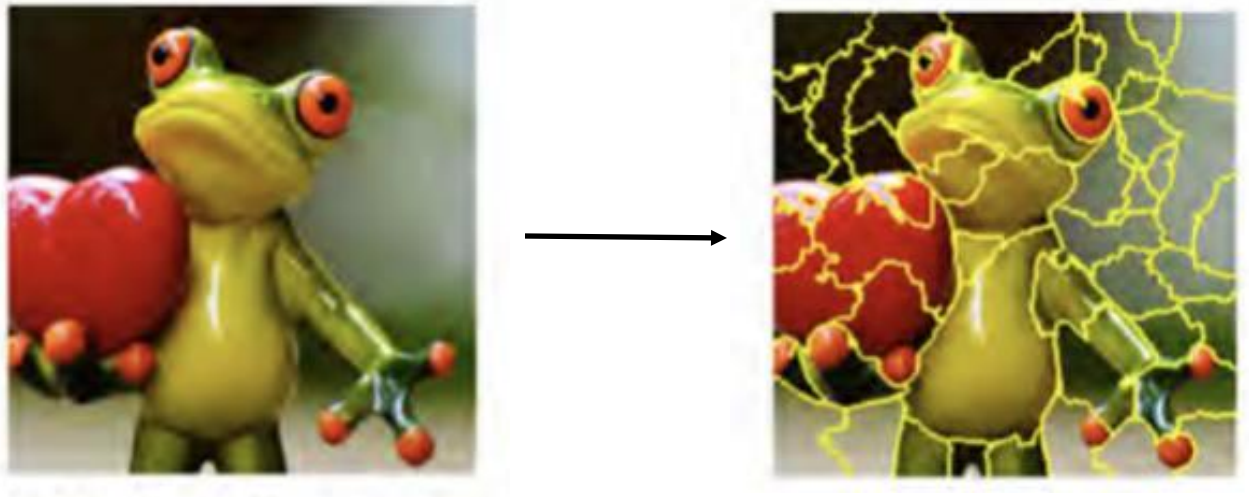
Local Interpretable Model-Agnostic (LIME)

Explanations (LIME

LIME Pseudocode

Explanations can be calculated with a following instructions.

Let $x' = h(x)$ be a version of x in the interpretable data space



Local Interpretable Model-Agnostic (LIME)

Explanations (LIME

LIME Pseudocode

Explanations can be calculated with a following instructions.

Let $x' = h(x)$ be a version of x in the interpretable data space

for i in 1…N {

z'[i] = `sample_around`(x')

selecting randomly coordinates that will be flipped to zero

Local Interpretable Model-Agnostic (LIME)

Explanations (LIME

LIME Pseudocode

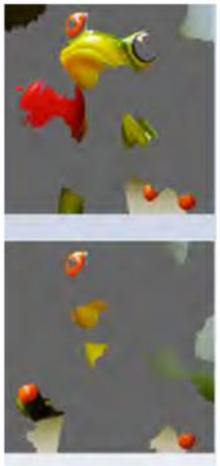
Explanations can be calculated with a following instructions.

Let $x' = h(x)$ be a version of x in the interpretable data space

for i in 1...N {

z'[i] = `sample_around`(x')

y'[i] = $f(z'[i])$



P(frog)

0.8

0.01

Local Interpretable Model-Agnostic (LIME)

Explanations (LIME

LIME Pseudocode

Explanations can be calculated with a following instructions.

Let $x' = h(x)$ be a version of x in the interpretable data space

for i in 1…N {

z'[i] = `sample_around`(x')

y'[i] = f(z'[i])

w'[i] = `similarity`(x', z'[i])

Local Interpretable Model-Agnostic (LIME)

Explanations (LIME

LIME Pseudocode

Explanations can be calculated with a following instructions.

Let $x' = h(x)$ be a version of x in the interpretable data space

```
for i in 1...N {
    z'[i] = sample_around(x')
    y'[i] = f(z'[i])
    w'[i] = similarity(x', z'[i])
}
return K-LASSO(y', x', w')
```

Train a weighted, interpretable (e.g. linear) model on the dataset with the variations.

Local Interpretable Model-Agnostic (LIME)

Explain the prediction by interpreting the local model.

$$y = w_1 x_1 + \cdots + w_m x_m + \cdots + w_M x_M$$

$$x_m = \begin{cases} 0 & \text{Segment m is deleted.} \\ 1 & \text{Segment m exists.} \end{cases}$$

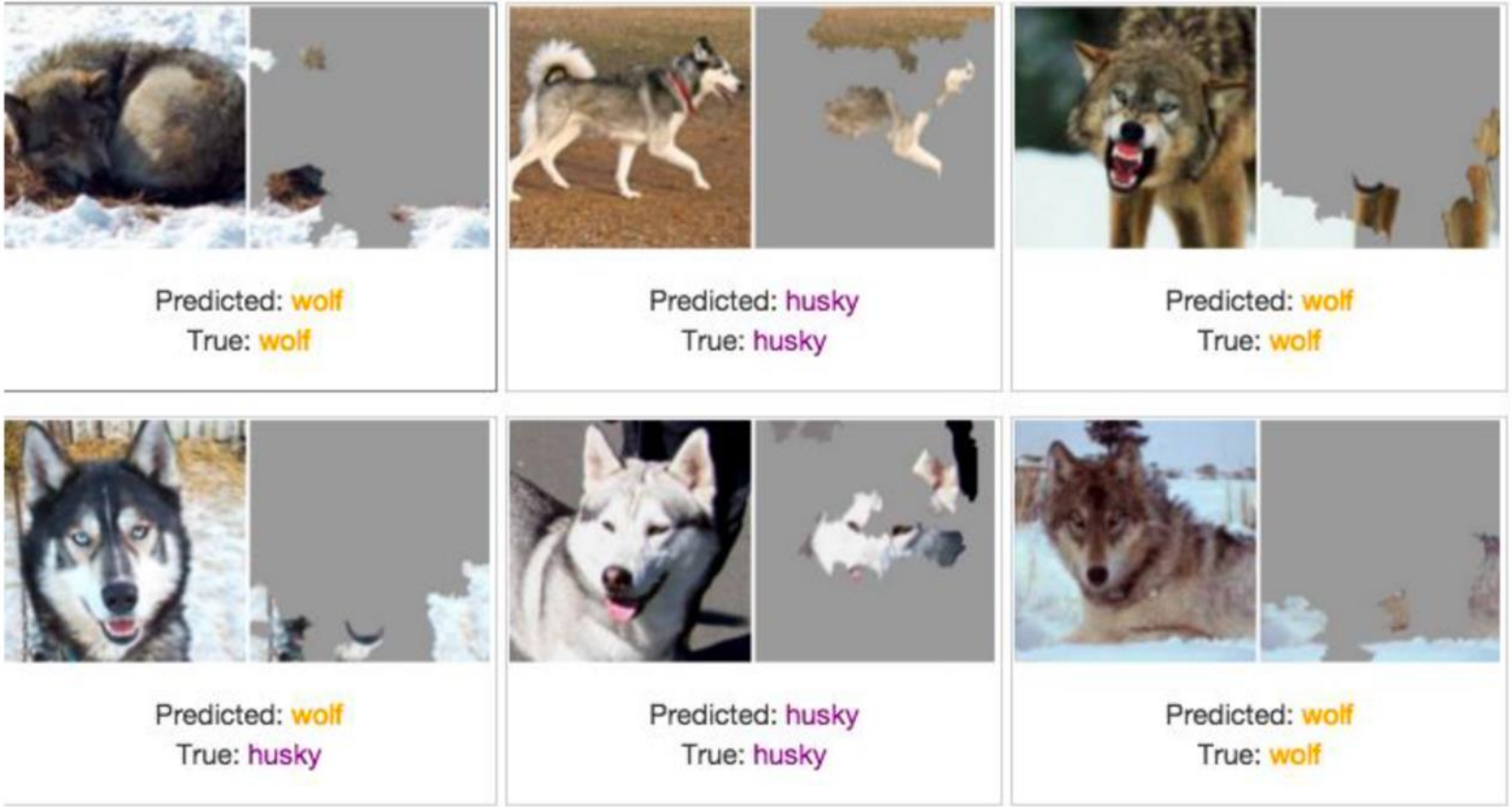
M is the number of segments.

If $w_m \approx 0$ ➡ segment m is not related to “frog”

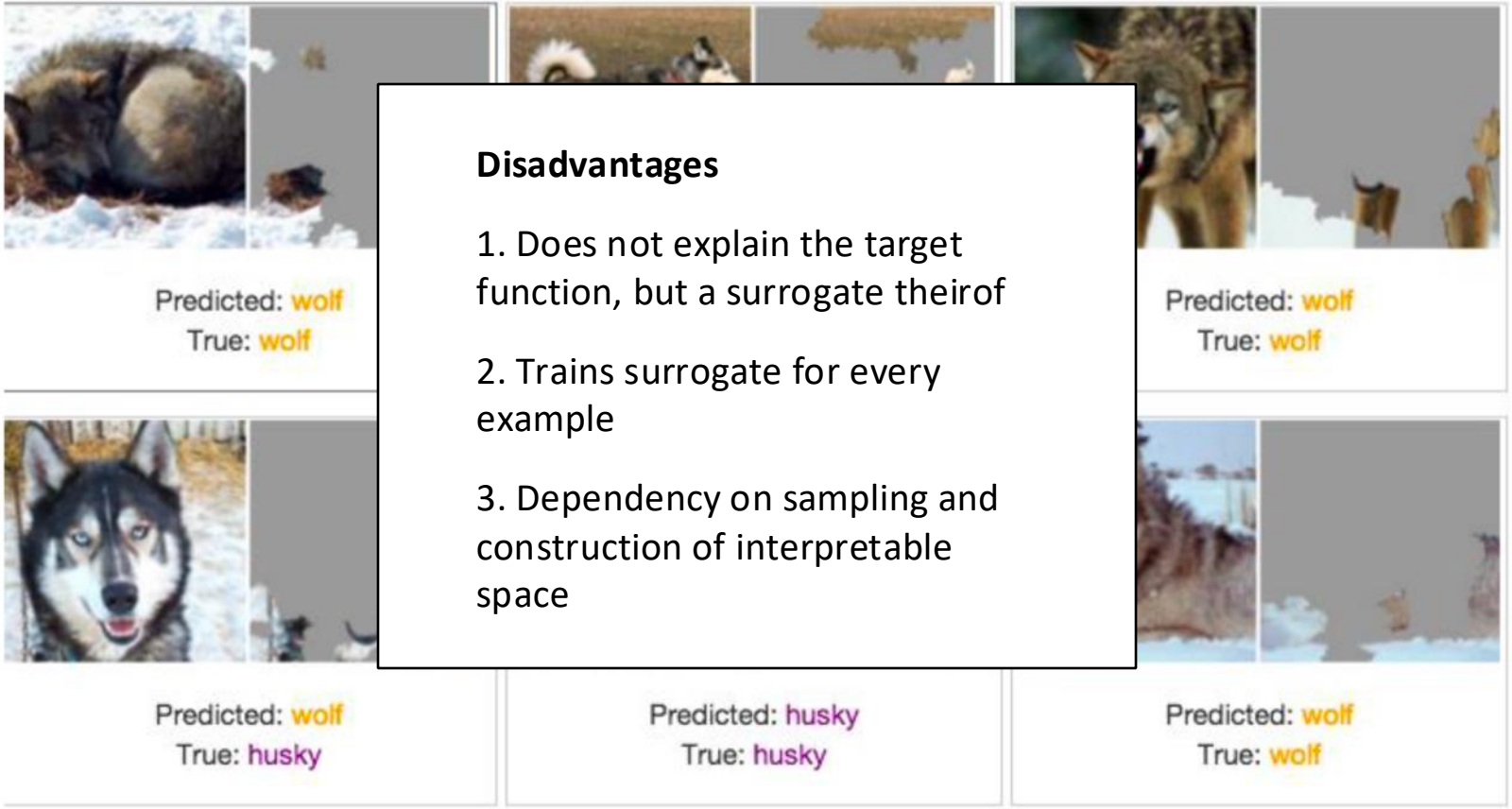
If w_m is positive ➡ segment m indicates the image is “frog”

If w_m is negative ➡ segment m indicates the image is not “frog”

Local Interpretable Model-Agnostic (LIME)



Local Interpretable Model-Agnostic (LIME)



Disadvantages

1. Does not explain the target function, but a surrogate theirof

2. Trains surrogate for every example

3. Dependency on sampling and construction of interpretable space

Explanation Methods

Perturbation-Based

Occlusion-Based (Zeiler & Fergus 14)

Meaningful Perturbations (Fong & Vedaldi 17)

…

Gradient-Based

Sensitivity Analysis (Simonyan et al. 14)

(Simple) Taylor Expansions

Gradient x Input (Shrikumar et al. 16)

…

Surrogate- / Sampling-Based

LIME (Ribeiro et al. 16)

SmoothGrad (Smilkov et al. 16)

…

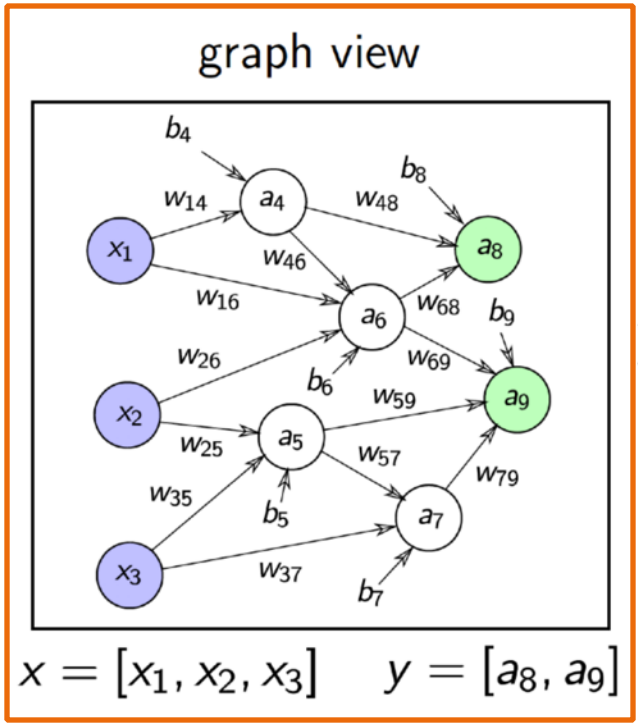
Propagation-Based

LRP (Bach et al. 15)

Deep Taylor Decomposition (Montavon et al. 17)

Excitation Backprop (Zhang et al. 16)

Model Aware Explanation Methods

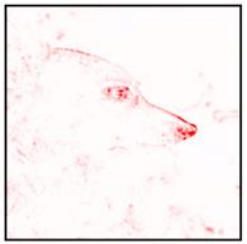


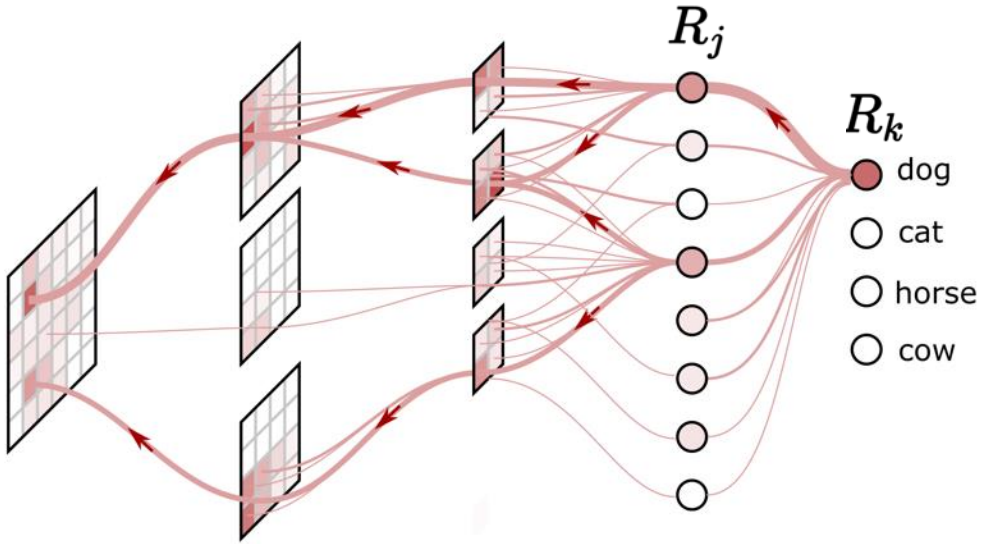
Ideas:

- Use the structure of the neural network to robustly compute relevance scores for the input features.
- Propagate the output of the network backwards by means of propagation rules.

Price to pay: Not model agnostic anymore

Layer-wise Relevance Propagation





(1) decompose

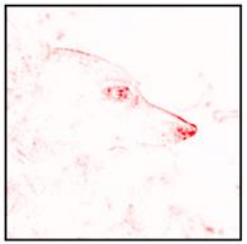
$$R_{j \leftarrow k} = \frac{z_{jk}}{z_k} R_k$$

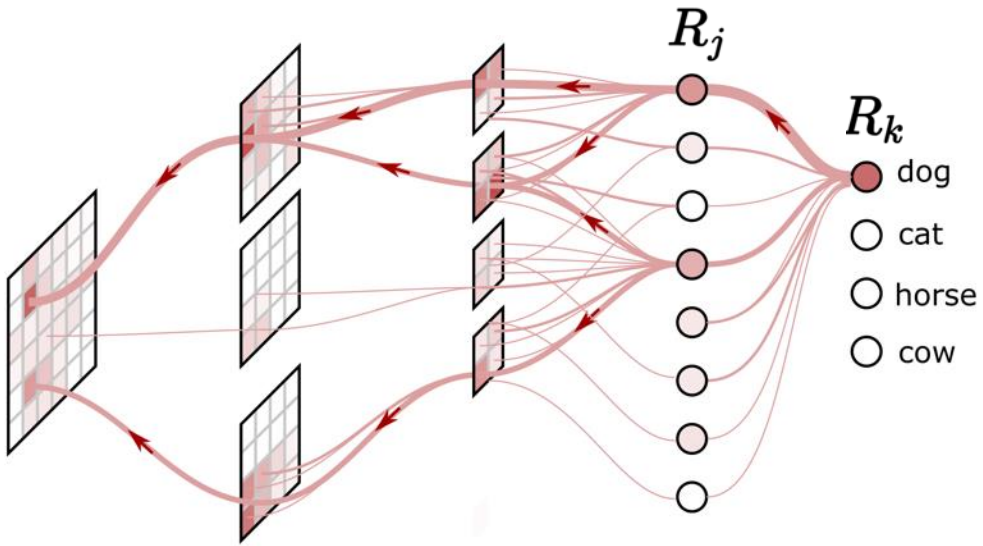
(2) aggregate

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

zjk measures how much has j contributed to activation of k

Layer-wise Relevance Propagation





Layer-wise relevance conservation

$$\sum_i R_i = \ldots = \sum_i R_i^{(l)} = \sum_j R_j^{(l+1)} = \ldots = f(x)$$

Advantages

- efficient
- relevance values for all elements of NN
- applicable to non-differentiable layers (no gradient shattering)

Which redistribution rule is the right one (i.e. how to best measure zjk)?

3rd Take Home Message

"There is no one best LRP rule"

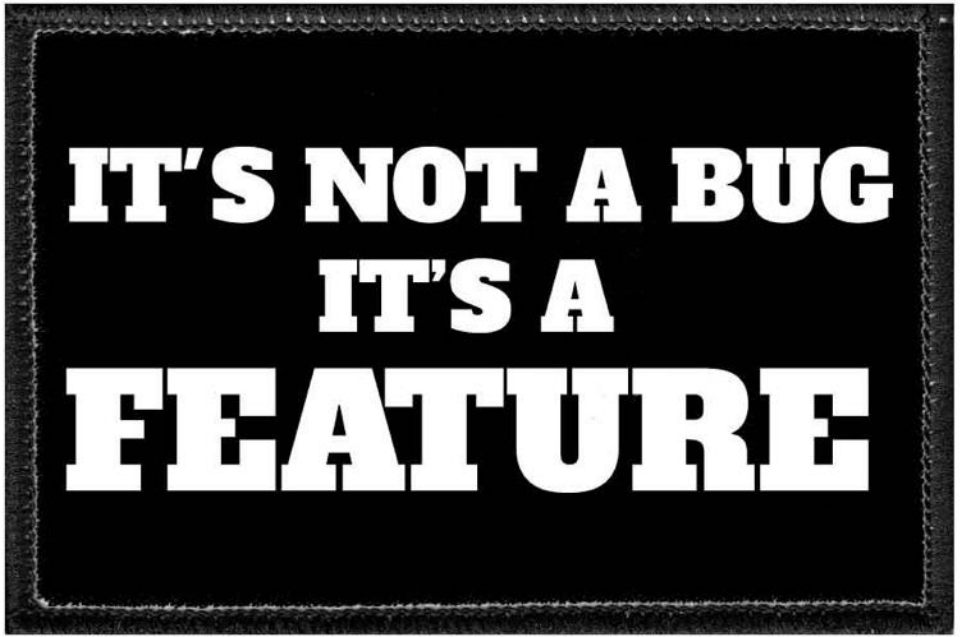
3rd Take Home Message

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 $\alpha = 1, \beta = 0$.)

Don't worry about the flexiblity

Name	Formula	Usage	DTD
LRP-		ers	✓
LRP-		ers	✓
LRI		ers	✓
LRP-α		ers	$\times^\star$
flat [		ers	$\times$
w^2-rul		r	✓
$z^\mathcal{B}$-rul		r	✓

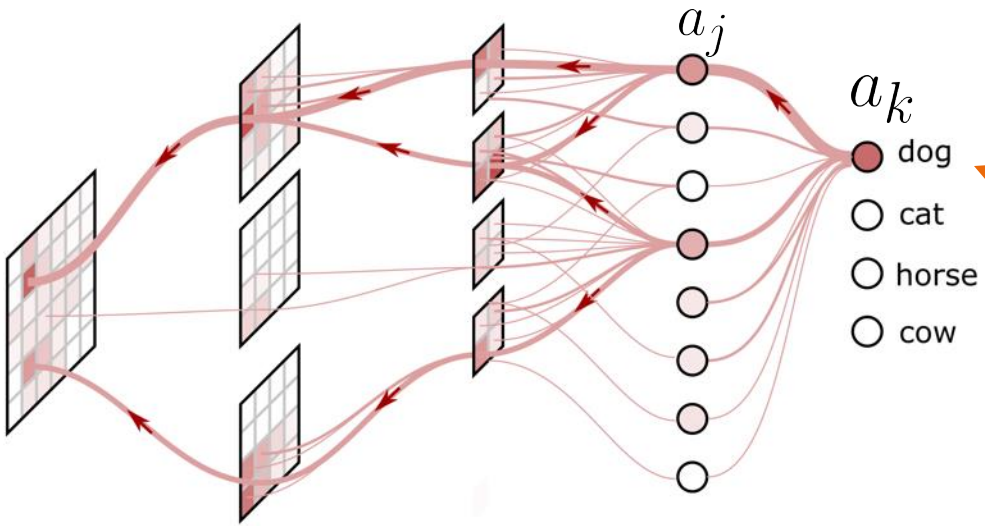


(⋆ DTD interpretation only for the case $\alpha = 1, \beta = 0$.)

Deep Taylor Decomposition Framework

(Montavon et al., Pattern Recognition, 2017)

LRP & Deep Taylor Decomposition

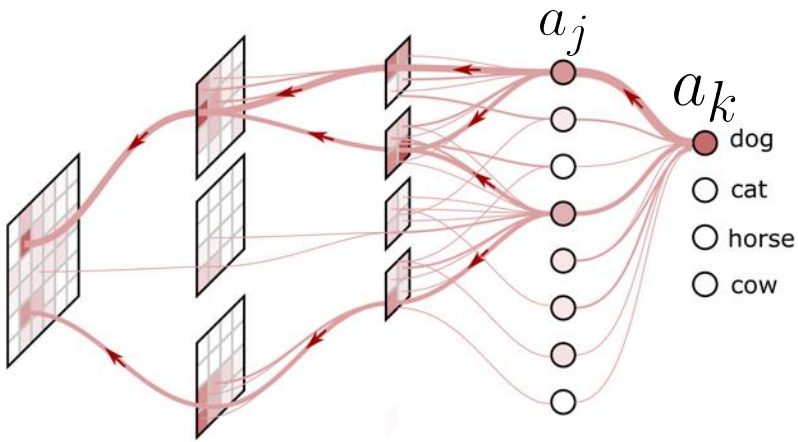


$$a_k = \max(0, \sum_j a_j \, w_{jk})$$

R_k relevance to be redistributed

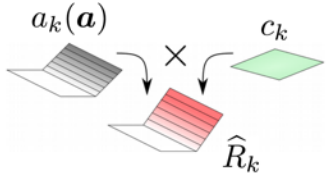
Key question: How much relevance should be redistributed from k to j ?

LRP & Deep Taylor Decomposition



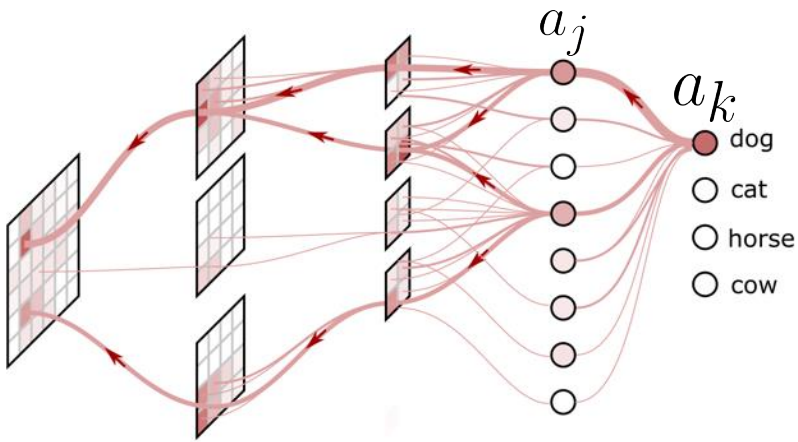
1. Relevance model

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



Key question: How much relevance should be redistributed from k to j ?

LRP & Deep Taylor Decomposition

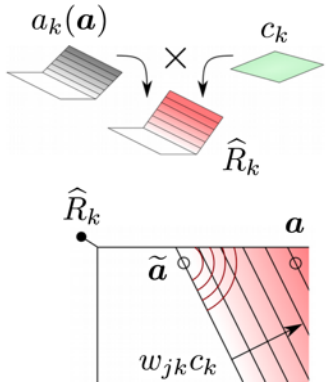


1. Relevance model

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

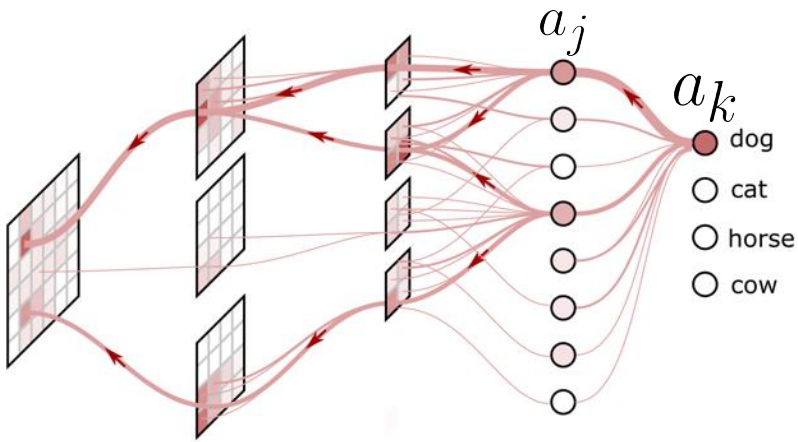
2. Taylor expansion

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



Key question: How much relevance should be redistributed from k to j ?

LRP & Deep Taylor Decomposition

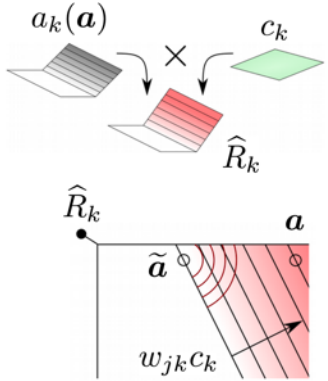


1. Relevance model

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

2. Taylor expansion

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



3. Choosing the reference point

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

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

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

large distance = less contextualized explanation (more contradicting variables)

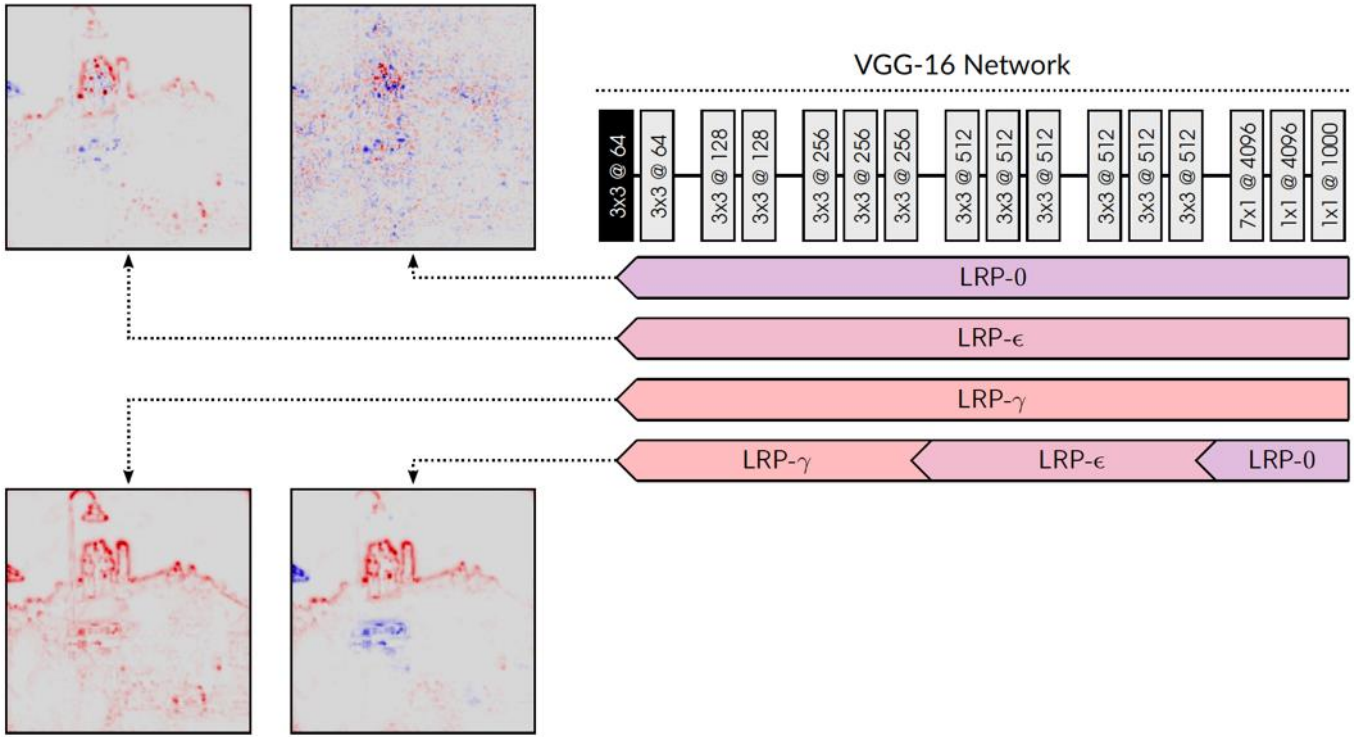
LRP & Deep Taylor Decomposition

Name	Formula	Usage	DTD	root point	trade off +/-
LRP-0 [7]	$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$	upper layers	✓	0	
LRP-ϵ [7]	$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} R_k$	middle layers	✓	$\mathbb{R}^+$	
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	✓	$\mathbb{R}^+$	✓
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	$\times^\star$	$\mathbb{R}^+$	✓
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 ($\mathbb{R}^d$)	✓	$\mathbb{R}$	
$z^\mathcal{B}$-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)	✓	[l ... h]	

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

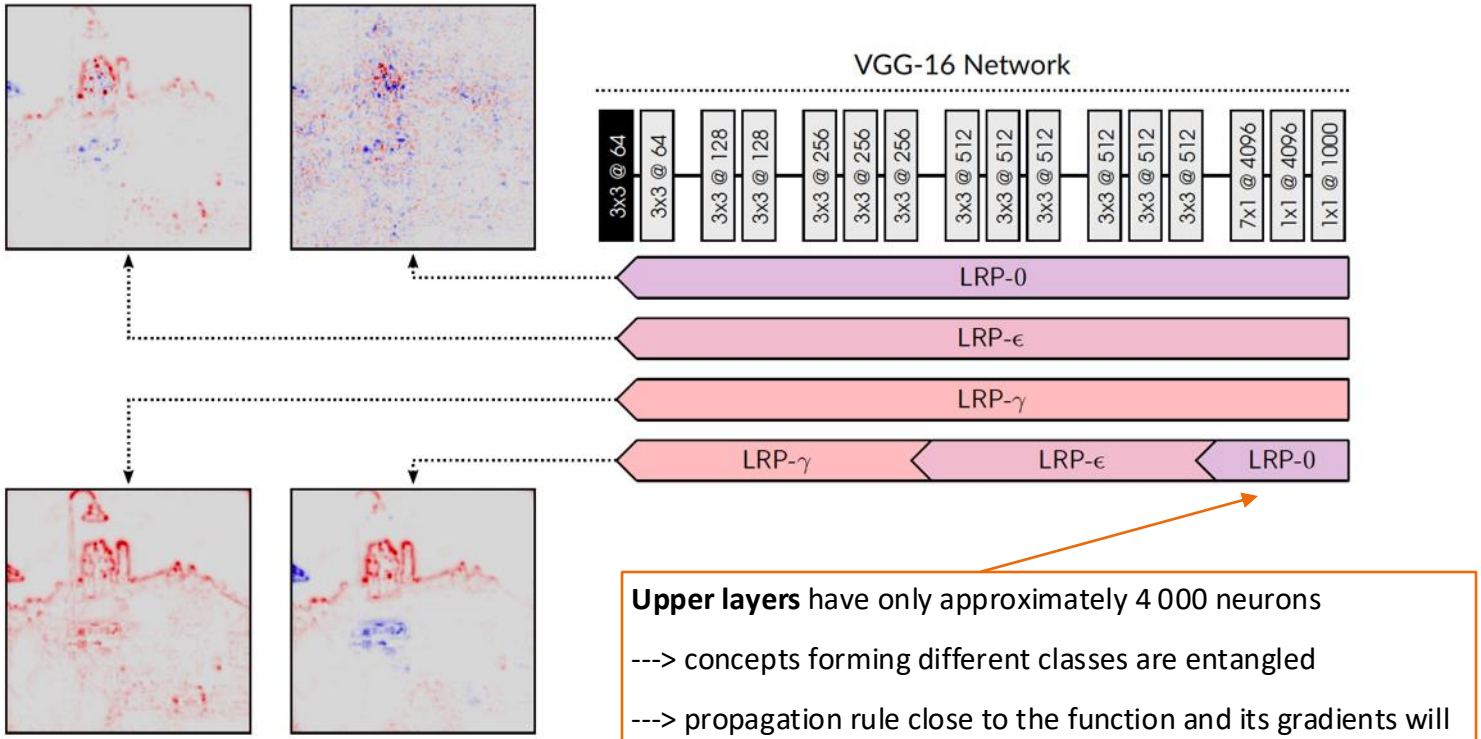
No gradient shattering problem (except LRP-0).

Effect of LRP Rules on Explanation



(Montavon et al., 2019)

Effect of LRP Rules on Explanation

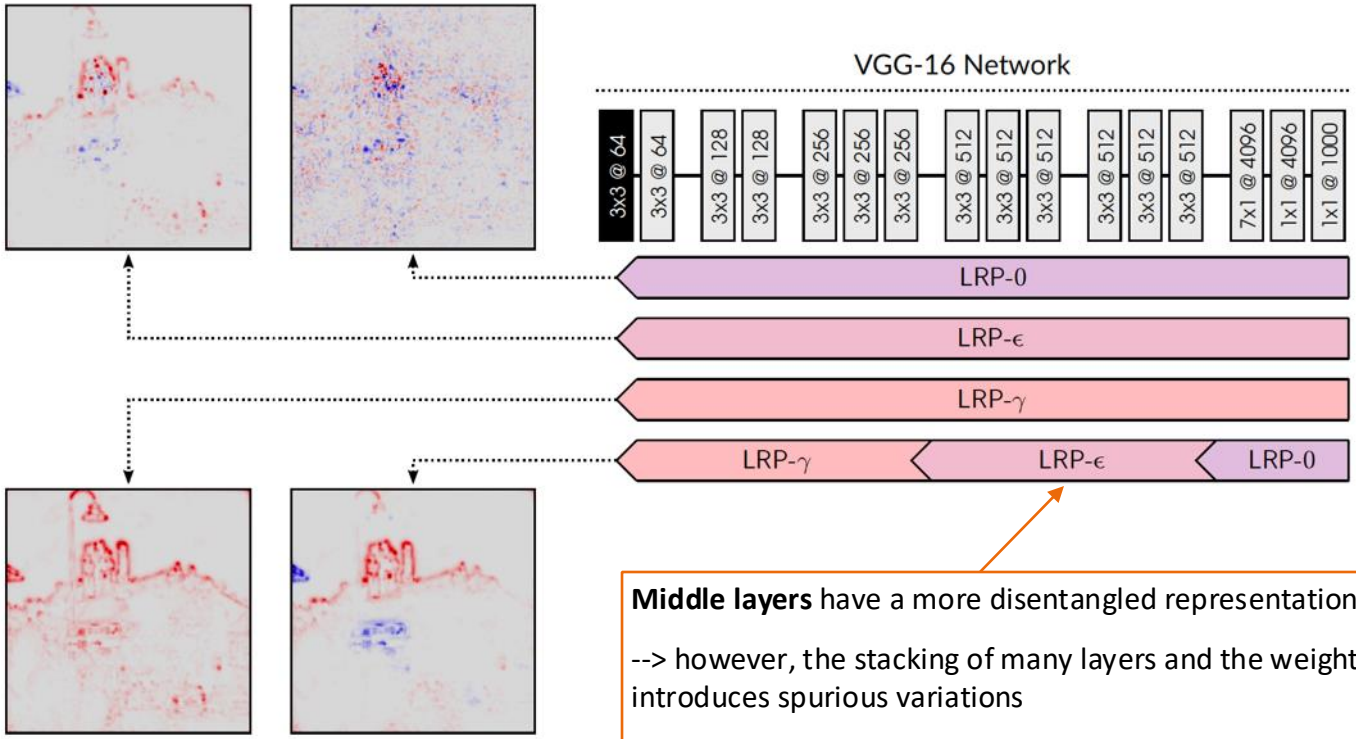


Upper layers have only approximately 4 000 neurons

---> concepts forming different classes are entangled

---> propagation rule close to the function and its gradients will be insensitive to these entanglements

Effect of LRP Rules on Explanation

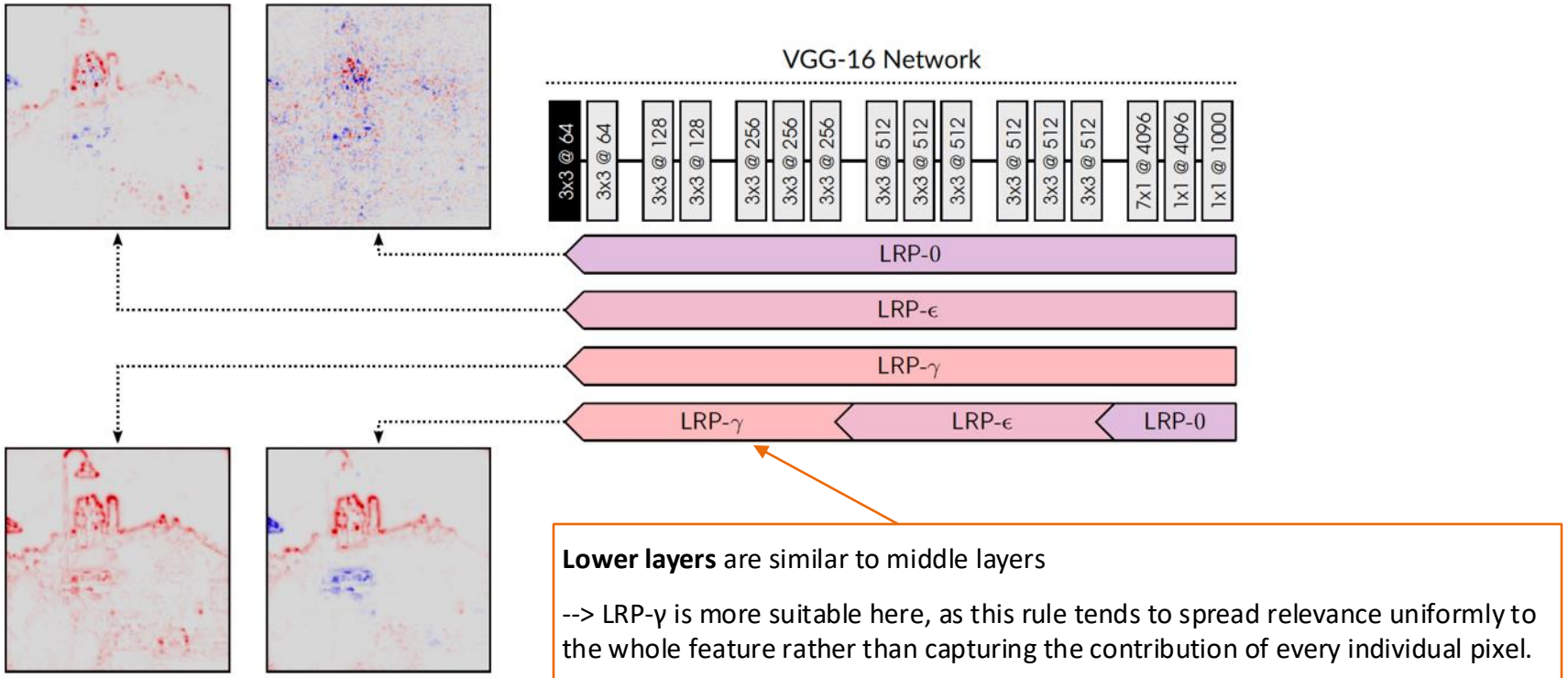


Middle layers have a more disentangled representation

--> however, the stacking of many layers and the weight sharing in convolutions introduces spurious variations

---> eps-rule filters out these spurious variations and retains only the most salient explanation factors.

Effect of LRP Rules on Explanation



Lower layers are similar to middle layers

--> LRP-γ is more suitable here, as this rule tends to spread relevance uniformly to the whole feature rather than capturing the contribution of every individual pixel.

--> This makes the explanation more understandable for a human.

Alternative: Optimization-Based Choice of LRP Rules

Optimizing Explanations by Network Canonization and Hyperparameter Search

Frederik Pahde[1] **Galip Ümit Yolcu**[1,2] **Alexander Binder**[3,4] **Wojciech Samek**[1,2,5] **Sebastian Lapuschkin**[1]
[1]Department of Artificial Intelligence, Fraunhofer Heinrich-Hertz-Institute
[2]Technische Universität Berlin
[3]ICT Cluster, Singapore Institute of Technology
[4]University of Oslo
[5]BIFOLD – Berlin Institute for the Foundations of Learning and Data

Need metrics to evaluate explanation quality (we have them)

Comparison

Method	Examples	Agnostic	Efficient	Determ.	Challenges
Pert.-Based	[ZF14, Sha53, FV17]	YES	NO	YES	OOD
Grad.-Based	[BSH+10, SVZ14, STY17]	PARTLY	YES	YES	Shattering
Sur.-Based	[RSG16, RSG18]	YES	NO	NO	Surrogate
Prop.-Based	[BBM+15, MLB+17b, ZBL+18]	NO	YES	YES	Rules*

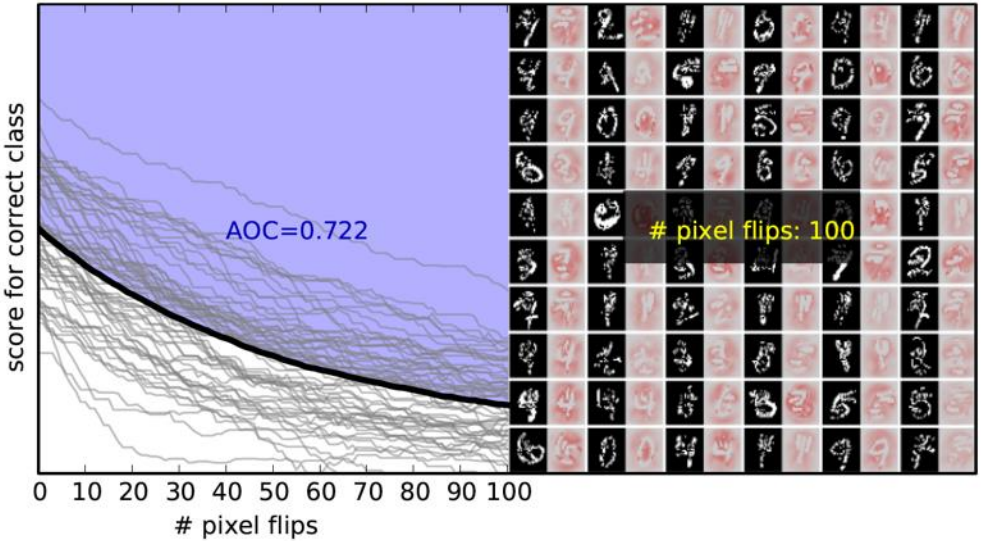
*Deep Taylor Decomposition [MLB+17b] offers a theoretical framework to design these rules.

Evaluating Explanations

Many Metrics

Faithfulness metrics (Samek'17):

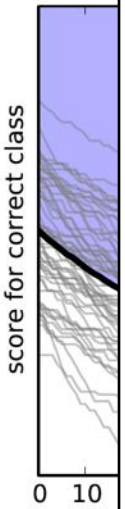
"If input features are deemed relevant, removing them should reduce evidence at the output of the network."



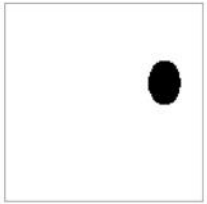
Many Metrics

Faithfulness metrics (Samek'17):

"If inp
shoul



Ground truth based metrics (Arras'22):

What is the material of the large ball? *metal*			GT Single Object
LRP [20]			0.98

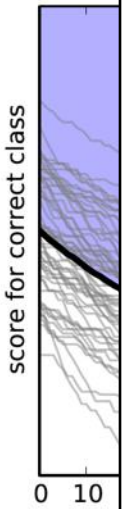
https://github.com/ahmedmagdiosman/clevr-xai

Many Metrics

Faithfulness metrics (Samek'17):

"If inp

shoul



Ground truth based metrics (Arras'22):

What
of the
meta

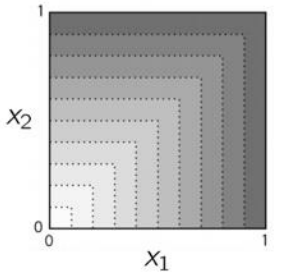
LRP

Axioms based metrics (Montavon'18):

If two inputs are the almost the same, and the prediction is also almost the same, then the explanation should also be almost the same.

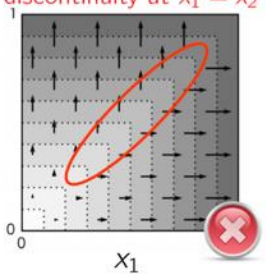
Example:

$f(x) = \max(x_1, x_2)$

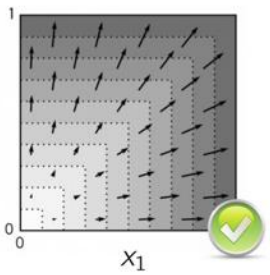


Method 1

discontinuity at $x_1 = x_2$

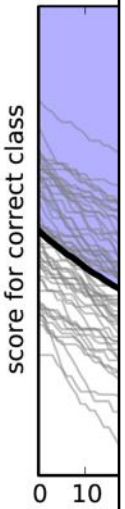


Method 2



Many Metrics

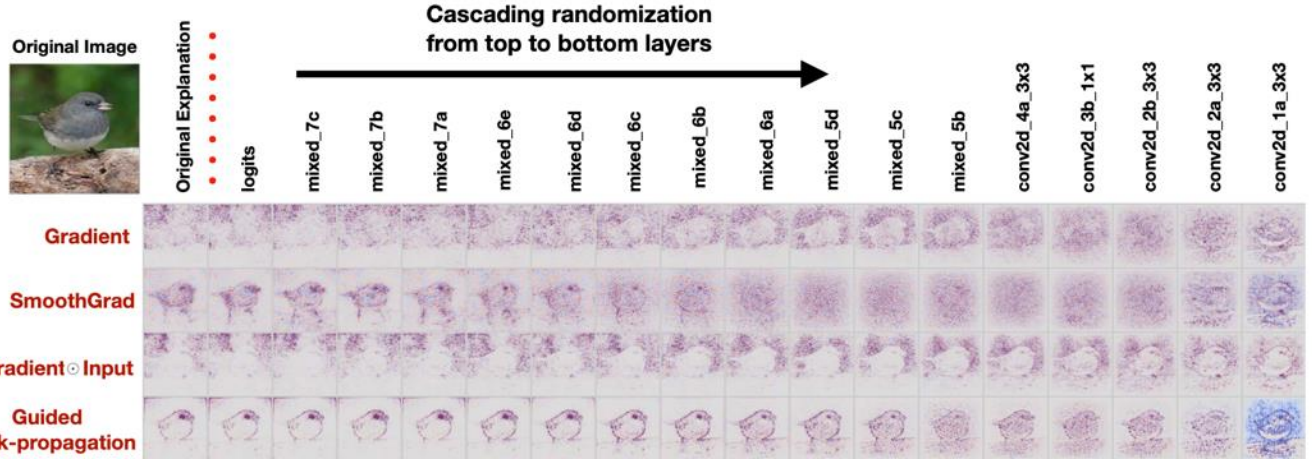
Faithfulness metrics (Samek'17):



Ground truth based metrics (Arras'22):

Axi

Randomization based metrics (Adebayo'18):



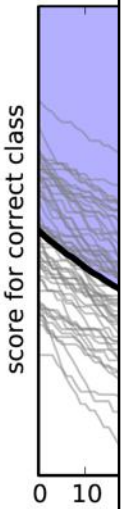
Many Metrics

"We show that some existing saliency methods are independent both of the model and of the data generating process."

Faithfulness metrics (Samek'17):

"If inp

shoul



Ground truth based metrics (Arras'22):

What
of the
meta

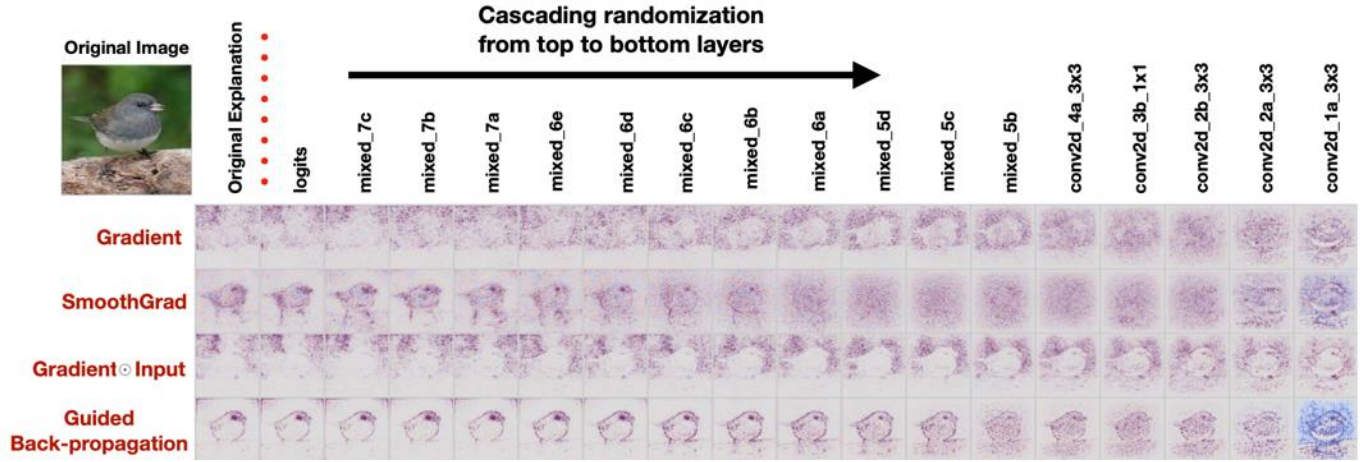
LRP

Axi

If t
aln
sar

Ex

Randomization based metrics (Adebayo'18):



Can We Explain This Observation?

Findings are due to very specific choices

--> ignoring the signs & including pixels that have zero attributions by choice of the baseline (for IG)

When both factors are accounted for, IG **attributions** for a random network and the actual network **are uncorrelated**.

A NOTE ABOUT: LOCAL EXPLANATION METHODS FOR DEEP NEURAL NETWORKS LACK SENSITIVITY TO PARAMETER VALUES

Mukund Sundararajan & Ankur Taly
Google Inc.
Mountain View, CA 94043, USA
{mukunds,ataly}@google.com

Can We Explain This Observation?

Findings are due to very specific choices

--> ignoring the signs & including pixels that have zero attributions by choice of the baseline (for IG)

--> specific randomization order (top down)

One can show that this order induces only modest alternations in the forward pass, where

(i) irrelevant features from lower, non-randomised layers persist in higher, randomised layers

(ii) high activations in lower layers are relatively likely to continue to dominate the network's response

(iii) architectures with skip connections maintain a baseline explanation that stays constant even after randomisation

Shortcomings of Top-Down Randomization-Based Sanity Checks for Evaluations of Deep Neural Network Explanations

Alexander Binder[1,2][0000−0001−9605−6209], Leander Weber[3], Sebastian Lapuschkin[3][0000−0002−0762−7258], Grégoire Montavon[4,5][0000−0001−7243−6186], Klaus-Robert Müller[5,6,7,8], and Wojciech Samek[3,5,6][0000−0002−6283−3265]

Sanity Checks Revisited: An Exploration to Repair the Model Parameter Randomisation Test

Anna Hedström[1,3*], Leander Weber[2,*], Sebastian Lapuschkin[2,†], and Marina Höhne[3,4,5†]

Can We Explain This Observation?

Findings are due to very specific choices

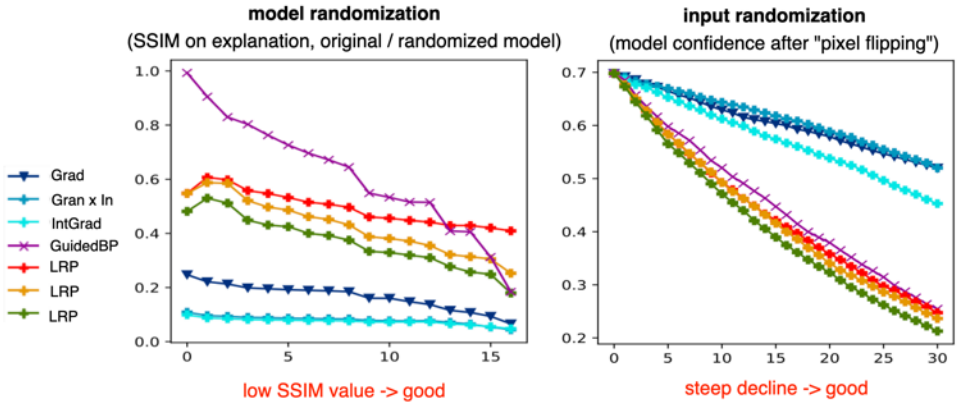
--> ignoring the signs & including pixels that have zero attributions by choice of the baseline (for IG)

--> specific randomization order (top down)

--> a similarity measure (SSIM), which varies with the noise level of an explanation

---

As a consequence more noisy explanations (e.g. gradient-based) perform better as they are less similar after randomization.



Shortcomings of Top-Down Randomization-Based Sanity Checks for Evaluations of Deep Neural Network Explanations

Alexander Binder[1,2][0000−0001−9605−6209], Leander Weber[3], Sebastian Lapuschkin[3][0000−0002−0762−7258], Grégoire Montavon[4,5][0000−0001−7243−6186], Klaus-Robert Müller[5,6,7,8], and Wojciech Samek[3,5,6][0000−0002−6283−3265]

4th Take Home Message

"There is no one best criterion to evaluate explanations"

Evaluation Toolbox

30+ evaluation metrics

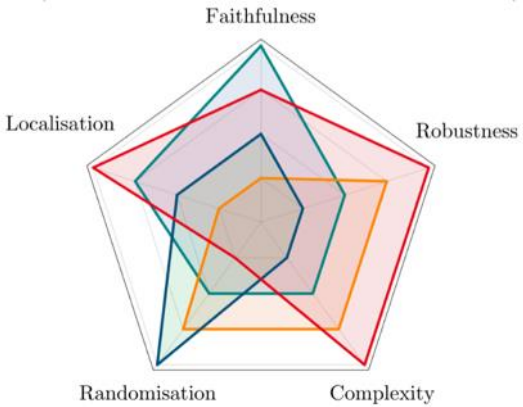
(Hedström et al. 2022)

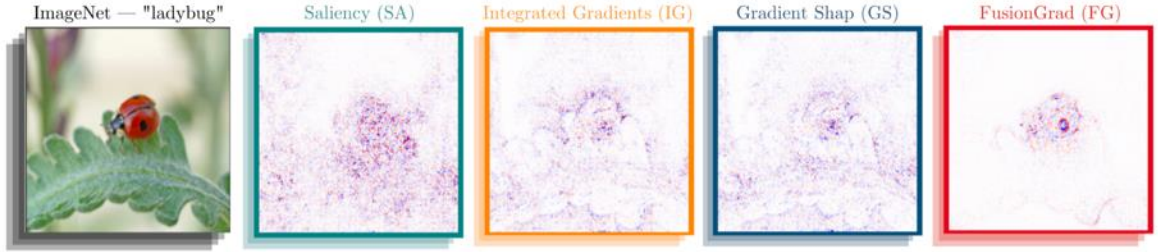
A toolkit to evaluate neural network explanations

https://github.com/understandable-machine-intelligence-lab/Quantus

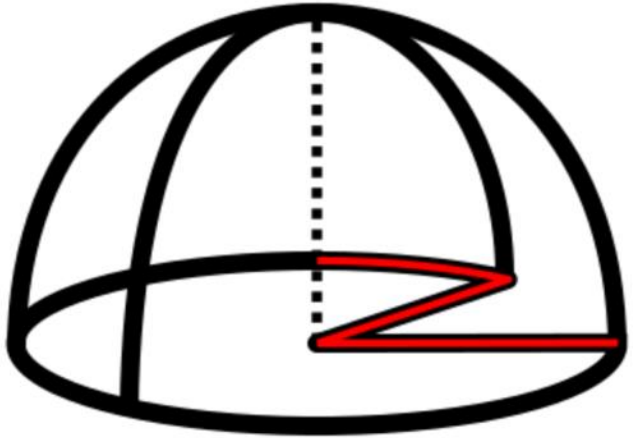
Evaluation Toolbox

Library	Faithfulness	Robustness	Localisation	Complexity	Axiomatic	Randomisation
`Captum` (2)	1	1	0	0	0	0
`AIX360` (2)	2	0	0	0	0	0
`TorchRay` (1)	0	0	1	0	0	0
`Quantus` (27)	**9**	**4**	**6**	**3**	**3**	**2**





Zennit Toolbox



https://github.com/chr5tphr/zennit

Zennit registers hooks at Pytorch's Module level, to modify the backward pass to produce rule-based attributions like LRP.

(Anders et al. 2021)

Zennit (**Z**ennit **e**xplains **n**eural **n**etworks **i**n **t**orch) is a high-level framework in Python using Pytorch for explaining/exploring neural networks.

Thank you for your attention

http://www.heatmapping.org

- Tutorials
- Software
- Online Demos