Tutorial on Interpreting and Explaining Deep Models in Computer Vision


Wojciech Samek
(Fraunhofer HHI)


Grégoire Montavon
(TU Berlin)


Klaus-Robert Müller
(TU Berlin)

08:30 - 09:15	Introduction KRM
09:15 - 10:00	Techniques for Interpretability GM
10:00 - 10:30	Coffee Break ALL
10:30 - 11:15	Applications of Interpretability WS
11:15 - 12:00	Further Applications and Wrap-Up KRM

2018 CVPR
SALT LAKE CITY • JUNE 18-22

Why interpretability?

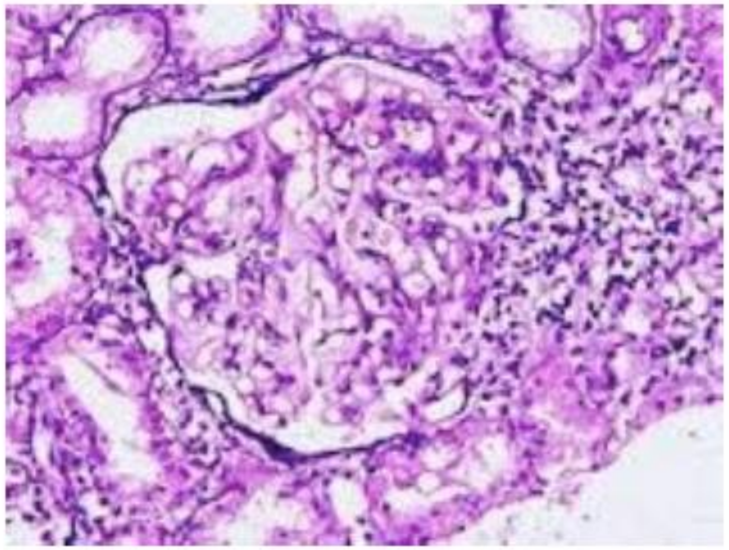
Why interpretability?

1) Verify that classifier works as expected

Wrong decisions can be costly
and dangerous

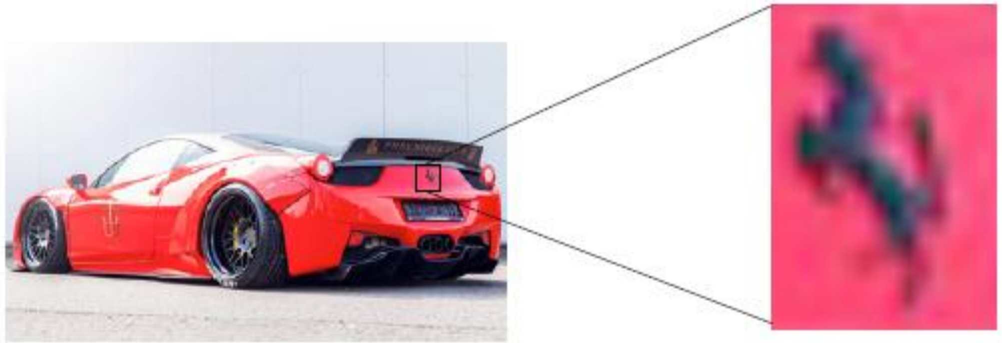
"Autonomous car crashes, because it wrongly recognizes …"

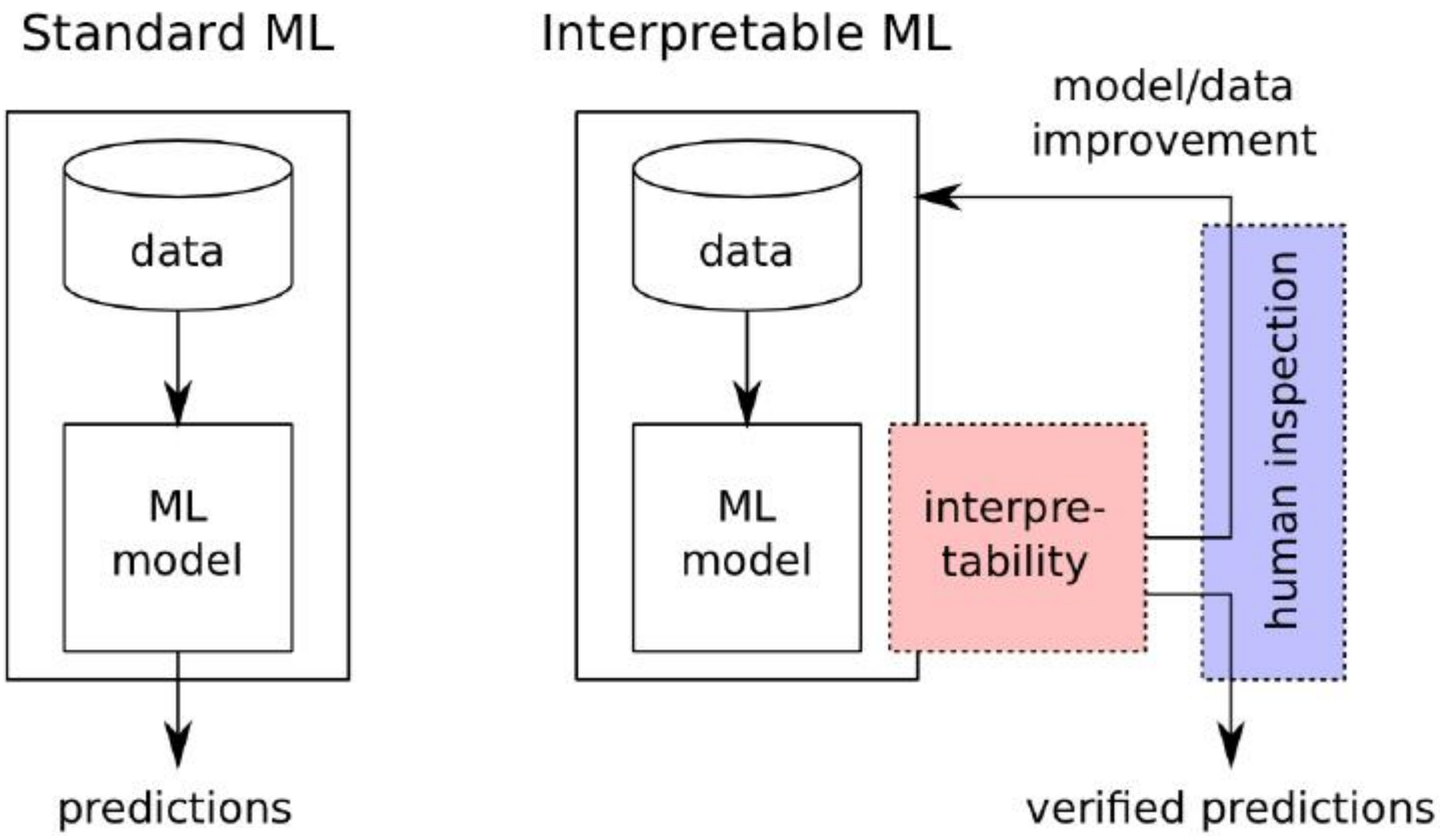
"AI medical diagnosis system misclassifies patient's disease …"



Why interpretability?

2) Improve classifier





Generalization error

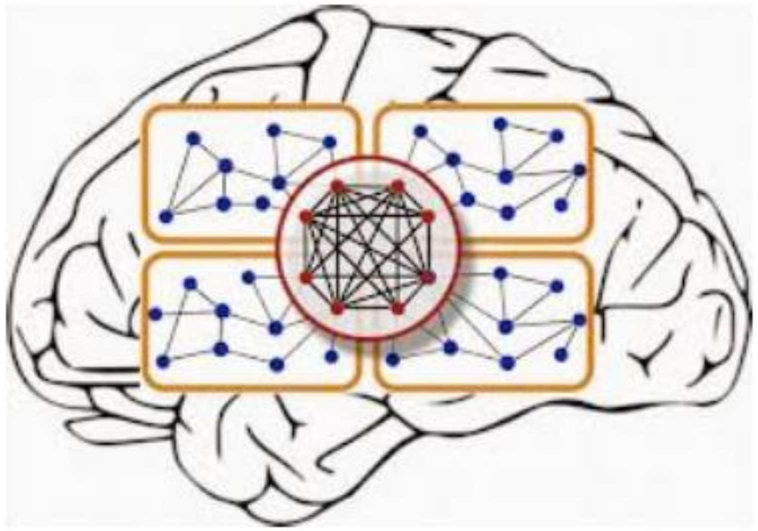
Generalization error + human experience

Why interpretability?

3) Learn from the learning machine

"It's not a human move. I've never seen a human play this move." (Fan Hui)

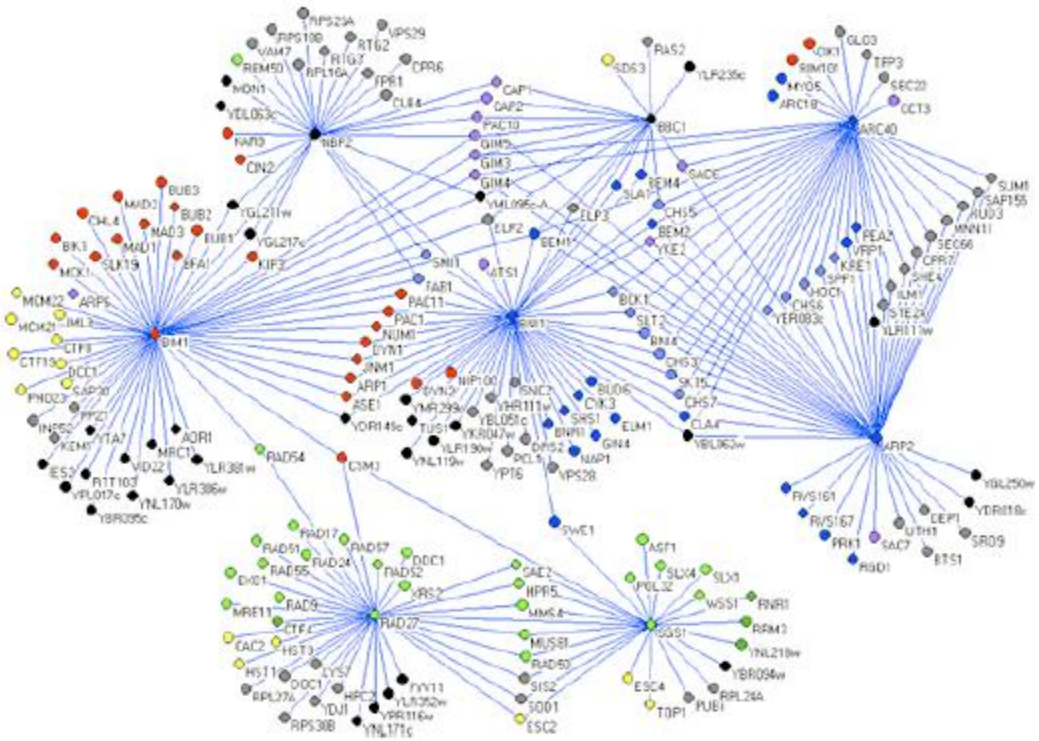
Old promise:
"Learn about the human brain."

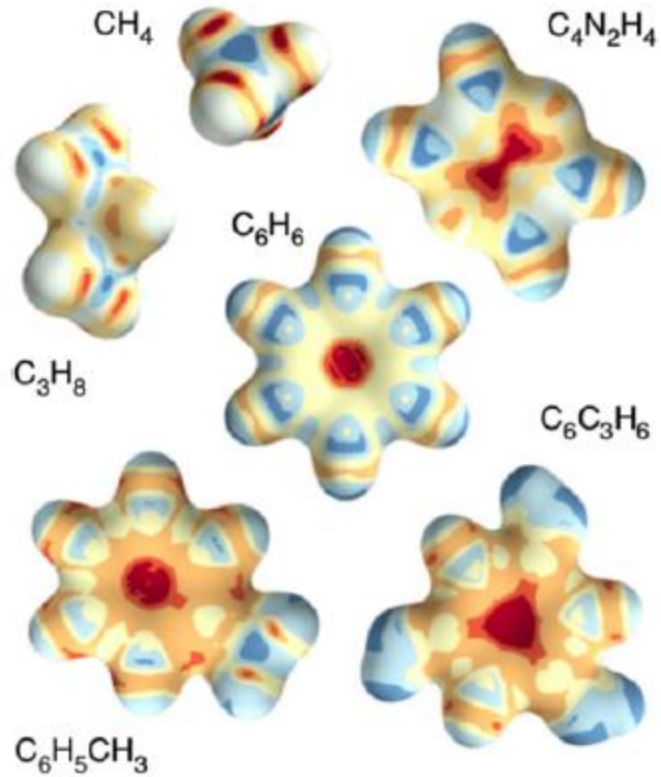


Why interpretability? Insights!

4) Interpretability in the sciences

Learn about the physical / biological / chemical mechanisms.
(e.g. find genes linked to cancer, identify binding sites …)





Why interpretability?

5) Compliance to legislation

European Union's new General Data Protection Regulation → "right to explanation"

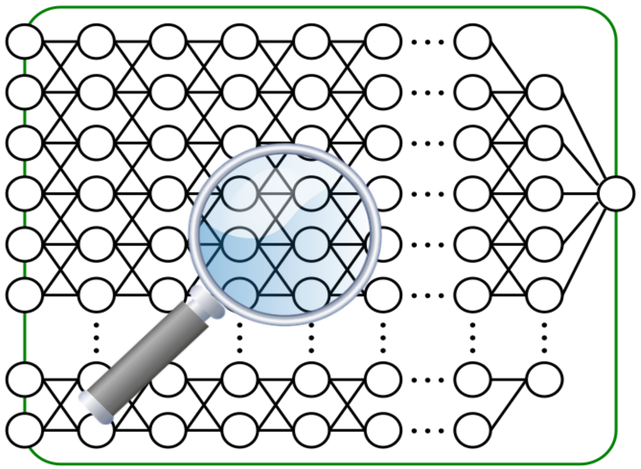
Retain human decision in order to assign responsibility.

"With interpretability we can ensure that ML models work in compliance to proposed legislation."

Overview and Intuition for different Techniques: sensitivity, deconvolution, LRP and friends.

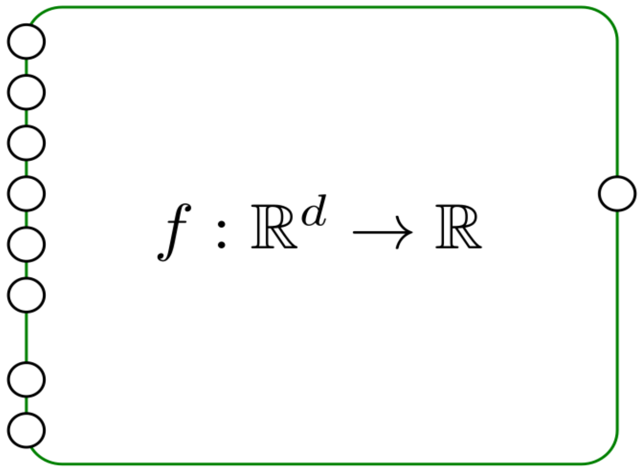
Understanding Deep Nets: Two Views

mechanistic understanding



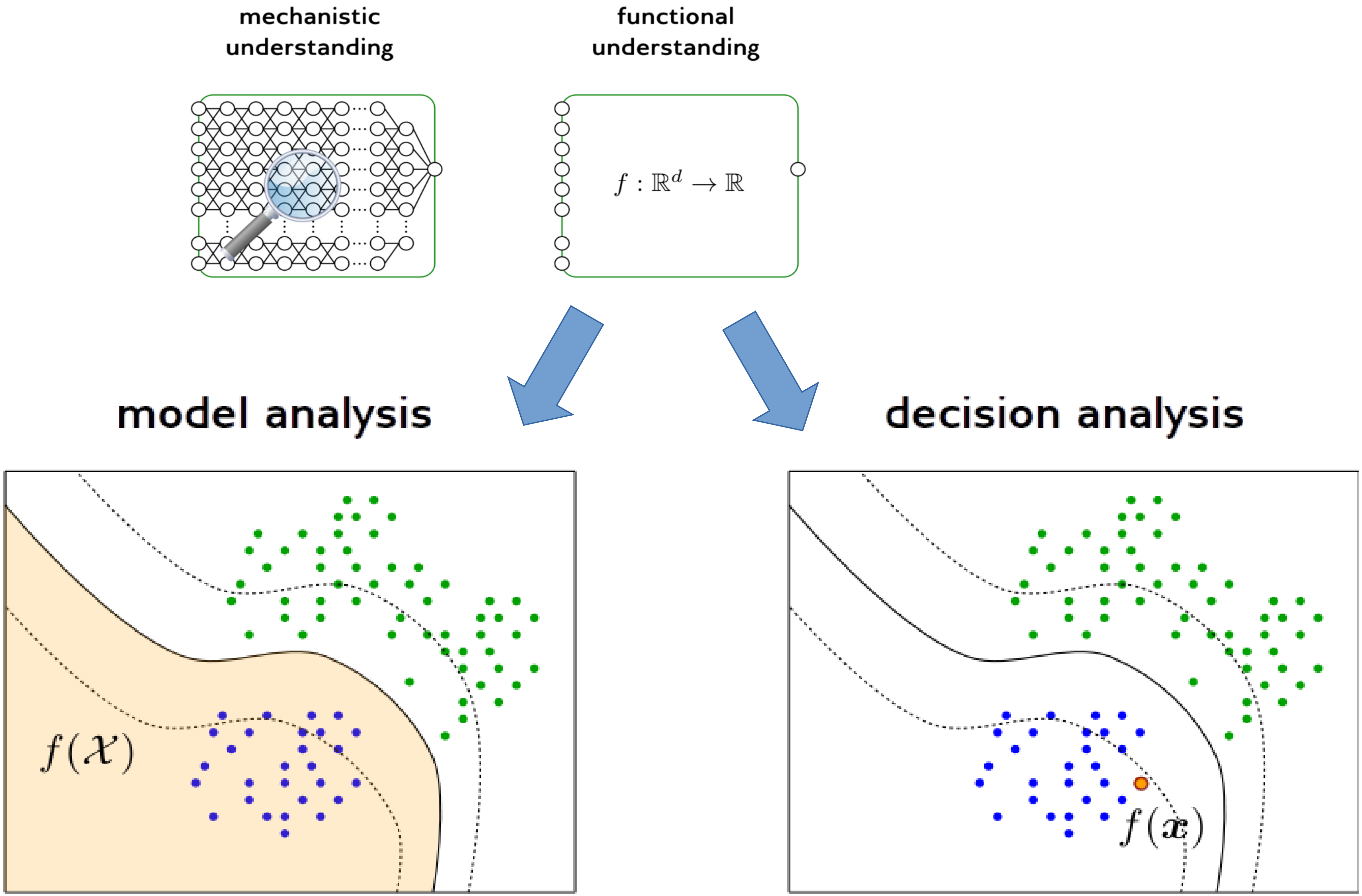
Understanding what mechanism the network uses to solve a problem or implement a function.

functional understanding



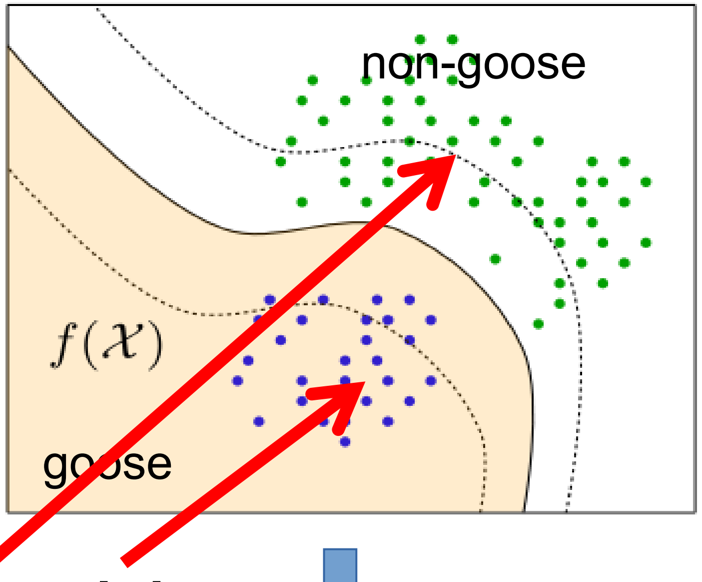
$$f : \mathbb{R}^d \to \mathbb{R}$$

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


mechanistic understanding
functional understanding
$f : \mathbb{R}^d \to \mathbb{R}$
model analysis
decision analysis
$f(\mathcal{X})$
$f(\boldsymbol{x})$

Approach 1: Class Prototypes

"How does a goose typically look like according to the neural network?"



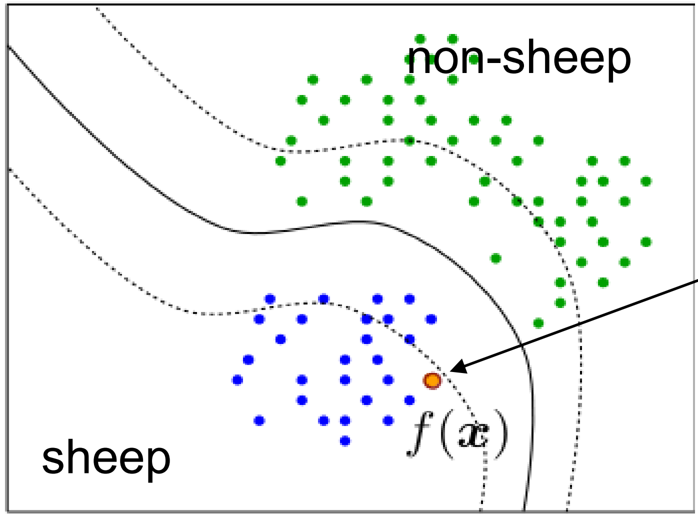
Class prototypes

$$\arg\max_{x} f(x) + \text{reg.}$$

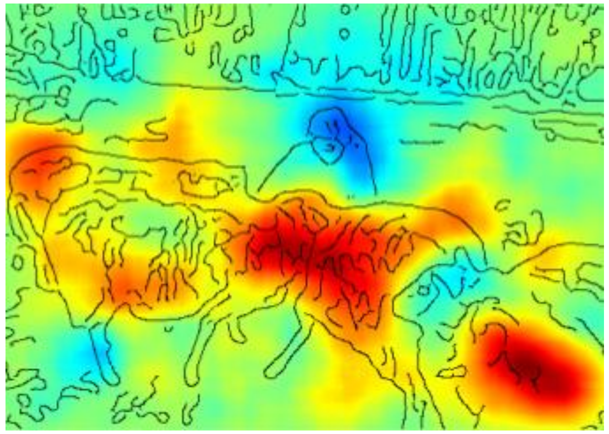
Image from **Symonian'13**

Approach 2: Individual Explanations

"Why is a given image classified as a sheep?"



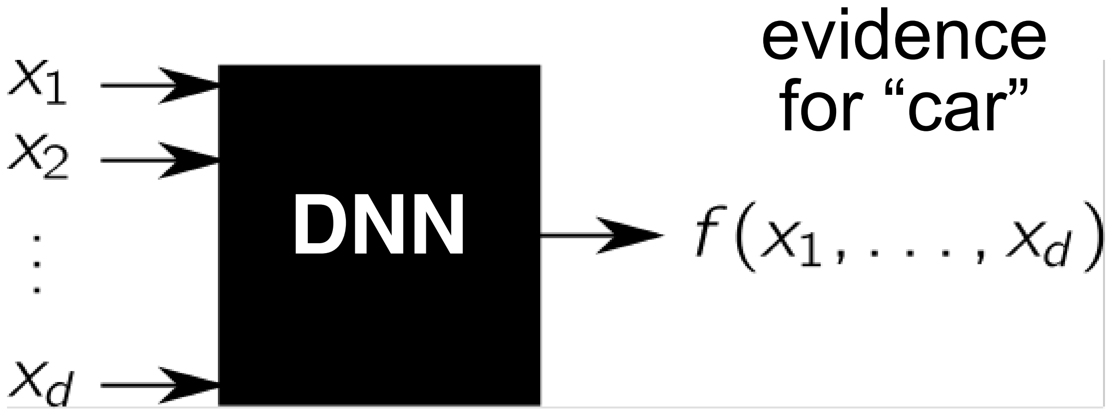
$$\text{heatmap} = LRP(x, f)$$



Images from **Lapuschkin'16**

3. Sensitivity analysis

input



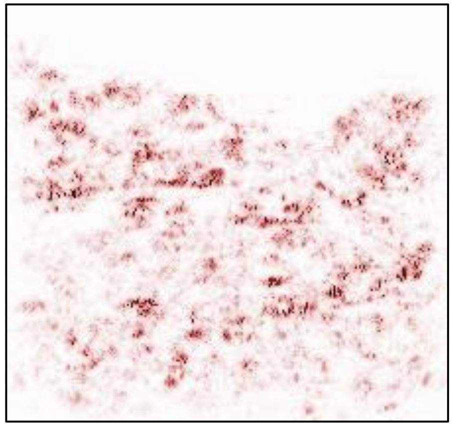
Sensitivity analysis: The relevance of input feature *i* is given by the squared partial derivative:

$$R_i = \left(\frac{\partial f}{\partial x_i}\right)^2$$

Understanding Sensitivity Analysis

Sensitivity analysis:

$$R_i = \left(\frac{\partial f}{\partial x_i}\right)^2$$



Problem: sensitivity analysis does not highlight cars

Observation:

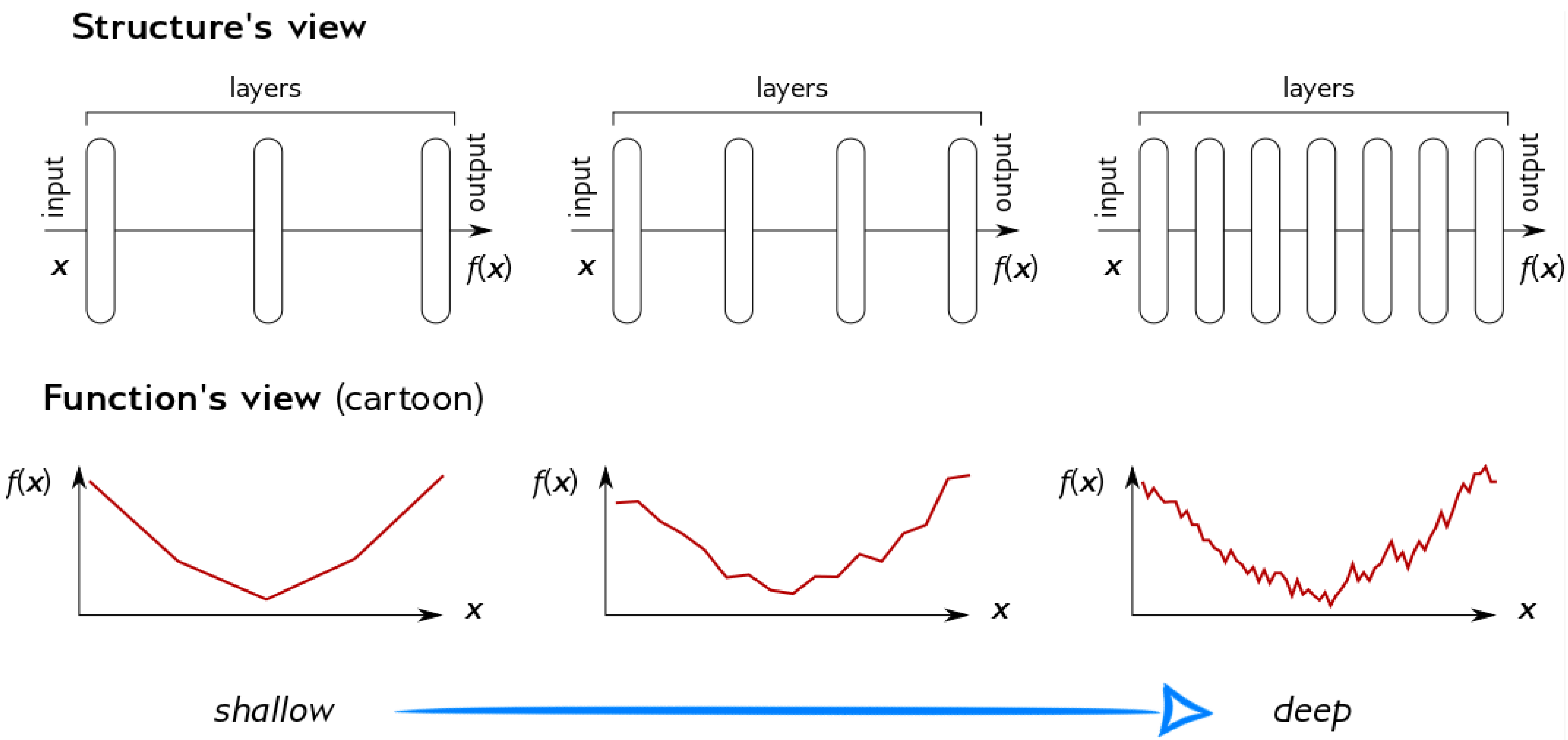
$$\sum_{i=1}^{d} \left(\frac{\partial f}{\partial x_i}\right)^2 = \|\nabla_x f\|^2$$

Sensitivity analysis explains a *variation* of the function, not the function value itself.

Sensitivity Analysis Problem: Shattered Gradients

[Montufar'14, Balduzzi'17]

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

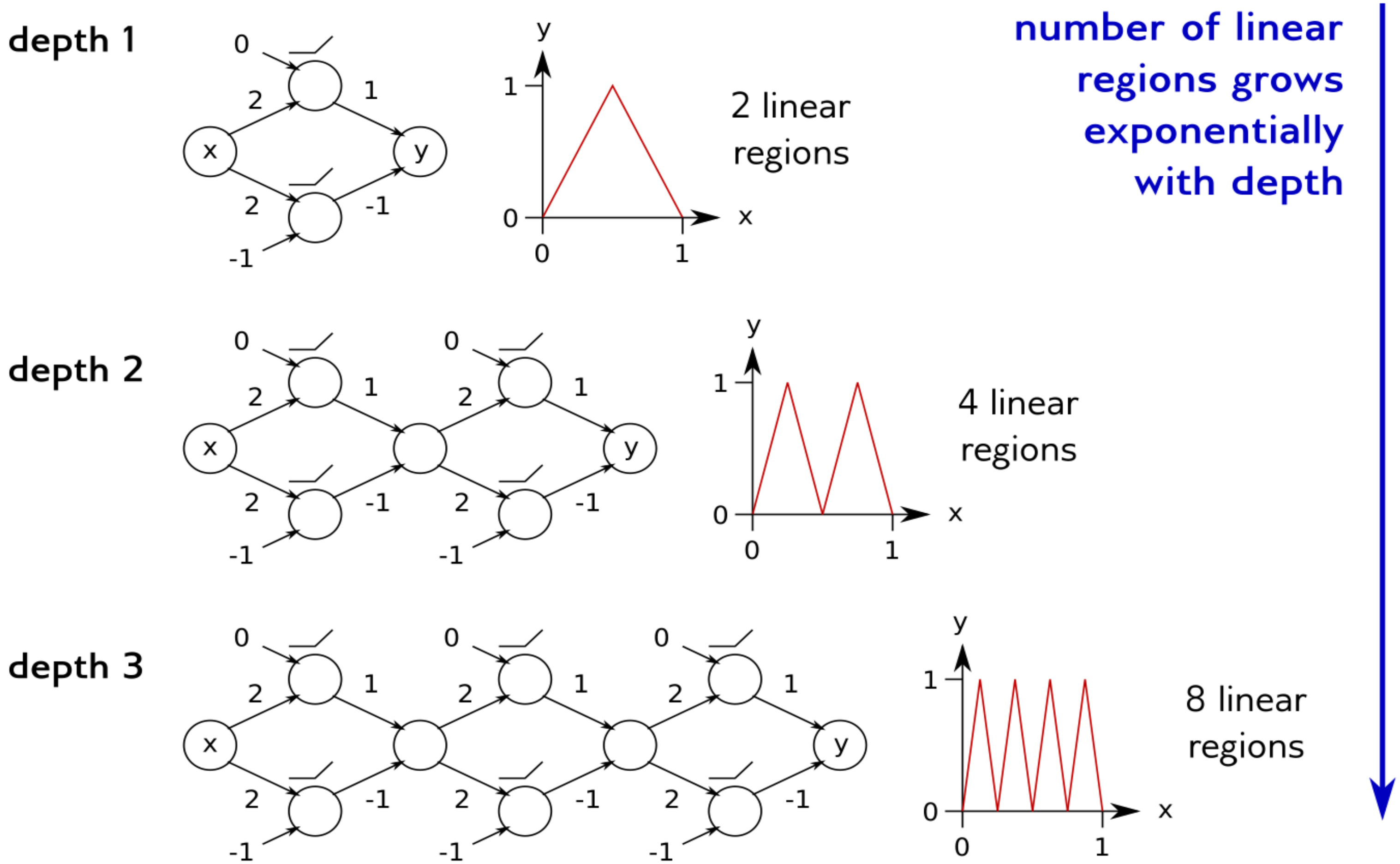


Shattered Gradients II

[Montufar'14, Balduzzi'17]

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

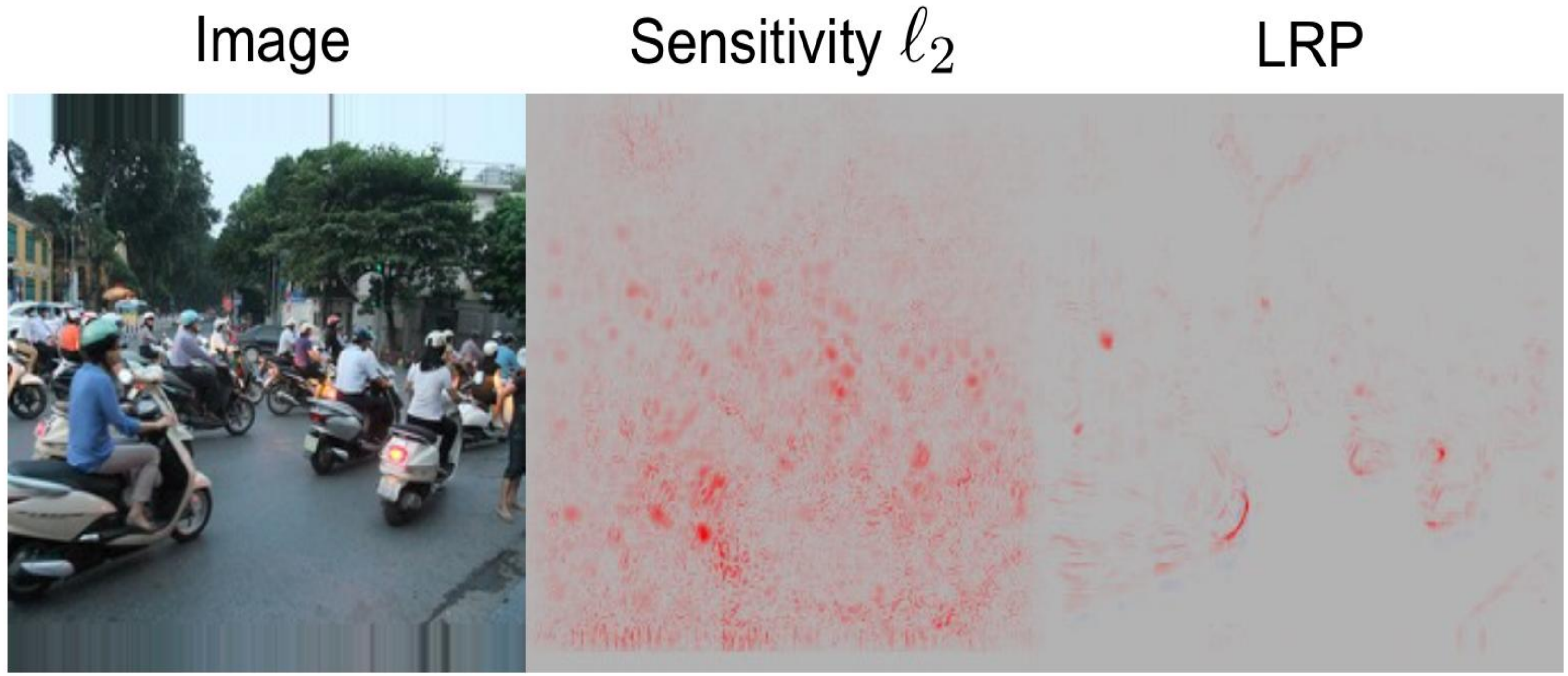
Example in [0,1]:



LPR is not sensitive to gradient shattering

$$r_i = x_i \cdot \underbrace{\sum_j \frac{w_{ij}^{+} r_j}{\sum_i x_i w_{ij}^{+}}}_{c_i}$$

LRP $\neq$ Gradient $\times$ Input



Explaining Neural Network Predictions

Layer-wise relevance Propagation (LRP, **Bach et al 15**) first method to ***explain*** nonlinear classifiers

- based on generic **theory** (related to Taylor decomposition – deep taylor decomposition **M et al 16**)

- applicable to any NN with monotonous activation, BoW models, Fisher Vectors, SVMs etc.

Explanation: "*Which pixels contribute how much to the classification*" *(**Bach et al 2015**)*

(what makes this image to be classified as a car)

$$f(x) = \sum_p h_p$$

Sensitivity / Saliency: "*Which pixels lead to increase/decrease of prediction score when changed*" *(what makes this image to be classified more/less as a car) (Baehrens et al 10, **Simonyan et al 14**)*

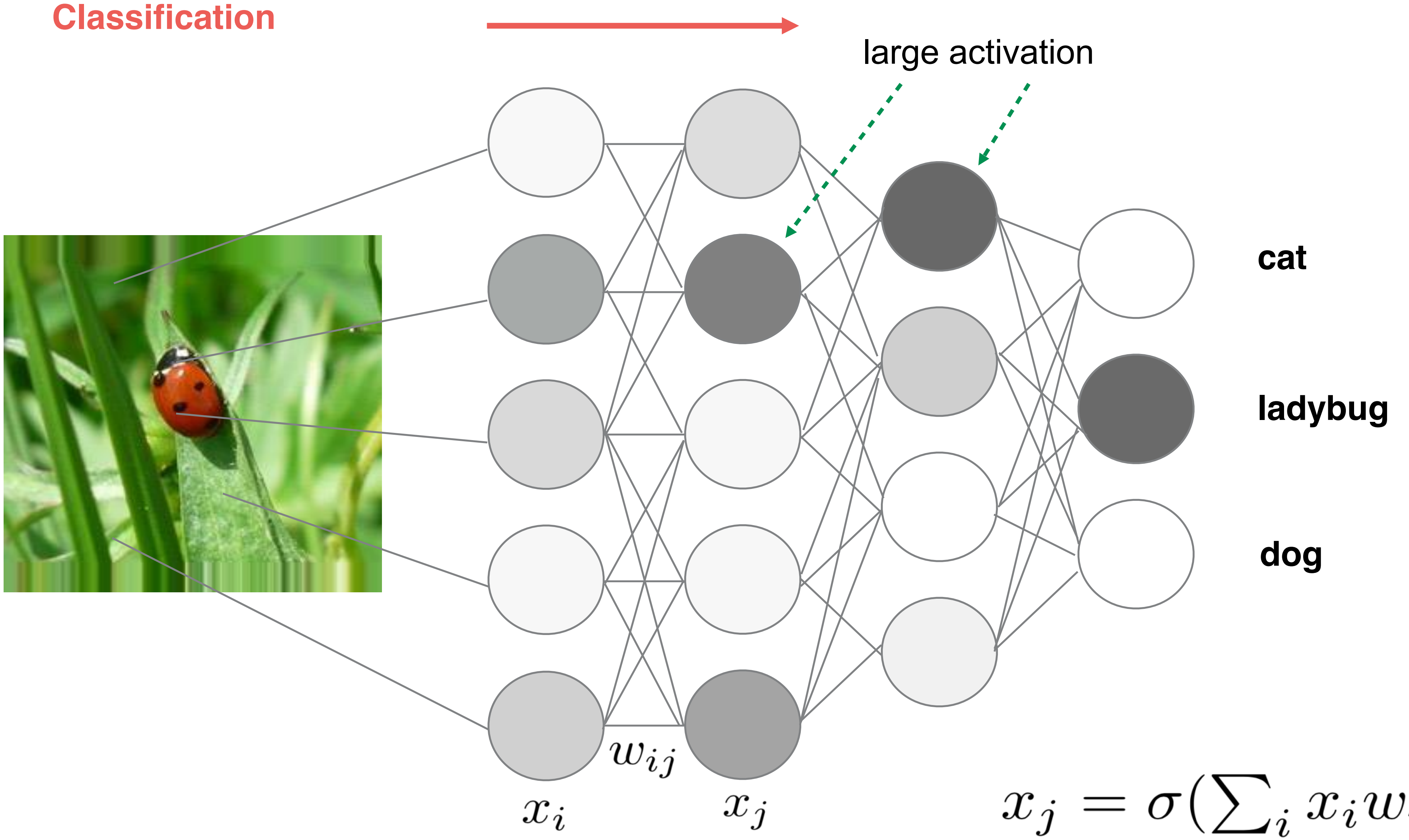
$$h_p = \left\| \frac{\partial}{\partial x_p} f(x) \right\|_\infty$$

Cf. Deconvolution: "Matching input pattern for the classified object in the image" *(**Zeiler & Fergus 2014**)*

(relation to f(x) not specified)

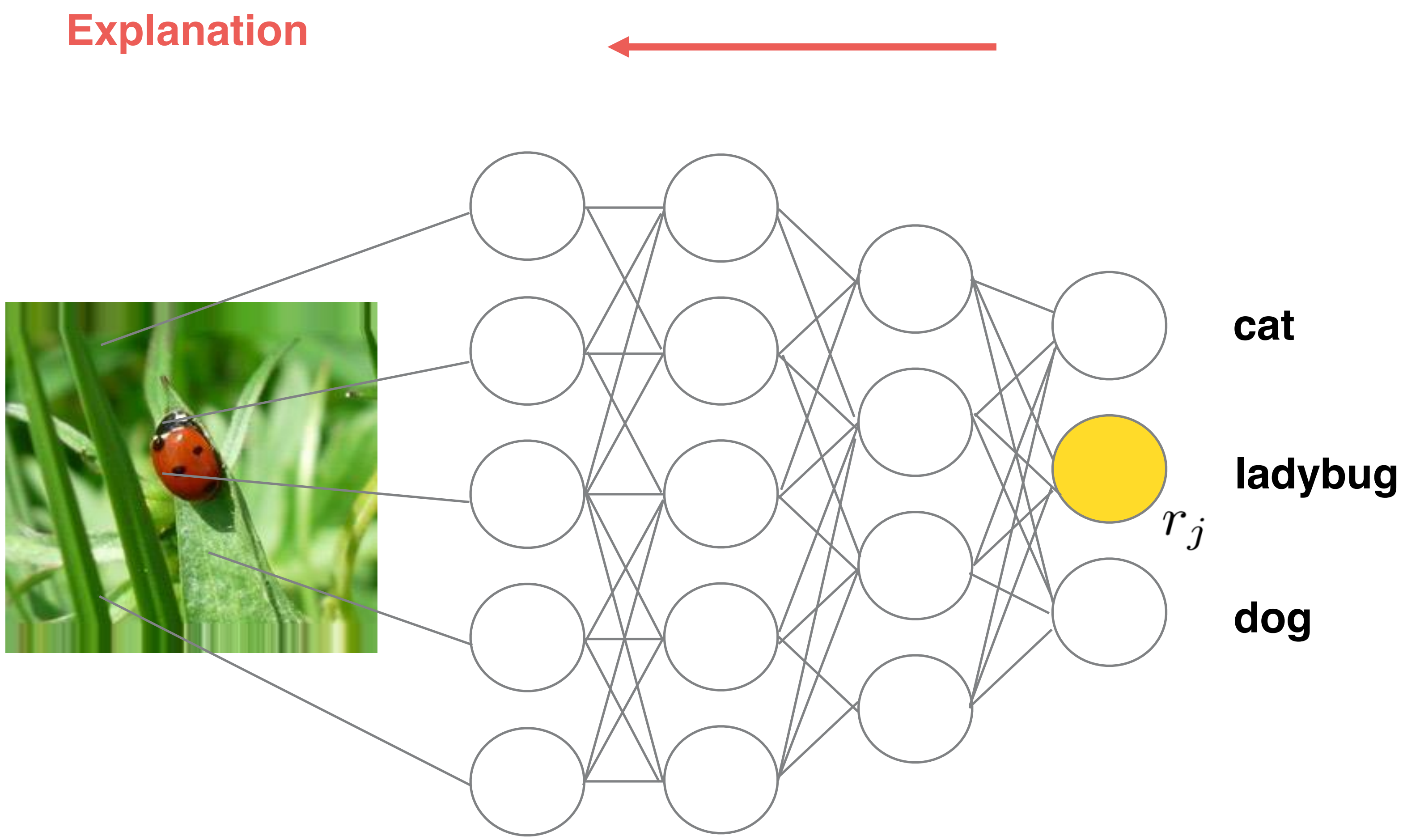
Each method solves a **different** problem!!!

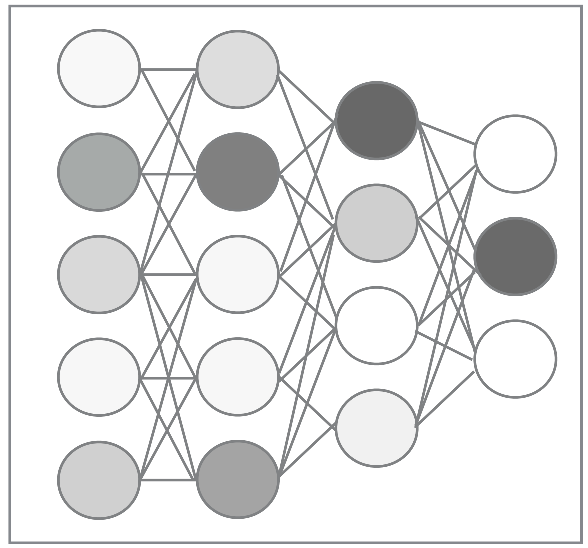
Explaining Neural Network Predictions



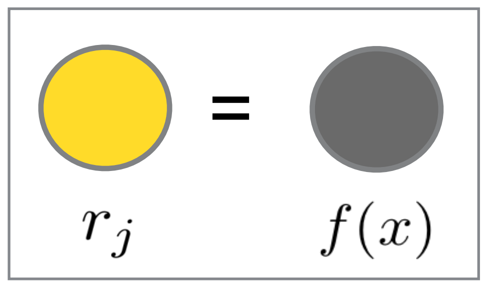
$$x_j = \sigma(\sum_i x_i w_{ij} + b_j)$$

Explaining Neural Network Predictions

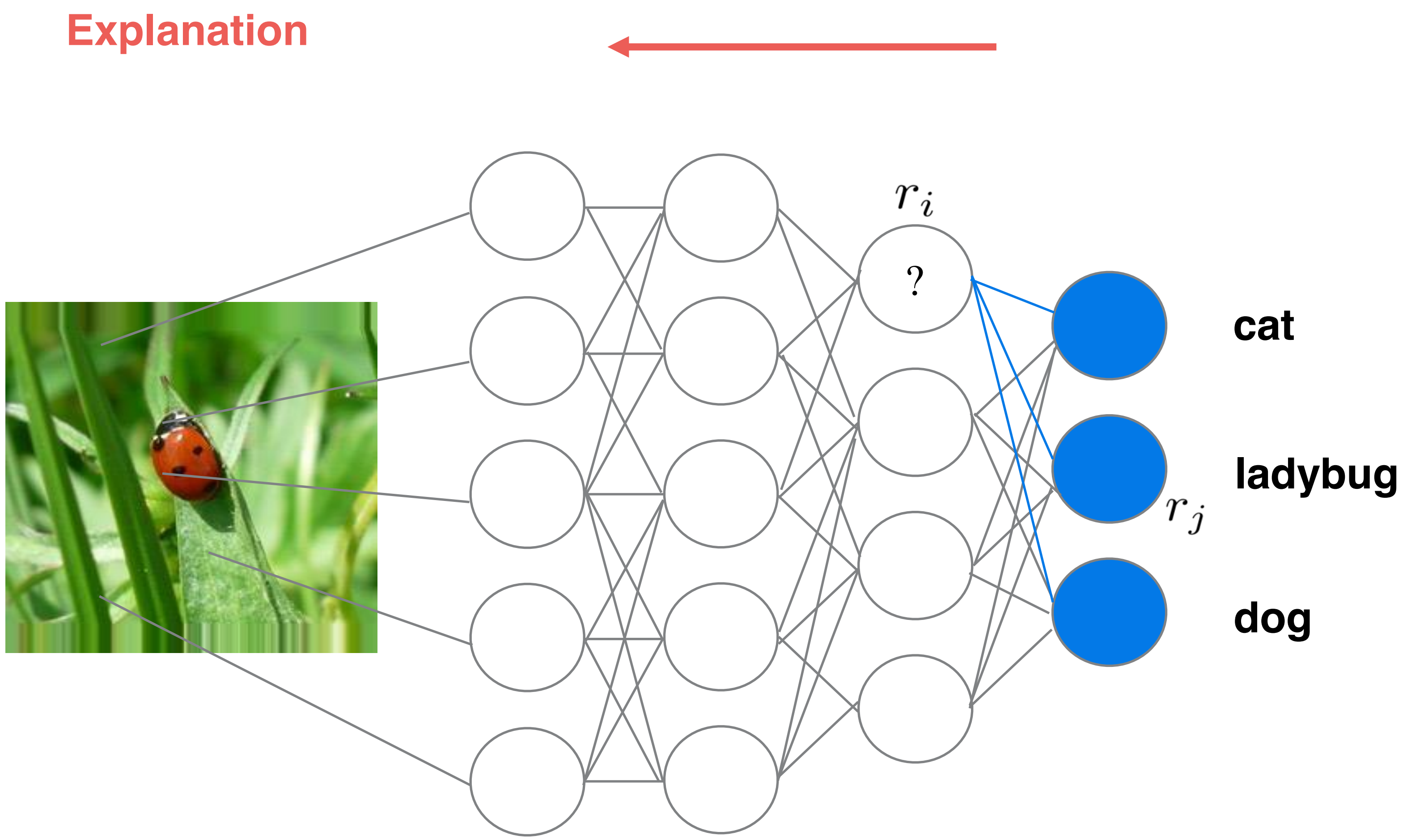


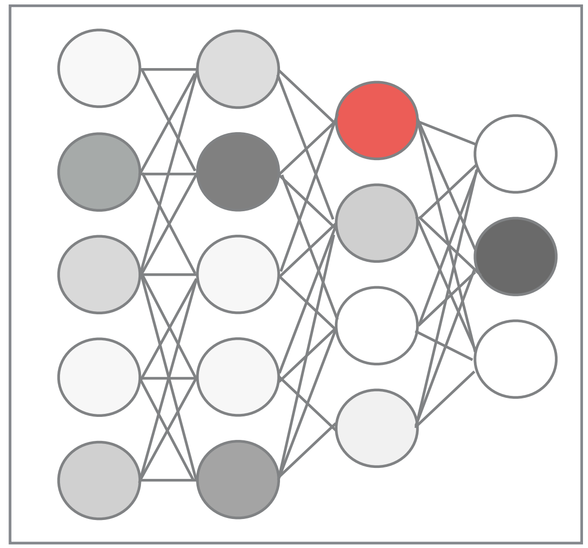


Initialization



Explaining Neural Network Predictions



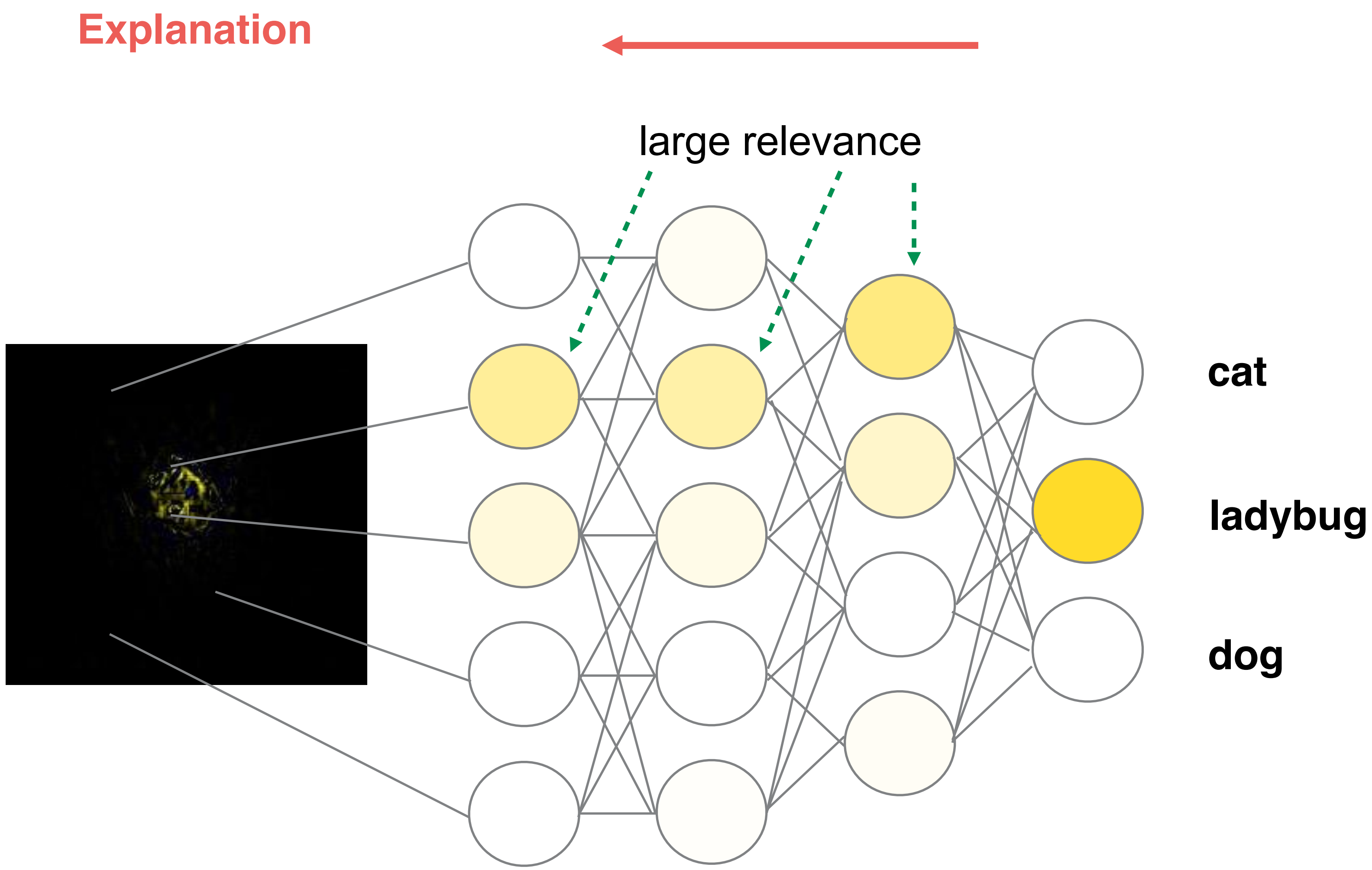


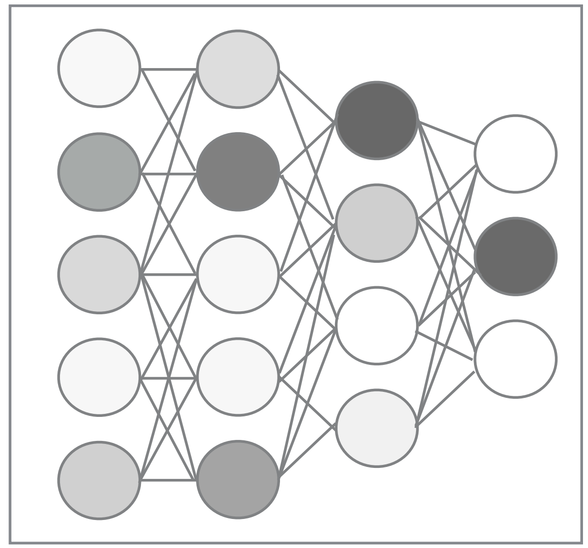
$$r_i = x_i \cdot \underbrace{\sum_j \frac{w_{ij}^+ r_j}{\sum_i x_i w_{ij}^+}}_{c_i}$$

Theoretical interpretation
Deep Taylor Decomposition

r_i deper

Explaining Neural Network Predictions





Relevance Conservation Property

$$\sum_p r_p = \ldots = \sum_i r_i = \sum_j r_j = \ldots = f(x)$$

Historical remarks on Explaining Predictors