

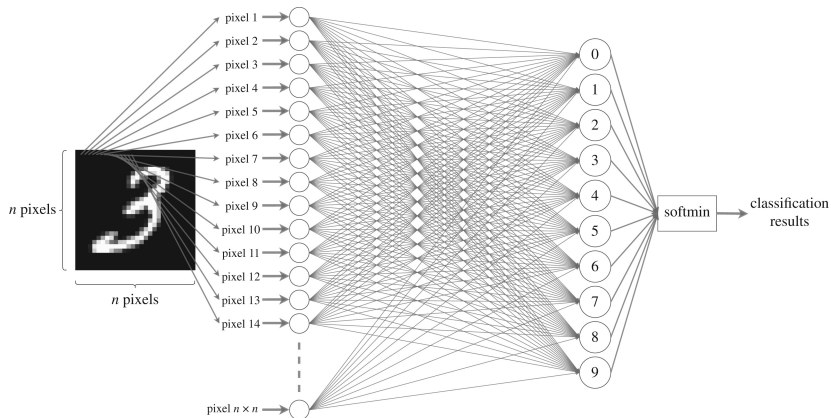
# Explaining NonLinear Classification Decisions with Deep Taylor Decomposition by Montavon et al.

Marcel Pommer

Ludwig-Maximilians-Universität München

26. Juni 2022

# Explainability



Deep neural networks perform great on a variety of problems  
**but** how can we explain decisions made by complex deep architectures?

# Table of Contents

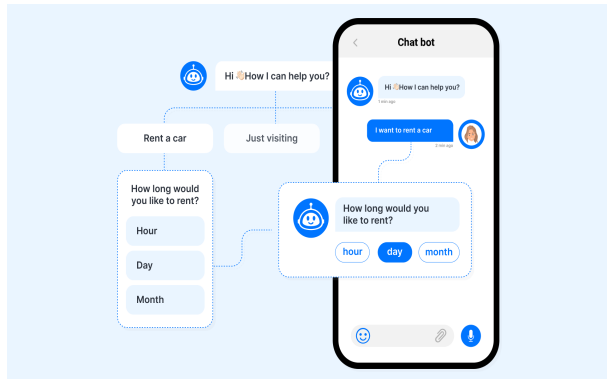
- 1 Introduction
- 2 Pixel-Wise Decomposition
  - Mathematical Framework and Definitions
  - Taylor Expansion and Sensitivity Analysis
- 3 Application to One-Layer Networks and Root Finding
  - $\omega^2$ -rule
  - $z^+$ -rule
  - $z^b$ -rule
  - Example MNIST
- 4 Application to Deep Networks
- 5 Python Programming Example

# Table of Contents

- 1 Introduction
- 2 Pixel-Wise Decomposition
  - Mathematical Framework and Definitions
  - Taylor Expansion and Sensitivity Analysis
- 3 Application to One-Layer Networks and Root Finding
  - $\omega^2$ -rule
  - $z^+$ -rule
  - $z^b$ -rule
  - Example MNIST
- 4 Application to Deep Networks
- 5 Python Programming Example

# Introduction

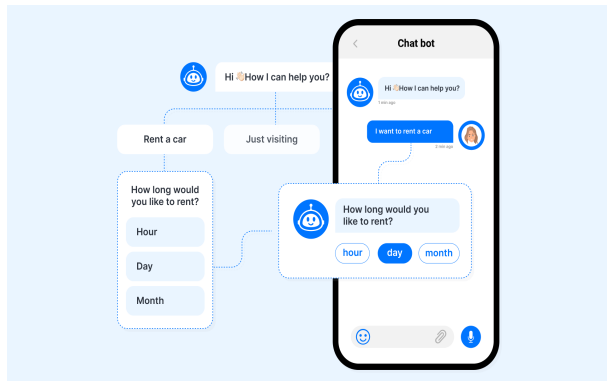
Deep neural networks revolutionized amongst others the field of



- Image recognition
- Natural language processing
- Human action recognition
- Physics
- Finance
- ...

# Introduction

Deep neural networks revolutionized amongst others the field of



- Image recognition
- Natural language processing
- Human action recognition
- Physics
- Finance
- ...

With one major drawback → **lack of transparency**

# General Idea

To accomplish the task of explainability we map relevance from the output to the input features

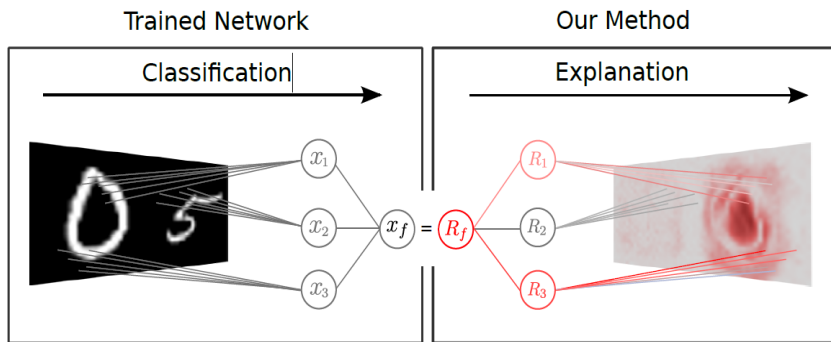


Figure 1: Neural Network Detecting 0 while Distracted by 5

# Table of Contents

- 1 Introduction
- 2 Pixel-Wise Decomposition
  - Mathematical Framework and Definitions
  - Taylor Expansion and Sensitivity Analysis
- 3 Application to One-Layer Networks and Root Finding
  - $\omega^2$ -rule
  - $z^+$ -rule
  - $z^b$ -rule
  - Example MNIST
- 4 Application to Deep Networks
- 5 Python Programming Example



# Mathematical Framework

In the context of image classification we define the following mathematical framework

- Positive valued function  $f : \mathbb{R}^d \rightarrow \mathbb{R}^+$ , where the output  $f(x)$  defines either the probability that the object is present or the quantity of the object in question
- $f(x) > 0$  expresses the presence of the object

# Mathematical Framework

In the context of image classification we define the following mathematical framework

- Positive valued function  $f : \mathbb{R}^d \rightarrow \mathbb{R}^+$ , where the output  $f(x)$  defines either the probability that the object is present or the quantity of the object in question
- $f(x) > 0$  expresses the presence of the object
- Input  $x \in \mathbb{R}^d$ , decomposable in a set of pixel values  $x = \{x_p\}$
  - Relevance score  $R_p(x)$  indicating the relevance of each pixel
- The relevance score can be displayed in a heatmap denoted by  $R(x) = \{R_p(x)\}$



# Definitions

## Definition 1

A heatmapping  $R(x)$  is conservative if the sum of assigned relevances in the pixel space corresponds to the total relevance detected by the model, that is

$$\forall x : f(x) = \sum_p R_p(x) \quad (1)$$

# Definitions

## Definition 1

A heatmapping  $R(x)$  is conservative if the sum of assigned relevances in the pixel space corresponds to the total relevance detected by the model, that is

$$\forall x : f(x) = \sum_p R_p(x) \quad (1)$$

## Definition 2

A heatmapping  $R(x)$  is positive if all values forming the heatmap are greater or equal to zero, that is:

$$\forall x, p : R_p(x) \geq 0 \quad (2)$$

# Definitions

All algorithms are shall comply with definition 1 and 2

## Definition 3

A heatmapping  $R(x)$  is consistent if it is *conservative* and *positive*. That is, it is consistent if it complies with Definitions 1 and 2.

# Definitions

All algorithms are shall comply with definition 1 and 2

## Definition 3

A heatmapping  $R(x)$  is consistent if it is *conservative* and *positive*. That is, it is consistent if it complies with Definitions 1 and 2.

But consistency is not a measure of quality which can be seen on the following example which complies with definition 3

$$\forall p : R_p(x) = \frac{1}{d} \cdot f(x),$$

where  $d$  denotes the number of pixels

# Taylor Expansion

First order Taylor expansion at root point  $\tilde{x}$

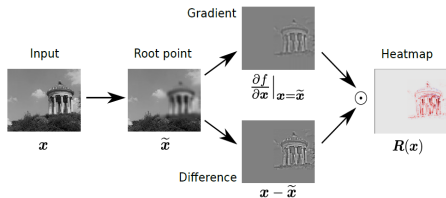
$$\begin{aligned} f(x) &= f(\tilde{x}) + \left( \frac{\partial f}{\partial x} \Big|_{x=\tilde{x}} \right)^T \cdot (x - \tilde{x}) + \epsilon \\ &= 0 + \sum_p \underbrace{\frac{\partial f}{\partial x_p} \Big|_{x=\tilde{x}} \cdot (x_p - \tilde{x}_p)}_{R_p(x)} + \epsilon \end{aligned}$$

# Taylor Expansion

First order Taylor expansion at root point  $\tilde{x}$

$$\begin{aligned}
 f(x) &= f(\tilde{x}) + \left( \frac{\partial f}{\partial x} \Big|_{x=\tilde{x}} \right)^T \cdot (x - \tilde{x}) + \epsilon \\
 &= 0 + \sum_p \underbrace{\frac{\partial f}{\partial x_p} \Big|_{x=\tilde{x}} \cdot (x_p - \tilde{x}_p)}_{R_p(x)} + \epsilon
 \end{aligned}$$

The challenge of finding a root point



- Potentially more than one root point
- Remove object but deviate as few as possible

$$\rightarrow \min_{\xi} \|\xi - x\|^2 \text{ subject to } f(\xi) = 0$$



# Sensitivity Analysis

Choose a point at infinitesimally small distance from the actual point, i.e.

$\xi = x - \delta \frac{\partial f}{\partial x}$ , where  $\delta$  is small

# Sensitivity Analysis

Choose a point at infinitesimally small distance from the actual point, i.e.  $\xi = x - \delta \frac{\partial f}{\partial x}$ , where  $\delta$  is small

If we assume a locally constant function we get

$$\begin{aligned} f(x) &= f(\xi) + \left( \frac{\partial f}{\partial x} \Big|_{x=\xi} \right)^T \cdot (x - (x - \delta \frac{\partial f}{\partial x})) + 0 \\ &= f(\xi) + \delta \left( \frac{\partial f}{\partial x} \right)^T \cdot \frac{\partial f}{\partial x} + 0 \\ &= f(\xi) + \underbrace{\sum_p \delta \left( \frac{\partial f}{\partial x} \right)^2}_{R_p} + 0 \end{aligned}$$

# Sensitivity Analysis

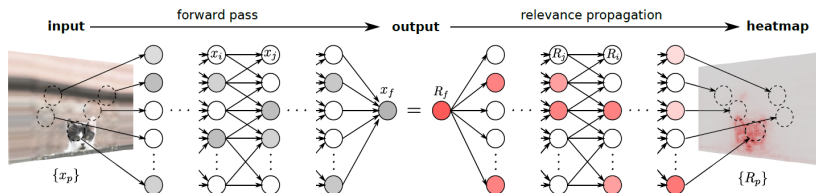
Choose a point at infinitesimally small distance from the actual point, i.e.  $\xi = x - \delta \frac{\partial f}{\partial x}$ , where  $\delta$  is small

If we assume a locally constant function we get

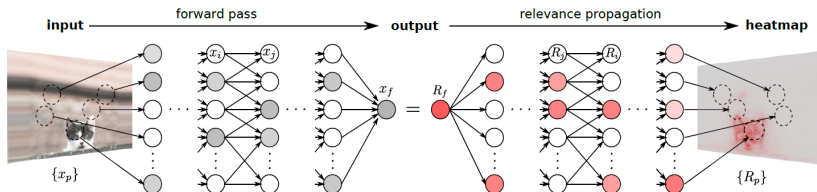
$$\begin{aligned} f(x) &= f(\xi) + \left( \frac{\partial f}{\partial x} \Big|_{x=\xi} \right)^T \cdot (x - (x - \delta \frac{\partial f}{\partial x})) + 0 \\ &= f(\xi) + \delta \left( \frac{\partial f}{\partial x} \right)^T \cdot \frac{\partial f}{\partial x} + 0 \\ &= f(\xi) + \underbrace{\sum_p \delta \left( \frac{\partial f}{\partial x} \right)^2}_{R_p} + 0 \end{aligned}$$

- The heatmap is positive but not conservative
- The heatmap only measures a local effect

# Deep Taylor Decomposition



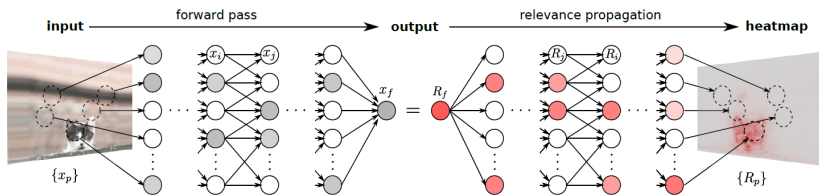
# Deep Taylor Decomposition



We view each layer as a separate function and write the Taylor decomposition of  $\sum_j R_j$  at  $\{x_i\}$  as

$$\begin{aligned} \sum_j R_j &= \left( \frac{\partial(\sum_j R_j)}{\partial\{x_i\}} \Big|_{\{\tilde{x}_i\}} \right)^T \cdot (\{x_i\} - \{\tilde{x}_i\}) + \epsilon \\ &= \underbrace{\sum_i \sum_j \frac{\partial R_j}{\partial x_i} \Big|_{\tilde{x}_i}}_{R_i} \cdot (x_i - \tilde{x}_i) + \epsilon \end{aligned}$$

# Deep Taylor Decomposition



- If each local Taylor decomposition is *conservative* then the chain of equalities is also *conservative* (layer-wise relevance conservation)
- $R_f = \dots = \sum_i R_i = \dots = \sum_p R_p$
- If each local Taylor decomposition is *positive* then the chain of equalities is also *positive*
- $R_f, \dots, \{R_i\}, \dots, \{R_p\} \geq 0$
- If each local Taylor decomposition is *consistent* then the chain of equalities is also *consistent*

# Table of Contents

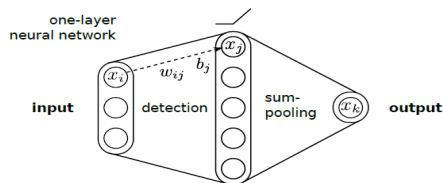
- 1 Introduction
- 2 Pixel-Wise Decomposition
  - Mathematical Framework and Definitions
  - Taylor Expansion and Sensitivity Analysis
- 3 Application to One-Layer Networks and Root Finding
  - $\omega^2$ -rule
  - $z^+$ -rule
  - $z^b$ -rule
  - Example MNIST
- 4 Application to Deep Networks
- 5 Python Programming Example

# Setting

Consider a simple detection-pooling one layer neural network with

$$x_j = \max(0, \sum_i x_i w_{ij} + b_j)$$

$$x_k = \sum_j x_j, \quad b_j \leq 0, \forall j$$



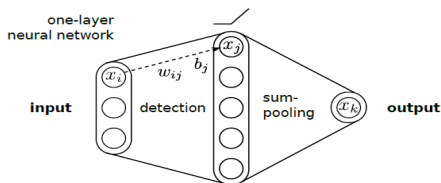


# Setting

Consider a simple detection-pooling one layer neural network with

$$x_j = \max(0, \sum_i x_i w_{ij} + b_j)$$

$$x_k = \sum_j x_j, \quad b_j \leq 0, \forall j$$



- 1  $R_k = x_k$  and thus  $R_k = \sum_j x_j$
- 2 Chose a root point and redistribute  $R_k$  on neurons  $x_j$   
 $\rightarrow R_j = \left. \frac{\partial R_k}{\partial x_j} \right|_{\{\tilde{x}_j\}} \cdot (x_j - \tilde{x}_j), \text{ with } \{\tilde{x}_j\} = 0$
- 3 Since  $\left. \frac{\partial R_k}{\partial x_j} \right|_{\{\tilde{x}_j\}} = 1$  we obtain  $R_j = x_j$
- 4 Apply Taylor decomposition another time and get  
 $R_i = \sum_j \left. \frac{\partial R_j}{\partial x_i} \right|_{\{\tilde{x}_i\}^{(j)}} \cdot (x_i - \tilde{x}_i^{(j)})$

# Derivation of Propagation Rules

Given  $R_j = \max(0, \sum_i x_i w_{ij} + b_j)$  and  $b_j \leq 0$  and a search direction  $\{v_i\}^{(j)}$

$$\tilde{x}_i^{(j)} = x_i + t v_i^{(j)} \Leftrightarrow t = \frac{\tilde{x}_i^{(j)} - x_i}{v_i^{(j)}} \quad (3)$$

# Derivation of Propagation Rules

Given  $R_j = \max(0, \sum_i x_i w_{ij} + b_j)$  and  $b_j \leq 0$  and a search direction  $\{v_i\}^{(j)}$

$$\tilde{x}_i^{(j)} = x_i + t v_i^{(j)} \Leftrightarrow t = \frac{\tilde{x}_i^{(j)} - x_i}{v_i^{(j)}} \quad (3)$$

If  $\sum_i x_i w_{ij} + b_j > 0$  the nearest root along the search direction  $\{v_i\}^{(j)}$  is given by the intersection of equation (3) and  $\sum_i \tilde{x}_i^{(j)} w_{ij} + b_j = 0$

$$\begin{aligned} 0 &= \sum_i x_i w_{ij} + b_j + \sum_i v_i^{(j)} t w_{ij} \\ \Leftrightarrow -t &= \frac{\sum_i x_i w_{ij} + b_j}{\sum_i v_i^{(j)} w_{ij}} \\ \Leftrightarrow x_i - \tilde{x}_i^{(j)} &= \frac{\sum_i x_i w_{ij} + b_j}{\sum_i v_i^{(j)} w_{ij}} v_i^{(j)} \end{aligned}$$

# Derivation of Propagation Rules

Starting from the Taylor expansion we can plug in

$$x_i - \tilde{x}_i^{(j)} = \frac{\sum_i x_i w_{ij} + b_j}{\sum_i v_i^{(j)} w_{ij}} v_i^{(j)}$$

To get

$$\begin{aligned} R_i &= \sum_j \frac{\partial R_j}{\partial x_i} \Big|_{\{\tilde{x}_i^{(j)}\}} \cdot (x_i - \tilde{x}_i^{(j)}) = \sum_j w_{ij} \frac{\sum_i x_i w_{ij} + b_j}{\sum_i v_i^{(j)} w_{ij}} v_i^{(j)} \\ &= \sum_j \frac{v_i^{(j)} w_{ij}}{\sum_i v_i^{(j)} w_{ij}} R_j \end{aligned} \quad (4)$$

# Derivation of Propagation Rules

Starting from the Taylor expansion we can plug in

$$x_i - \tilde{x}_i^{(j)} = \frac{\sum_i x_i w_{ij} + b_j}{\sum_i v_i^{(j)} w_{ij}} v_i^{(j)}$$

To get

$$\begin{aligned} R_i &= \sum_j \frac{\partial R_j}{\partial x_i} \Big|_{\{\tilde{x}_i^{(j)}\}} \cdot (x_i - \tilde{x}_i^{(j)}) = \sum_j w_{ij} \frac{\sum_i x_i w_{ij} + b_j}{\sum_i v_i^{(j)} w_{ij}} v_i^{(j)} \\ &= \sum_j \frac{v_i^{(j)} w_{ij}}{\sum_i v_i^{(j)} w_{ij}} R_j \end{aligned} \quad (4)$$

The relevance propagation rule can be calculated with the following steps

- ➊ Define a segment with search direction  $\{v_i\}^{(j)}$
- ➋ The line lies inside the input domain and contains a root point
- ➌ Inject search direction in equation (4)

$\omega^2$ -rule  $\mathcal{X} = \mathbb{R}^d$

Choose a root point which is nearest in the euclidean sense

- ① Search direction  $\{v_i\}^{(j)} = w_{ij}$  (gradient of  $R_j$ )
- ② No domain restriction and for  $\tilde{x}_i^{(j)} = x_i - \frac{R_j(x_i)}{\sum_{i'} w_{i'j}^2} w_{ij}$

$$\begin{aligned}
 R_j(\tilde{x}) &= \max(0, \sum_i (x_i - \frac{R_j(x_i) w_{ij}}{\sum_{i'} w_{i'j}^2}) w_{ij} + b_j) \\
 &= \max(0, \underbrace{\sum_i (x_i w_{ij} + b_j)}_{=R_j(x_i)} \underbrace{\frac{\sum_i w_{ij}^2}{\sum_{i'} w_{i'j}^2}}_{=1}) = 0
 \end{aligned}$$

- ③ Inject search direction in equation (4)

$\omega^2$ -rule  $\mathcal{X} = \mathbb{R}^d$

Choose a root point which is nearest in the euclidean sense

- ① Search direction  $\{v_i\}^{(j)} = w_{ij}$  (gradient of  $R_j$ )
- ② No domain restriction and for  $\tilde{x}_i^{(j)} = x_i - \frac{R_j(x_i)}{\sum_{i'} w_{i'j}^2} w_{ij}$

$$\begin{aligned}
 R_j(\tilde{x}) &= \max(0, \sum_i (x_i - \frac{R_j(x_i) w_{ij}}{\sum_{i'} w_{i'j}^2}) w_{ij} + b_j) \\
 &= \max(0, \underbrace{\sum_i (x_i w_{ij} + b_j)}_{=R_j(x_i)} - R_j(x_i) \underbrace{\frac{\sum_i w_{ij}^2}{\sum_{i'} w_{i'j}^2}}_{=1}) = 0
 \end{aligned}$$

- ③ Inject search direction in equation (4)

$\omega^2$ -rule

$$R_i = \sum_j \frac{\partial R_j}{\partial x_i} \Big|_{\{\tilde{x}_i\}^{(j)}} \cdot (x_i - \tilde{x}_i^{(j)}) = \sum_j \frac{w_{ij}^2}{\sum_{i'} w_{i'j}^2} R_j$$

$\omega^2$ -rule  $\mathcal{X} = \mathbb{R}^d$

### Proposition 1

For all functions  $g \in G$ , the deep Taylor decomposition with the  $\omega^2$ -rule is consistent.



$\omega^2$ -rule  $\mathcal{X} = \mathbb{R}^d$

### Proposition 1

For all functions  $g \in G$ , the deep Taylor decomposition with the  $\omega^2$ -rule is consistent.

### Proof

#### ① Conservative

$$\begin{aligned} \sum_i R_i &= \sum_i \left( \sum_j \frac{w_{ij}^2}{\sum_{i'} w_{i'j}^2} R_j \right) \\ &= \sum_j \underbrace{\frac{\sum_i w_{ij}^2}{\sum_{i'} w_{i'j}^2}}_{=1} R_j = \sum_j R_j = \sum_j x_j = f(x) \end{aligned}$$

#### ② Positive

$$R_i = \sum_j \frac{w_{ij}^2}{\sum_{i'} w_{i'j}^2} R_j = \sum_j \underbrace{\frac{w_{ij}^2}{\sum_{i'} w_{i'j}^2}}_{\geq 0} \underbrace{\frac{1}{\sum_{i'} w_{i'j}^2}}_{>0} \underbrace{R_j}_{\geq 0} \geq 0$$

$z^+$ -rule  $\mathcal{X} = \mathbb{R}_+^d$

Search for a root point on the segment  $(\{x_i 1_{w_{ij} < 0}\}, \{x_i\}) \subset \mathbb{R}_+^d$

- ① Search direction  $\{v_i\}^{(j)} = x_i - x_i 1_{w_{ij} < 0} = x_i 1_{w_{ij} \geq 0}$
- ② If  $\{x_i\} \in \mathbb{R}_+^d$  so is the whole domain, further for  $w_{ij}^- = \min(0, w_{ij})$  and  $w_{ij}^+ = \max(0, w_{ij})$

$$\begin{aligned} R_j(\{x_i 1_{w_{ij} < 0}\}) &= \max(0, \sum_i x_i 1_{w_{ij} < 0} w_{ij} + b_j) \\ &= \max(0, \underbrace{\sum_i x_i w_{ij}^-}_{\leq 0} + b_j) = 0 \end{aligned}$$

- ③ Inject search direction in equation (4)

$z^+$ -rule  $\mathcal{X} = \mathbb{R}_+^d$

Search for a root point on the segment  $(\{x_i 1_{w_{ij} < 0}\}, \{x_i\}) \subset \mathbb{R}_+^d$

- ① Search direction  $\{v_i\}^{(j)} = x_i - x_i 1_{w_{ij} < 0} = x_i 1_{w_{ij} \geq 0}$
- ② If  $\{x_i\} \in \mathbb{R}_+^d$  so is the whole domain, further for  $w_{ij}^- = \min(0, w_{ij})$  and  $w_{ij}^+ = \max(0, w_{ij})$

$$\begin{aligned} R_j(\{x_i 1_{w_{ij} < 0}\}) &= \max(0, \sum_i x_i 1_{w_{ij} < 0} w_{ij} + b_j) \\ &= \max(0, \underbrace{\sum_i x_i w_{ij}^-}_{\leq 0} + b_j) = 0 \end{aligned}$$

- ③ Inject search direction in equation (4)

$z^+$ -rule

$$R_i = \sum_j \frac{x_i w_{ij}^+}{\sum_{i'} x_{i'} w_{i'j}^+} R_j$$

$z^+$ -rule  $\mathcal{X} = \mathbb{R}_+^d$

## Proposition 2

For all functions  $g \in G$  and data points  $\{x_i\} \in \mathbb{R}_+^d$ , the deep Taylor decomposition with the  $z^+$ -rule is consistent.

$$z^+\text{-rule } \mathcal{X} = \mathbb{R}_+^d$$

## Proposition 2

For all functions  $g \in G$  and data points  $\{x_i\} \in \mathbb{R}_+^d$ , the deep Taylor decomposition with the  $z^+$ -rule is consistent.

### Proof

If  $\sum_i x_i w_{ij}^+ > 0$  the same proof as for the  $w^2$ -rule applies, if  $\sum_i x_i w_{ij}^+ = 0$  it follows that  $\forall i : x_i w_{ij} \leq 0$  and

$$R_j = x_j = \max(0, \underbrace{\sum_i x_i w_{ij}}_{\leq 0} + b_j) = 0$$

and there is no relevance to redistribute to the lower layers.

$z^b$ -rule  $\mathcal{X} = \mathcal{B}$ 

Given a bounded input space  $\mathcal{B} = \{\{x_i\} : \forall_{i=1}^d l_i \leq x_i \leq h_i\}$ , with  $l_i \leq 0$  and  $h_i \geq 0$  and we search on the segment  $(\{l_i 1_{w_{ij}>0} + h_i 1_{w_{ij}<0}\}, \{x_i\}) \subset \mathcal{B}$

- ① Search direction  $\{v_i\}^{(j)} = x_i - l_i 1_{w_{ij}>0} - h_i 1_{w_{ij}<0}$
- ② If  $\{x_i\} \in \mathcal{B}$  so is the whole domain and for  $w_{ij}^- = \min(0, w_{ij})$ ,  
 $w_{ij}^+ = \max(0, w_{ij})$

$$\begin{aligned} R_j(\{l_i 1_{w_{ij}>0} + h_i 1_{w_{ij}<0}\}) &= \max(0, \sum_i l_i 1_{w_{ij}>0} w_{ij} + h_i 1_{w_{ij}<0} w_{ij} + b_j) \\ &= \max(0, \sum_i \underbrace{l_i w_{ij}^+}_{\leq 0} + \underbrace{h_i w_{ij}^-}_{\leq 0} + b_j) = 0 \end{aligned}$$

- ③ Inject search direction in equation (4)

$z^b$ -rule  $\mathcal{X} = \mathcal{B}$ 

Given a bounded input space  $\mathcal{B} = \{\{x_i\} : \forall_{i=1}^d l_i \leq x_i \leq h_i\}$ , with  $l_i \leq 0$  and  $h_i \geq 0$  and we search on the segment  $(\{l_i 1_{w_{ij}>0} + h_i 1_{w_{ij}<0}\}, \{x_i\}) \subset \mathcal{B}$

- 1 Search direction  $\{v_i\}^{(j)} = x_i - l_i 1_{w_{ij}>0} - h_i 1_{w_{ij}<0}$
- 2 If  $\{x_i\} \in \mathcal{B}$  so is the whole domain and for  $w_{ij}^- = \min(0, w_{ij})$ ,  
 $w_{ij}^+ = \max(0, w_{ij})$

$$\begin{aligned}
 R_j(\{l_i 1_{w_{ij}>0} + h_i 1_{w_{ij}<0}\}) &= \max(0, \sum_i l_i 1_{w_{ij}>0} w_{ij} + h_i 1_{w_{ij}<0} w_{ij} + b_j) \\
 &= \max(0, \sum_i \underbrace{l_i w_{ij}^+}_{\leq 0} + \underbrace{h_i w_{ij}^-}_{\leq 0} + b_j) = 0
 \end{aligned}$$

- 3 Inject search direction in equation (4)

 $z^b$ -rule

$$R_i = \sum_j \frac{x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-}{\sum_{i'} x_{i'} w_{i'j} - l_{i'} w_{i'j}^+ - h_{i'} w_{i'j}^-} R_j$$

$z^b$ -rule  $\mathcal{X} = \mathcal{B}$

### Proposition 3

For all function  $g \in G$  and data points  $\{x_i\} \in \mathcal{B}$ , the deep Taylor decomposition with the  $z^b$ -rule is consistent.

### Proof

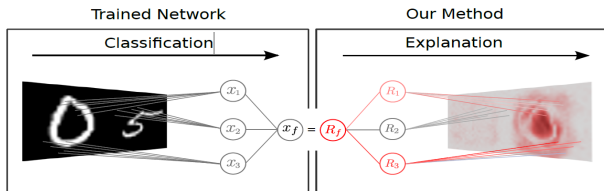
Since the proof is similar to the proofs of proposition 1 and 2 but lengthy I refer to the literature.



# Example MNIST: Setting

Training of a neural network to detect a handwritten digit between 0-3 next to a distracting digit from 4-9 given the following setting:

- Images of size  $28 \times 56$  pixels and 1568 input neurons  $\{x_i\}$
- One hidden layer with 400 neurons  $\{x_j\}$  and one output  $x_k$
- Random initialized weights  $\{w_{ij}\}$  and non-negative bias  $\{b_j\}$  initialized to zero
- Training with 300000 iterations of stochastic gradient descent with a batch size of 20



# Example MNIST: Heatmaps

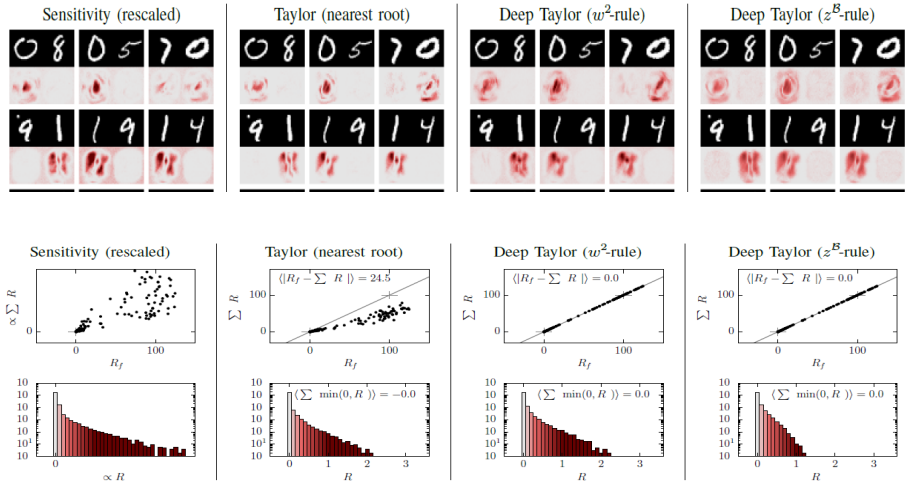


Figure 2: Heatmap and Empirical Results of Consistency

# Table of Contents

- 1 Introduction
- 2 Pixel-Wise Decomposition
  - Mathematical Framework and Definitions
  - Taylor Expansion and Sensitivity Analysis
- 3 Application to One-Layer Networks and Root Finding
  - $\omega^2$ -rule
  - $z^+$ -rule
  - $z^b$ -rule
  - Example MNIST
- 4 Application to Deep Networks
- 5 Python Programming Example

# Deep Networks

Many problems require very complex deep architectures

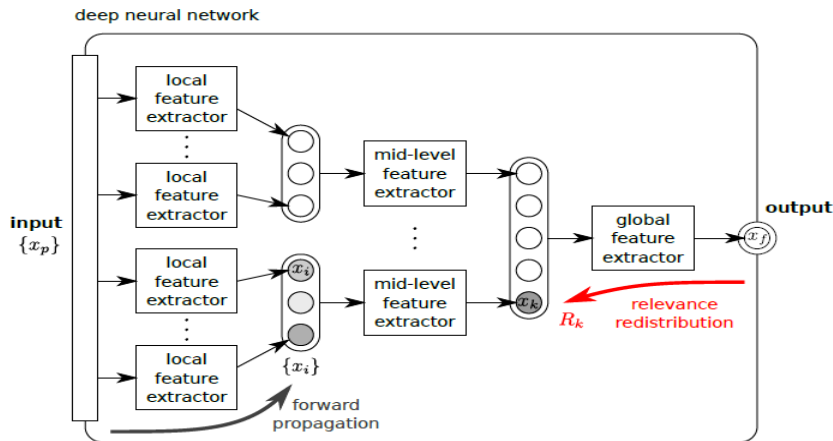


Figure 3: Example Deep Network

# Min-Max Relevance Model

Trainable relevance model defined as

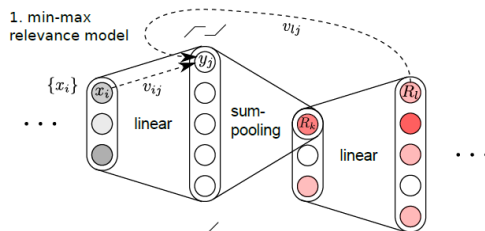
$$y_j = \max(0, \sum_i x_i v_{ij} + a_j)$$

$$\hat{R}_k = \sum_j y_j,$$

where  $a_j = \min(0, \sum_l R_l v_{lj} + d_j)$  is a negative bias

→ Compute  $\{v_{ij}, v_{lj}, d_j\}$  by minimizing

$$\min \langle (\hat{R}_k - R_k)^2 \rangle$$



# Min-Max Relevance Model

Due to the similar structure we can apply the propagation rules for the one-layer neural network

- Pooling layer

$$R_j = y_j$$

- Detection layer

$$R_i = \sum_j \frac{q_{ij}}{\sum_{i'} q_{i'j}} R_j$$

where  $q_{ij} = v_{ij}^2$ ,  $q_{ij} = x_i v_{ij}^+$  or  $q_{ij} = x_i v_{ij} - l_i v_{ij}^+ - h_i v_{ij}^-$  for the  $w^2$ -rule,  $z^+$ -rule and the  $z^b$ - rule respectively

→ The Min-Max relevance model is due to the minimization only approximately consistent

# Training-Free Relevance Model

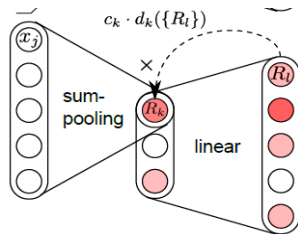
Consider the original network structure

$$x_j = \max(0, \sum_i x_i w_{ij} + b_j)$$

$$x_k = \|\{x_j\}\|_p$$

If the upper layer was explained by the  $z^+$ -rule, relevance  $R_k$  can be written as

$$\begin{aligned} R_k &= \sum_l \frac{x_k w_{kl}^+}{\sum_{k'} x_{k'} w_{k'l}^+} R_l \\ &= \left( \sum_j x_j \right) \cdot \frac{\|\{x_j\}\|_p}{\|\{x_j\}\|_1} \sum_l \frac{w_{kl}^+ R_l}{\sum_{k'} x_{k'} w_{k'l}^+} \end{aligned}$$



# Training-Free Relevance Model

As before we can apply the propagation rules for the one-layer neural network

- Pooling layer

$$R_j = \frac{x_j}{\sum_{j'} x_{j'}} R_k$$

- Detection layer

$$R_i = \sum_j \frac{q_{ij}}{\sum_{i'} q_{i'j}} R_j$$

where  $q_{ij} = w_{ij}^2$ ,  $q_{ij} = x_i w_{ij}^+$  or  $q_{ij} = x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-$  for the  $w^2$ -rule,  $z^+$ -rule and the  $z^b$ -rule respectively

→ The training-free relevance model is consistent

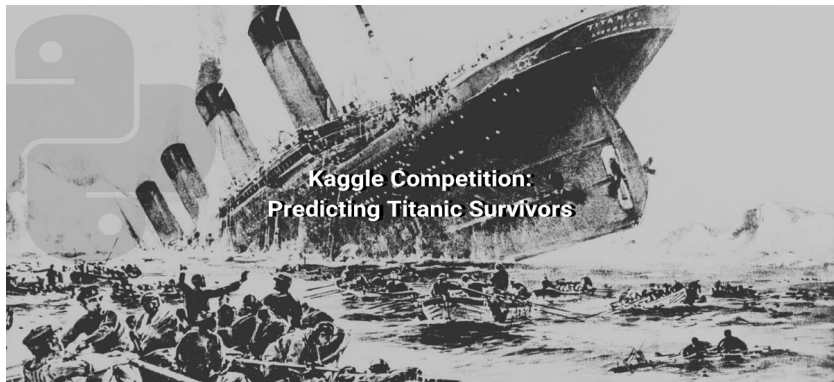
When using the training-free model for the whole network all but the first layer need to be decomposed using the  $z^+$ -rule



# Table of Contents

- 1 Introduction
- 2 Pixel-Wise Decomposition
  - Mathematical Framework and Definitions
  - Taylor Expansion and Sensitivity Analysis
- 3 Application to One-Layer Networks and Root Finding
  - $\omega^2$ -rule
  - $z^+$ -rule
  - $z^b$ -rule
  - Example MNIST
- 4 Application to Deep Networks
- 5 Python Programming Example

# Relevance Distribution on the Titanic Dataset



<https://github.com/mpommer/Deep-Taylor-Decomposition-Python>

# References I

John P. Eaton, Charles A. Haas, and John Maxtone-Graham. *Titanic: Triumph and tragedy*. 2nd ed. Nr. Yeovil, Somerset: Patrick Stephens, 1998. ISBN: 185260493X.

Yann LeCun, Corinna Cortes, and CJ Burges. “MNIST handwritten digit database”. In: *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist> 2 (2010).

Maximilian Alber et al. “iNNvestigate Neural Networks!” In: *Journal of Machine Learning Research* 20.93 (2019), pp. 1–8. URL: <http://jmlr.org/papers/v20/18-540.html>.

Grégoire Montavon et al. “Explaining NonLinear Classification Decisions with Deep Taylor Decomposition”. In: *Pattern Recognition* 65.2 (2017), pp. 211–222. ISSN: 00313203. DOI: 10.1016/j.patcog.2016.11.008. URL: <http://arxiv.org/pdf/1512.02479v1>.

# References II

Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. ISSN: 0920-5691. DOI: [10.1007/s11263-015-0816-y](https://doi.org/10.1007/s11263-015-0816-y).