Machine Learning Journal Club

November 28, 2019

Paper Discussion:

“Methods for interpreting and understanding deep neural networks”

G. Montavon, W. Samek, and K.-R. Müller, Feb. 2018.

Sunanda Gamage

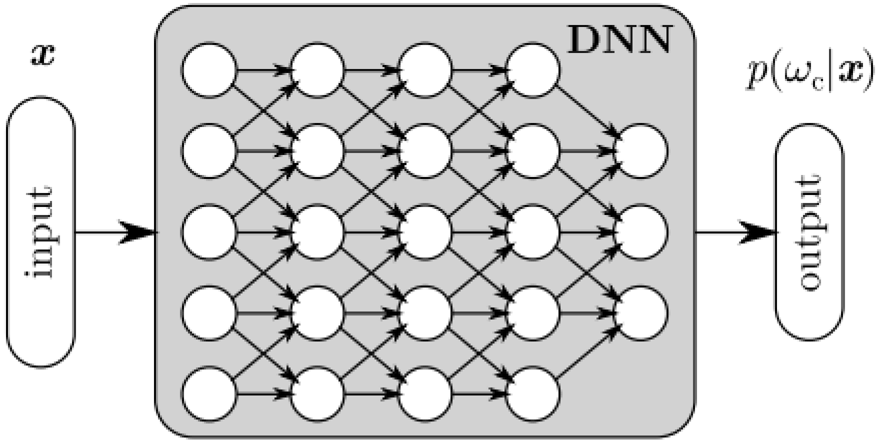
November 28, 2019

Outline

- The problem of interpretability in neural nets. Why is interpretability important?
- Definitions and types of interpretations
- Type 1: Interpretation of a DNN model: prototype
  - Method: Activation maximization (AM)
- Type 2: Explanation of a DNN prediction
  - Method 1: Sensitivity analysis
  - Method 2: Taylor decomposition based
  - **Method 3: Layer-wise relevance propagation (LRP)**

The problem of interpretability in deep neural networks

Deep neural network (DNN)



$f(\boldsymbol{x})$

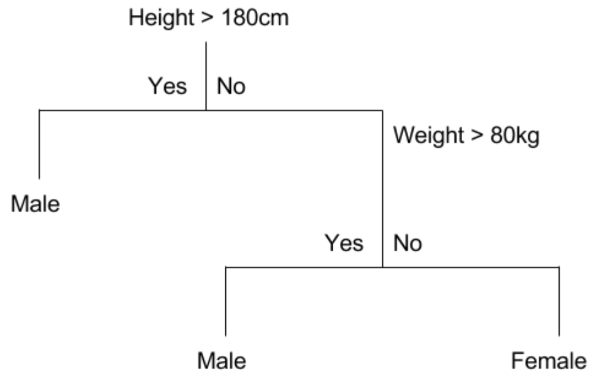
complex nonlinear function
(a composition of nonlinear functions)

vs.

Linear regression

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$$

Tree-based models/ graphical models

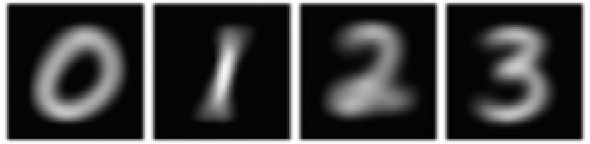


Why is interpretability important?

- **Trust**
  - Is the model relying on correct features, instead if statistical artifacts?
  - Esp. for critical applications like medicine or self-driving cars
  - Legal and ethical concerns
- **Insights to understand causality**
  - Extract new insights from complex physical, chemical, or biological systems
- **Informativeness**
  - Explain the reasons for decisions → informs the human user of the ML system
  - Useful to improve model

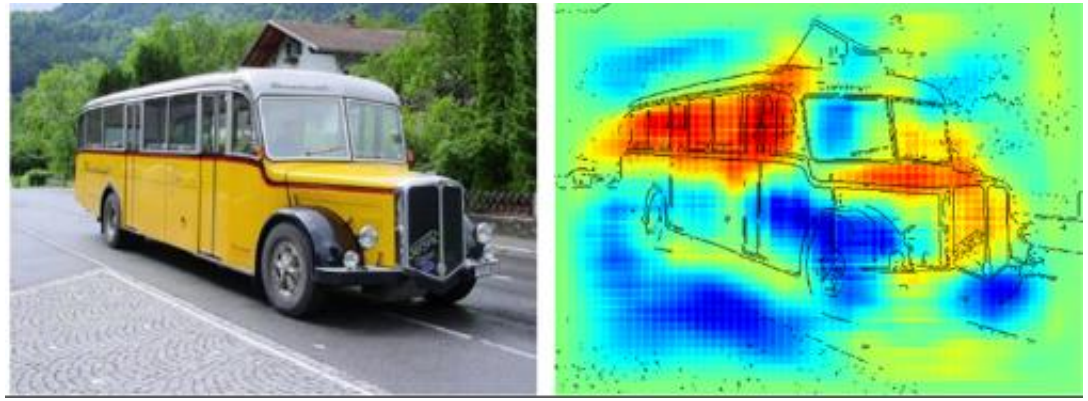
Definitions and types of interpretations

- **Type 1: interpretation [of a modeled concept]**
  - Mapping of the modeled concept (e.g. a class) into a **domain** that the human can make sense of
  - Interpretation = prototype of the class learned by the model, described in the input domain
  - Examples:

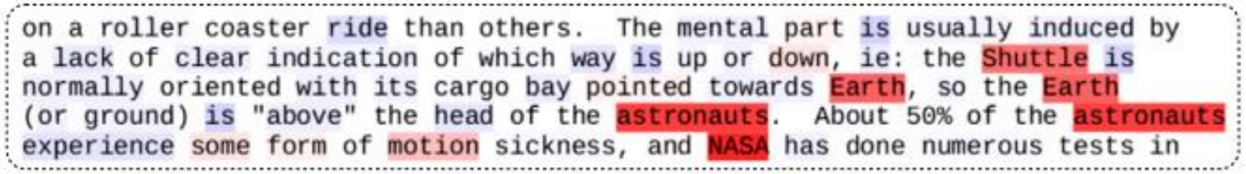


Definitions and types of interpretations

- **Type 2: explanation [of a prediction by the model]**
  - Explanation = which features of the input have contributed the most to the prediction?
    - Or, explanation = what's the significance level of each feature?
  - Visualization: heatmap(red = relevant, blue = negatively relevant [evidence against the class])



Why does the model classify this picture as a "bus"?

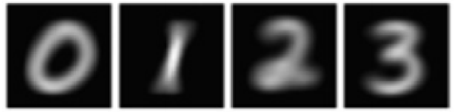


on a roller coaster ride than others. The mental part is usually induced by a lack of clear indication of which way is up or down, ie: the Shuttle is normally oriented with its cargo bay pointed towards Earth, so the Earth (or ground) is "above" the head of the astronauts. About 50% of the astronauts experience some form of motion sickness, and NASA has done numerous tests in

Why does the model classify this document as a "science.space"?

Type 1: Interpretation (prototype) of a model

- **Method: Activation maximization (AM)** (2006)



  - Simple AM

$$\text{prototype } \boldsymbol{x}^\star = \max_{\boldsymbol{x}} \underbrace{\log p(\omega_c|\boldsymbol{x})}_{f(\boldsymbol{x})}$$

Which input will cause the model to give peak output (highest probability for the class)?

  - AM + expert

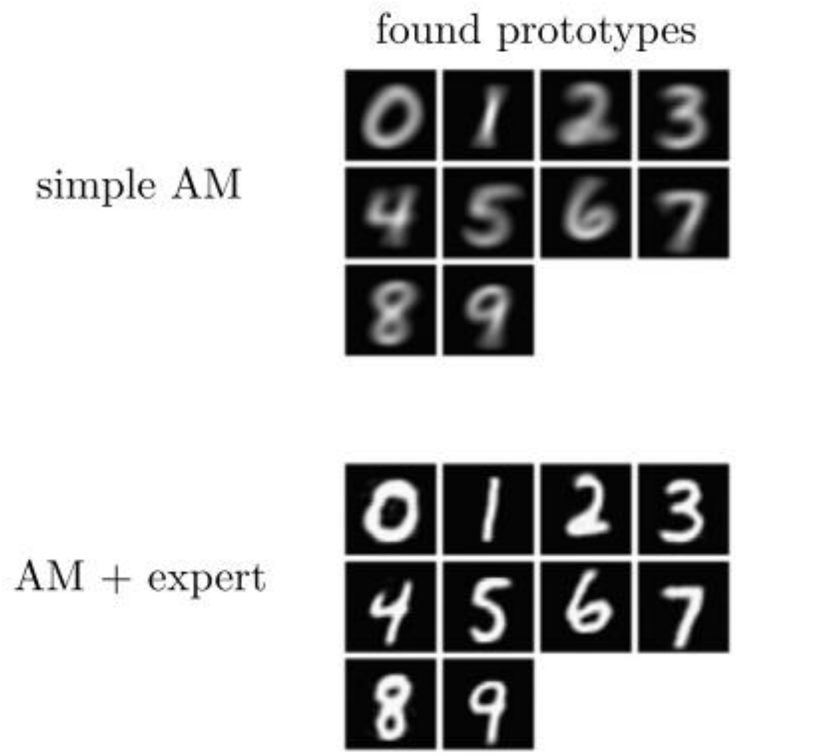
$$\text{prototype } \boldsymbol{x}^\star = \max_{\boldsymbol{x}} \log p(\omega_c|\boldsymbol{x}) + \log p(\boldsymbol{x})$$

Which input, *that also looks like other input*, will cause the model to give peak output?

  - Can solve using a gradient-decent like optimization algorithm

Type 1: Interpretation (prototype) of a model

- **Method: Activation maximization (AM)** (2006)

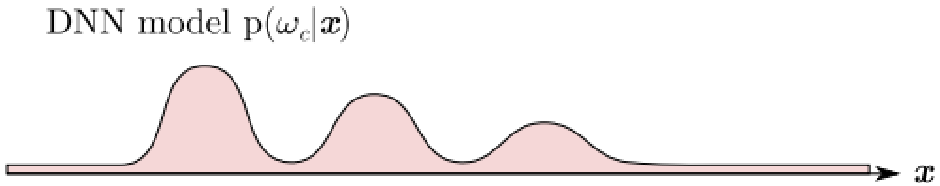


Type 1: Interpretation (prototype)

- **Problem 1 – computationally expensive**

$$\boldsymbol{x}^\star = \max_{\boldsymbol{x}} f(\boldsymbol{x})$$

- **Problem 2- a single protype may not exist**
  - Probability distributions $p(\omega_c|\boldsymbol{x})$ and $p(\boldsymbol{x})$ might be **<u>multimodal</u>** or strongly **<u>elongated</u>**
  - So that no single prototype $\boldsymbol{x}^*$ fully represents the modeled class

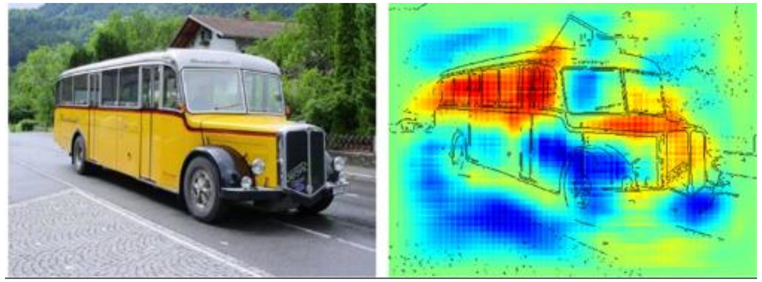


Prototype of an ostrich learnt by a ConvNet, trained on the ILSVRC-2013 dataset

ostrich

Type 2: Explanation of a prediction

- **Better question to ask:**
  - Which features of input **x** are the most significant/ relevant to the prediction?
  - Must assign a relevance score R_i to each feature x_i in the input **x**
  - Additionally, we would like the relevance scores to **fully explain** the function output f(x) (relevance conservation property)
    - i.e. $\sum_{i=1}^{d} R_i(\boldsymbol{x}) = f(\boldsymbol{x})$



- **Method 1: Sensitivity analysis (1990s)**
- **Method 2: Taylor decomposition based (~2013)**
- **Method 3: Layer-wise relevance propagation (LRP) (2015)**

Method 1: Sensitivity analysis

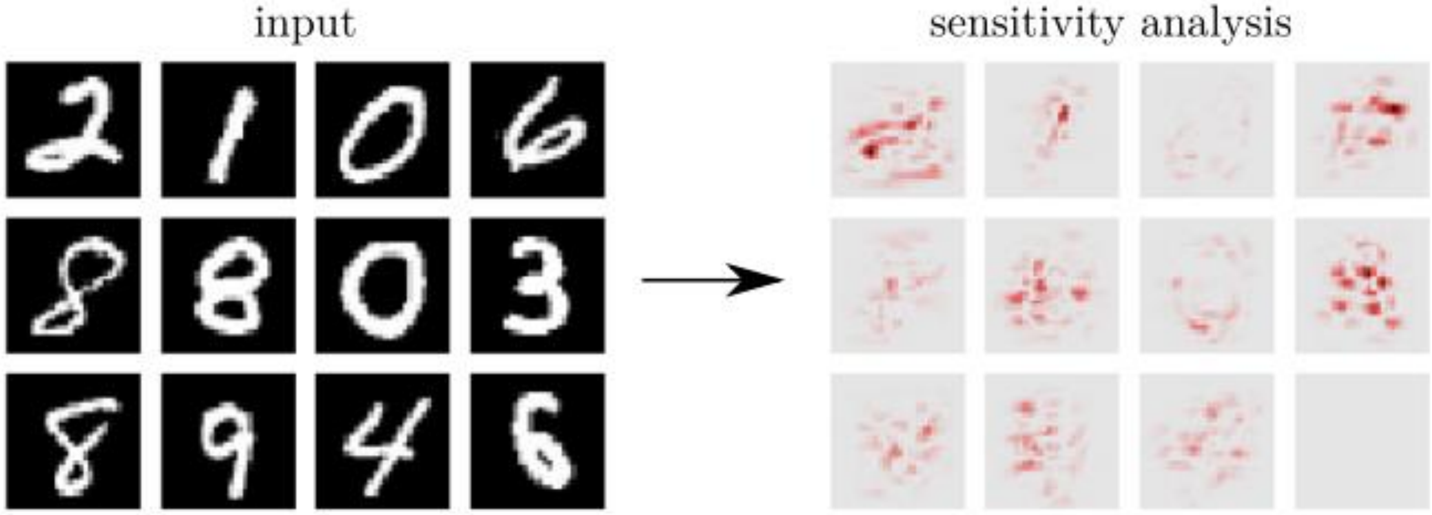
$$R_i(\boldsymbol{x}) = \left(\frac{\partial f}{\partial x_i}\right)^2$$

- The most relevant input features are those to which the output is <u>most sensitive</u>
- Note that

  $\sum_{i=1}^{d} R_i(\boldsymbol{x}) = \|\nabla f(\boldsymbol{x})\|^2$

  - i.e. sensitivity is an explanation of the gradient (local slope) of f(x), but not the function f(x)

Method 1: Sensitivity analysis



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

- **Pros**
  - Easy to implement: the gradient can be computed using backpropagation in deep nets
- **Cons**
  - Heatmaps are spatially discontinuous and scattered
  - Does not provide an explanation of the function f(x), but of its <u>local slope</u>
    - Highlights the pixels that make the digit belong <u>more or less</u> to the target class

Method 2: Taylor decomposition based

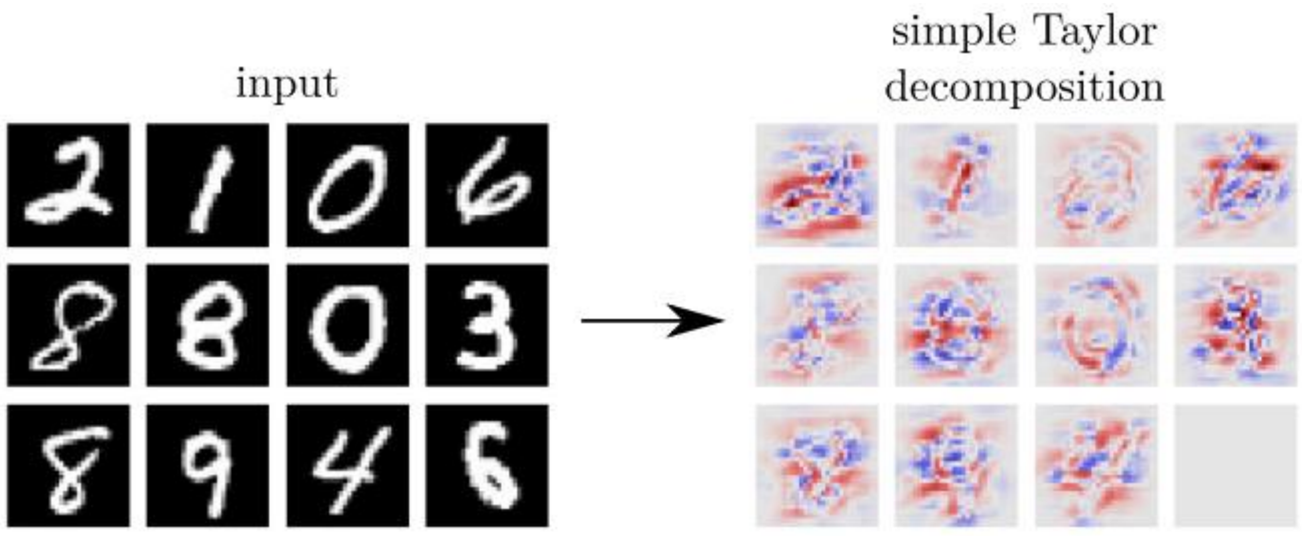
- First order Taylor decomposition of f(x) around the root

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} R_i(\boldsymbol{x})$$

where the relevance scores simplify to

$$R_i(\boldsymbol{x}) = \frac{\partial f}{\partial x_i} \cdot x_i .$$

red = relevant
blue = negatively relevant

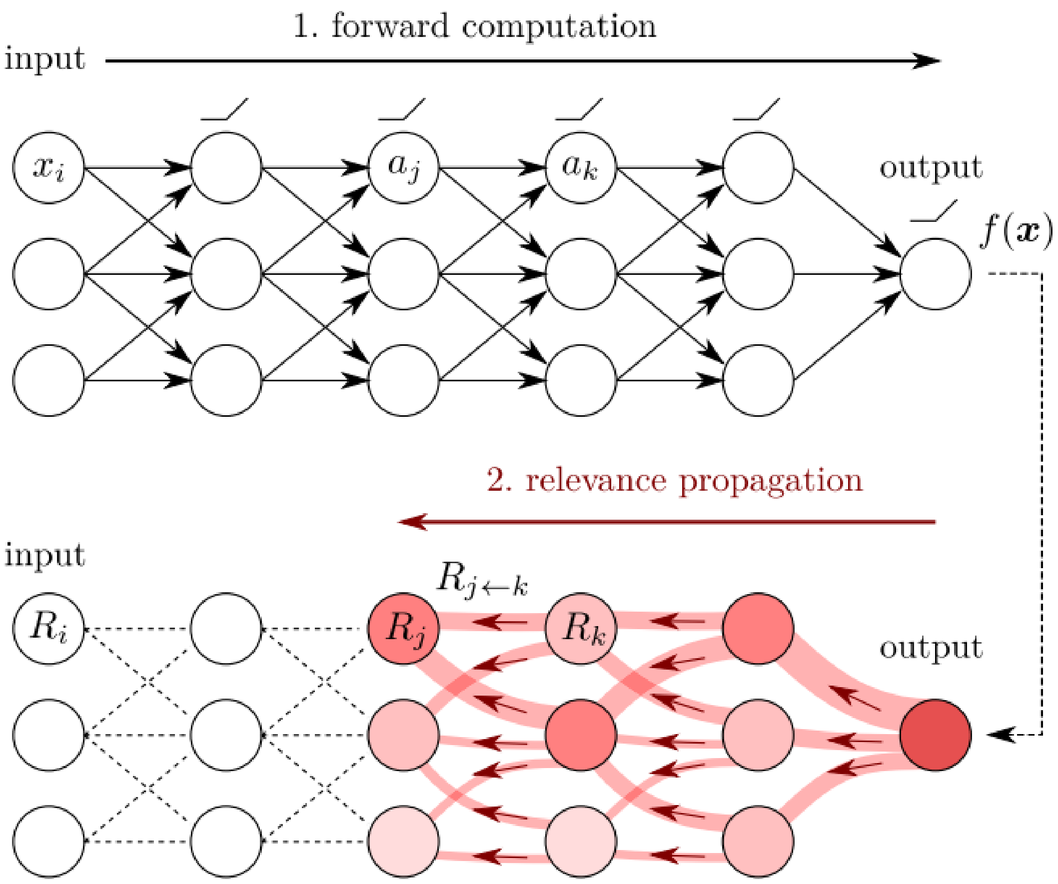


aka, saliency maps

Method 2: Taylor decomposition based

- Pros
  - By definition, the relevance scores explain the function f(x). $\sum_{i=1}^{d} R_i(\boldsymbol{x}) = f(\boldsymbol{x})$
  - More complete heatmaps than sensitivity analysis
- Cons
  - Unusually high amount of negative relevance
    - Because the root image is too dissimilar from the actual images ***x***

Method 3: Layer-wise relevance propagation (LRP)



Relevance backpropagation rule

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

LRP general rule for fully connected

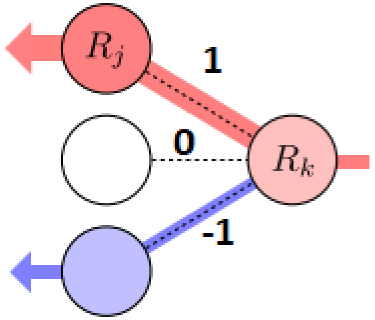
$$R_j = \sum_k \Bigg(\underbrace{\alpha \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+}}_{\text{positive relevance}} - \underbrace{\beta \frac{a_j w_{jk}^-}{\sum_j a_j w_{jk}^-}}_{\text{negative relevance (evidence against)}} \Bigg) R_k,$$

$\Downarrow$

Special case: LRP-$\alpha_1\beta_0$

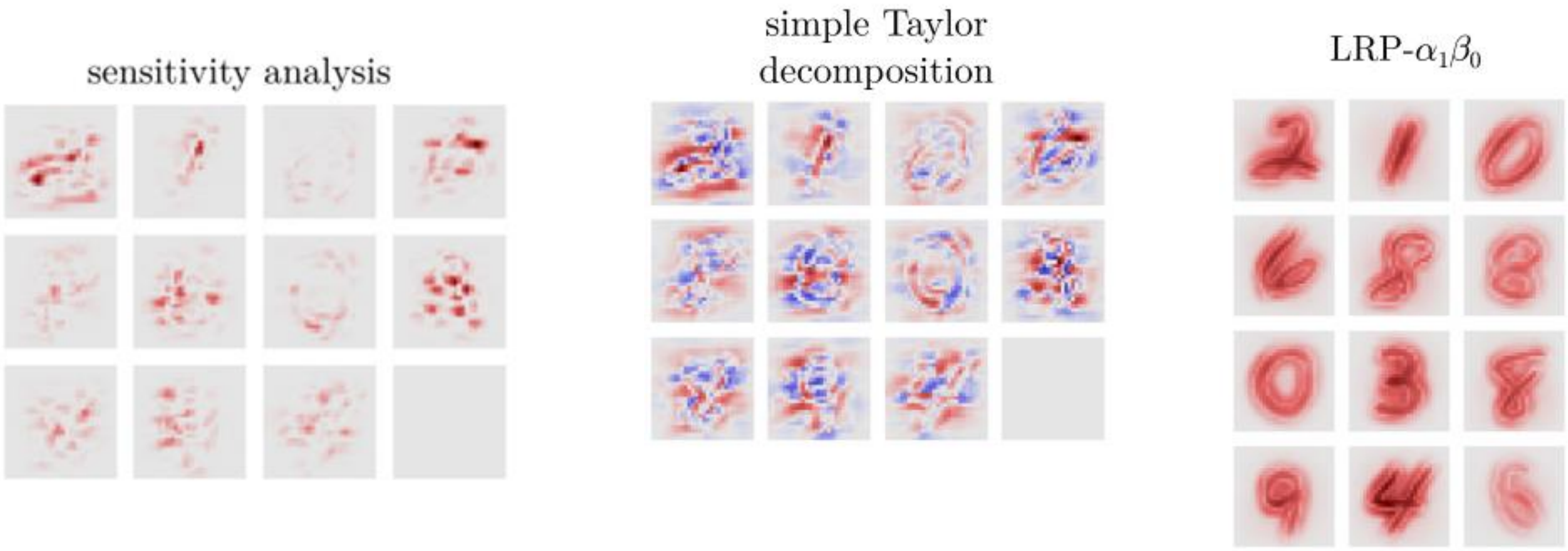
$$R_j = \sum_k \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} R_k.$$

LRP-$\alpha_2\beta_1$



- Other backprop rules for different types of layers (max-pooling, convolution, LSTM)

LRP vs. other methods



LRP pros and cons

- **Pros**
  - Principled approach: derives from Taylor decomposition applied to graph structures
  - Conservation property holds for all layers (every layer fully explains the next layer)

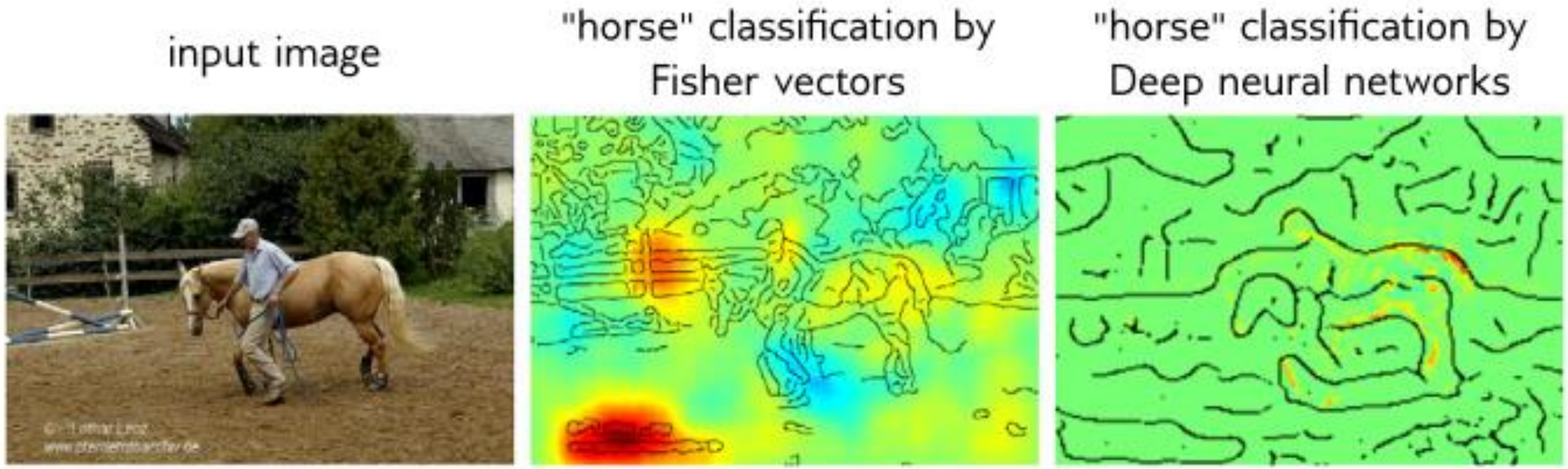
$$\sum_{i=1}^{d} R_i = \cdots = \sum_j R_j = \sum_k R_k = \cdots = f(\boldsymbol{x})$$

  - Continuous (smooth) heatmaps
  - High selectivity
    - When highly relevant features/ pixels are destroyed, model accuracy goes down drastically
  - LRP rules can be derived for other ML models too (eg: SVM)
  - Easy to implement, not computationally expensive

- **Cons**
  - Greedy procedure (“layer-wise”)
    - But the method seems to work well enough in practice

LRP applications

- **Model validation**
  - Check whether the model is focusing on meaningful features instead of statistical artifacts



Model is focusing on copyright text at bottom left

Model is focusing on pixels on the horse

LRP applications

- **Model validation**
  - Check whether the model is focusing on meaningful features instead of statistical artifacts

LRP heatmap Raw pixels of the screen

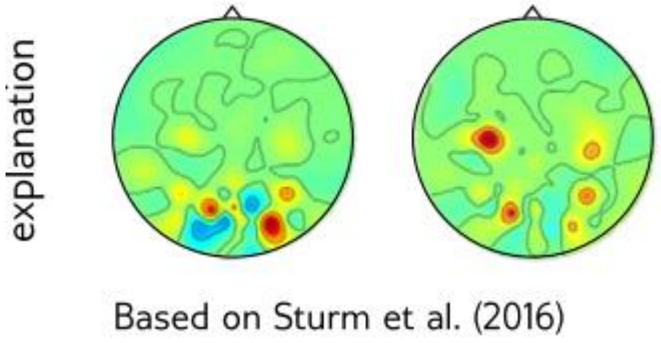
LRP applications

- **Improving a model based on LRP heatmaps**
  - If artifacts are present, remove them and retrain model
  - If the model is relying on too many input features, retrain with a sparsity penalty
  - Check relevance heatmaps as the training progresses
  - Feature engineering
    - If too many input features are irrelevant, remove them: a form of feature selection
    - Design new feature representations based on relevance
  - Do LRP for incorrect predictions, and understand what features are driving the incorrect decision
  - **Relevance analysis will likely strengthen the results/ claims in your papers!**

LRP applications

- **Analysis of scientific data**
    - Shed light on scientific problems where human intuition and domain knowledge is limited

EEG probe locations on skull



right hand foot

Which areas of the brain are responsible for different “movement thoughts”?

Papers on interpretability

- Paper presented:
  - "Methods for interpreting and understanding deep neural networks", G. Montavon, W. Samek, and K.-R. Müller, Feb. 2018.
- Activation Maximization, sensitivity analysis papers:
  - Erhan, Dumitru, et al. "Visualizing higher-layer features of a deep network."
  - Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps."
- LRP papers:
  - Montavon, Grégoire, et al. "Layer-wise relevance propagation: an overview."
  - Bach, Sebastian, et al. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation."
- Other interpretability papers:
  - Yosinski, Jason, et al. "Understanding neural networks through deep visualization."
  - Olah, Chris, et al. "The building blocks of interpretability." *Distill* 3.3 (2018)

Questions and Discussion

- Do you do relevance analysis in your models/ papers?
- Or any other kind of interpretations to validate the models?