Relevance Propagation for Deep Neural Networks

Zwischenvortrag 3

Theo Conrads, Robin Kühling, Marc Bremser

12. Juli 2020

Überblick

Section 1

Nähere Analyse der Implementierungsergebnisse

Wiederholung

- **Konservierung**
  Eine LRP-Regel ist **konservativ** genau dann, wenn für die Relevanzwerte der Inputschicht und jeden Input x gilt:

$$\sum_i R_i(x) = f(x)$$

- **Positivität**
  Eine LRP-Regel ist **positiv** genau dann, wenn gilt:

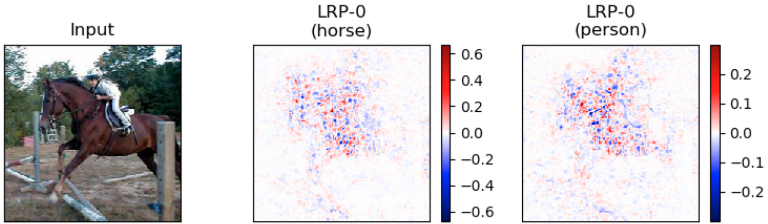
$$\forall x, i \quad R_i \geq 0$$

- **Konsistenz**
  Eine LRP-Regel ist **konsistent** genau dann, wenn sie **konservativ** und **positiv** ist.

LRP-0

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_i a_i w_{ik}} R_k$$

- **konservativ:** ✓[1]
- **positiv:** X
- **konsistent:** X



[1] Gilt hier und im Folgenden nur unter der Vorussetzung, dass der Bias nicht hinzuaddiert wird. Alle Bilder des Kapitels wurden unter Verwendung des Bias erstellt.

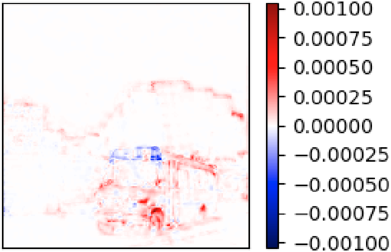
LRP-ϵ

$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_i a_i w_{ik}} R_k$$

- **konservativ: X**
- **positiv: X**
- **konsistent: X**

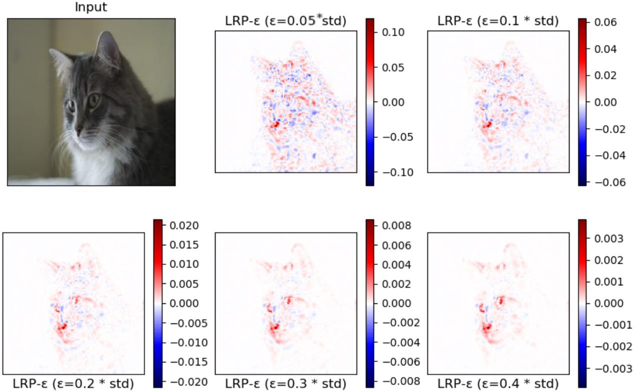
Input

LRP-ε (ε=0.3 * std)
(bus)



LRP-ϵ - Abhängigkeit von ϵ

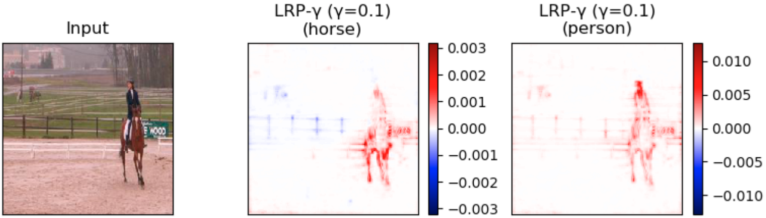
- Mit wachsendem ϵ verteilt sich die zurückgegebene Relevanz auf weniger Pixel
- Das Rauschen der Relevanz nimmt ab und Konturen werden deutlicher



LRP-γ

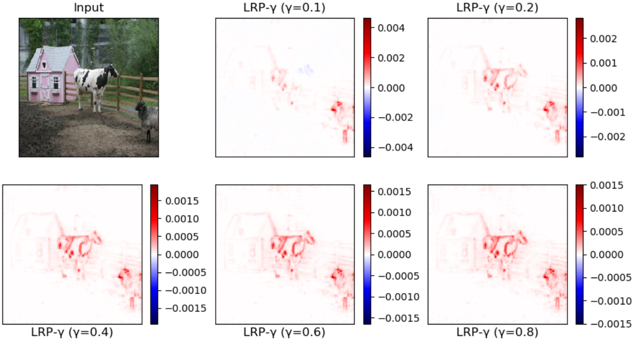
$$R_j = \sum_k \frac{a_j \cdot \left(w_{jk} + \gamma \cdot w_{ij}^+\right)}{\sum_i a_i \cdot \left(w_{jk} + \gamma \cdot w_{ij}^+\right)} R_k$$

- **konservativ:** ✓
- **positiv:** **X**
- **konsistent:** **X**



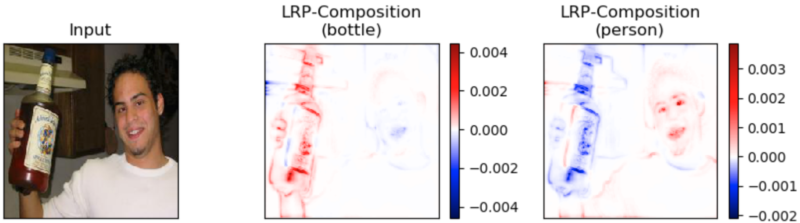
LRP-γ - Abhängigkeit von γ

- Mit wachsendem γ verlieren negative Relevanzen an Wert
- Bildbereiche, die nicht zum klassifizierten Objekt gehören, werden "relevanter"
- Es gilt LRP-$\gamma \xrightarrow{\gamma \to \infty} z^+$-Regel



LRP-Komposition

- Kombinierte Anwendung der vorherigen drei Regeln
- Aus den Eigenschaften der einzelnen Regeln folgt, dass die Komposition weder positiv, noch konservativ ist.
- Subjektiv betrachtet liefert diese Regel die interpretierbarsten Visualisierungen.



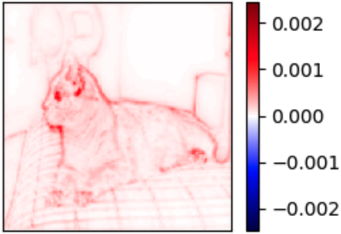
z^+-Regel

$$R_j = \sum_k \frac{a_j \cdot w_{ij}^+}{\sum_i a_i \cdot w_{ij}^+} R_k$$

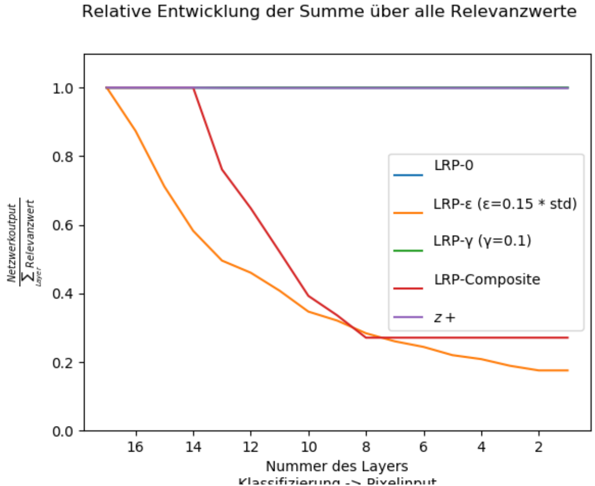
- **konservativ:** ✓
- **positiv:** ✓
- **konsistent:** ✓

Input

z+
(cat)



Übersicht Konservierung

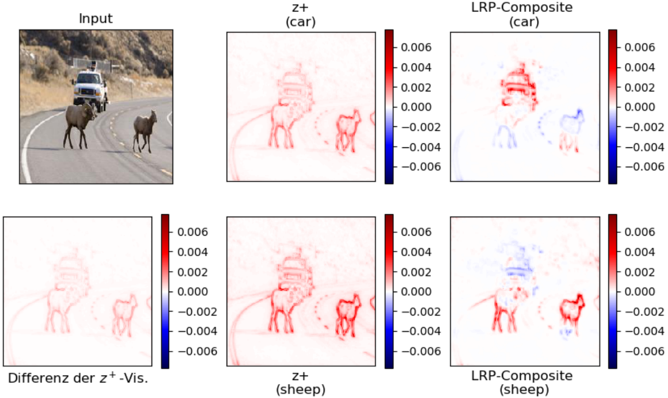


Section 2

Nähere Analyse der z^+-Regel

Beobachtung

- Die visualisierten Erklärungen für zwei verschiedene Klassen auf einem Bild ähneln sich sehr stark.
- Bereits im drittletzten Dense-Layer wurde die Relevanz für beide Klassifikationen auf die gleichen Neuronen verteilt.



Vermutung

- Negative Gewichte tragen stark zur Klassifizierung eines Objektes bei.
- Durch das Ignorieren der Gewichte geht Information verloren.
- Markante Features, die nicht oder negativ zur Klassifizierung beitragen, werden in der Backpropagation nicht gehemmt.

Abbildung: Implementierung der z^+-Regel durch Tool des Fraunhofer Instituts, angewendet auf Klassifizierung als *Motorroller* und *Motorradhelm*[2].

[2] https://lrpserver.hhi.fraunhofer.de/image-classification

Quellen

- Quellen für Bilder, Implementierungshinweise:
- Montavon, Binder, Lapuschkin, Samek, Müller :
  "Layer-Wise Relevance Propagation: An Overview"
  Gefunden auf:
  $\rightarrow$ `http://iphome.hhi.de/samek/pdf/MonXAI19.pdf`

Quellen

- Quellen für Bilder, Implementierungshinweise:
- Montavon et al. :
  "Explaining nonlinear classification decisions with deep Taylor decomposition"
  Version mit Appendix, gefunden unter:
  → `https://arxiv.org/pdf/1512.02479v1.pdf`

Quellen

- Quellen zum weiteren Verständnis:
- Montavon:
  "Deep Taylor Decomposition, Conference Talk"
  Gefunden auf:
  $\rightarrow$
  `https://www.youtube.com/watch?v=gy_Cb4Do_YE&t=939s`

- Montavon, Samek, Müller:
  "Methods for interpreting and understanding deep neural networks"
  Gefunden auf:
  $\rightarrow$ `https://doi.org/10.1016/j.dsp.2017.10.011`