LRP für CNNs

- Basisformel

$$R_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_{i'} z_{i'j}} R_j^{(l+1)} \quad \text{mit} \quad \mathrm{z_{ij}} = \mathrm{x_i^{(1)} w_{ij}^{(\mathit{l},\mathit{l}+1)}}$$

- Problematisch bei komplexeren Layern (*Beispiel: ConvLayer*)
- Betrachte alternative Implementierung

LRP für CNNs

- Algorithmus [QUELLE]

$$
\begin{aligned}
&\forall_k : z_k = \textstyle\sum_{0,j} x_j \cdot \rho\left(w_{jk}\right) && \text{(Forward Pass)}\\
&\forall_k : s_k = R_k / z_k && \text{(Elementweise Division)}\\
&\forall_j : c_j = \textstyle\sum_k \rho\left(w_{jk}\right) \cdot s_k && \text{(Backward Pass)}\\
&\forall_j : R_j = x_j c_j && \text{(Elementweises Produkt)}
\end{aligned}
$$

- Ergebnis identisch
- Was ist der Vorteil dieser neuen Reihenfolge?

LRP für CNNs

- Schritt 3 kann auch als Gradient ausgedrückt werden:

$$\forall_j : c_j = \sum_k \rho\left(w_{jk}\right) \cdot s_k \quad \text{(Backward Pass)},$$

- denn:

$$\begin{aligned} c_j &= \left[\nabla\left(\sum_k z_k(\boldsymbol{x}) \cdot s_k\right)\right]_j \\ &= \left[\nabla\left(\sum_k \left(\sum_{j'} (\boldsymbol{x_{j'}} \cdot w_{j'k})\right) \cdot s_k\right)\right]_j \\ &= \left[\sum_k w_{jk} \cdot s_k\right]_j , \end{aligned}$$

  wobei s_k als Konstante behandelt wird.
- Implementierung ist nun unabhängig der Art des Layers anwendbar.