Relevance Propagation for Deep Neural Networks

Zwischenvortrag 3

Theo Conrads, Robin Kühling, Marc Bremser

29. Juni 2020

Überblick

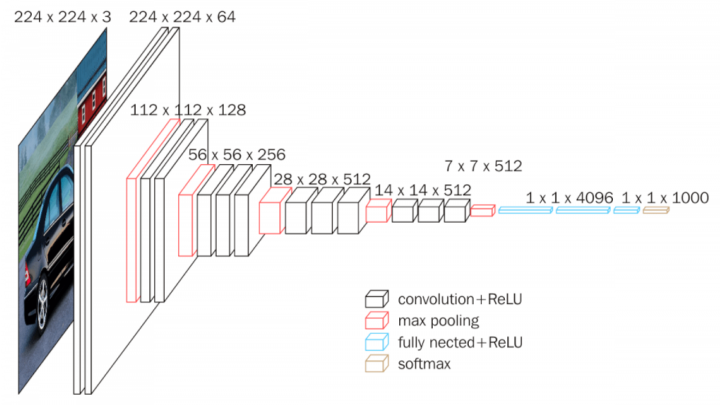
Aufgaben

1. Arbeit an einem CNN für den Pascal VOC 2012 Datensatz fortsetzen
2. Implementierung des Ansatz der Deep Taylor Decomposition für DNN (insbesondere CNN)
3. Vergleich LRP $\leftrightarrow$ Deep Taylor Decomposition

Section 2

Implementierung eines CNN für den Pascal VOC

Das VGG-Modell



Finetuning

Wir entfernen die Dense Layer aus dem bereits trainierten VGG16 und trainieren diese neu

```
if model_name == 'vgg16_finetuned':
    vgg16_model = VGG16()
    model = Sequential()
    for layer in vgg16_model.layers[:-1]:
        if type(layer) != Dense:
            model.add(layer)
    for layer in model.layers:
        layer.trainable=False
    model.add(BatchNormalization())
    model.add(Dense(4096, activation='relu'))
    model.add(Dropout(0.25))
    model.add(BatchNormalization())
    model.add(Dense(2048, activation='relu'))
    model.add(Dropout(0.25))
    model.add(BatchNormalization())
    model.add(Dense(output_shape))
    model.add(Activation(final_activation))
```

Regularisierung

- Hier besoners wichtig weil:
  1. der Datensatz klein ist
  2. die Klassen ungleich viele Bilder enthalten
- Methoden:
  1. Dropout
  2. BatchNormalization
  3. Data Augmentation
  4. Sample anderer Klassen

Sample anderer Klassen

- Ein NN muss lernen, was zu einer Klasse gehört und was nicht
- Problem: Falsche Entscheidungen für eine Klasse anhand von Merkmalen die häufig mit dieser Klasse auftreten
- Idee: Hinzufügen von Bildern, die keine der zu trainierenden Klassen enhalten

Ein eigener Model Checkpoint

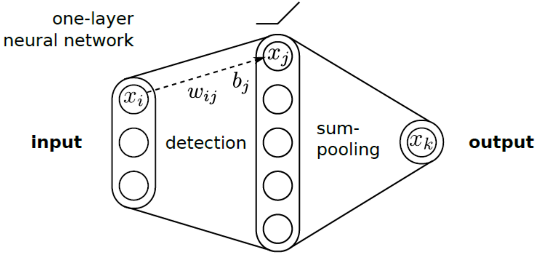
- Idee: Speichere das Model nicht zum Zeitpunkt minimalen Fehlers sondern anhand spezieller Metriken
  1. Precision:
     $$\frac{true\_positives}{true\_positives + false\_positives}$$
     Welcher Anteil positiv vorhergesagter Label war korrekt?
  2. Recall:
     $$\frac{true\_positives}{true\_positives + false\_negatives}$$
     Welcher Anteil positiver Label wurde korrekt vorhergesagt?

Section 3

Taylor Decomposition

Taylor Decomposition - Rückblick

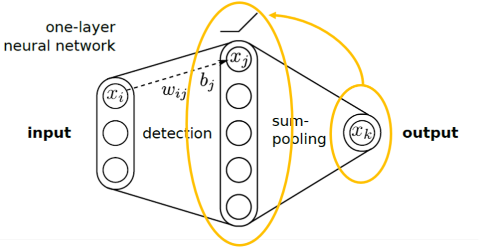
- Einfaches Netzwerk mit einem Hidden Layer, ReLU Aktivierung und Sum-Pooling als Output.
- Zusätzliche Voraussetzung: $b_j \leq 0$.



- Für das Outputneuron x_k gilt: $x_k = max(0, \sum_j x_j)$

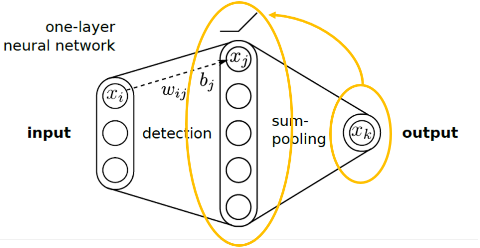
Taylor Decomposition - Rückblick

- Suche eine Nullstelle für die Taylorentwicklung von $R_k(\boldsymbol{x}) = \sum_j x_j$.



Taylor Decomposition - Rückblick

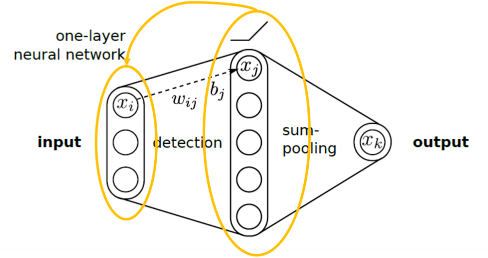
- Suche eine Nullstelle für die Taylorentwicklung von $R_k(\boldsymbol{x}) = \sum_j x_j$.



- Wg. ReLU Aktivierung im vorherigen Layer und $\sum_j x_j \stackrel{!}{=} 0$ ist $\tilde{\boldsymbol{x}} = \boldsymbol{0}$ die einzige Nullstelle von R_k.
- Wegen $R_j = \frac{\partial R_k}{\partial x_j}(x_j - \tilde{x}_j) = 1 \cdot (x_j - 0)$ gilt also
- $R_j = x_j = \max\left(0, \sum_i x_i w_{ij} + b_j\right)$

Taylor Decomposition - Generische Regel

- Es gilt $R_j = x_j = \max\left(0, \sum_i x_i w_{ij} + b_j\right)$



- Unterscheide nun 2 Fälle:
  1. $R_j = 0$: Nicht aktivierte Neuronen sollen keine Relevanz zurückverteilen. Insbesondere gilt hier $\tilde{\boldsymbol{x}} = \boldsymbol{x}$.
  2. $R_j > 0$: Hierfür wird ein Richtungsvektor $\boldsymbol{v}^{(j)}$ definiert. $\tilde{\boldsymbol{x}}$ soll von der Form $\boldsymbol{x} + t \cdot \boldsymbol{v}^{(j)}$, mit $t \in \mathbb{R}$ sein.

Taylor - Entwicklungspunkt

- Allgemeine Vorgehensweise:
- Durch Einsetzen von $\tilde{\boldsymbol{x}} = \boldsymbol{x} + t \cdot \boldsymbol{v}^{(j)}$ in die Ebenengleichung $\sum_i \tilde{x}_i w_{ij} + b_j = 0$ lässt sich eine allgemeine Formel für t finden.
- Somit gilt:

$$0 = \sum_i \left(x_i + t v_i^{(j)} \right) w_{ij} + b_j$$

$$\Leftrightarrow -t = \frac{\sum_i x_i w_{ij} + b_j}{\sum_i v_i^{(j)} w_{ij}}$$

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

Taylor - Entwicklungspunkt

- Für die Umverteilung von der $l+1$-ten Schicht in die l-te Schicht gilt

$$
\begin{aligned}
R_i^l = \sum_j R_{i\leftarrow j}^l &= \sum_j \frac{\partial R_j^{l+1}}{\partial x_i^l}(x_i - \tilde{x}_i) \\
&= \sum_{j:R_j=0} \frac{\partial R_j^{l+1}}{\partial x_i^l} \cdot 0 + \sum_{j:R_j>0} w_{ij} \frac{\sum_i x_i w_{ij} + b_j}{\sum_i v_i^{(j)} w_{ij}} v_i^{(j)} \\
&= \sum_j \frac{v_i^{(j)} w_{ij}}{\sum_i v_i^{(j)} w_{ij}} R_j^{l+1}
\end{aligned}
$$

- $\Rightarrow$ Allgemeine Formel in Abhängigkeit von $v_i^{(j)}$

Section 4

Herleitung zweier z-Regeln

z^+-Regel

- Grundannahme der Deep Taylor Decomposition ist die Anwendung der ReLU-Aktivierungsfunktion.
  $\Rightarrow$ für Input eines Layers gilt $\boldsymbol{x} \in \mathbb{R}_+^d$
  $\Rightarrow$ zulässige Nullstelle sollte auch aus zulässigem Bereich kommen
- Gesucht wird Nullstelle für R_j auf dem Intervall $[\{x_i \mathbb{1}_{w_{ij}<0}\}, \{x_i\}]$
- Mindestens eine Nullstelle existiert bei $\{x_i 1_{w_{ij}<0}\}$

z^+-Regel

- Wähle $v_i^{(j)} = x_i - x_i \cdot \mathbb{1}_{w_{ij} \leq 0} = x_i \cdot \mathbb{1}_{w_{ij} > 0}$
- Einsetzen in die Gleichung liefert:

$$R_i = \sum_j \frac{x_i \mathbb{1}_{w_{ij}>0} w_{ij}}{\sum_i x_i \mathbb{1}_{w_{ij}>0} w_{ij}} R_j^{l+1} = \sum_j \frac{z_{ij}^+}{\sum_i z_{ij}^+} R_j^{l+1}$$

- Mit $z_{ij} = x_i \cdot w_{ij}$

z^B-Regel

- Für Inputwerte des ersten Layers (z.B. Pixelwerte) gilt Positivität i.A. nicht
- Diese sind meistens beschränkt und können auch Werte kleiner 0 annehmen
- Formal: $x \in \mathcal{B}$ mit

$$\mathcal{B} = \{x \in \mathbb{R}^d : l_i \leq x_i \leq h_i \quad \forall i \in \{0, ..., d\}\},$$

wobei $l_i \leq 0$ und $h_i \geq 0$

z^B-Regel

- Gesucht wird Nullstelle für R_j auf dem Intervall

$$\left[\{l_i \mathbb{1}_{w_{ij}>0} + h_i \mathbb{1}_{w_{ij}<0}\}, \{x_i\}\right]$$

- Auf diesem Intervall existiert mindestens eine Nullstelle, denn

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

z^B-Regel

- Dieses Intervall ist also für die Suche zulässig und wir fügen die Richtung $v_i^{(j)}$, mit

$$v_i^{(j)} = x_i - l_i \mathbb{1}_{w_{ij}>0} - h_i \mathbb{1}_{w_{ij}<0},$$

- in die Grundformel

$$\sum_j \frac{v_i^{(j)} w_{ij}}{\sum_i v_i^{(j)} w_{ij}} R_j$$

- ein und erhalten unsere finale $z^{\mathcal{B}}$-Regel

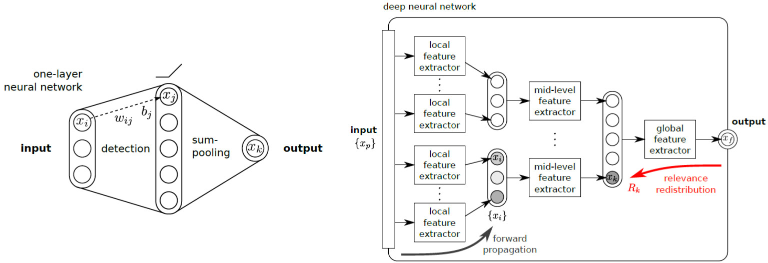
$$R_i = \sum_j \frac{z_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-}{\sum_i z_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-} R_j$$

Section 5

Deep Taylor Decomposition

Erweiterung auf tiefe Netze

- Bei tiefen Netzen, insbesondere Convolutional Layern ist die Relevanzfunktion nicht unbedingt explizit angegeben.



- Ein Feature kann vorhanden sein, aber bei der Bilderkennung keine Rolle spielen

1

[1] Grafik entnommen aus 36

Erweiterung auf tiefe Netze

- Gesucht ist eine Approximation der Relevanz Funktion, die leicht zu analysieren ist.
- Führe das Konzept **Relevanz-Modell** ein, um bei tieferen Netzen die Relevanzfunktion $R_j^{l+1}(\boldsymbol{x}^l)$ zu approximieren.
- Nehme an, die Relevanzfunktion R_j lässt sich in der Form $R_j = x_j \cdot c_j$ schreiben, wobei c_j konstant ist.
- Im Paper als "Training Free" Ansatz vorgestellt

Erweiterung auf tiefe Netze

- Nehme an, die Relevanzfunktion R_j lässt sich in der Form $R_j = x_j \cdot c_j$ schreiben, mit c_j konstant.
- Betrachte die generische Redistributionsregel

$$
\begin{aligned}
R_i &= \sum_j \frac{x_i \cdot \rho(w_{ij})}{\sum_i x_i \cdot \rho(w_{ij})} R_j \\
&= x_i \sum_j \frac{\rho(w_{ij})}{\sum_i x_i \cdot \rho(w_{ij})} x_j \cdot c_j \\
&= x_i \sum_j \rho(w_{ij}) \frac{\max\left(0, \sum_i x_i w_{ij}\right)}{\sum_i x_i \cdot \rho(w_{ij})} c_j
\end{aligned}
$$

Erweiterung auf tiefe Netze

- Nehme an, die Relevanzfunktion R_j lässt sich in der Form $R_j = x_j \cdot c_j$ schreiben, mit c_j konstant.
- Betrachte die generische Redistributionsregel

$$
\begin{aligned}
R_i &= \sum_j \frac{x_i \cdot \rho(w_{ij})}{\sum_i x_i \cdot \rho(w_{ij})} R_j \\
&= x_i \sum_j \frac{\rho(w_{ij})}{\sum_i x_i \cdot \rho(w_{ij})} x_j \cdot c_j \\
&= x_i \underbrace{\sum_j \rho(w_{ij}) \frac{\max\left(0, \sum_i x_i w_{ij}\right)}{\sum_i x_i \cdot \rho(w_{ij})} c_j}_{:\approx c_i}
\end{aligned}
$$

Erweiterung auf tiefe Netze

- Es gilt also

$$R_i = x_i \underbrace{\sum_j \rho(w_{ij}) \frac{\max\left(0, \sum_i x_i w_{ij}\right)}{\sum_i x_i \cdot \rho(w_{ij})} c_j}_{:\approx c_i} = \sum_j \frac{\rho(w_{ij}) \cdot x_i}{\sum_i x_i \cdot \rho(w_{ij})} R_j$$

- D.h. R_i^l lässt sich wieder schreiben als $x_i^l \cdot c_i^l$, mit c_i^l annähernd konstant.
- Ausgehend von der letzten Schicht kann die Relevanz somit auch gemäß der hergeleiteten Regeln zurück zum Input verteilt werden.
- Der Parameter c_i^l wird dabei durch die betrachtete Regel "induktiv" gebildet.

Zusammenhang mit LRP

- Die klassische $LRP - 0$ Formel kann als Deep Taylor Entwicklung im Nullpunkt gesehen werden
- Betrachte O.B.d.A. ein Neuron x_j mit $R_j > 0$.
- Die Suchrichtung $\boldsymbol{v}$ ist hierbei der Punkt $\boldsymbol{x}$ selbst, und somit gilt:

Zusammenhang mit LRP

- Die klassische $LRP - 0$ Formel kann als Deep Taylor Entwicklung im Nullpunkt gesehen werden
- Betrachte O.B.d.A. ein Neuron x_j mit $R_j > 0$.
- Die Suchrichtung $\boldsymbol{v}$ ist hierbei der Punkt $\boldsymbol{x}$ selbst, und somit gilt:

$$
\begin{aligned}
R^l_{i \leftarrow j} &= \frac{\partial R_j^{l+1}}{\partial x_i^l}(x_i - \tilde{x}_i) = w_{ij} \cdot c_j \cdot (x_i - \tilde{x}_i) \\
&= w_{ij} \cdot c_j \cdot \frac{\sum_i x_i w_{ij} + b_j}{\sum_i x_i w_{ij}} x_i \\
&= \frac{x_i \cdot w_{ij}}{\sum_i x_i w_{ij}} x_j \cdot c_j
\end{aligned}
$$

Zusammenhang mit LRP

- Die klassische $LRP - 0$ Formel kann als Deep Taylor Entwicklung im Nullpunkt gesehen werden
- Betrachte O.B.d.A. ein Neuron x_j^{l+1} mit $R_j > 0$.
- Die Suchrichtung $\boldsymbol{v}$ ist hierbei der Punkt $\boldsymbol{x}$ selbst, und somit gilt:

$$
\begin{aligned}
R_{i \leftarrow j}^{l} &= \frac{\partial R_j^{l+1}}{\partial x_i^{l}} (x_i - \tilde{x}_i) = w_{ij} \cdot c_j \cdot (x_i - \tilde{x}_i) \\
&= w_{ij} \cdot c_j \cdot \frac{\sum_i x_i w_{ij} + b_j}{\sum_i x_i w_{ij}} x_i \\
&= \frac{x_i \cdot w_{ij}}{\sum_i x_i w_{ij}} \underbrace{x_j \cdot c_j}_{R_j}
\end{aligned}
$$

Zusammenhang mit LRP

- Für die totale Relevanz von x_i gilt:

$$R_i^l = \sum_j R_{i \leftarrow j}^l = \sum_j \frac{x_i \cdot w_{ij}}{\sum_i x_i w_{ij}} R_j^{l+1} = \sum_j \frac{z_{ij}}{\sum_i z_{ij}} R_j^{l+1}$$

- Mit der gleichen Vorgehensweise können auch die $LRP - \varepsilon$ Regel sowie $LRP - \gamma$ hergeleitet werden.
- Deep Taylor Decomposition als Basis für LRP
- Wesentlicher Unterschied: Geforderte Konsistenz von Montavon et al. im Vergleich zu den LRP Regeln

Section 6

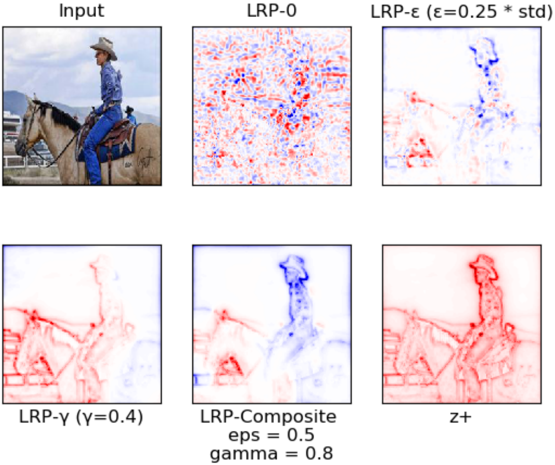
Ergebnisse auf Pascal-Datensatz

Anwendung auf den Pascal-Datensatz

- VGG16 trainiert für Multilabel-Klassifizierung für die Klassen *Mensch* und *Pferd*
- $z^{\mathcal{B}}$-Regel wurde bei Anwendung aller Regeln für das Inputlayer verwendet
- LRP-Composition:
  - LRP-0 auf Dense-Layern
  - LRP-ϵ auf mittleren sechs Conv-Layern
  - LRP-γ auf den letzten sechs Conv-Layern vor Inputlayer

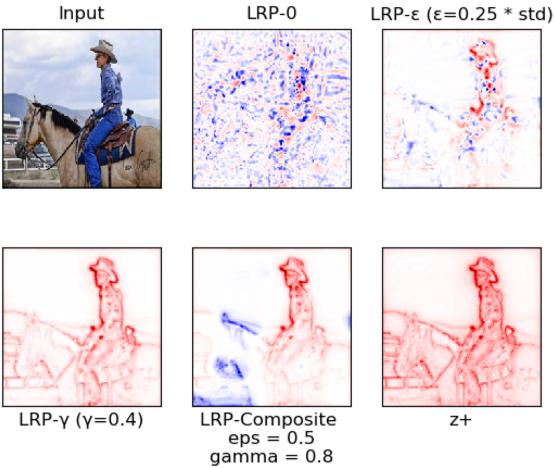
Visualisierungen der bisher vorgestellten Regeln

- Erklärung für das Erkennen des Labels *Pferd*



Visualisierungen der bisher vorgestellten Regeln

- Erklärung für das Erkennen des Labels *Mensch*



Quellen

- Quellen für Bilder, Implementierungshinweise:
- Montavon, Binder, Lapuschkin, Samek, Müller :
  "Layer-Wise Relevance Propagation: An Overview"
  Gefunden auf:
  → `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:
  →
  `https://www.youtube.com/watch?v=gy_Cb4Do_YE&t=939s`

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