OST
Ostschweizer
Fachhochschule

8 XAI: Attributionsmethoden

Systemtechnik BSc
HS 2026

Aufgaben

ANN im Modul ANN WUCH

1 XAI für CNN: Attributionsmethoden zur Bildanalyse

Model Predictive Control (MPC) excels when an accurate model of the system is available. However, in many real-world scenarios, systems are subject to significant stochasticity and modeling uncertainty. This chapter introduces Stochastic Value Gradients (SVG), a powerful class of policy gradient algorithms from reinforcement learning that provides a robust framework for learning control policies in such environments. We will show how SVG elegantly combines a learned dynamics model with value function approximation, creating a spectrum of algorithms that blend the predictive foresight of model-based methods with the robustness of model-free learning.

2 Einführung in die Attribution bei CNNs

Convolutional Neural Networks (CNNs) haben eine herausragende Leistungsfähigkeit in Aufgaben der Bilderkennung und -regression erzielt. Ihre komplexe, hierarchische Struktur macht sie jedoch zu "Black Boxes": Es ist oft unklar, auf welche Merkmale im Eingangsbild sich das Modell für seine Entscheidung stützt. Explainable AI (XAI) zielt darauf ab, diese Black Box zu öffnen und die Entscheidungsfindung von Modellen nachvollziehbar zu machen.

Ein zentraler Ansatz hierfür sind *Attributionsmethoden*. Die grundlegende Idee ist, die Vorhersage eines Modells auf seine Eingabemerkmale "zurückzuführen" (zu attribuieren). Für Bilddaten bedeutet dies, jedem Pixel des Eingangsbildes einen Relevanz- oder

Wichtigkeitswert zuzuordnen. Das Ergebnis ist eine Heatmap, oft als *Saliency Map* bezeichnet, die visuell hervorhebt, welche Bildbereiche für die Ausgabe des Netzwerks (z.B. die Klassifizierung als "Hund") am einflussreichsten waren.

2.1 Mathematische Definition der Attribution

Sei $F : \mathbb{R}^d \to \mathbb{R}$ eine Funktion, die ein neuronales Netzwerk repräsentiert. Für ein Eingangsbild $x \in \mathbb{R}^d$, das als Vektor von d Pixeln betrachtet wird, gibt $F(x)$ einen Skalar aus. Dieser Skalar kann der Logit-Wert für eine bestimmte Klasse bei einer Klassifikationsaufgabe oder der vorhergesagte Wert bei einer Regressionsaufgabe sein.

Eine *Attribution* ist eine Zuweisung eines Relevanzwertes $A_i(x)$ zu jedem Eingabemerkmal (Pixel) x_i. Das Ziel ist die Erstellung einer Attributionskarte (oder Vektor) $A(x) \in \mathbb{R}^d$.

$$A(x) = (A_1(x), A_2(x), \ldots, A_d(x)) \tag{1}$$

Diese Karte $A(x)$ soll die Wichtigkeit jedes Pixels x_i für den finalen Output $F(x)$ quantifizieren.

2.2 Standard-Attribution: Sensitivity Maps

Der direkteste Weg, die SSensitivität"des Outputs in Bezug auf eine kleine Änderung eines Input-Pixels zu messen, ist die Berechnung des Gradienten.

Definition (Sensitivity Map): Die Attribution eines Pixels x_i wird als der partielle Ableitungswert der Output-Funktion F nach diesem Pixel definiert.

$$A_i^{\mathsf{Sens}}(x) = \frac{\partial F(x)}{\partial x_i} \tag{2}$$

Die gesamte Attributionskarte ist somit der Gradient des Outputs bezüglich des Inputs:

$$A^{\mathsf{Sens}}(x) = \nabla_x F(x) \tag{3}$$

Interpretation: Der Wert $\frac{\partial F(x)}{\partial x_i}$ gibt an, wie stark sich der Output $F(x)$ ändert, wenn das Pixel x_i infinitesimal klein verändert wird. Ein hoher absoluter Wert bedeutet eine hohe Relevanz des Pixels für die Entscheidung. Zur Visualisierung wird oft der Absolutbetrag oder das Quadrat des Gradienten verwendet.

2.3 Integrated Gradients (IG)

Ein Problem der einfachen Gradientenmethode ist die Sättigung. Wenn ein Neuron bereits stark aktiviert ist (z.B. durch eine ReLU-Aktivierungsfunktion), kann sein Gradient null sein, obwohl das Neuron entscheidend für das Ergebnis ist. Integrated Gradients (IG) löst dieses Problem, indem es die Gradienten entlang eines Pfades von einem Referenzbild (Baseline) x' zum eigentlichen Bild x integriert. Die Baseline x' ist typischerweise ein informationsloses Bild, z.B. ein komplett schwarzes Bild.

Definition (Integrated Gradients): Die Attribution eines Pixels x_i mittels IG ist definiert als:

$$A_i^{\mathsf{IG}}(x) ::= (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} \, d\alpha \tag{4}$$

Eigenschaften und Interpretation:

- **Pfadintegral:** Die Formel integriert die Gradienten entlang der geraden Linie im Merkmalsraum von der Baseline x' zum Bild x.
- **Vollständigkeit (Completeness):** Eine wichtige Eigenschaft von IG ist, dass die Summe aller Attributionswerte der Differenz der Modellvorhersage zwischen dem Bild x und der Baseline x' entspricht:

  $$\sum_{i=1}^{d} A_i^{\mathsf{IG}}(x) = F(x) - F(x') \tag{5}$$

  Dies macht die Attributionen "vollständigünd direkt interpretierbar als Beiträge zur Gesamtänderung des Outputs.

—

3 Gradienten-basierte Saliency-Methoden

Diese Methoden basieren alle auf der Rückpropagierung von Gradienten vom Output zum Input.

3.1 Saliency Maps (nach Simonyan et al., 2014)

Historisch gesehen ist dies eine der ersten und einfachsten Methoden. Sie ist in ihrer reinsten Form identisch mit der oben definierten Sensitivity Map.

Algorithm 1 Berechnung einer Saliency Map

1: **Input:** Modell F, Eingangsbild x, Zielklasse c.
2: Führe einen Forward-Pass mit x durch, um alle Aktivierungen zu berechnen.
3: Berechne den Score $S_c(x)$ für die Zielklasse c. Dies ist der Output $F(x)$.
4: Berechne den Gradienten des Scores bezüglich des Eingangsbildes:

$$M(x) = \nabla_x S_c(x) = \frac{\partial S_c(x)}{\partial x}$$

5: **Visualisierung:**
6: Aggregiere die Gradienten über die Farbkanäle, z.B. durch den Maximalwert des Absolutbetrags für jedes Pixel: $m_{ij} = \max_k |M_{ijk}|$.
7: Normalisiere die resultierende 2D-Karte m zur Darstellung als Heatmap.
8: **Output:** Saliency Map m.

3.2 SmoothGrad (Smilkov et al., 2017)

Standard-Gradientenkarten sind oft visuell verrauscht, was die Interpretation erschwert. SmoothGrad reduziert dieses Rauschen durch einen einfachen, aber effektiven Mittelungsprozess. Die Intuition ist, dass der wahre Relevanz-Signal bei leichten Störungen des Bildes stabil bleibt, während das Rauschen im Gradienten zufällig ist und sich bei Mittelung herauskürzt.

Algorithm 2 SmoothGrad

Input: Modell F, Eingangsbild x, Anzahl der Samples n, Rauschlevel (Standardabweichung) σ.
Initialisiere eine leere Akkumulator-Karte $M_{avg} \leftarrow 0$.
for $i = 1$ to n **do**
    Erzeuge einen zufälligen Rauschvektor $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.
    Erstelle ein gestörtes Bild: $x_{noisy} = x + \epsilon_i$.
    Berechne die Gradienten-basierte Saliency Map für das gestörte Bild: $M_i = \nabla_x F(x_{noisy})$.
    Addiere die Karte zum Akkumulator: $M_{avg} \leftarrow M_{avg} + M_i$.
end for
Berechne den Durchschnitt: $M_{smooth} = \frac{1}{n} M_{avg}$.
Output: Geglättete Saliency Map M_{smooth}.

4 SUMMIT: Skalierbare Interpretierbarkeit durch Aktivierungs- und Attributions-Zusammenfassungen

Während Saliency Maps die Wichtigkeit von Pixeln für ein *einzelnes* Bild erklären, zielt SUMMIT (SUMmarization of Activations and Attributions) darauf ab, die interne Funktionsweise eines CNNs über einen *gesamten Datensatz* hinweg zu aggregieren und zu visualisieren. Der Kern von SUMMIT ist die Erstellung eines **Attributionsgraphen**.

4.1 Die Idee des Attributionsgraphen

Ein Attributionsgraph ist ein gerichteter azyklischer Graph (DAG), $G = (V, E)$, der die kausalen Einflüsse zwischen den internen "Konzepten", die von den Neuronen des Netzwerks gelernt wurden, darstellt.

- **Knoten (Nodes)** V: Jeder Knoten repräsentiert eine Gruppe von semantisch ähnlichen Neuronenaktivierungen innerhalb eines Layers. Ein Knoten steht also nicht für ein einzelnes Neuron, sondern für ein wiederkehrendes Muster oder "Konzept"(z.B. Äuge", "Felltextur").
- **Kanten (Edges)** E: Eine gerichtete Kante von einem Knoten u in Layer l_i zu

einem Knoten v in einem späteren Layer l_j $(j > i)$ quantifiziert, wie stark das von u repräsentierte Konzept zur Aktivierung des von v repräsentierten Konzepts beiträgt.

4.2 Attribution in SUMMIT

SUMMIT erweitert den Begriff der Attribution. Statt die Relevanz von *Input-Pixeln* für den *finalen Output* zu messen, misst SUMMIT die Relevanz der *Aktivierung eines Neurons* in einem früheren Layer für die *Aktivierung eines Neurons* in einem späteren Layer. Hierfür wird das Framework der Integrated Gradients (IG) verwendet.

Mathematische Definition: Sei $a_k^l(x)$ die Aktivierung des k-ten Neurons im Layer l für das Eingangsbild x. Die Attribution der Aktivierung des Neurons i in Layer l_1 auf die Aktivierung des Neurons j in einem späteren Layer l_2 wird definiert als:

$$\text{Attribution}(a_i^{l_1}, a_j^{l_2}) = \int_{\alpha=0}^{1} \frac{\partial a_j^{l_2}(\text{path}(\alpha))}{\partial a_i^{l_1}} \, d\alpha \tag{6}$$

Hierbei ist der Integrationspfad im Aktivierungsraum des Layers l_1 definiert, typischerweise von einem Baseline-Aktivierungsvektor (z.B. Nullvektor) zum tatsächlichen Aktivierungsvektor des Layers l_1.

4.3 Konstruktion des Attributionsgraphen

Bestimmung der Knoten (Nodes)

Die Knoten werden durch Clustering von Neuronenaktivierungen über einen gesamten Datensatz (z.B. alle Bilder der Klasse "Katze") ermittelt.

(a) **Aktivierungen sammeln:** Für einen gegebenen Layer l und einen Datensatz D werden alle Aktivierungsvektoren $\{a^l(x) \mid x \in D\}$ gesammelt.

(b) **Dimensionalitätsreduktion:** Da die Anzahl der Neuronen pro Layer sehr hoch sein kann, wird typischerweise eine Dimensionsreduktion wie PCA (Principal Component Analysis) auf die gesammelten Aktivierungen angewendet.

(c) **Clustering:** Auf den dimensionalitätsreduzierten Aktivierungen wird ein Clustering-Algorithmus (z.B. k-Means) ausgeführt.

(d) **Knotenerstellung:** Jedes resultierende Cluster C_k^l bildet einen Knoten im Graphen. Dieser Knoten repräsentiert eine Gruppe von Neuronen, die auf ähnliche Merkmale im Datensatz ansprechen.

Bestimmung der Kantengewichte (Edge Weights)

Das Gewicht einer Kante $w(u, v)$ von einem Knoten $u = C_k^{l_1}$ zu einem Knoten $v = C_m^{l_2}$ misst den aggregierten Einfluss.

(a) **Paarweise Attribution:** Für jedes Bild im Datensatz wird die paarweise Attribution (mittels IG) zwischen allen Neuronen im Quell-Cluster u und allen Neuronen im Ziel-Cluster v berechnet.

(b) **Aggregation:** Das Kantengewicht ist die Summe dieser Attributionen, gemittelt über den gesamten Datensatz.

Formel: Das Gewicht der Kante vom Knoten u (in Layer l_1) zum Knoten v (in Layer l_2) ist:

$$w(u, v) = \mathbb{E}_{x \in D} \left[\sum_{i \in u} \sum_{j \in v} \text{Attribution}(a_i^{l_1}(x), a_j^{l_2}(x)) \right] \tag{7}$$

wobei $\mathbb{E}_{x \in D}[\cdot]$ den Erwartungswert (Durchschnitt) über alle Bilder x im Datensatz D bezeichnet. Ein hohes Kantengewicht bedeutet, dass das von Knoten u repräsentierte niedrigstufige Merkmal ein starker kausaler Faktor für die Erkennung des von Knoten v repräsentierten höherstufigen Merkmals ist.

Literatur

[1] Smilkov, Daniel, Nikhil Thorat, Been Kim, Fernanda B. Viégas and Martin Wattenberg. *SUMMIT: Scaling Deep Learning Interpretability by Visualizing Activation and Attribution Summarizations.* ArXiv, abs/1704.03313, 2017.

[2] Smilkov, Daniel, Nikhil Thorat, Been Kim, Fernanda B. Viégas and Martin Wattenberg. *SmoothGrad: removing noise by adding noise.* ArXiv, abs/1706.03825, 2017.

[3] Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. *Axiomatic Attribution for Deep Networks.* Proceedings of the 34th International Conference on Machine Learning, 2017.

[4] Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps.* ArXiv, abs/1312.6034, 2014.

Lösungen