ERKLÄRBARE KÜNSTLICHE INTELLIGENZ

Vorlesung 4

Inhalte der Vorlesung

- Ersatztermine für Vorlesung
  - TINF22B5
  - Klausur auf den 13.12. verlegen
- Vorstellung der Projektaufgabe
- Verfahren
  - Counterfactual Explanations im Bildbereich
  - Saliency Maps
  - Integrated Gradients

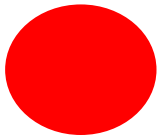
Projektaufgabe – Vorstellung der Projektergebnisse

- Pro Team
  - Training eines KI-Modells auf einem Datensatz
  - Implementierung und Anwendung zweier XAI Verfahren auf dem KI-Modell
  - Jede Person aus dem Team präsentiert einen Teil
  - Datensatz ist frei wählbar (Bild/tabellarische Daten, UCI oder Kaggle oder andere)
- Teilaufgaben im Team aufteilen:
  - Aufbau Datensatz und Preprocessing
  - KI-Modelltraining erklären
  - **Beide XAI-Verfahren erklären**
  - Notebook und Ergebnisse vorstellen
    - Was überzeugt an den Verfahren?
    - Was fällt negativ auf?
    - Was sollte verbessert werden (Ausblick?)
    - Welche Unterschiede haben die Verfahren in Hinblick auf die Ergebnisse?
- Präsentation der Ergebnisse in Vorlesung (Pro Team 30 Min)

Saliency Maps

- Saliency Maps sind eine Technik der Explainable AI (XAI), die genutzt wird, um die Entscheidungsprozesse von tiefen neuronalen Netzen, insbesondere Convolutional Neural Networks (CNNs), zu visualisieren.
- Sie helfen zu verstehen, welche Bereiche eines Bildes für die Vorhersage eines Modells besonders relevant sind. Dies wird häufig für Bildklassifikationsaufgaben verwendet, um zu erkennen, welche Bildbereiche das Netzwerk am meisten beeinflussen.

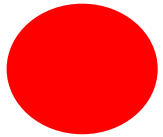
Saliency Maps



- Das Verfahren läuft im Wesentlichen wie folgt ab:

1. **Vorwärtsdurchlauf**: Ein Bild wird durch das neuronale Netzwerk geschickt, und das Modell trifft eine Vorhersage über die Klassenwahrscheinlichkeiten. Dabei handelt es sich oft um die Wahrscheinlichkeiten für verschiedene Objektklassen im Bild.
2. **Gradientenberechnung**: Nach der Vorhersage wird ein Rückwärtsdurchlauf initiiert, bei dem der Grad der Änderung (Gradient) der Ausgabe eines bestimmten Neurons bezüglich der Eingabepixel berechnet wird. Typischerweise wird der Gradient der Vorhersagewahrscheinlichkeit der Zielklasse (z. B. "Katze" in einem Bild) bezüglich der Pixelintensitäten berechnet.
3. **Erstellung der Saliency Map**: Die berechneten Gradienten geben an, wie empfindlich die Vorhersage auf Veränderungen in jedem Pixel reagiert. Hohe Gradientenwerte zeigen an, dass kleine Änderungen dieser Pixel die Vorhersage stark beeinflussen würden – also sind diese Pixel für die Entscheidung wichtig. Die resultierenden Gradienten werden in einer Saliency Map visualisiert, die die relevanten Bildbereiche hervorhebt.
4. **Visualisierung**: Die Saliency Map wird als Wärmebild (oft in Graustufen oder mit Farbskalen) über das Eingangsbild gelegt, um zu zeigen, welche Bildbereiche für das Modell besonders relevant waren. Dies sind die "auffälligen" Bereiche, die das Netzwerk zur Entscheidungsfindung herangezogen hat.

Saliency Maps



- **Hauptunterschiede zwischen Saliency Maps und Grad-CAM:**

1. **Berechnungsebene:**
    1. **Saliency Maps:** Diese Technik berechnet die Gradienten direkt bezüglich der Eingabepixel. Das bedeutet, dass die Empfindlichkeit der Ausgabe für die Klasse in Bezug auf jedes einzelne Pixel des Eingabebildes berechnet wird. Daher liefert eine Saliency Map eine pixelgenaue Visualisierung der wichtigsten Bildbereiche.
    2. **Grad-CAM:** Grad-CAM berechnet die Gradienten hingegen in den tieferen Schichten des Netzwerks, speziell in den letzten Convolutional Layers, und gewichtet dann die Aktivierungen in diesen Schichten. Dadurch entsteht eine „Karte“, die zeigt, welche Regionen des Bildes für die Entscheidung auf einer höheren Abstraktionsebene wichtig sind, und nicht nur auf Pixelebene.

Saliency Maps

- **Hauptunterschiede zwischen Saliency Maps und Grad-CAM:**

1. **Interpretation und Visualisierung:**
    1. **Saliency Maps:** Das Ergebnis ist oft feinkörnig, und die Karte hebt die Pixel hervor, die besonders empfindlich für die Klassenvorhersage sind. Die Saliency Map wirkt oft „rauschiger“, da sie direkt auf den Gradienten der Eingabe basiert.
    2. **Grad-CAM:** Das Ergebnis ist gröber und konzentriert sich auf die Regionen des Bildes, die für die Klassifikation relevant sind. Grad-CAM liefert eine besser interpretierbare Visualisierung, da es die Fokusregionen des Modells in Form von „Hotspots“ oder Wärmebildern anzeigt, die direkt über dem Eingangsbild angezeigt werden können.

Saliency Maps

- **Hauptunterschiede zwischen Saliency Maps und Grad-CAM:**
- **Einsatzgebiet und Genauigkeit:**
- **Saliency Maps:** Diese sind unabhängig von der Architektur und können auf jede Art von neuronalen Netzwerk angewendet werden. Allerdings sind sie anfälliger für Rauschen und liefern weniger intuitive Ergebnisse.
- **Grad-CAM:** Grad-CAM eignet sich besonders gut für Convolutional Neural Networks (CNNs), da es die räumlichen Informationen aus den Convolutional Layers nutzt und eine aussagekräftigere Visualisierung der Bildregionen bietet. Daher wird Grad-CAM oft bevorzugt, wenn eine verständlichere und intuitivere Erklärung der Modellentscheidung gewünscht ist.

Übung zu Saliency Maps

- Übung: Modifikationen zur Saliency Map und Bildverarbeitung

1. Aufgabe
   - Ändern Sie das Bild: Ersetzen Sie das verwendete Bild durch ein anderes Bild Ihrer Wahl, z. B. ein Bild eines Hundes, Autos oder eines Objekts, das sich deutlich von einer Katze unterscheidet.
   - Dies hilft, die Unterschiede in den Saliency Maps für verschiedene Bildinhalte zu beobachten und zu verstehen, wie das Modell verschiedene Objekte erkennt.
2. Aufgabe
   - Ändern Sie das Modell: Verwenden Sie statt VGG16 ein anderes vortrainiertes Modell, z. B. ResNet50 oder InceptionV3.
   - Ändern Sie dazu einfach die Zeile model = VGG16(weights="imagenet") in model = ResNet50(weights="imagenet").
   - Beobachten Sie, ob sich die Saliency Map oder die Erkennung der wichtigen Bildbereiche ändert.

Übung zu Saliency Maps

3. Aufgabe

- Ändern Sie das Farbkonzept der Saliency Map:
- Modifizieren Sie das cmap-Argument in plt.imshow(saliency, cmap='hot') zu einem anderen Farbkonzept, z. B. viridis, plasma oder gray, um die Saliency Map auf unterschiedliche Weise darzustellen.
- Vergleichen Sie die Lesbarkeit und Klarheit der Saliency Maps bei verschiedenen Farbschemata.

4. Aufgabe

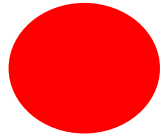
- Normalisierung der Saliency Map:Skalieren Sie die Saliency Map, um sie besser sichtbar zu machen.
- Sie könnten beispielsweise alle Werte auf einen Bereich von 0 bis 1 normalisieren
- Fügen Sie dazu folgenden Codesnip ein: saliency = (saliency - tf.reduce_min(saliency)) / (tf.reduce_max(saliency) - tf.reduce_min(saliency))

Übung zu Saliency Maps

5. Aufgabe

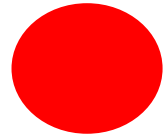
- Visualisieren Sie die Saliency Map über dem Originalbild. Fügen Sie dazu folgendes ein
  - plt.imshow(img)
  - plt.imshow(saliency, cmap='hot', alpha=0.5) # Setze Transparenz für die Saliency Map
  - plt.title("Saliency Map over Original Image")
  - plt.axis("off")
  - plt.show()
- Reflexionsfragen
  - Wie unterscheiden sich die Saliency Maps zwischen verschiedenen Modellen und Bildern?
  - In welchen Bildbereichen konzentriert sich das Modell und wie interpretiert es visuelle Merkmale?
  - Wie verändert sich die Saliency Map für verschiedene Zielklassen und wie wirkt sich die Auswahl der Farbschemata auf die Lesbarkeit aus?

Integrated Gradients



- **Integrated Gradients** ist ein Verfahren der Explainable AI (XAI), das genutzt wird, um die Entscheidungsprozesse neuronaler Netze verständlicher zu machen.
- Es hilft dabei, zu erklären, welche Merkmale oder Pixelwerte besonders stark zur Vorhersage des Modells beigetragen haben.
- Integrated Gradients wurde speziell entwickelt, um die Empfindlichkeit und Vollständigkeit von Erklärungen sicherzustellen und ist besonders nützlich bei Modellen, bei denen eine lineare Ableitung, wie sie bei Saliency Maps verwendet wird, zu wenig aussagekräftig ist.

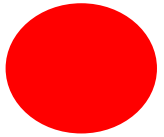
Integrated Gradients



- Vorgehen:

1. **Auswahl einer Basislinie**:
    1. Man startet mit einer neutralen Version der Eingabe, genannt Basislinie. Bei Bildern könnte das ein komplett schwarzes oder graues Bild sein, das keine erkennbaren Merkmale enthält. Die Basislinie dient als Vergleichspunkt und zeigt, wie die Eingabe ohne jegliche Merkmale aussehen würde.
2. **Erstellen von Zwischenbildern**:
    1. Man erstellt eine Serie von Bildern, die in kleinen Schritten von der Basislinie bis zum tatsächlichen Bild reichen. Diese Zwischenbilder sehen so aus, als ob das Bild langsam von schwarz zu seiner vollen Form „aufgeblendet“ wird.
3. **Bewertung der Modellreaktion**:
    1. Für jedes Zwischenbild berechnet das Modell, wie stark jedes Detail (z. B. ein bestimmter Pixel oder ein Bereich) die Vorhersage beeinflusst. Diese Berechnungen zeigen, wie sich das Modell verhält, wenn Teile des Bildes stufenweise „sichtbar“ werden.

Integrated Gradients



4. **Zusammenführen der Informationen**:
    1. Man kombiniert alle Erkenntnisse über die Zwischenbilder, um zu verstehen, welche Bildbereiche insgesamt am wichtigsten für die Vorhersage sind. Dabei geht es um die Frage: Welche Bilddetails führen dazu, dass das Modell sich immer mehr zur Zielklasse (z. B. "Katze") hin „entscheidet“?
5. **Erstellen der Erklärungs-Heatmap**:
    1. Am Ende erzeugt man eine Art Karte des Bildes (Heatmap), die zeigt, welche Bereiche des Bildes den größten Einfluss auf die Entscheidung des Modells hatten. Je stärker der Einfluss, desto intensiver wird der Bereich markiert.

Übung zu Integrated Gradients

1. Aufgabe

- Ändern Sie das Basisbild (Baseline), das aktuell ein schwarzes Bild ist, zu einem weißen Bild oder einem grauen Bild.
- Ersetzen Sie baseline = np.zeros_like(x) durch baseline = np.ones_like(x) * 255 für ein weißes Bild oder baseline = np.ones_like(x) * 128 für ein graues Bild.
- Lernziel: Verstehen, wie sich die Wahl des Basisbilds auf die Integrated Gradients auswirkt und welche Bereiche des Bildes das Modell als relevant einstuft.

2. Aufagbe

- Variieren Sie die Anzahl der Schritte (m_steps) in der Funktion integrated_gradients und beobachten Sie die Ergebnisse.
- Setzen Sie m_steps auf einen kleineren Wert (z. B. 10) oder einen größeren Wert (z. B. 100) und vergleichen Sie die Qualität der Saliency Map.
- Lernziel: Erkennen, wie die Anzahl der Schritte die Genauigkeit und Aussagekraft der Integrated Gradients beeinflusst.

Übung zu Integrated Gradients

3. Aufgabe

- Normalisieren Sie die Werte der Integrated Gradients auf einen Bereich von 0 bis 1, bevor Sie sie visualisieren.
- Fügen Sie den Code integrated_grads = (integrated_grads - integrated_grads.min()) / (integrated_grads.max() - integrated_grads.min()) ein, um die Werte zu normalisieren.
- Lernziel: Besseres Verständnis für die Normalisierung von Daten und die Verbesserung der visuellen Klarheit bei der Darstellung der Saliency Map.

4. Aufgabe

- Ändern die die Farbpalette analog zu den Saliency Maps

5. Aufgabe

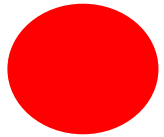
- Berechnen Sie die Gradienten, ohne die Zielklasse zu fixieren, und analysieren Sie, welche Bereiche für alle Klassen relevant sind.
- Ersetzen Sie loss = preds[:, target_class_idx] durch loss = tf.reduce_sum(preds, axis=-1) in der Funktion compute_gradients.Lernziel:
- Einblick in die allgemeinen Sensitivitätsbereiche des Modells erhalten, ohne eine spezifische Zielklasse zu setzen.

Counterfactual Explanations im Bildbereich

- Eine **Counterfactual Explanation** (kontrafaktische Erklärung) im Bildbereich ist eine Methode, um zu zeigen, welche Änderungen an einem Bild erforderlich wären, damit ein Modell seine Klassifikation zu einer anderen, gewünschten Klasse ändert. Es ist ein Ansatz der Explainable AI (XAI), der hilft, die Entscheidungsprozesse eines Modells besser zu verstehen, indem er aufzeigt, wie das Bild verändert werden müsste, um eine alternative Vorhersage zu erhalten.
- Stellen wir uns vor, ein neuronales Netzwerk klassifiziert ein Bild als „Katze“. Eine kontrafaktische Erklärung würde nun ein ähnliches Bild erzeugen, das so modifiziert ist, dass das Modell es als eine andere, vorgegebene Zielklasse (z.B. „Hund“) klassifiziert. Dieses modifizierte Bild wird als **kontrafaktisches Bild** bezeichnet.
- **Vorgehensweise bei der Erstellung kontrafaktischer Bilder**

1. **Modellvorhersage und Zielklassendefinition:**
    1. Zuerst wird die aktuelle Vorhersage des Modells für das Originalbild ermittelt.
    2. Eine alternative Zielklasse wird festgelegt, z. B. „Hund“ anstelle der ursprünglichen Vorhersage „Katze“.

Counterfactual Explanations im Bildbereich



- **Vorgehensweise bei der Erstellung kontrafaktischer Bilder**

2. **Optimierung zur Erzeugung des kontrafaktischen Bildes:**
    1. Das Originalbild wird als Eingabe verwendet und in kleinen Schritten verändert, bis das Modell es als die Zielklasse erkennt.
    2. Die Optimierung minimiert die „Distanz“ zur Zielklasse und erzeugt eine Bildversion, die möglichst nah am Original ist, jedoch für das Modell so verändert wurde, dass es die Zielklasse statt der Originalklasse erkennt.
3. **Visualisierung der Änderungen:**
    1. Das kontrafaktische Bild zeigt, welche Merkmale das Modell als typisch für die Zielklasse betrachtet. Es zeigt also, welche Eigenschaften im Bild angepasst wurden, um die Klassifikation zu ändern.

Counterfactual Explanations im Bildbereich

- **Aufgabe: Kontrafaktische Erklärung – Bild einer Katze in einen Hund umwandeln**
  - In dieser Übung werden Sie ein Bild einer Katze mithilfe eines vortrainierten Modells und TensorFlow so modifizieren, dass das Modell das Bild letztendlich als Hund klassifiziert. Hierbei werden Sie schrittweise Änderungen am Bild vornehmen und dabei die kleinsten Anpassungen finden, die das Modell benötigen könnte, um die Klassifikation von „Katze“ zu „Hund“ zu ändern.
- **Ziel der Aufgabe**
  - Das Ziel ist, zu verstehen, wie man mit kontrafaktischen Erklärungen arbeitet, indem man das Bild durch Optimierung so verändert, dass die Bildinhalte für das Modell eher einem Hund entsprechen. Dabei nutzen Sie das VGG16-Modell, das auf dem ImageNet-Datensatz vortrainiert ist.

Übung zu Counterfactual Explanations im Bildbereich

- Führen Sie den Code aus und beobachten was passiert.
- Schafft es der Algorithmus das Bild zu ändern?
- Falls, ja auf welche Klasse?

Übung zu Counterfactual Explanations im Bildbereich

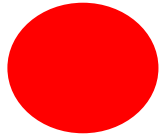
1. Aufgabe

- Erstellen und vergleichen Sie die Saliency Maps für das Originalbild und das kontrafaktische Bild. Hinweis: Berechnen Sie die Saliency Map für das Originalbild und das Counterfactual-Bild und stellen Sie beide gegenüber.
- Lernziel: Verstehen, wie sich die wichtigen Bildbereiche vor und nach der Transformation ändern.

2. Aufgabe

- Passen Sie die Lernrate (learning rate) von 0.01 auf 0.1 an und beobachten was passiert.
- Schafft es der Algorithmus nun die Anpassung durchzuführen?

Übung zu Counterfactual Explanations im Bildbereich



1. **Kleine Lernrate**:
    1. Eine niedrige Lernrate bewirkt, dass das Modell kleinere, feinere Anpassungen vornimmt.
    2. Dies führt zu einem **langsameren, aber stabileren Optimierungsprozess** und kann dabei helfen, ein genaueres Ergebnis zu erreichen, da der Prozess vorsichtig ist und nicht zu große Änderungen am Bild vornimmt.
    3. Nachteil: Wenn die Lernrate zu niedrig ist, kann die Optimierung sehr lange dauern oder sogar „steckenbleiben“, bevor das Ziel erreicht wird.
2. **Hohe Lernrate**:
    1. Eine hohe Lernrate bedeutet, dass das Modell größere Schritte macht und schnellere Fortschritte erzielt.
    2. Dies kann **den Optimierungsprozess beschleunigen**, da das Modell schneller auf die Zielklasse zusteuert.
    3. Nachteil: Wenn die Lernrate zu hoch ist, besteht das Risiko, dass das Modell „überschießt“ und instabile oder übertriebene Änderungen am Bild vorgenommen werden. Das Ergebnis kann dann ungenau oder „chaotisch“ wirken, und der Prozess erreicht möglicherweise nicht das gewünschte Ziel.

Übung zu Counterfactual Explanations im Bildbereich

- **Rolle der Lernrate in Counterfactual Explanations**
- In einem Counterfactual-Optimierungsprozess wie diesem, bei dem ein Bild so verändert werden soll, dass es vom Modell als andere Klasse erkannt wird, steuert die Lernrate, wie drastisch die Anpassungen am Bild sind:
  - Eine **zu niedrige Lernrate** kann bedeuten, dass das Bild nur leicht angepasst wird und möglicherweise nicht weit genug verändert wird, um als Hund klassifiziert zu werden.
  - Eine **zu hohe Lernrate** könnte das Bild so stark verändern, dass es unrealistisch aussieht oder instabil wird.

Explainable AI Verfahren

- Anchors
- GraphLime
- Testing with Concept Activation Vectors (TCAV)
- Scalable Bayesian Rule Lists
- Permutation Importance
- Partial Dependence Plot
- DeepLift

Programmierübung

- S. Moodle
- Abgabe ohne Bewertung!
- Grad Cam
- Counterfactual explanations
- Normalisierung

Rechenlaufzeit

Aufgabe Integrated Gradients reden

Tim, Denis und Juliana