Warsaw University of Technology
Faculty of Electrical Engineering

Explainability of Convolutional Neural Network: overview of methods

Authors: dr hab. inż. Krzysztof Siwek, prof. Uczelni [1]
Karim Sonbul [1]
Magdalena Markowicz [1]
Piotr Podgórski [1]
Jan Kruszyński [1]

Affiliations
[1] Warsaw University of Technology

Understanding and interpreting the decisions made by deep learning models has become an essential area of research in artificial intelligence. Convolutional neural networks (CNNs), despite their high performance in various tasks, often function as ”black boxes,” making it challenging to explain their predictions. This study focuses on applying and evaluating different explainability techniques to CNN models to gain more insight into their decision-making processes. Using multiple approaches, our aim was to assess the effectiveness and reliability of these methods in improving the transparency and interpretability of neural networks.

LRP - Layer-wise Relevance Propagation

The first explainability method considered in the study is Layer-wise Relevance Propagation (LRP). LRP works by propagating the given model’s output score backward through the layers of the network, assigning relevance scores to each neuron during the process. In effect, it highlights which input features contribute the most to a prediction. Layer-wise Relevance Propagation returns a relevance score that can be visualized as a heatmap, making the method intuitive and easy to understand.

LRP achieves its goal by making use of a given propagation rule. Propagation rules dictate how relevance is propagated backwards through the model’s layers. One of the most fundamental and most popular propagation rules is the Epsilon rule. The equation is as follows:

$$R_i^l = \sum_j \frac{z_{ij}}{\sum_i z_{ij} + \epsilon \cdot \text{sign}(\sum_i z_{ij})} R_j^{l+1}$$

By preventing small neuron activations from disproportionately influencing the relevance output, the Epsilon rule ensures stability within the explanation. However, it can lead to overtly sparse explanations in some cases. To improve the explainability of the method, we used a modified version of the Epsilon rule in the study, called the Epsilon-Alpha2Beta1-Flat rule. The rule allows the method to better handle positive and negative contributions to the network’s classification. The equation is as follows:

$$R_i^l = \sum_j \left(\alpha \frac{z_{ij}^+}{\sum_i z_{ij}^+ + \epsilon} - \beta \frac{z_{ij}^-}{\sum_i z_{ij}^- + \epsilon} \right) R_j^{l+1}$$

In the Epsilon-Alpha2Beta1-Flat rule, α is assigned the value of 2 and β is assigned the value of 1. Thus, positive contributions are emphasized over negative contributions; however, negative contributions are still accounted for. This allows the rule to achieve a more balanced and interpretable explanation.

GradCAM - Gradient-weighted Class Activation Map

The second explainability method considered in the study is Gradient-weighted Class Activation Map (GradCAM).The implementation of the Grad-CAM was adapted to the architecture of the applied CNN model and the specifics of the binary classification task. Grad-CAM relies on analyzing the activations and gradients from a selected convolutional layer. In the first step, activations of the target layer are recorded during the forward pass. Then, after backpropagation with respect to a chosen output class, gradients corresponding to the same layer are collected. These gradients are spatially averaged to produce weights, which are subsequently used to scale the corresponding activation maps. By summing these weighted channels, a class-discriminative activation map is obtained, indicating the importance of different regions of the input image. The resulting map is then filtered using the ReLU function to discard negative values and resized to the original input image dimensions via bilinear interpolation. Finally, the map is normalized to the [0, 1] range, allowing it to be visualized as a transparent overlay on the original image .In this experiment, the last convolutional layer of the model (referred to as model.20) was selected, as it captures the highest-level spatial features relevant for the model's decision-making. To prevent excessive memory usage, the activation and gradient hooks are removed after each Grad-CAM execution.

DeepLIFT

DeepLIFT (Deep Learning Important Features) is a backpropagation-based attribution method that explains neural network decisions by quantifying input feature contributions relative to a reference baseline. The method identifies features responsible for driving predictions away from a neutral baseline state through layer-wise deviation propagation. This method can be divided into three steps:

baseline state through layer-wise deviation propagation. This method can be divided into three steps:

- Reference Establishment: A baseline input (e.g., zero tensor for images, dataset mean) is propagated through the network to establish layer-wise reference activations.This represents a ”neutral” input producing minimal task-specific signal.
- Deviation Propagation: For the target input, differences between actual activations and reference values are com-puted at each layer. These deviations capture how theinput perturbs network behavior from baseline.
- Contribution Attribution: Modified backpropagation rules distribute output deviations backward through the computational graph:
  - Linear operations (e.g., dense/convolutional layers): Contributions split proportionally to learned weights.
  - Nonlinear activations (e.g., ReLU, sigmoid): Local linear approximations estimate deviation propagation.

The final attributions are computed via a single backwardpass, making DeepLIFT computationally efficient compared to perturbation-based approaches. DeepLIFT mitigates gradient saturation issues by focusing on input/reference differences rather than absolute gradients.

LIME

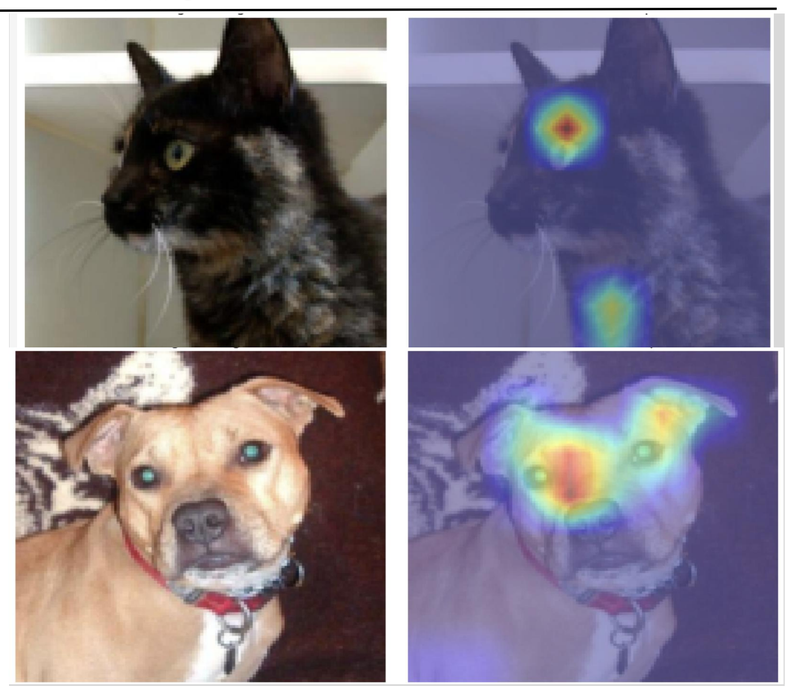
LIME (Local Interpretable Model-agnostic Explanations)for image classification is a perturbation-based technique meant to clarify the forecasts of any black-box machine learning model for particular cases by approximating the model locally with an interpretable one. It determines which areas of an image (superpixels) were most important in the model’s decision for that particular image. The procedure consists of these key steps:

- Interpretable Representation Generation: First, the input image is segmented into a series of contiguous, perceptually relevant patches known as superpixels (e.g.,using skimage.segmentation). LIME will explain using these superpixels as the understandable characteristics.
- Local Sample Perturbation: Within the interpretable(superpixel) space, a dataset of perturbed samples is produced in the vicinity of the original image. Randomly turning superpixel subsets ”on” (using original pixel values) or ”off” (replacing with a neutral color like gray or black, corresponding to hide_color=0 in the code)accomplishes this.
- Black-Box Model Prediction: Using its prediction function (the predict_fn implemented in the code), each perturbed sample is passed through the intricate black-box model after being reconstructed back into the original image pixel space. For every perturbed sample, this provides the model’s probability outputs for the classes of interest.

The significance or contribution of each superpixel to the particular prediction under explanation is represented by the coefficients that this local linear model has learned. High-lighting the super pixels on the original image that have the highest positive coefficients is a common way to visualize these importances. Although LIME is model-agnostic, meaning it can be used on a variety of complex models without requiring access to their internal structure, its explanations are essentially local, and it can be computationally more expensive to generate perturbations than gradient-based methods

Dataset

The neural network used in this study was designed with the classification of images of cats and dogs in mind. We had chosen the Kaggle Cats and Dogs dataset for this task. The dataset consists of 25 thousand labeled images of different cats and dogs. Before being passed through the model, the dataset was resized to 128x128, rotated up to 15 degrees to increase variance, flipped horizontally to increase generalization, normalized, and split into training and validation subsets.



Results

Metoda	IAUC	DAUC	IC	DC	AD	IIC
LRP	0.387	0.546	0.118	-0.241	-0.093	0.014
LIME	0.416	0.361	0.108	-0.069	0.107	0.035
GradCAM	0.489	0.392	0.070	-0.179	0.050	0.003
DeepLIFT	0.323	0.439	0.125	0.333	0.0193	0.001

In the table above, we gathered the metric results for each of the tested explainability methods for the first 200 images from the test image dataset. For IAUC, GradCAM got the best score, with a result of 0.489 and DeepLIFT got the worst score of the tested methods, with a score of 0.323. For the DAUC metric, the LIME method got the best score, namely 0.361 and LRP got by far the worst score, with a result of 0.546. GradCAM also had the best result for the Insertion Correlation metric, while DeepLIFT had the best score for Deletion Correlation. None of the methods scored particularly high for AD and IIC metrics, but LIME got the highest result of all the tested ones, 0.107 and 0.035 for AD and IIC respectively. LIME scored the highest out of all tested XAI methods for three of the six metrics tested. GradCAM came second, with two highest scores, for IAUC and IC. LRP seems to have performed the worst of the tested metrics, having scored the lowest for three different metrics (DAUC, DC and AD).