JOHANNES KEPLER
UNIVERSITY LINZ

Submitted by
Viktor Maximilian Loreth
k12006268

Submitted at
Institute of Computational Perception

Supervisor
Katharina Hoedt, PHD

August 28, 2023

EVALUATION OF IMAGE RECOGNITION NEURAL NETWORK INTERPRETATION METHODS: AN IN-DEPTH LOOK

Bachelor Thesis

to obtain the academic degree of

Bachelor of Science

in the Bachelor's Program

Artificial Intelligence

JOHANNES KEPLER UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Österreich
www.jku.at
DVR 0093696

JOHANNES KEPLER
UNIVERSITY LINZ

Sworn Declaration

I hereby declare under oath that the submitted Bachelor Thesis has been written solely by me without any third-party assistance, information other than provided sources or aids have not been used and those used have been fully documented. Sources for literal, paraphrased and cited quotes have been accurately credited.

The submitted document here present is identical to the electronically submitted text document.

Linz, August 28, 2023

JOHANNES KEPLER
UNIVERSITY LINZ

Abstract

This paper focuses on the accuracy of interpretability methods for machine learning models. The main research problem is the lack of ground truth for evaluation method in interpretability methods. While there are inherent interpretative models, black-box networks perform better and have developed rapidly. Existing interpretability methods for neural networks such as ROAR, KAR, BAM and Real Time Image Saliency for Black Box Classifiers provide a numerical evaluation method but they do still suffer from ambiguity. This paper summarizes the different evaluation methods, compares them and also calculate ROAR on 2 different saliency methods.

JOHANNES KEPLER
UNIVERSITY LINZ

Contents

List of Figures

1 Introduction

Artificial Intelligence (AI) has undergone rapid development in the last years. In today's modern era of mobile phones and computers, algorithm's are used on a daily basis to have quick access to information and improve the efficiency of the daily life.

While various Algorithms (e.g.: Decision Trees, Linear Regression, Support Vector Machines, etc.), which are comprehensible by design, have been developed, the spotlight has turned to Deep Neural Networks(DNN). This shift is attributed to the increase in computational power and the exponential increase in accessible data. Despite their remarkable accuracy, DNN remain opaque black-boxes, which we struggle to understand. Nevertheless, the immense improvement in performance and their ability to handle massive datasets have led to widespread adoption in contemporary devices. It is predicted, that algorithms based on Neural Networks will be becoming increasingly popular in the next years.

However, one of the primary difficulties with Neural Networks is the lack of reliable interpretation techniques. Numerous interpretation methods exist, yet a universally reliable method remains missing. Particularly in the domain of image analysis, encompassing critical applications like automated driving and facial recognition, no solution is present. The decision-making rationale of neural networks remain unclear, attributed to factors like background elements, peripheral objects or lighting conditions. Efforts to address this issue have given rise to gradient methods, aiming to assign significance values to pixels and represent their importance on neural network decisions. Another alternative option to mitigate the black-box nature of algorithms involves employing model-agnostic methods. These methods offer an computational linkage between inputs and outputs, irrespective which model is used. Although highly effective for smaller datasets, they begin to struggle as the data size and their complexity increases. Because of this, they do not offer a reliable way to quickly make Neural Networks interpretable.

In light of these prevalent problems, the object of this thesis is to recapitulate interpretation algorithms Neural networks in computer vision. Emphasis is placed on the evaluation of post-hoc interpretability techniques, forecasting potential future developments and focusing on the strengths and weaknesses of distinct techniques. Concluding the theoretical segment, a practical demonstration showcasing the application of ROAR is shown.

1.1 Structure of the thesis

1. The first part presents a overview of contemporary machine learning algorithms, categorizing them into two main groups: algorithms with inherent interpretability and those without. The goal is to make clear how supervised methods can be applied to image recognition tasks.
2. Subsequent sections delve into interpretability, emphasizing global and local model-agnostic techniques. These methods offer insights into overall model behavior, regardless of algorithm specifics.
3. Additionally, in the domain of interpretability techniques for Neural Networks, the paper explores ad-hoc methods for Neural Networks. Features visualization and Gradient-focused methods are explained.
4. After introducing existing interpretation methods, the paper's focus transitions to evaluating post-hoc interpretation methods. Various approaches to assess the effectiveness and dependability of these methods in offering meaningful insights into intricate models are introduced and discussed. Additionally, the advantages and disadvantages of these approaches are carefully examined to provide a comprehensive understanding of their applicability.
5. To exemplify the discussed concepts, the practical application of the ROAR methodology using the food101 data set [2] and MNIST dataset [4] is presented. This real-world instance illustrates the current state of art of interpretability techniques in image recognition.

2 Machine Learning and their Interpretability

Before going into detail into the different methods of interpretability an overview of current supervised machine learning is given. With those examples, it should be explained why interpretability is necessary.

2.1 Supervised Machine Learning

In supervised machine learning, there are several base methods of classification. They are shortly introduced and analyzed for their interpretability. This should give readers a simply structure on why this research is needed. This section analyzes the base functionality of each singular model. Furthermore, an analysis of the interpretability from a human perspective is made. Methods are developing rapidly, therefore only the most common methods are included.
There are several methods on how algorithms can be classified in their interpretability: Algorithm Transparency: How does the algorithm learn the model from the data? What relationships can it learn? [10] A linear regression model using least squares method is easier to understand than a deep learning model. Pushing a gradient through a network with millions of weights is less well understood and considered less transparent.

2.1.1 Linear Models

When it comes to predicting outcomes, a simple method is to use a linear regression model. This model predicts by adding up different features, each multiplied by a weight. The linear nature of this model makes it easy to understand. Mathematically, the predictive output, denoted as $\hat{y}$, is captured in the equation:

$$\hat{y} = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n + \epsilon$$

The alphas α_i indicate the significance of each feature. The initial coefficient α_0 is known as the intercept, signifying the baseline. The noise ϵ encapsulates the inevitable errors stemming from inherent non-linearity in real-world dynamics or measurement inaccuracies.

To train model, the MSE-Loss or the absolute loss can be applied. When using regularization methods, the absolute loss is taken to be more resilient to outliers.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\text{ABS} = \frac{1}{n}\sum_{i=1}^{n} y_i - \hat{y}_i$$

The interpretability of the model is very simple. The factors are given through the coefficient matrix. Each feature has a distinctive importance to the model and it can be seen easily how important each factor is, when the data is normalized.

$$\alpha = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \dots \\ \alpha_n \end{bmatrix}$$

Although linear models possess comprehensibility and provide a straightforward method for prediction and are inherent understandable, their application is limited to linear relationships and small datasets. One can use advanced techniques like L1 and L2 regularization [5] to achieve regularization against outliers and in case of correlation between factors. In the domain of image recognition it is not applicable.

2.1.2 Distance-Based methods

K-Nearest Neighbors is used for classification and uses the nearest neighbars as classification. KNN is not interpretable by default, as there are no parameters to learn and analyze. However KNN is inherently easy to understand, as it simply looks for the most similar samples. One is able to visualize fewer features or use clustering algorithms to reduce the dimensionality, but the relationship between input and output is unclear. KNN also struggles with many features, therefore it is not recommended.

Support Vector Machines (SVM) aim to find a hyperplane that maximizes the margin between different classes of data points.

$$\text{Minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{Subject to } y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \text{ for } i = 1, \dots, n$$

By using the kernel trick, SVM can be used for non-linear data. In higher dimensionality, SVM becomes non-interpretable, as displaying the weight matrix is not understandable. SVM also struggles with big datasets. Therefore also SVM is not a good choice for image classification.

2.1.3 Decision Tree-Based Methods

Decision Trees based on minimizing the gini-index are inherently interpretable. As the depth increases, the models become less understandable. An example for the titanic dataset is visible in 2.1 Their bad generalization method gave rise to ensemble methods.

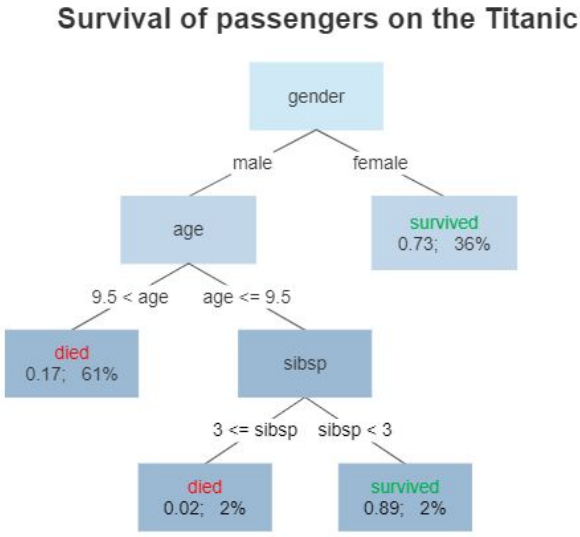

Figure 2.1: Decision Tree Example: By Gilgoldm - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=90405437

Random forests are an ensemble of multiple decision trees. Their advantage is a smoother predictive power. But it suffers from being less interpretable. Using special techniques like SHAP values [9] or partial dependence trees can make them more understandable.

Gradient boosting like XGBoost, LightGBM and CatBoost are similar to random forest and decision trees, but with differential learned weights for each decision. They suffer from the same interpretability issues as random forests.

While decision tree-based methods can be applied for image classification, their accuracy is poor.

2.1.4 LDA, QDA

2.1.5 Neural Networks

The rise of neural networks and their strong predictive power makes them a common choice for classification task. But as they increase in size, understandability by taking a look at the weights becomes impossible. With the rise of CNN in 2012 [8] Neural Networks have become state of art for image prediction. Special interpretability methods considering the learned weights are looked in detail in 3.3.

3 Interpretation of Image Recognition Neural Networks

In this chapter, evaluation method are described to analyze the behavior of not inherently interpretable models. Only evaluation methods which are commonly used for neural networks are introduced.

3.1 Global Model-Agnostic Methods

Global model-agnostic methods describe expected outcomes based on the distribution of the data. They can show a correlation between singular or multiple features and an outcome.

1. **Partial Dependency Plots:** Partial Dependency Plots (PDP) are easy to interpret but using it with several features becomes increasingly difficult. Therefore, using a PDP is not possible in Neural Networks.
2. **Accumulated Local Effects:** Accumulated Local Effects (ALE) are an an advancement of PDP. While it solves some problems PDP suffers from, it can't be applied to a Neural Network because of the complexity of the datasets.
3. **Feature Interaction:** Feature Interaction also analyzes how features correlate. Doesn't work due to the same problem of the complexity of the dataset.
4. **Functional Decomposition:** Functional Decomposition is commonly used in Neural Networks. See Chapter 3.3.1.
5. **Permutation Feature Importance:** Permutation Feature Importance is regularly used in visual machine learning tasks.
6. **Prototype and Criticism:** Prototype and Criticism is used as adversarial attacks in Neural Networks.

3.2 Local Model-Agnostic Methods

To explain individual predictions, Local model-agnostic methods are used. A single outcome is correlated to some features and explain the model.

1. **LIME: Local Interpretablle Model-agnostic Explanations:** Lime generates locally faithful explanations by training interpretable models on perterurbed instances of the original data.
2. **Local surrogate models:** Local surrogate models can be programmed to select a singular instance to explain.
3. **Scoped Rules (Anchors):** Find so called anchors to explain the predictions.
4. **Individual conditional expectation curves:** Practically not useful in image recognition, as the computation is too high.
5. **Counterfactual explanations:** Try to find a change while still resembling the original image.
6. **SHAPly:** Calculate SHAP values for an image prediction to determine how each pixel contributes to the prediction's deviation from the average prediction across all images. Positive SHAP values indicate pixels that push the prediction up, while negative values indicate pixels that pull it down.

3.3 Neural Network Interpretation

In the domain of Natural Language Processing (NLP) and Computer Vision, Deep Learning has proven very successful. By passing the input data through a sequence of layers, characterized by matrix-multiplications with kernel weights and nonlinear transformations functions, a prediction is computed. Depending on the specific task, additional elements like Long Short-Time Memory(LSTM) layers and Convolutional layers (CNN) are utilized. Given the immense amount of mathematical operations underlying a single prediction, humans are not fit to apprehend the mapping. To interpret predictions, we would have to decipher the intricate learned knowledge of numerous different kernels and weights. Recognizing that it's impossible for humans to grasp millions of weights, the demand for evaluation methods is high. To assess the behavior and predictions of Deep Neural networks, specific interpretation methods were developed. These methods calculate the likelihood of an input entry being responsible for the result.

While model-agnostic methods offer an approach to understand Neural Networks, the sheer size of the data used to train and test Neural Networks make this task extremely hard. For instance, an image with the dimensions of 3x224x224, as commonly encountered in Food-101,the data entries

exceed 150.000. In NLP tasks, where vocabularies often encompass around 20.000 words, the computational complexity renders most model-agnostic techniques as too expensive.
In the pursuit of comprehending the complexity of Deep Neural Networks it makes sense to utilize the weights in the model. The information saved in the hidden layers as learned weights can be used to evaluate the network. Moreover, the gradients can be taken into consideration as well. In the following subsections several concepts for understanding Deep Neural Networks are introduced.

3.3.1 Feature visualization and Network Dissection [11]

Modern Neural Networks like ResNet50 or Bard consist of several million layers. Network dissection attempts to overcome this challenge by breaking down separate layers and connecting them with ideas.

The higher-level features in these networks relate to clear concepts, shown in Figure 3.1. As the input image moves through layers, it changes at each layer. In each convolutional layer, the network gains new and more complex features. The smooth joining of fully connected layers then changes image-based data into predictions.

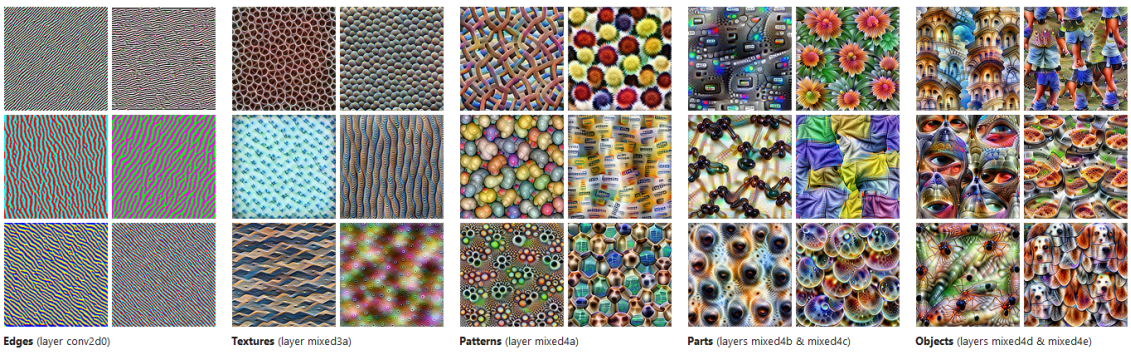

Figure 3.1: Feature Visualization [11]

The image explains this process. The first convolutional layers find simple features like edges and basic textures. Later, they recognize more detailed patterns. The deepest layers learn about parts and objects. This object information passes to the other hidden layers, which then finally make a prediction.

Feature visualization is based on activating one kernel in the network. This involves maximizing the activation of a specific neuron (Visible in 3.2). There are two methods for achieving this. First, we can make use of the training image that triggers the highest activation. Yet, this approach faces a significant problem. When an image contains multiple objects, it's hard to pinpoint which object causes the activation. Because of this an alternative route is adopted: generating new images

from random noise. This is accomplished through methods like Generative Adversarial Networks (GANs) or other diffusion-based techniques.

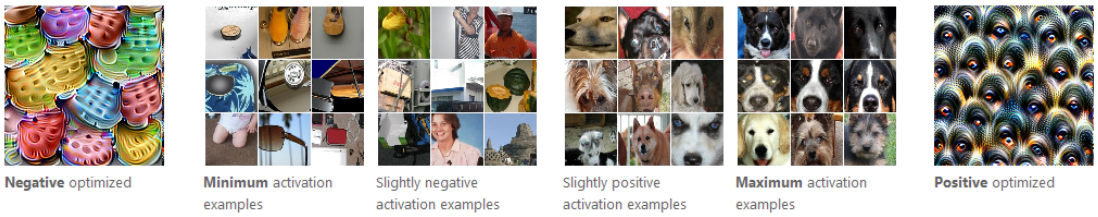

Figure 3.2: Activation Maximization [11]

Advantages of Feature Visualization:

1. **Initial Model Insights:** Feature visualization offer an initial view into a model's behavior, improving the understanding of its inner layers.
2. **Enhanced Domain Understanding:** It has the potential to enrich domain understanding by aligning learned features with domain-specific knowledge. An example can be seen in the medical industry
3. **Debugging and Improvement:** Feature visualization assists in debugging and refining models, contributing to their overall performance enhancement.

Disadvantages of Feature Visualization:

1. **Unclear Decision-Making:** While activations are evident, understanding the meaning behind them and how they contribute to decision-making remains challenging.
2. **Subjective Interpretation:** The interpretation of visualized features can be subjective, potentially leading to differing conclusions among observers.
3. **Limited Applicability to Visual Data:** Feature visualization's applicability is limited to visual data types.

3.3.2 Saliency Maps

Saliency maps are visualizations that highlight the regions of an input image that have the most significant impact on a model's output. By revealing the areas that strongly influence a prediction, saliency maps bridge the gap between the model's "black-box" nature and human understanding.

Saliency maps are commonly calculated using SHAP [9] or gradient methods. Saliency maps provide a direct and intuitive way to understand which parts of an input data are influencing a model's decision. The main difficulty is the generation of reliable saliency maps.

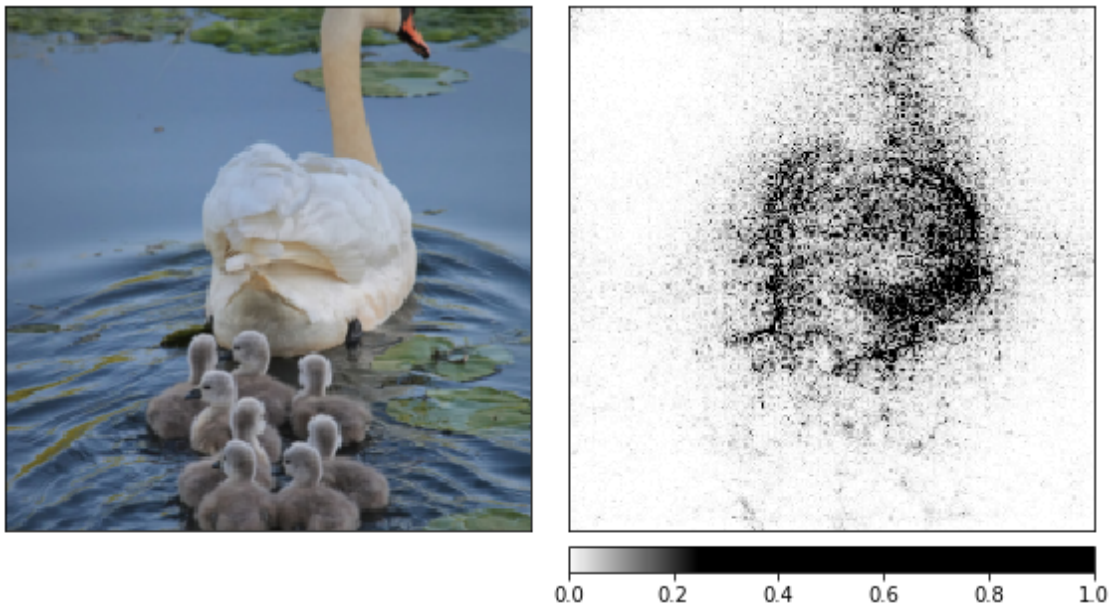

Figure 3.3: Saliency Map - Source: https://captum.ai/tutorials/Resnet_TorchVision_Interpret

3.3.3 Gradient-focused methods

Vanilla Gradient focuses on computing the gradients of the network. A forward pass of an image is generated. The gradients of the class score is computed. Then the gradients are visualized. This method has 2 problems: When ReLU is used and the gradients are negative, then information is lost. Furthermore, in pooling layers, there are no gradients and the information is lost.
Deconv Net [14] takes care of the gradient problem of Pooling and Convolutional Layers. Unpooling is done by recording the maxima in a set of switch variables. This preserves the initial structure of the stimulus. To obtain valid feature reconstructions, the reconstructed signal is passed through a relu non-linearity. The filtering using learned weights is done using transposed version of the same filter, applied to the rectified maps. This is similar to flipping the filter vertically and horizontally.

Option 2: Grad-CAM & Guided Grad-CAM

Grad-CAM provides visual explanations for CNN decisions. The goal is to understand at which parts of an image a convolutional layer looks for a certain classification. Grad-CAM analyzes which regions are activated in the feature maps of the last convolutional layer. The main problem of grad-CAM is that it's not very exact. Through the CNN's it's unclear which pixels are exactly responsible.

Guided Grad-CAM computes Grad-CAM with another method to have a better localization. It's basically multiplied with another method to achieve stable results.

Option 3: Top-Down Neural Attention by Excitation Backprop [15]

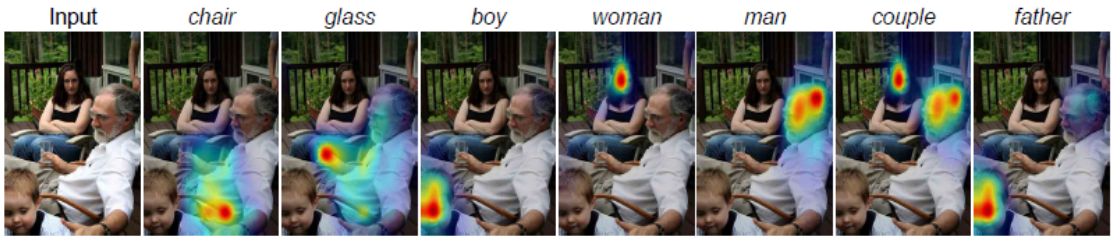

Figure 3.4: CNN classifier's top-down attention map [15]

By calculating the importance of a neuron set and saving it during the computational step, the gradient step can be combined with the importance and localize exact objects. The probabilistic WTA formulation produce well-normalized attention maps that enable direct subtraction.

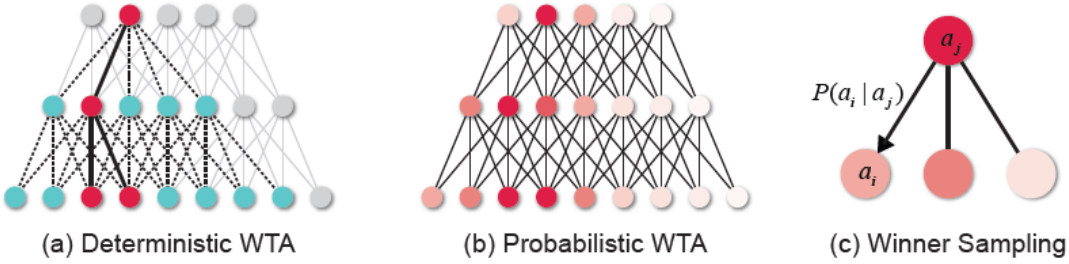

Figure 3.5: Identifying task-relevant neurons in the network. The red shading of a dot indicates its relative likelihood of winning against the other ones in the same layer. [15]

4 Evaluation of post-hoc interpretability methods

Gradient-methods which generate saliency maps are hard to measure. While humans can evaluate the maps by giving a general statement, this can not be applied to thousands of images. Despite many significant recent contributions to saliency maps, the valuable effort of explaning machine learning models face this methodological challenge: the difficulty of assessing the scope and quality of model explanations [1].

4.1 Evaluation metrics, ground truths [6]

Different evaluation methods exist to evaluate saliency methods. Evaluations can be extrinsic [6], involving human evaluations and comparing the results to certain ground truth explanations [15]. Intrinsic methods use computations involving the net itself and the saliency map, without human evaluation or retraining a net. These methods are based on creating a new composite input using the heat map and the original input. Then they are evaluated using the original trained model. E.g.[3] These methods suffer from violating one key assumption in machine learning: the training and evaluation data must come from the same distribution. [7] Without re-training it is not clear if the degrade in performance stems from the distribution shirt or because informative features were removed.

4.2 Completeness and Soundness

[6] proposes Soundness and Completeness, two concepts which are required for evaluation metrics which involve using a composition of a heat map and the original input.
Soundness is needed: The masked input method proves that a certain part of the image "caused" the networks output.

Completeness means, for any composition of an input image of label a with a well functioning saliency mask the model must still be able to identify the correct label.
$\forall x : f(x \rightarrow a)$
$x = input(a) \odot mask$
Soundness means, for any composition of an input image of label b with a well functioning saliency

mask them model must not return a wrong label.
$\nexists x : f(x \rightarrow a)$
$x = input(b) \odot mask$

The AUC metric [12] of the insertion game:
For s = [1,dim(x)] take the top s pixels as per saliency map m and plot the probability f(x,a) given by the model. The top s pixels of x are retained and the remaining pixels are assigned a default value). Return the area under the curve.

The method is α complete on f,x,a if:

$$g_{AUC}(x, a, m) >= \alpha f(x, a)$$

The method is β sound on f,x,a if:

$$g_{AUC}(x, a, m) <= \frac{1}{beta} \alpha f(x, a)$$

$$\alpha(x, a) = min(\frac{max(g_{AUC}(x, a, n), \epsilon_1)}{f(x, a)}, 1)$$

$$\alpha(x, a) = min(\frac{max(f(x, a), \epsilon_2)}{g_{AUC}(x, a, n)}, 1)$$

Then $\alpha(m) = E_x[min_a(\alpha(x, a))]$
Then $\beta(m) = E_x[min_a(\beta(x, a))]$

If alpha is close to 1: The blocked input predicts correctly.
If beta is close to 1: The blocked input mimics the behavior and is not overconfident.

x: input to the model
a: true label
f: model
m : saliency method.
ϵ = term if x is very small

Older evaluation methods only try to maximize the g_{AUC} curve. This completely ignores if the method is overconfident.

Critics to the method: No human interpretability is included. Furthermore, the masked images do suffer from not having the same distribution as in the original trained maps. This could cause a unwanted drop in performance.

4.3 Perturbation based [7552539]

4.4 A benchmark for interpretability methods in deep neural networks [7]

In this paper two methods to estimate the effectiveness of saliency maps are presented. "RemOve And Retrain (ROAR)" and "Keep And Retrain (KAR)".

ROAR proposes a numerical solution to evaluate attribution maps. It is a algorithm which estimates the effectiveness of saliency maps. This is done, by removing supposedly informative features from the input and observing the reaction of the neural network. ROAR is applicable to any visual domain. In Section 5, an example for MNIST and Food-101 is done.

RoaR works as follows:

1. **Selection of the most important pixels from an attribution map**
   An attribution map provides a mapping to the most important pixels. Those pixels are ranked based on their importance. Depending on the algorithm, different values differ on the importance. In the case of Integrated Gradient, the pixels with the highest absolute values are considered the most important.

2. **Replacement of x% of the pixels with the mean value**
   In the Roar algorithm, 0.1, 0.3, 0.5, 0.7 or 0.9 % of the most important pixels are identified and replaced by the mean value. Both the training and the testing data undergo this process.

3. **Retraining the model using the modified dataset**
   To ensure the consistency of the model, retraining is necessary. Afterwards, the model is evaluated using several seed runs on the test set. The same split should be used for all runs. This is crucial, because the training data and the test data must be drawn from the same distribution. Without retraining the model this property is ignored.

4. **Comparison of the attribution map with a random baseline**
   For each training setup, respectively with 0.1, 0.3, 0.5, 0.7 or 0.9 % pixels replaced, a random baseline is considered. The random baseline also replaces the same percentage of pixels. The random baseline is expected to perform worse than a sophisticated method.

5. **Evaluation**
   An attribution method is deemed effective, if it consistently outperforms the random baseline across various setups.

KAR works similar: Instead of removing the most important pixels, the least important pixels are identified. Because KAR performs worse than ROAR, it is not described further.

The conclusion of [7] is that commonly used based estimator, Gradients, Integrated Gradient and Guided BackProp are worse than a random assignment. The effectiveness of Smooth-Grad-Squared and VarGrad was proofed.

4.5 Benchmarking Attribution Methods (BAM) [13]

BAM with relative feature importance explores wether features are important or not. In reality, we do not know how important a feature is. However, we can calculate how important a feature is to model relative to another model. Given the relative feature importance, the metrics compare attributions between pairs of models and pairs of input.

The main idea of BAM is: If you paste a grey square to every training image for all classes, it is expected that this square matters less than the original image region being covered. The same expectation should hold if instead an object (which is not represented by the dataset) is pasted in every image. Any explanations that assign higher attribution to the inserted object than to the original pixels are false positives.

BAM solves the problem of an distribution shift by inserting objects by only inserting objects which don't strongly change the distribution of the image. E.g. an object with mean similar to the mean of the dataset is chosen.

The BAM algorithm works as follows:

1. **BAM dataset construction**
   The BAM dataset is constructed by pasting object pixels from MSCOCO [**lin2015microsoft**] into scene images from MiniPlaces [**zhou2016places**]. An object is re-scaled to between 1/3 to 1/2 of a scene image at a randomly chosen location. Resulting images have an object label and a scene label. Either can be used to train a classifier. Every object class appears in every scene class and vice versa. Scenes which contain original BAM objects are not used.

2. **Common features and commonality**
   Common features are defined as a set of pixels with semantic meaning (e.g. looks like a dog) which commonly appear in all examples of one or more classes. For example, a dog

which appears in all bamboo forests is less common than a dog which appears in all images of bamboo forests, bedroom and corn field.

3. **2 Classifiers are trained and attribution maps are created**
   An object detection classifier and a scene detection classifier are trained. Objects should be significantly more important than the scene to the object detector than to the scene detector. To verify this intuition, the objects are removed. With this knowledge, attribution maps can be tested by checking if the pixels are assigned noticeably higher attributions.

4. **Relative importance is calculated.**
   With this knowledge, the object pixels should be more important than any other pixels for the detection classifier. Additionally for the scene attribution, the attribution maps should be higher if the object pixels are replaced with the original pixels.

The advantage of BAM over ROAR [7] and other methods is the lower computational cost. No retraining or perturbation is required.

4.6 Sanity Checks for Saliency Maps [1]

In this paper an actionable methodology is proposed, which kinds of explanation a given method can and cannot provide. It shows that visual inspection by humans does not provide good information if the explanation is sensitive to the underlying model and data.

Two instances of the framework are tested:

1. **Model Parameter randomization test**
   The model parameter randomization test compares the output of a salinecy method on a trained model with the output of a randomly initalized untrrained network of the same architecture. The output should differ, otherwise the saliency map is deemed as not helpful.

2. **Data randomization test**
   The data randomization test randomly shuffles the labels of the data and the model is trained on this method. If the saliency maps does not differ to a normally trained model, then the method does not depend on the relationship of the images and labels.

   If any of the 2 hypothesis is failed, then the method can safely be rejected. This is a so called sanity check.

   Extensive experiments on several explanation methods are done across data sets and model architectures.

   On the tested methods, Gradients & GradCam pass the Sanity checks, while Guided BackProp & Guided GradCAM fail.

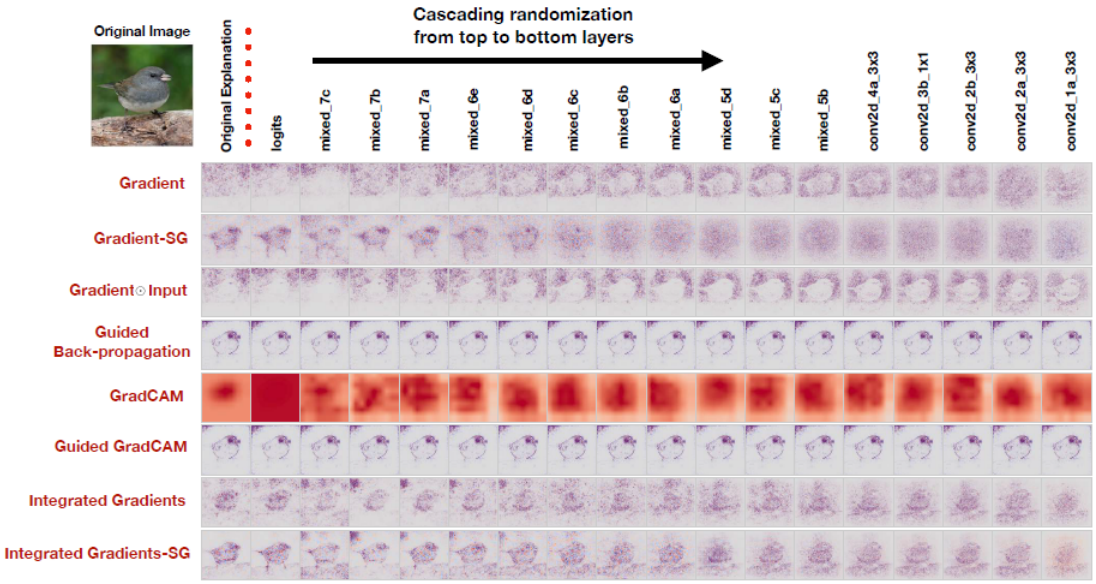

Figure 4.1: Cascading Randomization on Image Net. The figure shows the original explanations. Progression from left to right indicate complete randomization of network weights up to that block inclusive. The last block corresponds to a network with completely reinitialized weights.[1]

4.7 Comparison of the Evaluation Methods & critique

While the described methods both offer a numerical evaluation method, they do still lack a clear evaluation structure. They show, that some evaluation methods are indeed correct, but they still suffer ambuigity from different datasets.

5 Project work

5.1 Project Goal

In RoaR[7] the paper does not list the standard deviation of the trained nets. We expect to validate the results by achieving similar results.

As training 25 image nets requires high computational power we do not have right now, we limited our research to evaluating food-101 [2] using only 2 interpretation methods and comparing it to the baseline.

Additionally, we also add a mini evaluation using the MNIST dataset.

5.1.1 Project Setup

5.1.2 Results and Plots

5.1.3 Discussion of results

Bibliography

[1] Julius Adebayo et al. *Sanity Checks for Saliency Maps*. 2020. arXiv: 1810.03292 [cs.CV].

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. "Food-101 – Mining Discriminative Components with Random Forests". In: *European Conference on Computer Vision*. 2014.

[3] Piotr Dabkowski and Yarin Gal. *Real Time Image Saliency for Black Box Classifiers*. 2017. arXiv: 1705.07857 [stat.ML].

[4] Li Deng. "The mnist database of handwritten digit images for machine learning research". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[5] Jerome H. Friedman, Trevor Hastie, and Rob Tibshirani. "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1 (2010), pp. 1–22. DOI: 10.18637/jss.v033.i01. URL: https://www.jstatsoft.org/index.php/jss/article/view/v033i01.

[6] Arushi Gupta et al. *New Definitions and Evaluations for Saliency Methods: Staying Intrinsic, Complete and Sound*. 2022. arXiv: 2211.02912 [stat.ML].

[7] Sara Hooker et al. *A Benchmark for Interpretability Methods in Deep Neural Networks*. 2019. arXiv: 1806.10758 [cs.LG].

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Commun. ACM* 60.6 (2017), pp. 84–90.

[9] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI].

[10] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. 2022. URL: https://christophm.github.io/interpretable-ml-book.

[11] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. "Feature Visualization". In: *Distill* (2017). https://distill.pub/2017/feature-visualization. DOI: 10.23915/distill.00007.

[12] Vitali Petsiuk, Abir Das, and Kate Saenko. *RISE: Randomized Input Sampling for Explanation of Black-box Models*. 2018. arXiv: 1806.07421 [cs.CV].

[13] Mengjiao Yang and Been Kim. *Benchmarking Attribution Methods with Relative Feature Importance*. 2019. arXiv: 1907.09701 [cs.LG].

[14] Matthew D. Zeiler, Graham W. Taylor, and Rob Fergus. "Adaptive deconvolutional networks for mid and high level feature learning". In: *2011 International Conference on Computer Vision* (2011), pp. 2018–2025. URL: https://api.semanticscholar.org/CorpusID:975170.

[15] Jianming Zhang et al. "Top-Down Neural Attention by Excitation Backprop". In: *International Journal of Computer Vision* 126 (Oct. 2018). DOI: 10.1007/s11263-017-1059-x.