

Axiomatic Attribution for Deep Networks*

Explainable Machine Learning
Master Seminar SS24



UNIVERSITÄT
DES
SAARLANDES

MAX PLANCK INSTITUTE
FOR INFORMATICS



*Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks." *International conference on machine learning*. PMLR, 2017.

Introduction

Motivation

Introduction

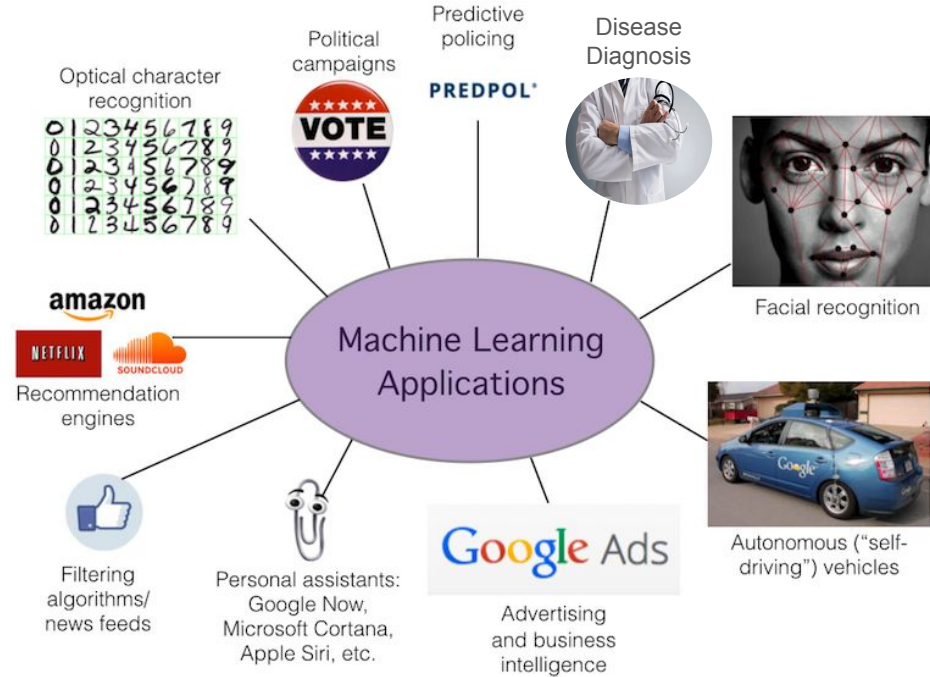


Image source: <https://www.redshiftzero.com/post/manipulation-ml/>

Motivation

Introduction

Understanding these model is absolutely critical in several domains particularly those involving high stakes decisions!



Autonomous ("self-driving") vehicles



Disease Diagnosis

Let's start with why?

Introduction





Top 5 labels and scores

fireboat: 0.999961

pier: 3.67026e-05

steel arch bridge: 6.33059e-07

crane: 3.84465e-07

liner: 2.96321e-07

Why did the network label this image as “fireboat”?

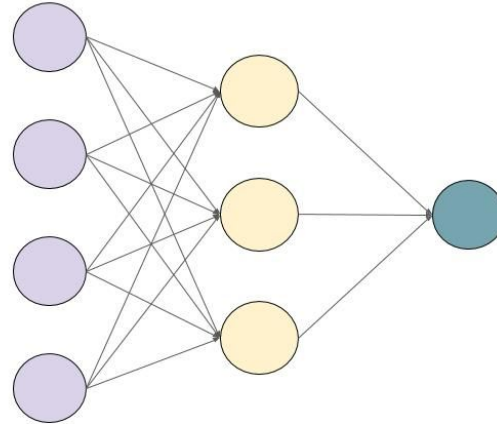
Why did the model label this image as “clog”?

Input image



Introduction

Clog



Text

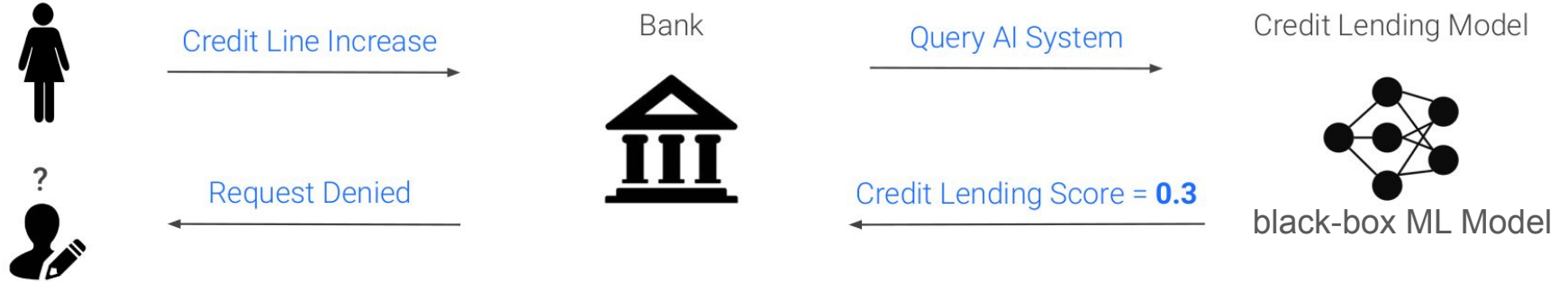
Introduction

Beautiful design and execution, both the space itself and the food. Many interesting options beyond the excellent traditional Italian meatball. Great for the Financial District.

Why did the network predict **positive sentiment** for this review?

Credit Lending

Introduction



Why? Why not? How?

Confusion & Doubt for End-User

Introduction

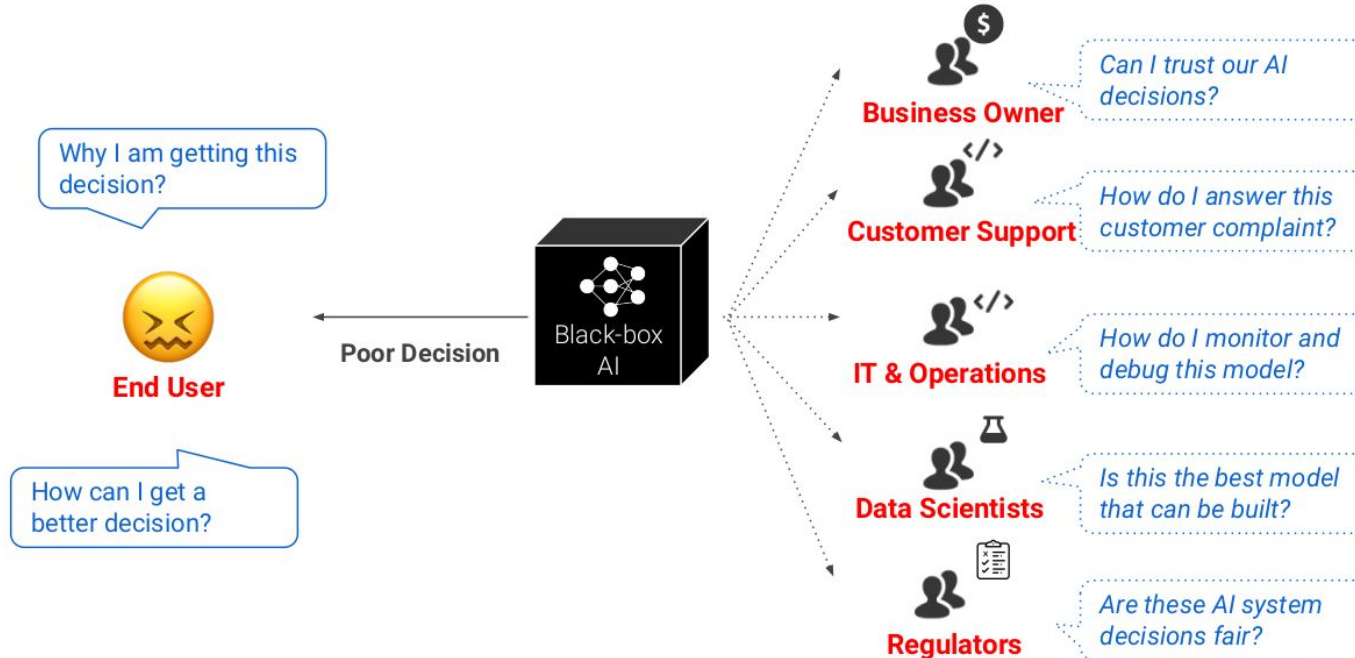


Image source: <https://www.fiddler.ai/blog/series-a-why-now-and-whats-next-for-fiddler>

Attributing the prediction of deep network to its input features.

- E.g. Attribute an Object Recognition Network's prediction to its pixels
- E.g. Attribute a text sentiment network's prediction to individual words
- E.g. Attribute a lending model's prediction to its features

A reductive formulation of “why this prediction” **but surprisingly useful :-)**

- **Debugging model prediction**
 - Eg. Attribution an image misclassification to the pixel responsible for it
- **Generating an explanation for the end-user**
 - Eg. Expose attribution for a lending prediction to the end-user
- **Analyzing model robustness**
 - Eg. Craft adversarial examples using weaknesses surfaced by attributions
- **Extract rule from the model**
 - Eg. Combine attribution to craft rule (pharmacophores) capturing prediction logic of drug screening network

Gradients

Feature*Gradients

Gradients

Reasonable starting point:

Product of gradients (output w.r.t to the input) and feature value.

$$x_i \times \frac{\partial Y}{\partial x_i}$$

Baehrens, David, et al. "How to explain individual classification decisions." (2010)

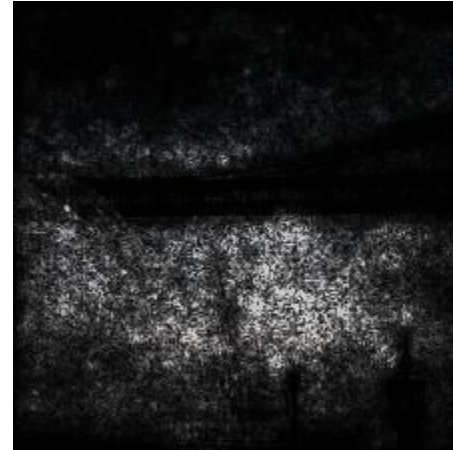
Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps." (2013).

Gradients

Original Image



Gradients

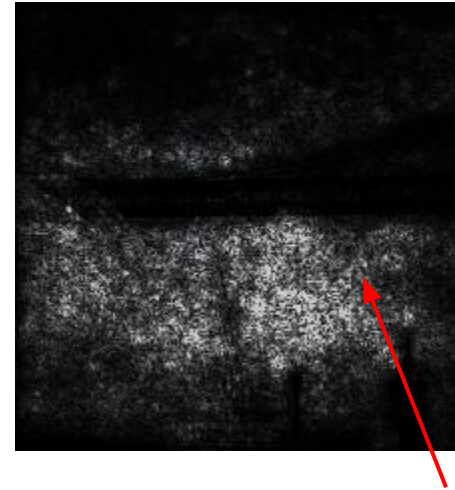


Gradients

Original Image



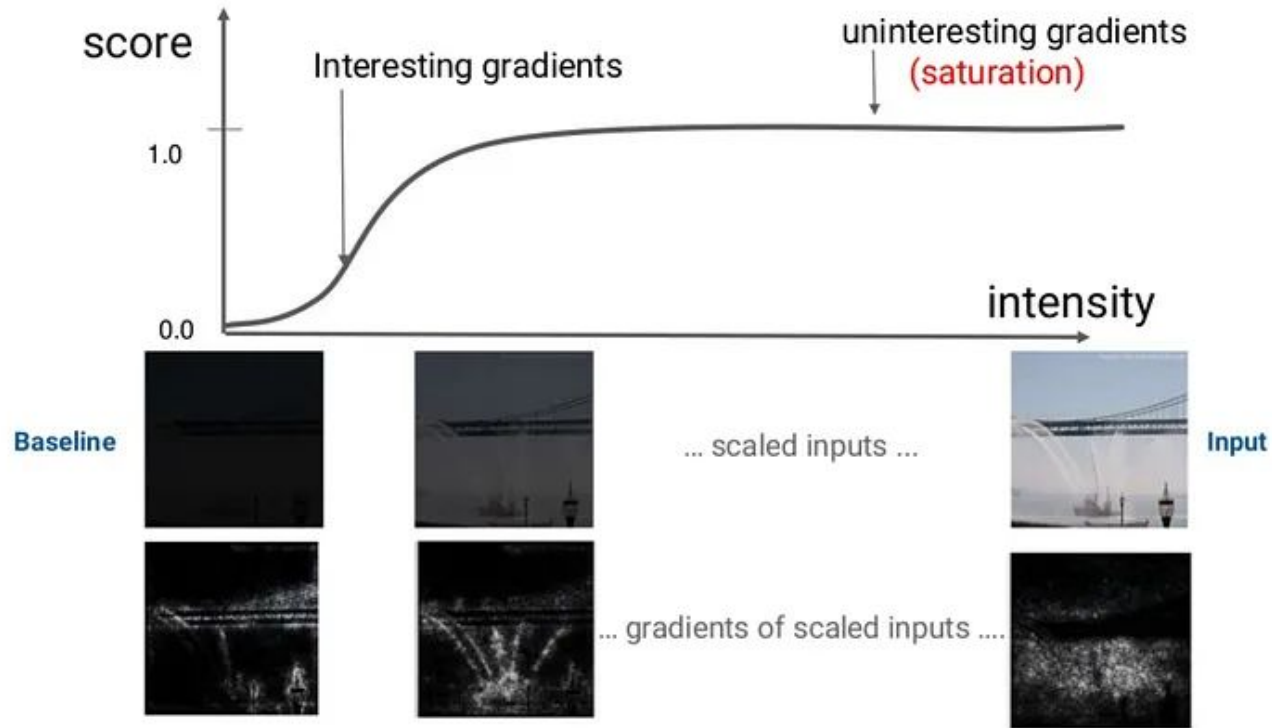
Gradients



Gradients in the vicinity of input
looks like noise

Feature*Gradients

Gradients



Integrated Gradients

Attribution method (Formally)

IG

$$F : \mathbb{R}^n \rightarrow [0, 1]$$

↳ deep network

$x = (x_1, \dots, x_n) \in \mathbb{R}^n \rightarrow$ the input at hand

$x' \in \mathbb{R}^n \rightarrow$ baseline input (e.g., the black image or
zero embedding vector for text)

$A_F(x, x') = (a_1, \dots, a_n) \in \mathbb{R}^n \rightarrow$ attribution of the prediction at
input x relative to a baseline input x'

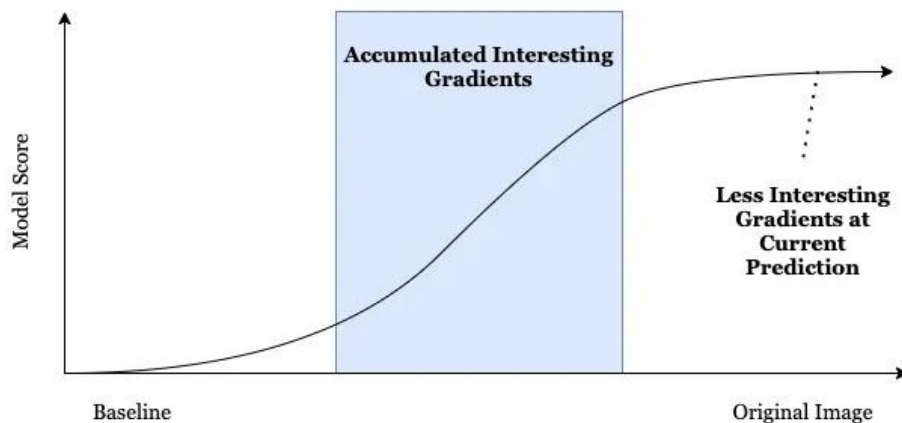
$a_i \rightarrow$ contribution of x_i to the prediction $F(x)$

Interesting Gradients

IG

The gradient at baseline and at data point are going to be close to 0.

All the interesting — and informative — gradients are in between the two data points, so ideally we would find a way to capture all of that information too.



Accumulated Grads.

IG

Baseline
Image



Input
Image

$$\phi_i^{IG}(f, x, x') = \underbrace{(x_i - x'_i)}_{\text{Difference from baseline}} \times \underbrace{\int_{\alpha=0}^1 \frac{\delta f(x' + \alpha(x - x'))}{\delta x_i} d\alpha}_{\substack{\text{From baseline to input...} \\ \dots \text{accumulate local gradients}}}$$

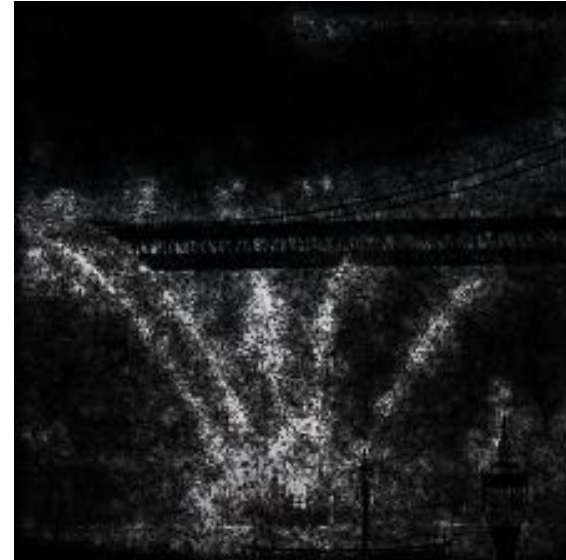
Visualization

IG

Original Image



Integrated Gradients



Baseline

Need of baseline

Baseline

Counterfactual intuition:

Assign blame (or credit) to a certain cause and implicitly consider the absence of the cause as a baseline for comparing outcomes

Eg. For most deep networks, a natural baseline in input space is where the prediction is natural.

Eg. For object recognition, it can be black image.

Baseline Example

Baseline

The probability Bob wants ice cream

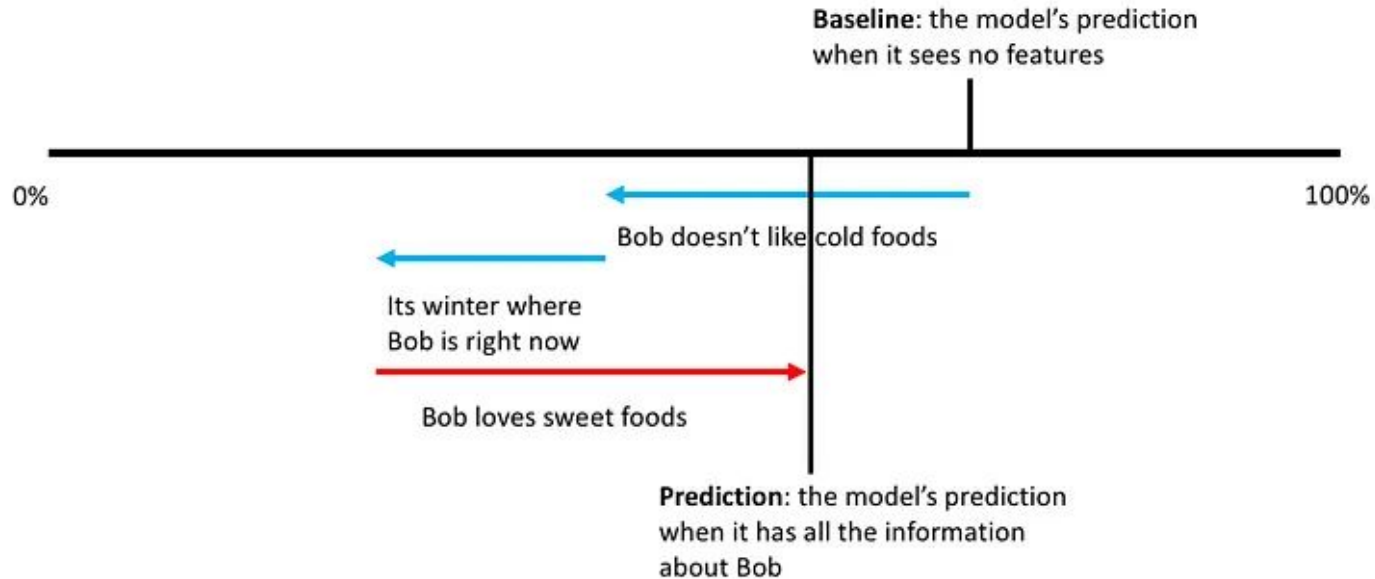


Image Source: <https://towardsdatascience.com/interpretable-neural-networks-45ac8aa91411>

Evaluation

Pixel Perturbation Method

Evaluation

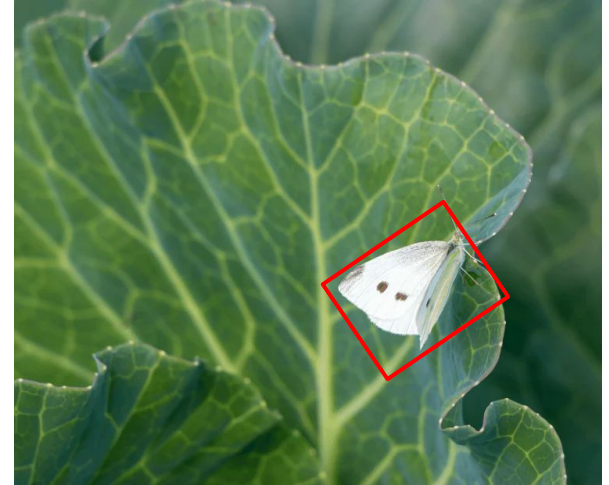
- Evaluates attribution by varying top k pixel intensities and measuring score drop; large drop indicates good attribution.
- **Issue:** Perturbed images may appear unnatural, leading to score drops because the network hasn't seen similar images during training.



Human-Drawn Bounding Boxes

Evaluation

- Uses human-drawn bounding boxes to compute pixel attribution percentage within the box, assuming object pixels are most important.
- **Issue 1**: Context pixels, such as background elements, can also be important, complicating the interpretation of attribution results.
- **Issue 2** : Confirmation bias



- Prior work is characterized by intuitive design + empirical eval
- Challenges with evaluation
 - Evals from prior work are specific to object recognition
 - Hard to separate model behavior, attribution errors or artifacts from perturbing from data
 - Attributions may “look” incorrect because of unintuitive network behavior
 - Perturbations could change score for artifactual reasons like if they create a “new object” or a “crazy” input

Prior work

Evaluation



Axioms

~ desirable criteria for an attribution method

When two neural networks compute the same mathematical function $F(x)$, regardless of how differently they are implemented, the attributions to all features should always be identical.

E.g. $f1 = x*(y*z)$ and $f2 = (y*x)*z$ should get the same attributions

- DeepLift and Layerwise relevance propagation (LRP) break Implementation Invariance!

Score Back-Propagation based Methods

Re-distribute the prediction score through the neurons in the network

- LRP [JMLR 2017], DeepLift [ICML 2017], Guided BackProp [ICLR 2014]

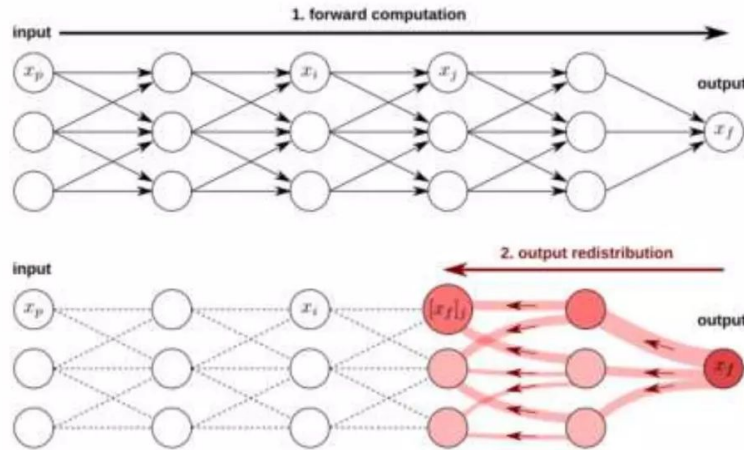


Image credit heatmapping.org

Tricky case: Output of a neuron is a **non-linear** function, e.g., ReLU, Sigmoid, etc.

- **Guided BackProp:** Only consider ReLUs that are on (linear regime), and which contribute positively
- **LRP:** Use first-order Taylor decomposition to linearize activation function
- **DeepLift:** Distribute activation difference relative a reference point in proportion to edge weights

43

- (a) For every input and baseline that differ in one feature but have different predictions, the differing feature should be given a non-zero attribution.
 - (b) If the function implemented by the deep network does not depend (mathematically) on some variable, then the attribution to that variable is always zero.
-
- DeConvNets and Guided back-propagation violate Sensitivity (a)!

Sensitivity

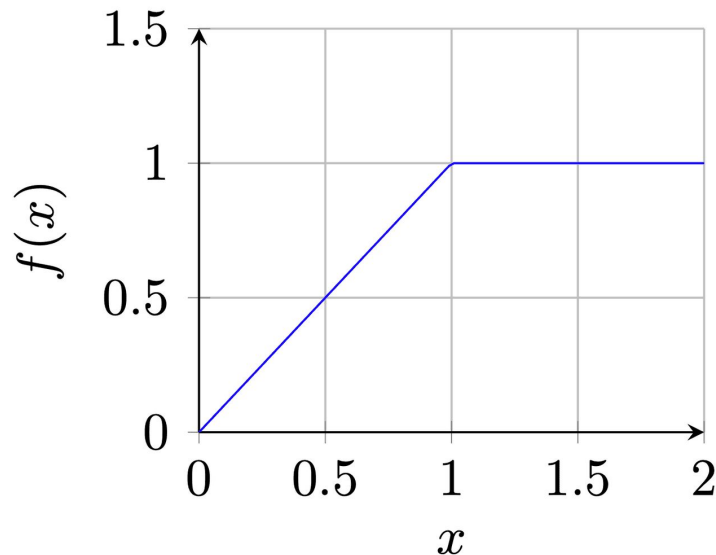
Axioms

$$F(x) = 1 - \text{ReLU}(x)$$

Baseline $x' = 0$

Input $x = 2$

- functions' output changes from 0 to 1 but after $x = 1$, it becomes flat and causes the gradient to equal zero.
- x attributes to the result, but the function is flat at the $x = 2$, results in invalid attribution and breaks the Sensitivity



The attributions preserve any linearity within the network.

If we linearly compose two neural networks computing F and G to form a new neural network that computes $H = aF + bG$, then for any feature x_i , its attribution for H is a weighted sum of the attributions for F and G , with weights a and b respectively

$$A_i^H(\mathbf{x}, \text{base}) = a \cdot A_i^F(\mathbf{x}, \text{base}) + b \cdot A_i^G(\mathbf{x}, \text{base})$$

Rationale: Attributions have additive semantics. It is best to respect any existing linear structure.

$$\sum_i \phi_i^{IG}(f, x, x') = f(x) - f(x')$$

Theorem [Friedman 2004] : Every method that satisfies Linearity preservation, Sensitivity and Implementation invariance, and Completeness is a **path integral** of a gradient.

Symmetric variables with identical values get equal attributions

E.g. For $f = x*y + 1$, the "optimal" attribution at $x,y = (1, 1)$ should be equal for x and y .

Theorem: Integrated Gradients is the unique path-integral method satisfying:
Sensitivity, Insensitivity, Linearity preservation, Implementation invariance,
Completeness, and Symmetry.

Integrated Gradients In Practice

Integrated Gradients is incalculable (because of the integral). So, approximate value by replacing the integral with the summation:

$$\text{IntegratedGrads}_i^{\text{approx}}(x) ::= (x_i - x'_i) \times \sum_{k=1}^m \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m}$$

m defines a number of interpolation steps

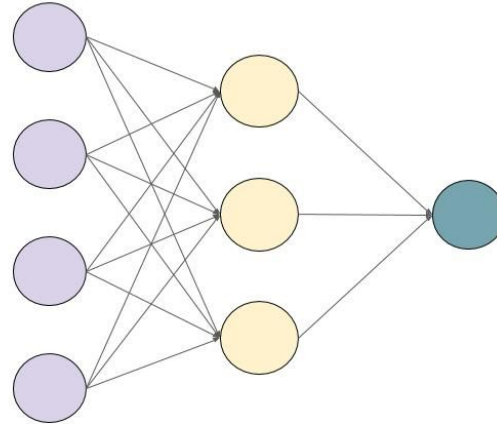
```
def integrated_gradients(inp, baseline, label, steps=range(50)):  
    t_input = input_tensor() # input tensor  
    t_prediction = prediction_tensor(label) # output tensor  
    t_gradients = tf.gradients(t_prediction, t_input)[0] # gradients  
    path_inputs = [baseline + (i/steps)*(inp-baseline) for i in steps]  
    grads = run_network(t_gradients, path_inputs)  
    return (inp-baseline)*np.average(grads, axis=0) # integration
```

Application

Why this image labeled as “Clog”?

Application

Input image



Clog



Why this image labeled as “Clog”

Application

Input image

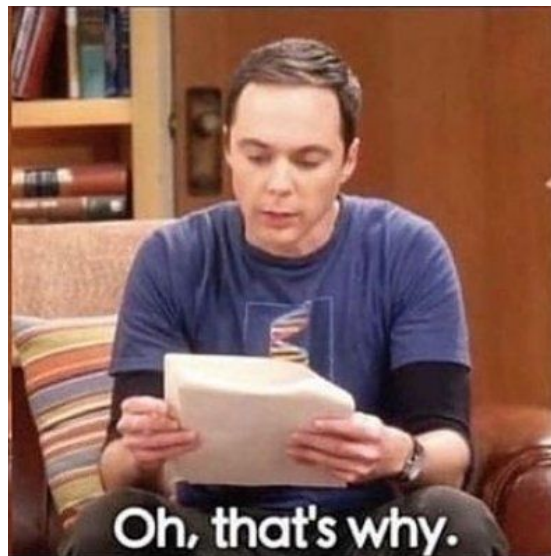


IG for label “Clog”



Clog

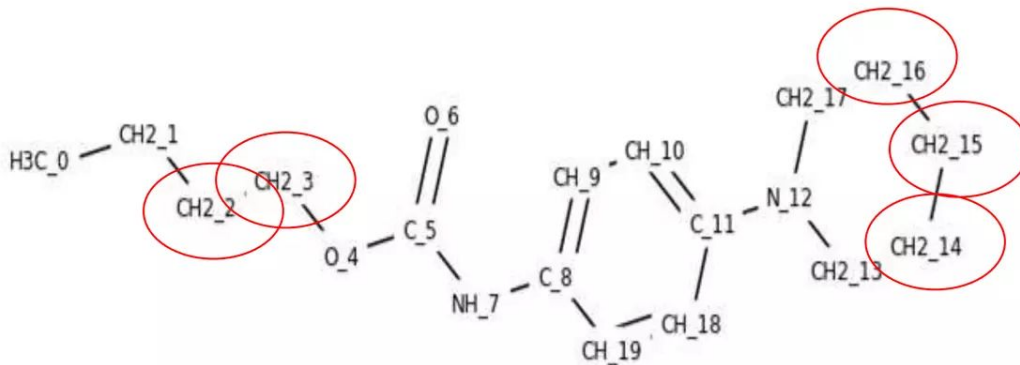




Bug Detection

Application

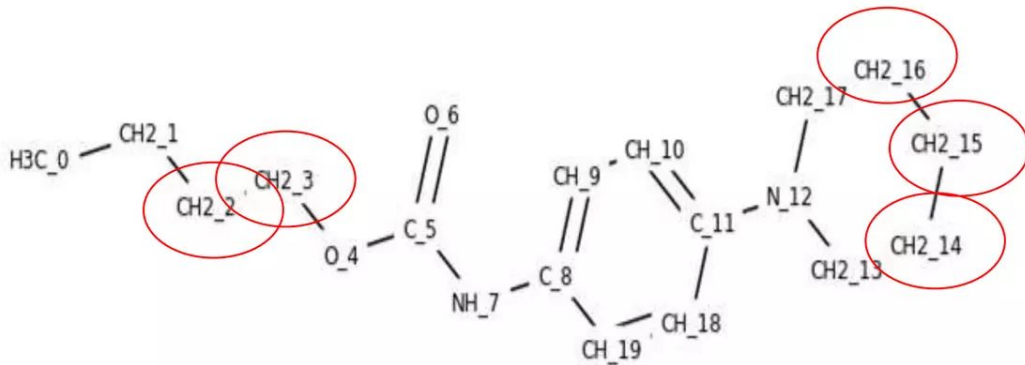
- Deep network [Kearns, 2016], predicts if a molecule binds to certain DNA site
- Finding: Some atoms had identical attributions despite different connectivity



Bug Detection

Application

- Deep network [Kearns, 2016], predicts if a molecule binds to certain DNA site
- Finding: Some atoms had identical attributions despite different connectivity



Bug: The architecture had a bug due to which the convolved bond features did not affect the prediction

Detecting a data issue

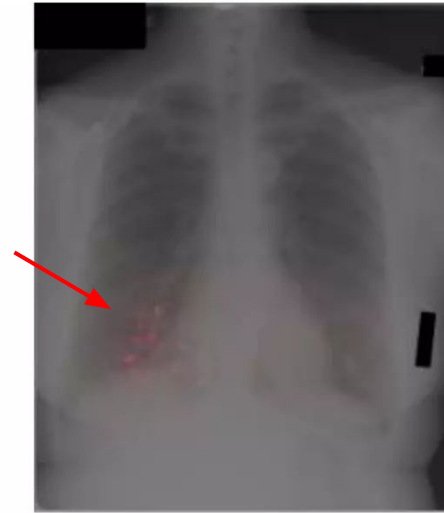
Application

- Deep network predicts various diseases from chest x-rays

Original image



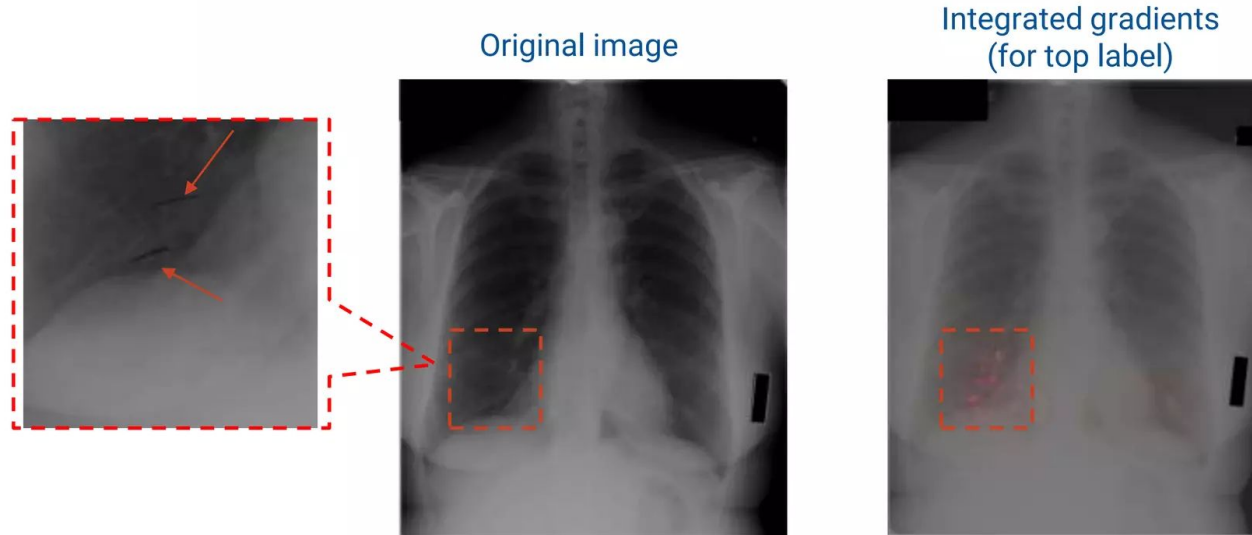
Integrated gradients
(for top label)



Detecting a data issue

Application

- Deep network predicts various diseases from chest x-rays
- Finding: Attributions fell on radiologist's marking (rather than the pathology)

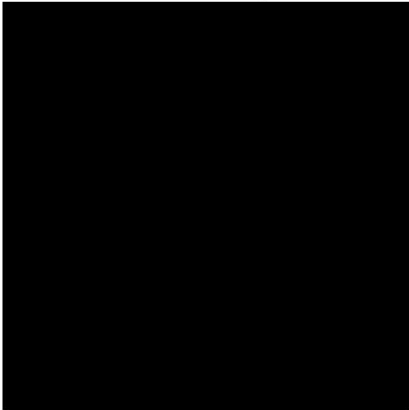


Limitation

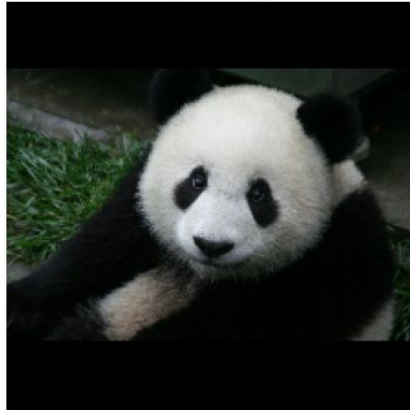
Picking baseline

Limitation

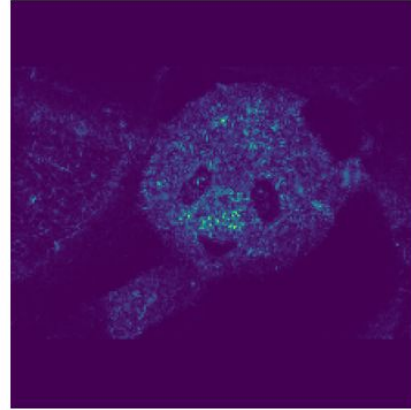
Baseline image



Original image



Attribution mask



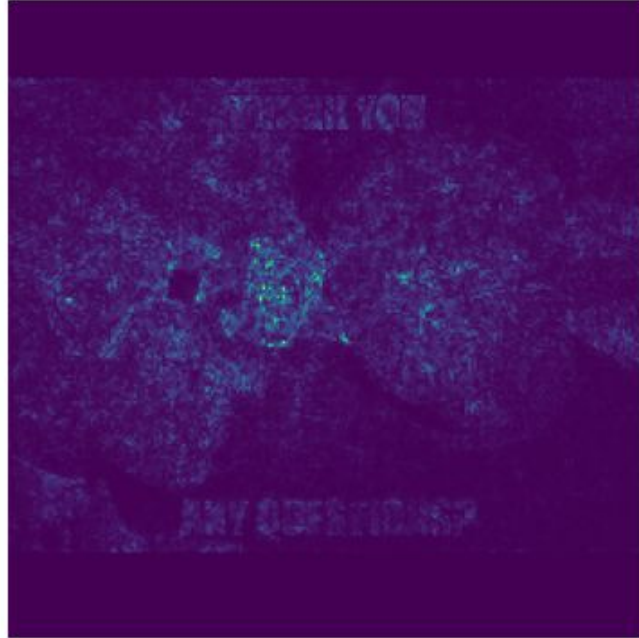
Overlay



- **Local, Not Global Interpretability**
 - IG provides feature importance for individual examples but does not offer global feature importance across datasets.
- **Explains Individual Features, Not Interactions**
 - IG explains network predictions in terms of individual features using a first-order linear approximation, but it does not reveal how features interact or are combined by the network.
- **Applicable Only to Differentiable Models**
 - IG can be used with differentiable models like neural networks but not with non-differentiable models such as tree-based models without modifications.
- **Baseline Selection and Visual Inspection Limitations**
 - Accurate IG interpretation depends on selecting an appropriate baseline.

Any guesses? What could be the label for this image?

Attribution mask

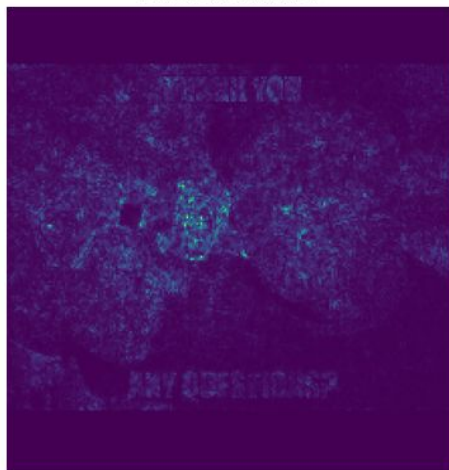


revolver: 33.1%, can opener: 13.5%, loupe: 3.9%

Original image



Attribution mask



Overlay



Discussion