

# Challenging deep learning models in real-world applications

Looking for explainability

Céline Hudelot  
Ecole d'été Peyresq 2022

# Outline

Motivations and Overview

XAI and Deep Learning

Explanation with examples

Explainable deep models

Conclusion



# Explanations in AI

Explanation in AI aims to create a suite of techniques that produce **more explainable models**, while maintaining a high level of searching, learning, planning, reasoning performance, optimization, accuracy, precision ; and enable human users to **understand, appropriately trust, and effectively manage the emerging generation of AI systems.**

# The Need of explanations ?

# Does an AI have to explain itself?



**Yann LeCun** @ylecun · Feb 5, 2020

We often hear that AI systems must provide explanations and establish causal relationships, particularly for life-critical applications. Yes, that can be useful. Or at least reassuring....

1/n



**Yann LeCun** @ylecun · Feb 5, 2020

But sometimes people have accurate models of a phenomenon without any intuitive explanation or causation that provides an accurate picture of the situation. In many cases of physical phenomena, "explanations" contain causal loops where A causes B and B causes A.

2/n



**Yann LeCun** @ylecun · Feb 5, 2020

A good example is how a wing causes lift. The computational fluid dynamics model, based on Navier-Stokes equations, works just fine. But there is no completely-accurate intuitive "explanation" of why airplanes fly.

3/n



# Does an AI have to explain itself?

## Round Table XAI, DigiHall Days 2019

*Why would an AI need to explain its decisions? When you go to the doctor, you don't ask him why he/she is prescribing you this or that medication and why he/she has made that diagnosis.*

Anonymous

# Does an AI have to explain itself?

However, humans do not like black boxes for decision making

- ▶ *The Black Box Society* - Franck Pasquale- *The Secret Algorithms That Control Money and Information*
- ▶ *Le temps des algorithmes* - Gilles Dowek, Serge Abiteboul
- ▶ ...

# The Need of explanations

- ▶ AI systems are often involved in **decision-making tasks**, a task which, in its nature, need explanations.
  - ▶ More than the output(s) of a prediction : provide elements of evidence, explanatory elements.
- ▶ **Legal aspects** : RGPD (right to explanation)
- ▶ **Human agency and oversight** : Make the decision transparent, understandable and explainable : to allow interaction and communication with users.
- ▶ **For trust and acceptance.**

# Explainability as part of a growing Global AI Policy and Regulation

- ▶ UNESCO : Elaboration of a Recommendation on the ethics of artificial intelligence<sup>1</sup>
- ▶ European Commission : Ethics guidelines for trustworthy AI<sup>2</sup>
- ▶ European GDPR : Article 22 empowers individuals with the **right to demand an explanation** of how an automated system made a decision that affects them.
- ▶ Algorithmic Accountability Act 2019<sup>3</sup> : Requires companies to provide an **assessment of the risks** posed by the automated decision system to the privacy or security and the risks that contribute to inaccurate, unfair, biased, or discriminatory decisions impacting consumers.
- ▶ ...

---

1. <https://unesdoc.unesco.org/ark:/48223/pf0000373434>

2. <https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai>

3. <https://www.congress.gov/bill/116th-congress/house-bill/2231/all-info>

# What is eXplainable Artificial Intelligence ?



# XAI :eXplainable Artificial Intelligence

## The renewal : The DARPA program

- ▶ In 2016, DARPA is launching a new program to fund the research in AI, with a constraint of explainability.
- ▶ David Gunning, a researcher involved in this program introduced the name **eXplainable Artificial Intelligence** and the acronym **XAI**.

## Definition

The goal of an XAI system is to make its behavior more intelligible for humans by providing them with explanations. Such a system must be capable of :

- ▶ Explaining its rationale ;
- ▶ Characterizing its strengths and weaknesses ;
- ▶ Conveying an understanding of how it will behave in the future.

## Definition of Arrieta et al <sup>a</sup>

a. <https://www.sciencedirect.com/science/article/pii/S1566253519308103>

Given a certain audience, explainability refers to the details and reasons a model gives to make its functioning clear or easy to understand.

# XAI : eXplainable Artificial Intelligence

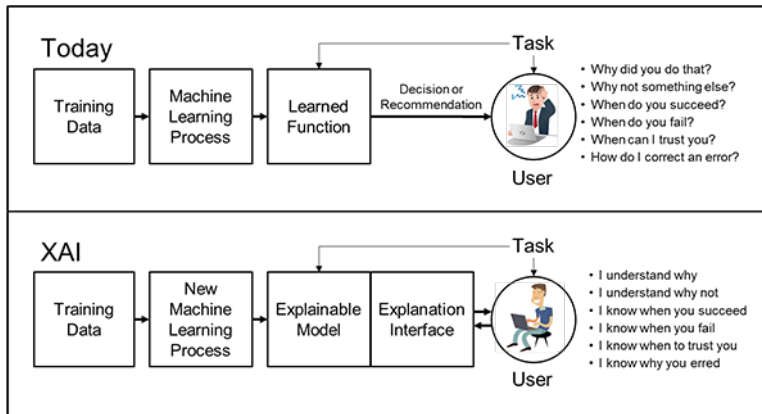


FIGURE – Source : DARPA

# XAI : a multi-disciplinary research area

- ▶ For millennia, philosophers have asked the questions about what constitutes an explanation, what is the function of explanations, and what are their structure.
- ▶ Various concepts and terms (e.g. interpretability, comprehensibility, transparency...)

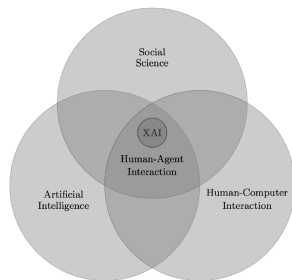


FIGURE – Source :XAI kdd Tutorial 2019

Additional Reading - *Tim Miller, Explanation in Artificial Intelligence : Insights from the Social Sciences*<sup>4</sup>.

4. <https://arxiv.org/abs/1706.07269>

# eXplainable Artificial Intelligence : terminology

An interchangeable mis-use of interpretability and explainability in the literature

## Interpretability

- ▶ (Doshi-Velez & Kim, 2017)<sup>a</sup> - Interpret means to **explain or to present in understandable terms to a human**.
- ▶ (Arrieta et al, 2020)<sup>b</sup> - A **passive characteristic of a model** referring to the level at which a given model **makes sense for a human observer**.

---

a. A Roadmap for a Rigorous Science of Interpretability :  
<https://arxiv.org/abs/1702.08608v1>

b. Explainable Artificial Intelligence (XAI) : Concepts, taxonomies, opportunities and challenges toward responsible AI : <https://arxiv.org/abs/1910.10045>

# eXplainable Artificial Intelligence : terminology

An interchangeable mis-use of interpretability and explainability in the litterature

## Explainability

- ▶ (Guidotti et al,2018)<sup>a</sup> - Ability to provide an **explanation**, i.e. an **interface** between humans and a decision maker that is at the same time **both an accurate proxy of the decision maker and comprehensible to humans**.
- ▶ (Arrieta et al, 2020)<sup>b</sup> - An **active characteristic of a model**, denoting any action or procedure taken by a model with the intent of clarifying or detailing its internal functions.

---

*a.* A Survey Of Methods For Explaining Black Box Models :  
<https://arxiv.org/abs/1802.01933>

*b.* Explainable Artificial Intelligence (XAI) : Concepts, taxonomies, opportunities and challenges toward responsible AI : <https://arxiv.org/abs/1910.10045>

# eXplainable Artificial Intelligence : terminology

Others current terms.

## Understandability (or intelligibility)

Characteristic of a model to make a human understands its function –how the model works –without any need for explaining its internal structure or the algorithmic means by which the model processes data internally (Montavon et al, 2018)<sup>a</sup>.

---

a. <https://arxiv.org/abs/1706.07979>

## Comprehensibility

Ability of a model to represent its inferred knowledge in a human understandable fashion (Gleicher et al, 2016)<sup>a</sup>

---

a. <https://www.liebertpub.com/doi/10.1089/big.2016.0007>

## Transparency

A model is considered to be transparent if by itself it is understandable (Lipton, 2017)<sup>a</sup>.

---

a. <https://arxiv.org/abs/1606.03490>

# eXplainable Artificial Intelligence : Who and For Whom ?

# eXplainable Artificial Intelligence : multi-stakeholder

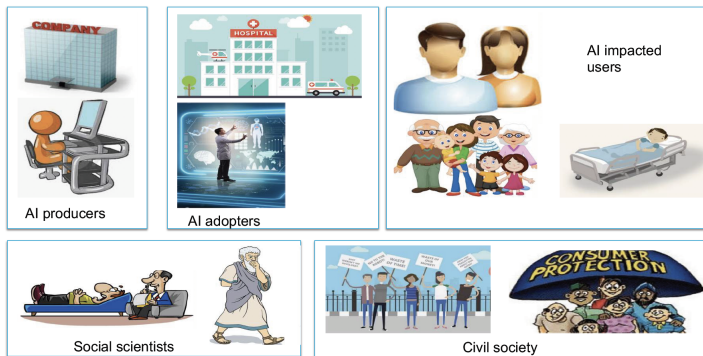


FIGURE – Source : IBM. Talk of Francesca Rossi<sup>5</sup>

## Additional Reading : *Preece et al, Stakeholders in Explainable AI*<sup>6</sup>

5. [https://economics.harvard.edu/files/economics/files/rossi-francesca\\_4-22-19\\_ai-ethics-for-enterprise-ai\\_ec3118-hbs.pdf](https://economics.harvard.edu/files/economics/files/rossi-francesca_4-22-19_ai-ethics-for-enterprise-ai_ec3118-hbs.pdf)

6. <https://arxiv.org/abs/1810.00184>



# Target audience in XAI

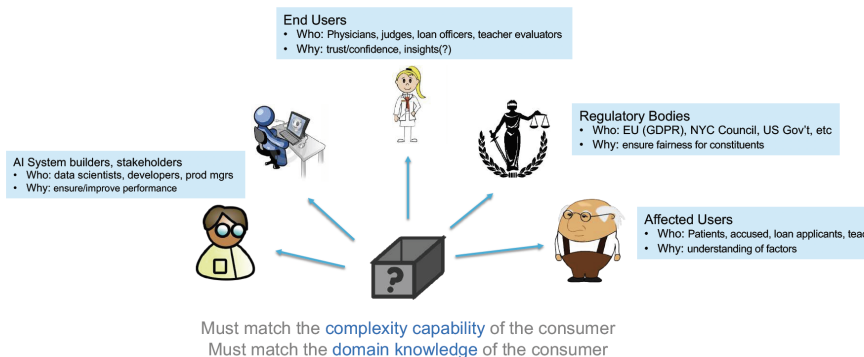


FIGURE – Source : IBM. Talk of Francesca Rossi<sup>7</sup>

7. [https://economics.harvard.edu/files/economics/files/rossi-francesca\\_4-22-19\\_ai-ethics-for-enterprise-ai\\_ec3118-hbs.pdf](https://economics.harvard.edu/files/economics/files/rossi-francesca_4-22-19_ai-ethics-for-enterprise-ai_ec3118-hbs.pdf)

# eXplainable Artificial Intelligence : What to explain ?

# Common types of explanations in XAI

- ▶ **How explanations** : demonstrate a **holistic representation** of a model to explain **How** the model works
- ▶ **Why explanations** : describe **Why** a prediction is made or a decision is taken for a **particular input**.
- ▶ **Why-not or contrastive explanations** : characterize the reasons for **differences** between a model prediction and the user's expected outcome.
- ▶ **What-If explanations** : involve demonstration of how different algorithmic and data changes affect model output given new inputs, manipulation of inputs, or changing model parameters.
- ▶ **How-to or counterfactual explanations** : spell out hypothetical adjustments to the input or model that would result in a different output
- ▶ **What-else explanations or explanations by examples** : present users with similar instances of input that generate the same or similar outputs from the model.

Source and additional reading : *Mosheni et al : A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems, 2020*<sup>8</sup>

---

8. <https://arxiv.org/abs/1811.11839>

# Common types of explanations in XAI

Table 1: Examples of explanations divided for different data type and explanation



TABULAR	IMAGE	TEXT																																																									
<p><b>Rule-Based (RB)</b></p> <p>A set of premises that the record must satisfy in order to meet the rule's consequence.</p> $r = \text{Education} \leq \text{College}$ $\rightarrow \leq 50k$	<p><b>Saliency Maps (SM)</b></p> <p>A map which highlight the contribution of each pixel at the prediction.</p> 	<p><b>Sentence Highlighting (SH)</b></p> <p>A map which highlight the contribution of each word at the prediction.</p> <p>the movie is not that bad</p>																																																									
<p><b>Feature Importance (FI)</b></p> <p>A vector containing a value for each feature. Each value indicates the importance of the feature for the classification.</p> <table border="1"> <tbody> <tr> <td>capitalgain</td> <td>0.00</td> </tr> <tr> <td>education-num</td> <td>14.00</td> </tr> <tr> <td>relationship</td> <td>1.00</td> </tr> <tr> <td>hoursperweek</td> <td>3.00</td> </tr> </tbody> </table>	capitalgain	0.00	education-num	14.00	relationship	1.00	hoursperweek	3.00	<p><b>Concept Attribution (CA)</b></p> <p>Compute attribution to a target “concept” given by the user. For example, how sensitive is the output (a prediction of zebra) to a concept (the presence of stripes)?</p> 	<p><b>Attention Based (AB)</b></p> <p>This type of explanation gives a matrix of scores which reveal how the word in the sentence are related to each other.</p> <table border="1"> <thead> <tr> <th></th> <th>the</th> <th>movie</th> <th>is</th> <th>not</th> <th>that</th> <th>bad</th> </tr> </thead> <tbody> <tr> <th>the</th> <td>0.1</td> <td>0.2</td> <td>0.1</td> <td>0.1</td> <td>0.1</td> <td>0.1</td> </tr> <tr> <th>movie</th> <td>0.2</td> <td>0.4</td> <td>0.2</td> <td>0.2</td> <td>0.2</td> <td>0.2</td> </tr> <tr> <th>is</th> <td>0.1</td> <td>0.2</td> <td>0.3</td> <td>0.1</td> <td>0.1</td> <td>0.1</td> </tr> <tr> <th>not</th> <td>0.1</td> <td>0.1</td> <td>0.1</td> <td>0.4</td> <td>0.1</td> <td>0.1</td> </tr> <tr> <th>that</th> <td>0.1</td> <td>0.1</td> <td>0.1</td> <td>0.1</td> <td>0.3</td> <td>0.1</td> </tr> <tr> <th>bad</th> <td>0.1</td> <td>0.1</td> <td>0.1</td> <td>0.1</td> <td>0.1</td> <td>0.4</td> </tr> </tbody> </table>		the	movie	is	not	that	bad	the	0.1	0.2	0.1	0.1	0.1	0.1	movie	0.2	0.4	0.2	0.2	0.2	0.2	is	0.1	0.2	0.3	0.1	0.1	0.1	not	0.1	0.1	0.1	0.4	0.1	0.1	that	0.1	0.1	0.1	0.1	0.3	0.1	bad	0.1	0.1	0.1	0.1	0.1	0.4
capitalgain	0.00																																																										
education-num	14.00																																																										
relationship	1.00																																																										
hoursperweek	3.00																																																										
	the	movie	is	not	that	bad																																																					
the	0.1	0.2	0.1	0.1	0.1	0.1																																																					
movie	0.2	0.4	0.2	0.2	0.2	0.2																																																					
is	0.1	0.2	0.3	0.1	0.1	0.1																																																					
not	0.1	0.1	0.1	0.4	0.1	0.1																																																					
that	0.1	0.1	0.1	0.1	0.3	0.1																																																					
bad	0.1	0.1	0.1	0.1	0.1	0.4																																																					

FIGURE – Source : Bodria et al, Benchmarking and Survey of Explanation Methods for Black Box Models, 2021<sup>10</sup>

9. <https://arxiv.org/abs/2102.13076>

10. <https://arxiv.org/abs/2102.13076>

# Common types of explanations in XAI

## Prototypes (PR)

The user is provided with a series of examples that characterize a class of the black box

$p = \text{Age} \in [35, 60], \text{Education} \in [\text{College}, \text{Master}] \rightarrow \text{“}\geq 50k\text{”}$ 
 $p =$ 



 $\rightarrow \text{“cat”}$ 
 $p = \text{“... not bad ...”} \rightarrow \text{“positive”}$

---

## Counterfactuals (CF)

The user is provided with a series of examples similar to the input query but with different class prediction

$q = \text{Education} \leq \text{College} \rightarrow \text{“}\leq 50k\text{”}$   
 $c = \text{Education} \geq \text{Master} \rightarrow \text{“}\geq 50k\text{”}$

$q =$ 

 $\rightarrow \text{“3”}$ 
 $c =$ 

 $\rightarrow \text{“8”}$

$q =$   
 The movie is not that bad  $\rightarrow$  “positive”  
 $c =$   
 The movie is that bad  $\rightarrow$  “negative”

---

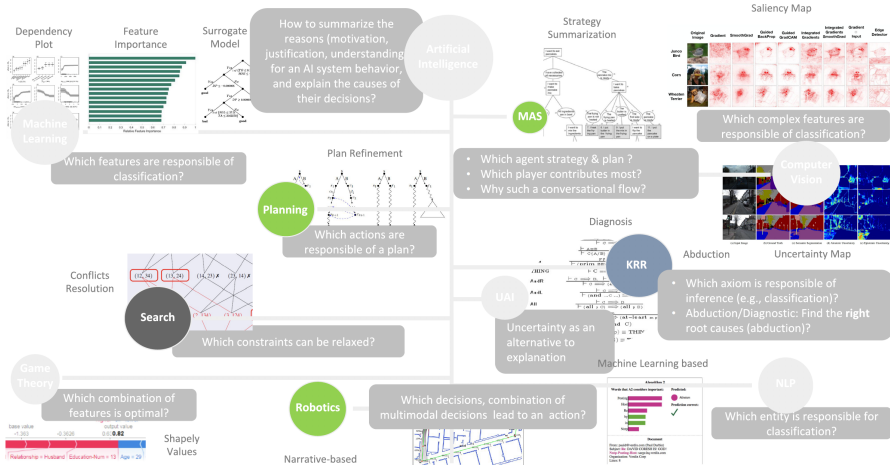
**FIGURE** – Source : Bodria et al, Benchmarking and Survey of Explanation Methods for Black Box Models, 2021 <sup>12</sup>

11. <https://arxiv.org/abs/2102.13076>

12. <https://arxiv.org/abs/2102.13076>

# eXplainable Artificial Intelligence in Artificial Intelligence

# One Objective ; many AIs, many XAIs



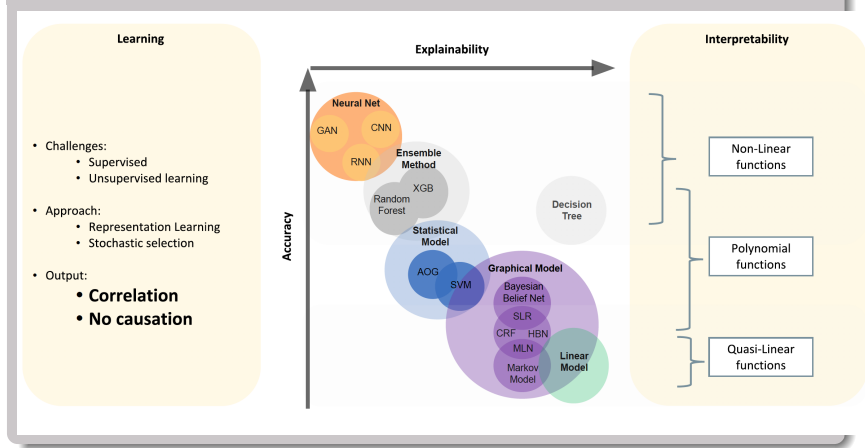
From AAAI 2020 Explainable AI Tutorial <https://xaitutorial2020.github.io/>

# eXplainable Artificial Intelligence : some issues



# XAI : some issues

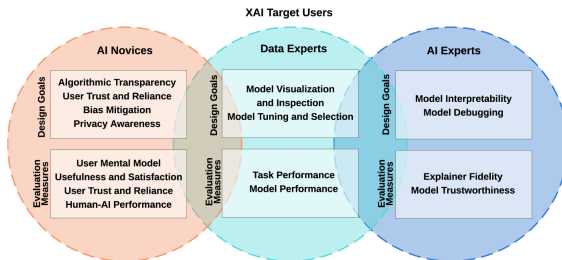
## A trade-off between accuracy and explainability (in ML)



# XAI : some issues

## How to evaluate explainability ?

- ▶ Difficult and not an unified view.
- ▶ Different design goals, different stakeholders so different evaluation methods and measures.



Source : Mohseni et al, 2021 <https://arxiv.org/abs/1811.11839>

# Two main approaches

## Approach 1 : Build an interpretable or transparent model

Provide a model which is locally or globally interpretable on its own.

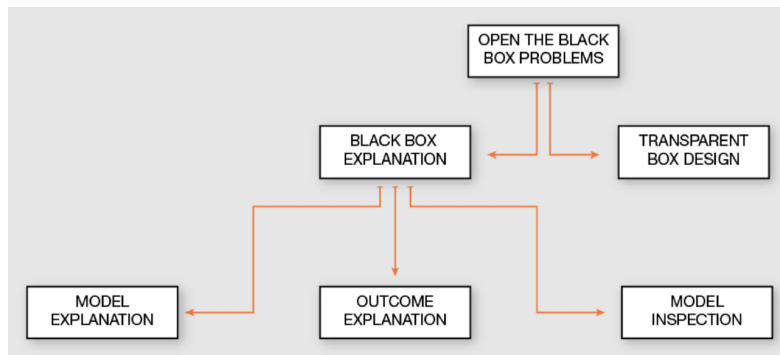
- ▶ Logistic regression, Decision trees, Decision lists and sets, Generalized Additive Models (GAMs)...

## Approach 2 : Post-hoc explain a model

Start with a black box model and probe into it with a companion model to create interpretations.

- ▶ **Model-Agnostic** or **Model-specific**
- ▶ **Individual** prediction explanations, **Global** prediction explanations or model **inspection**.

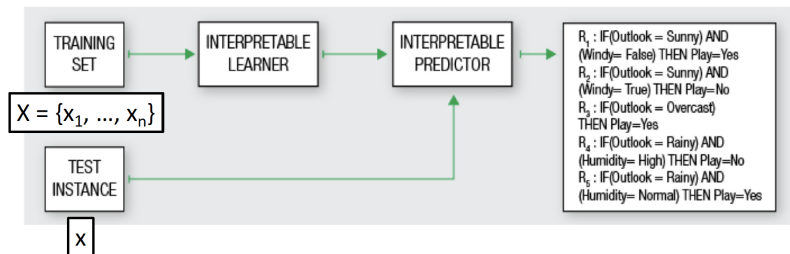
# Two main approaches



**FIGURE** – Source : Guidotti et al, A Survey of Methods for Explaining Black Box Models

# Build an interpretable or transparent model

Provide a model which is locally or globally interpretable on its own.



**FIGURE** – Source : Guidotti et al, A Survey of Methods for Explaining Black Box Models

# Three levels of transparency

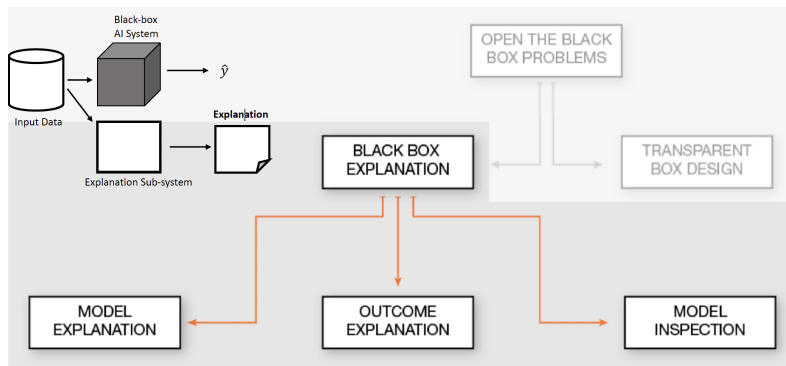
- ▶ **Simulatability** : denotes the ability of a model of being simulated or thought about strictly by a human. The model can be easily presented to a human by means of text and visualizations.
- ▶ **Decomposability** : stands for the ability to explain each of the parts of a model (input, parameter and calculation).
- ▶ **Algorithmic transparency** : deals with the ability of the user to understand the process followed by the model to produce any given output from its input data. The model has to be fully explorable by means of mathematical analysis and methods.

# Build an interpretable or transparent model

## Several transparent models

- ▶ Logistic / Linear models.
- ▶ Decision Trees, Decision Lists and Sets.
- ▶ K-Nearest Neighbors.
- ▶ Rule-base Learners.
- ▶ General Additive Models.
- ▶ Bayesian Models.

# Post-hoc model explanation

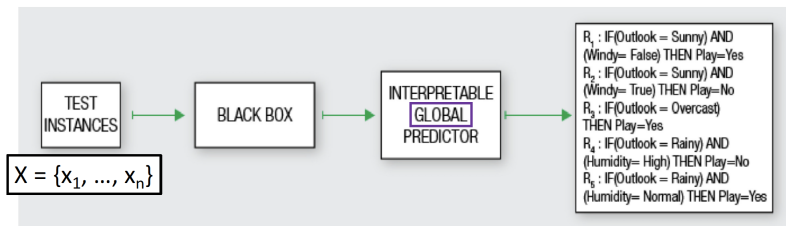


**FIGURE** – Source : Guidotti et al, A Survey of Methods for Explaining Black Box Models



# Post-hoc model explanation : model explanation

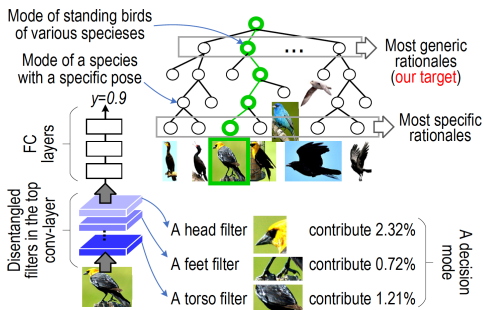
Provide an interpretable model able to mimic the *overall logic/behavior* of the black box and to explain its logic.



**FIGURE** – Source : Guidotti et al, A Survey of Methods for Explaining Black Box Models

# Post-hoc model explanation : model explanation

## Example : Interpreting CNNs with Decision Trees



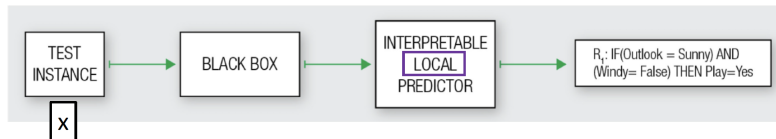
Zhang et al - Interpreting CNNs via Decision Trees. CVPR 2019<sup>13</sup>

Based on Filter Loss in the top convlayer to push the filter towards the representation of an object part- From (Zhang et al, 2018) Interpretable convolutional neural networks

13. [https://openaccess.thecvf.com/content\\_CVPR\\_2019/papers/Zhang\\_Interpreting\\_CNNs\\_via\\_Decision\\_Trees\\_CVPR\\_2019\\_paper.pdf](https://openaccess.thecvf.com/content_CVPR_2019/papers/Zhang_Interpreting_CNNs_via_Decision_Trees_CVPR_2019_paper.pdf)

# Post-hoc model explanation : outcome explanation

Provide an interpretable outcome, i.e., an **explanation** for the outcome of the black box for a **single instance**.

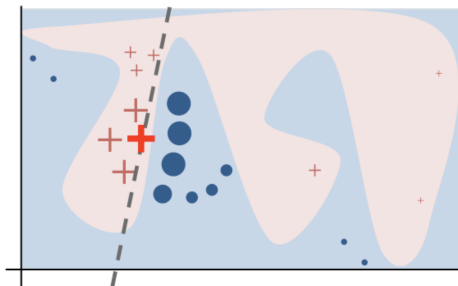


**FIGURE** – Source : Guidotti et al, A Survey of Methods for Explaining Black Box Models

# Post-hoc model explanation : outcome explanation

Example : LIME, Local Interpretable Model-agnostic Explanations  
(Ribeiro et al. KDD 2016)

- The overall decision boundary is complex
- In the neighborhood of a single decision, the boundary is simple
- A single decision can be explained by auditing the black box around the given instance and learning a *local* decision.



Ribeiro et al - Why Should I Trust You? : Explaining the Predictions of Any Classifier. KDD 2016 : 1135-1144

# Post-hoc model explanation : model inspection

Provide a representation (visual or textual) for understanding either how the black box model works or why the black box returns certain predictions more likely than others.

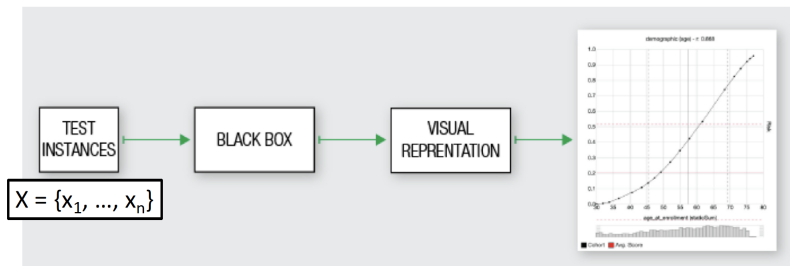
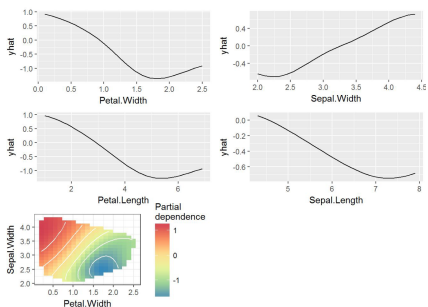


FIGURE – Source : Guidotti et al, A Survey of Methods for Explaining Black Box Models

# Post-hoc model explanation : model inspection

## Example : Partial Dependence Plots (Craven et al, 96)

Partial Dependence Plot: Shows the marginal effect one or two features have on the predicted outcome of a machine learning model

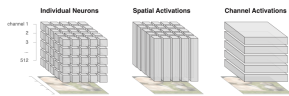


Krause et al, 2016 - Interacting with predictions : Visual inspection of black-box machine learning models, CHI.

See also <https://github.com/SauceCat/PDPbox>

# Post-hoc model explanation : model inspection

## The building block of interpretability



The cube of activations that a neural network for computer vision develops at each hidden layer. Different slices of the cube allow us to target the activations of individual neurons, spatial positions, or channels.



Making sense of these activations is hard because we usually work with them as abstract vectors:

$$a_{i,12} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 58.3, \dots]$$

With feature visualization, however, we can transform this abstract vector into a more meaningful "semantic dictionary".



There seem to be detectors for *happy ears*, *dog snouts*, *cat heads*, *furry legs*, and *grass*.

See also <https://distill.pub/2018/building-blocks/>

# Outline

Motivations and Overview

XAI and Deep Learning

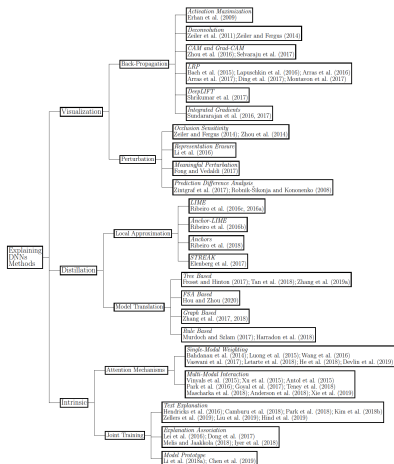
Explanation with examples

Explainable deep models

Conclusion



# Various taxonomies



(Xie et al, ) Explainable Deep Learning : A Field Guide for the Uninitiated <sup>14</sup>

14. <https://arxiv.org/abs/2004.14545>

# Various taxonomies

## Taxonomy of Xie et al

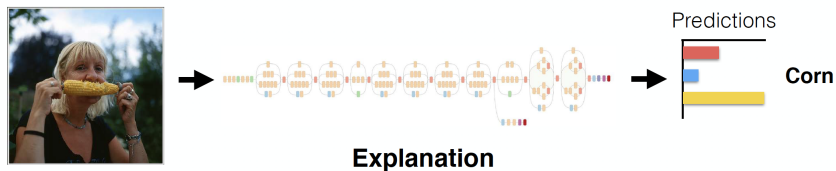
- ▶ **Visualization methods** : express an explanation by highlighting, through a scientific visualization, characteristics of an input that strongly influence the output of a DNN.
  - ▶ Back-propagation.
  - ▶ Perturbation.
- ▶ **Distillation** : develops a separate, *transparent* machine learning model that is trained to mimic the input-output behavior of the DNN.
  - ▶ Local approximations.
  - ▶ Model translation.
- ▶ **Intrinsic** : DNNs that have been specifically created to render an explanation along with its output
  - ▶ Attention Mechanisms.
  - ▶ Joint Training.

# Explaining DNN by visualizations

## Principle

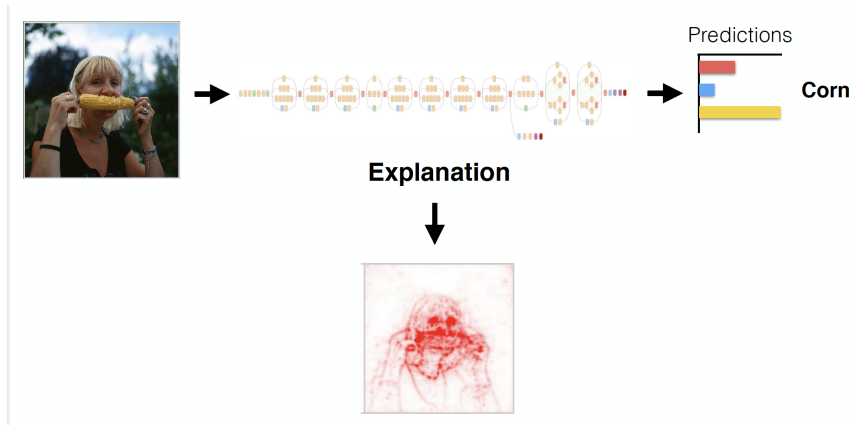
- ▶ Methods that associate the degree to which a DNN considers input features to a decision.
- ▶ Common explanatory form : [saliency maps](#)
  - ▶ Maps that identify input features that are the most salient for the decision.

# Explaining DNN by visualizations



Slide credit : J. Adebayo

# Explaining DNN by visualizations



Slide credit : J. Adebayo

# Explaining DNN by visualizations

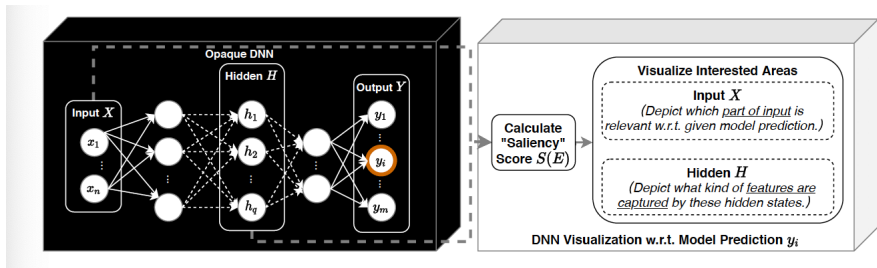


FIGURE – Source : Xie et al

# Explaining DNN by visualizations

## Two main approaches

- ▶ **Backpropagation-based** : visualize feature relevance based on volume of gradient passed through network layers during network training.
- ▶ **Perturbation-based** : Visualize feature relevance by comparing network output and a modified copy of the input.

# Formalisation

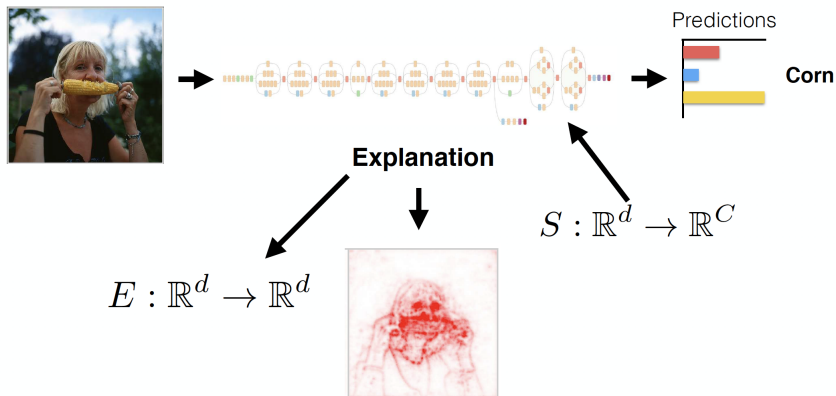
Given :

- ▶ an input : vector  $x \in \mathbb{R}^d$
- ▶ a model describing a function  $S : \mathbb{R}^d \rightarrow \mathbb{R}^C$
- ▶  $C$  the number of classes (classification problem)

An explanation method provides an **explanation map**  $E : \mathbb{R}^d \rightarrow \mathbb{R}^d$  that maps inputs to objects of the same shape.



# Formalisation



Slide credit : J. Adebayo

# Back-propagation based approaches

## Various families of approaches

- ▶ Activation Maximization.
- ▶ Deconvolution.
- ▶ Layer-Wise Relevance Propagation.
- ▶ DeepLift : Deep Learning Important FeaTures.
- ▶ Gradient-based approaches.

# Activation Maximization

(Erhan et al, 2009) Visualizing Higher-Layer Features of a Deep Network <sup>15</sup>

- ▶ One of the earliest works on visualization in deep architectures
- ▶ Visualization at **the feature level**

## Main idea

Given  $\theta$  the neural network parameters (weights and biases) and  $h_{ij}(\theta, x)$  the activation of a given unit  $i$  from a given layer  $j$  in the network, the objective is to maximize the activation of a unit :

$$x^* = \operatorname{argmax}_{x \text{ s.t. } \|x\|=r} h_{ij}(\theta, x)$$

The optimization is done with a gradient ascent in the input space, i.e. computing the gradient of  $h_{ij}(\cdot; x)$  and moving  $x$  in the direction of this gradient.

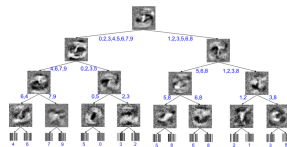



Fig. 2: This is a visualization of a soft decision tree of depth 4 trained on MNIST. The images at the inner nodes are the learned filters, and the images at the leaves are visualizations of the learned probability distribution over classes. The final most likely classification at each leaf, as well as the likely classifications at each edge are annotated. If we take for example the right most internal node, we can see that at that level in the tree the potential classifications are only 3 or 8, thus the learned filter is simply learning to distinguish between those two digits. The result is a filter that looks for the presence of two areas that would join the ends of the 3 to make an 8.

# Deconvolution

(Zeiler et al, 2014) Visualizing and Understanding Convolutional Networks<sup>16</sup>

- ▶ Introduced as an algorithm to learn image features in an unsupervised manner.
- ▶ Visualization at **the feature level** : recognize what features in the input image an intermediate layer of the network is looking for.
- ▶ Hypothesis : the model being explained is a neural network consisting of multiple **convolutional layers** and RELUs layers
- ▶ **Principle** : A deconvnet : a convnet model that uses the same components (filtering, pooling) but **in reverse**, so instead of mapping pixels to features does the opposite.
  - ▶ convolution layer → **deconvolution** layer
  - ▶ max-pooling layers → **unpooling**

---

16. <https://csulb-ml.github.io/pdf/visualizing.pdf> 

# Deconvolution

A module (**switch**) to recover the positions of maxima in the forward pass because the pooling operation is noninvertible.

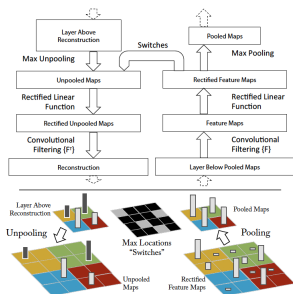


Figure 1. Top: A deconvnet layer (left) attached to a convnet layer (right). The deconvnet will reconstruct an approximate version of the convnet features from the layer beneath. Bottom: An illustration of the unpooling operation in the deconvnet, using *switches* which record the location of the local max in each pooling region (colored zones) during pooling in the convnet.

► Conv

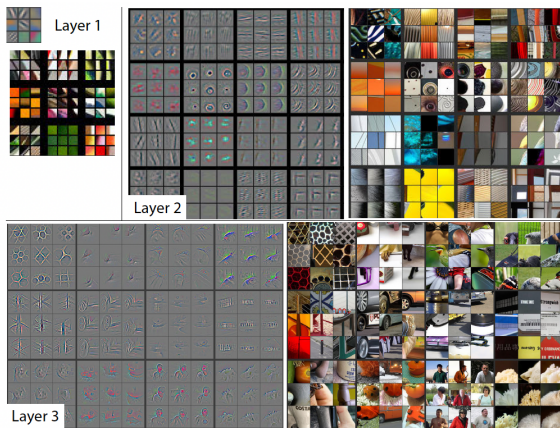
$$A^l, s^l = \text{maxpool}(\text{ReLU}(A^{l-1} * K^l + b^l))$$

with  $l$  the layer,  $A^l$  the output of the previous layer,  $K$  the learned filter,  $b$ , the bias and  $s^l$  that contains the indices of the maximum values ( for a later unpooling operation.)

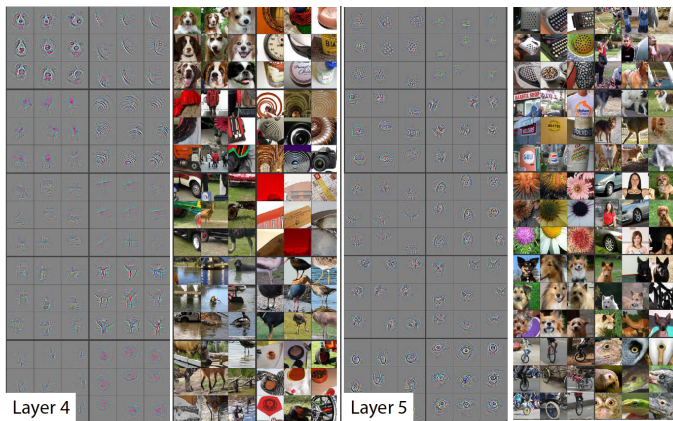
► Deconv

$$A^{l-1} = \text{unpool}(\text{ReLU}((A^l - b^l) * K^{lT}), s^l)$$

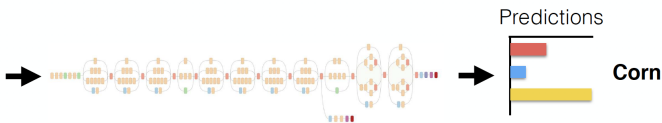
# Deconvolution



# Deconvolution



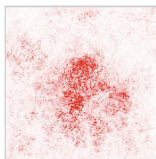
# Gradient explanation



## Attribution

$$E_{grad}(x) = \frac{\partial S_i}{\partial x}$$

Gradient



[SVZ'13]

Slide credit : J. Adebayo



# Gradient explanation

(Simonyan et al, 2014) Deep Inside Convolutional Networks : Visualising Image Classification Models and Saliency Maps.<sup>17</sup>

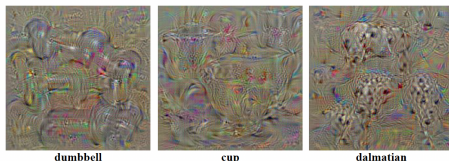
## Class Model Visualisation

**How does a typical class look like according to the neural network ?**

- ▶ Objective : finding the optimal image, constrained on its brightness, that would maximize the class score.

$$\operatorname{argmax}_I S_c(I) - \lambda \|I\|_2^2$$

with  $S_c$  score of class  $c$



17. <https://arxiv.org/pdf/1312.6034.pdf>

# Gradient explanation

(Simonyan et al, 2014) Deep Inside Convolutional Networks : Visualising Image Classification Models and Saliency Maps. <sup>18</sup>

Image-Specific Class Saliency Visualisation : spatial support of a class in a given image

**Given a model's scoring function for a class, which of the inputs has the most influence over the score ?** : gradient of the score with respect to each input

- ▶ For an input  $x$ , the gradient explanation is  $E_{grad}(x) = \frac{\partial S_c}{\partial x}$
- ▶ Quantify how much a change in each input dimension would change a prediction  $S_c(x)$  in a small neighborhood around the input.

18. <https://arxiv.org/pdf/1312.6034.pdf>

# Gradient explanation

## Justification

- ▶ In the case of DeepNets, the class score  $S_c$  is a highly non-linear function of  $x$ .
- ▶ Given an image  $x_0$ , we can approximate  $S_c(x)$  with a linear function in the neighbourhood of  $x_0$  with the first-order Taylor expansion :

$$S(x) \approx w^T x + b$$

with  $w$  the derivative of  $S_c$  with respect to the image  $x$  at the point (image)  $x_0$ .

$$w = \left. \frac{\partial S_c}{\partial x} \right|_{x_0}$$

# Gradient explanation

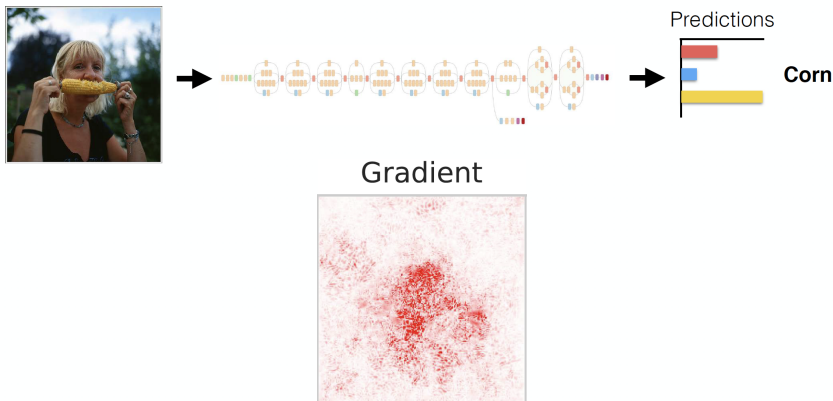
## Principle

- ▶ Perform a forward pass of the image of interest.
- ▶ Compute the gradient of class score of interest with respect to the input pixels :

$$E_{grad}(I_0) = \frac{\delta S_c}{\delta I} |_{I=I_0}$$

- ▶ Visualize the gradient.

# Gradient explanation

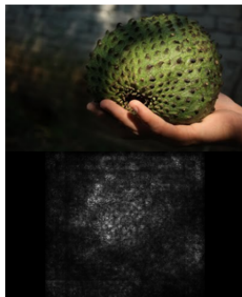
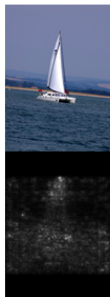


‘Visually noisy’, and can violate sensitivity w.r.t. a baseline input  
 [Sundararajan et. al., Shrikumar et. al., and Smilkov et. al.]

Slide credit : J. Adebayo

# Gradient explanation

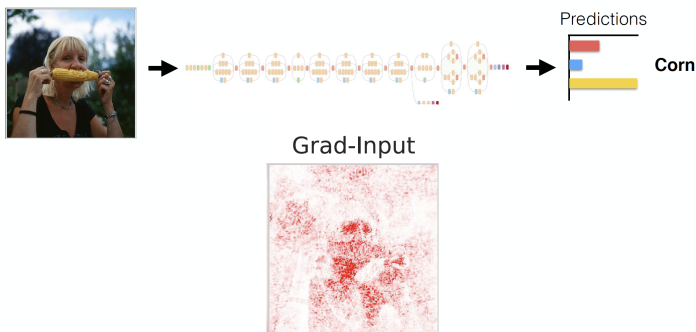
Some saliency maps.



# Gradient-Input

Address gradient saturation, and reduce visual diffusion with a pixel wise multiplication of the input and the gradient<sup>19</sup>.

$$E_{\text{grad-input}} = x \odot \frac{\partial S}{\partial x}$$



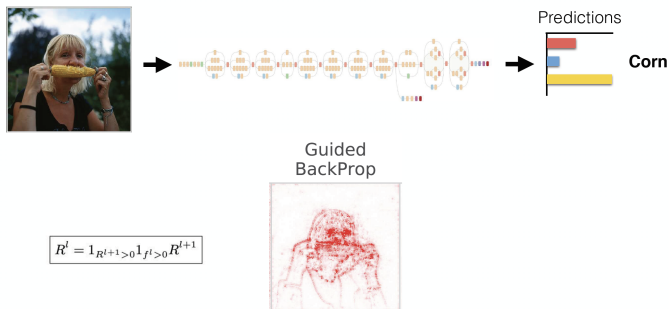
Element-wise product of gradient and input.

Slide credit : J. Adebayo

19. <https://arxiv.org/pdf/1605.01713.pdf>

# Guided Backpropagation (GBP)

Guided Backpropagation<sup>20</sup> : builds on DeConvNet and corresponds to the gradient explanation where negative gradient entries are set to zero while backpropagating through a ReLU unit.



Zero out 'negative' gradients and 'activations' while back-propagating.

Slide credit : J. Adebayo

20. (Springenberg et al, 2015) Striving for Simplicity : The All Convolutional Net  
<https://arxiv.org/abs/1412.6806>



# Guided Backpropagation (GBP)

## Combine DeconvNet and Gradient Based Backpropagation

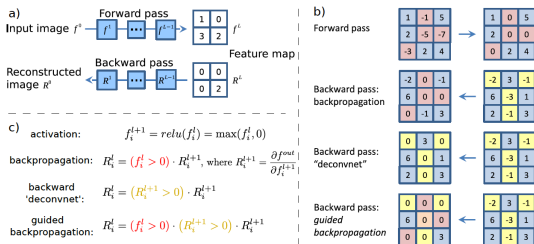


Figure 1: Schematic of visualizing the activations of high layer neurons. a) Given an input image, we perform the forward pass to the layer we are interested in, then set to zero all activations except one and propagate back to the image to get a reconstruction. b) Different methods of propagating back through a ReLU nonlinearity. c) Formal definition of different methods for propagating a output activation *out* back through a ReLU unit in layer *l*; note that the 'deconvnet' approach and guided backpropagation do not compute a true gradient but rather an imputed version.

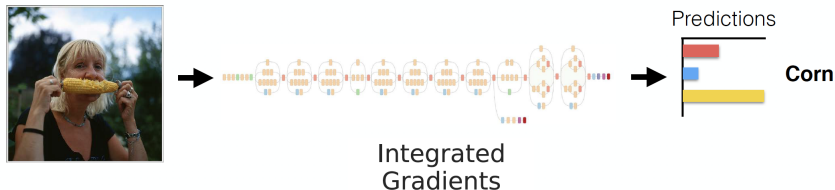
# Deep Lift : Deep Learning Important Features

- ▶ Assigns relevance scores to input features based on the difference between an input  $x$  and a **reference** input  $x^{\text{baseline}}$  : compare activation of a unit to a reference activation of the unit (when a specific input is fed forward).
- ▶ The reference is chosen according to the problem at hand.
- ▶ Consider the slope instead of the gradient

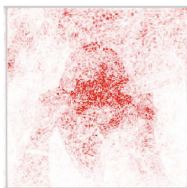
$$x_i \times \frac{\partial Y}{\partial x_i} \rightarrow (x_i - x_i^{\text{baseline}}) \times \frac{Y - Y^{\text{baseline}}}{x_i - x_i^{\text{baseline}}}$$

# Integrated Gradient

Addresses gradient saturation by summing over scaled versions of the input.



$$E_{IG}(x) = (x - \bar{x}) \times \int_0^1 \frac{\partial S(\bar{x} + \alpha(x - \bar{x}))}{\partial x} d\alpha$$



[STY'17]

Slide credit : J. Adebayo

# Integrated Gradient

(Sundararajan et al,17) Axiomatic Attribution for Deep Networks<sup>21</sup>

## Definition of attribution

Given a function  $F : \mathbb{R}^n \rightarrow [0, 1]$  representing a deep network, an input  $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ . An attribution of the prediction at input  $x$  relative to a baseline input  $x_0$  is a vector  $a_F(a_1, \dots, a_n) \in \mathbb{R}^n$  where  $a_i$  is the contribution of  $x_i$  to the prediction  $F(x)$

## Two Fundamental axioms (desirable characteristics) for attribution methods

- ▶ **Sensitivity(a)** : An attribution method satisfies Sensitivity(a) if for every input and baseline that differ in one feature but have different predictions then the differing feature should be given a non-zero attribution
- ▶ **Implementation Invariance** : Attribution methods should satisfy Implementation Invariance, i.e., the attributions are always identical for two functionally equivalent networks.

21. <https://arxiv.org/pdf/1703.01365.pdf>

# Integrated Gradient

(Sundararajan et al,17) Axiomatic Attribution for Deep Networks<sup>22</sup>

## Integrated Gradient

- ▶ Given  $F$ , the deep network, an input  $x$  and a baseline input  $x'^a$ , the relevance of feature  $x_i$  of input  $x$  over  $F$  is the integral of the gradients of  $F$  along the straight line path from  $x' \rightarrow x$

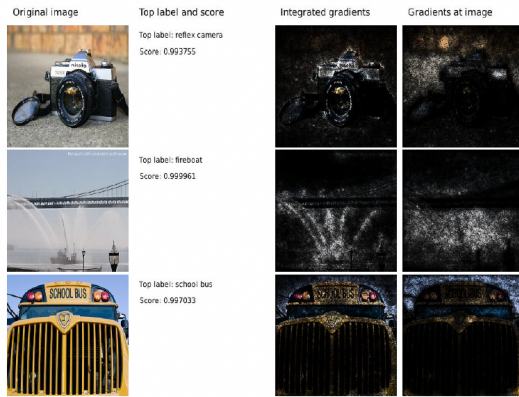
$$E_{IGi}(x) = (x_i - x'_i) \int_0^1 \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha$$

- ▶  $x'$  baseline input, represents the absence of a feature in the original input  $x$ .
- ▶ Interpretation : cumulative sensitivity of  $F$  to change in feature  $i$  in all inputs on a straight line between  $x'$  and  $x$  going in direction  $i$

a. black image, or zero embedding vector for text

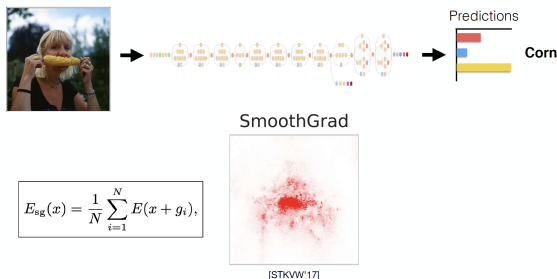
22. <https://arxiv.org/pdf/1703.01365.pdf>

# Integrated Gradient



# SmoothGrad

SmoothGrad<sup>23</sup> : Alleviate noise and visual diffusion for saliency maps by averaging over explanations of noisy copies of an input.



Average attribution of 'noisy' inputs.

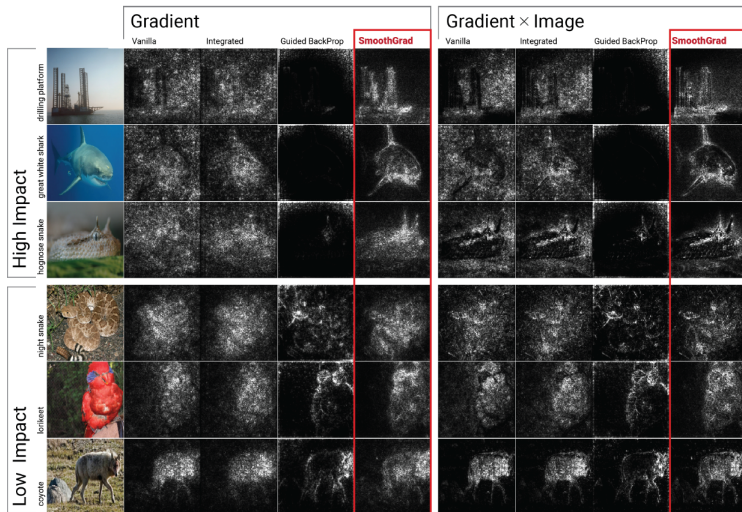
FIGURE – Source : J. Adebayo

with  $g_i \sim \mathcal{N}(0, \sigma^2)$  drawn i.i.d from a normal distribution and  $N$  the number of random samples in the neighborhood of the input  $x$

23. (Smilkov et al, 2017) SmoothGrad : removing noise by adding noise

<https://arxiv.org/pdf/1706.03825.pdf>

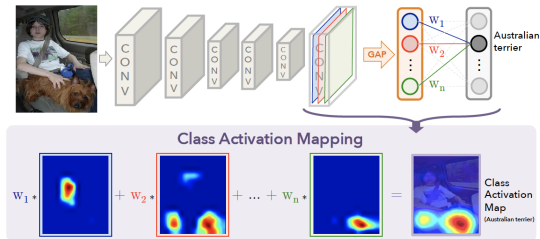
# SmoothGrad





# CAM and GradCAM

Visualization method for creating **Class Activation Maps (CAM)** using global average pooling in CNNs.



Idea : a weighted sum of the presence of (these) visual patterns at different spatial locations  
 Summation of the dot product of the last convolutional feature maps, and the class-wise weights of the FC layer

$$\text{map}_c = \sum_{i=1}^n w_{i,c} A_i$$

(Zhou et al, 2016) Learning Deep Features for Discriminative Localization<sup>24</sup>

24. <https://arxiv.org/pdf/1512.04150.pdf>

# CAM and GradCAM

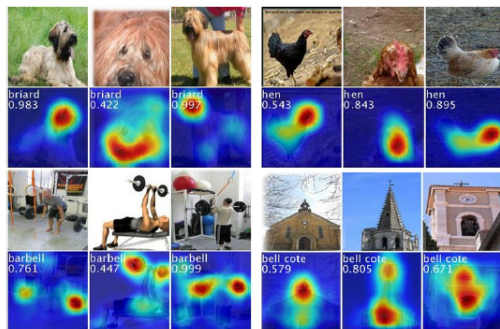
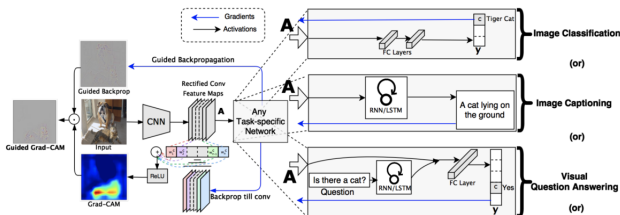


Figure 3. The CAMs of four classes from ILSVRC [20]. The maps highlight the discriminative image regions used for image classification e.g., the head of the animal for *briard* and *hen*, the plates in *barbell*, and the bell in *bell cote*.

# CAM and GradCAM

**Gradient-weighted Class Activation Map** : generalization<sup>25</sup> of CAM that uses the gradients of the network outputs with respect to the last convolutional layer.



(Selvaraju et al., 2017)) Grad-cam : Why did you say that ?<sup>26</sup>

25. applicable to a broader range of CNNs, the final activation function has to be differentiable

26. <https://arxiv.org/pdf/1611.07450.pdf>

# CAM and GradCAM

## Principle

- ▶ A gradient of the score  $y_c$  (logit, before softmax) of class  $c$  with respect to every node in the feature map  $A_k$  in the final convolution layer is computed and averaged to get an importance score  $\alpha_{k,c}$

$$\alpha_{k,c} = \frac{1}{m \cdot n} \sum_{i=1}^m \sum_{j=1}^n \frac{\partial y_c}{\partial A_{k,i,j}}$$

with  $A_{k,i,j}$  is a neuron positioned at  $(i,j)$  in the  $m \times n$  feature map  $A_k$

- ▶ Grad\_CAM linearly combines the importance scores of each feature map and passes them through a ReLU to obtain :

$$\text{map}_c = \text{ReLU}\left(\sum_k^K \alpha_{k,c} A_k\right)$$

Guided Grad-CAM = pointwise multiplication of Grad-CAM and Guided-Backpropagation.

# Outline

## Motivations and Overview

### eXplainable Artificial Intelligence

What is eXplainable Artificial Intelligence ?

Who and For Whom ?

### What to explain ?

eXplainable Artificial Intelligence in Artificial Intelligence

Some issues

### Achieving eXplainable Artificial Intelligence : main approaches

Build an interpretable model

Post-hoc model explanation

## XAI and Deep Learning

### Introduction

### Explaining by visualizations

Approaches based on backpropagation

Activation Maximization

Deconvolution

Gradient explanation

Perturbation-based approaches

Sanity Checks for Saliency maps

### Distillation Approaches

Model agnostic approaches

## Explanation with examples

# Perturbation-based approaches

## Principle

- ▶ Perturbation-based methods compute input feature relevance by **altering or removing** the input feature and comparing the difference in network output between the original and altered one.
- ▶ They can compute the **marginal relevance of each feature** with respect to how a network responds to a particular input

# Occlusion Sensitivity

(Zeiler et al, 2014) Visualizing and Understanding Convolutional Networks<sup>27</sup>

## Principle

- ▶ Sweep of a grey patch that occludes pixels over the image and see how the model prediction varies as the patch covering different positions.
- ▶ When the patch covers a critical area, the prediction performance drops significantly.
- ▶ The visualization depicts the area sensitivity of an image with respect to its classification label.

27. <https://csulb-ml.github.io/pdf/visualizing.pdf>

# Meaningful Perturbation.

(Fong et al, 2017) Interpretable Explanations of Black Boxes by Meaningful Perturbation <sup>28</sup>

## Principle

- ▶ **Explanations as a meta-predictor** : an explanation is a rule that predicts the response of black box  $f$  to certain inputs.
- ▶ Example : we can explain the behavior of a robin classifier by the rule :

$$Q_1(x; f) = \{x \in \mathcal{X}_c \iff f(x) = +1\}$$

where  $\mathcal{X}_c$  subset of robin images.

- ▶  $f$  is imperfect so the rule applies only approximately
- ▶ We can measure the faithfulness of the explanation as its expected prediction error

$$\mathcal{L}_1 = \mathbb{E}[1 - \delta_{Q_1(x;f)}]$$

where  $\delta_Q$  the indicator function of event  $Q$ .

- ▶ Use of machine learning to discover explanations automatically.

28. <https://arxiv.org/abs/1704.03296>



# Meaningful Perturbation.

(Fong et al, 2017) Interpretable Explanations of Black Boxes by Meaningful Perturbation <sup>29</sup>

## Learning explanations

Finding the most accurate explanation  $Q$  is similar to a traditional learning problem and can be formulated computationally as a regularized empirical risk minimization

$$\min_{Q \in \mathcal{Q}} \lambda \mathcal{R}(Q) + \frac{1}{n} \sum_{i=1}^n \mathcal{L}(Q(x_i; f), x_i, f), x_i \sim p(x)$$

The regularizer  $\mathcal{R}(Q)$  has two goals :

- ▶ allow the explanations to generalize beyond the  $n$  samples
- ▶ pick an explanation  $Q$  which is simple and more interpretable.

29. <https://arxiv.org/abs/1704.03296>

# Meaningful Perturbation.

## Local Explanation

- ▶ A **local explanation** is a rule  $Q(x; f, x_0)$  that predicts the response of  $f$  in a neighborhood of a certain point  $x_0$
- ▶ Construct  $Q$  by using the first-order Taylor expansion ( $f$  is smooth in  $x_0$ )
- ▶ The formulation provides an interpretation of saliency maps as gradient-based explanation.
- ▶ The meaning of explanations depends in large part on the **meaning of varying the input  $x$  to the black-box**
- ▶ Three types of perturbations :
  - ▶ replacing a region  $R$  with a constant value
  - ▶ injecting noise
  - ▶ Blurring the image

# Meaningful Perturbations

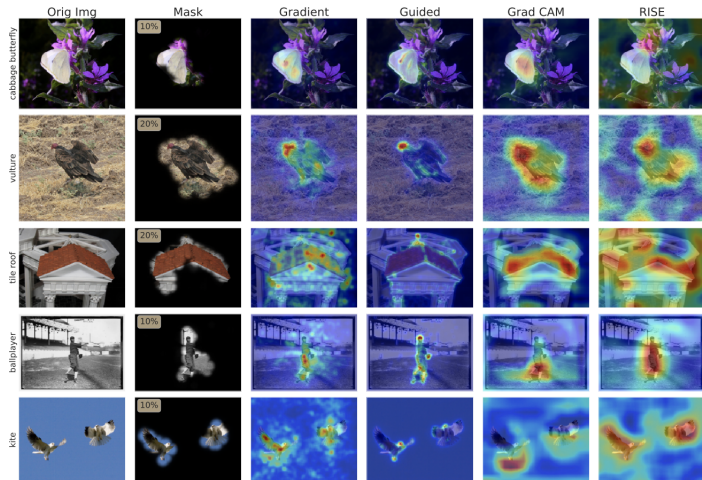
## Saliency revisited : deletion and preservation

- ▶ Given an image  $x_0$ , the goal is to summarize compactly the effect of deleting image regions in order to explain the behavior of the black box : find **deletion regions that are maximally informative**.
- ▶ A deletion game : find the smallest deletion mask  $m$  that causes the score to drop significantly  $f_c(\Psi(x_0; m)) \ll f_c(x_0)$  with  $c$  the target class.

$$m^* = \underset{m \in [0,1]^M}{\text{argmin}} \lambda \|1 - m\|_1 + f_c(\Psi(x_0; m))$$

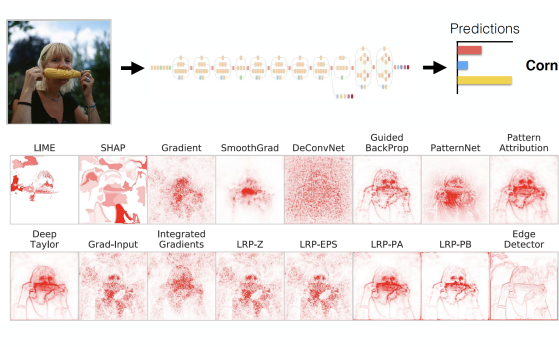
with  $\Psi$  the perturbation operator.

# Meaningful Perturbations



# Sanity Checks for Saliency maps

Actionable methodology based on randomization tests to evaluate the adequacy of explanation approaches



For a particular task and model, how should a developer/researcher select which method to use?

(Adebayo et al, 2018) Sanity Checks for Saliency Maps<sup>30</sup>

30. <https://arxiv.org/abs/1810.03292>

# Sanity Checks for Saliency maps

## Desirable properties

- ▶ Sensitivity to the **parameters** of a model to be explained.
- ▶ Depend on the labeling of the data, i.e., reflect the relationship between inputs and outputs

# Sanity Checks for Saliency maps

## Principle

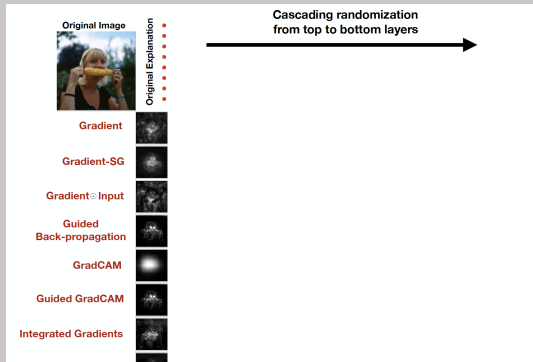
Randomization as a way to test both requirements

- ▶ **Model parameter randomization test** : randomize (reinitialize) the parameters of a model and now compare attribution maps for a trained model to those derived from a randomized model
- ▶ **Data randomization test** : compare attribution maps for a model trained with correct labels to those derived from a model trained with random labels.

# Sanity Checks for Saliency maps

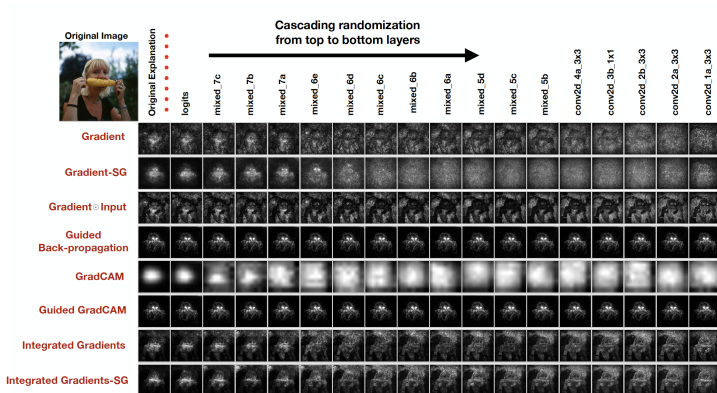
## Model Parameter Randomization

**Conjecture** : If a model captures higher level class concepts, then saliency maps should change as the model is being randomized.

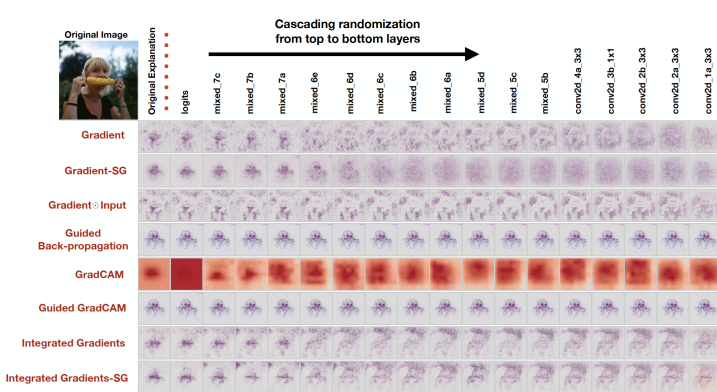




# Sanity Checks for Saliency maps

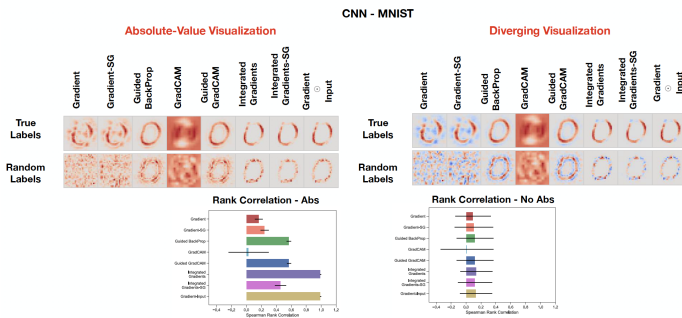


# Sanity Checks for Saliency maps



# Sanity Checks for Saliency maps

## Data randomization



# Sanity Checks for Saliency maps

- ▶ Sanity checks don't tell if a method is good, just if it is invariant.
- ▶ Sole visual inspection can be deceiving.

# Saliency maps are prone to attacks

## Are explainability methods black-box themselves?

### Attack on explanation via data change



A. K. Dombrowski et al. "Explanations can be manipulated and geometry is to blame" NeurIPS 2019

# Distillation Approaches

## Recap

Develops a separate, transparent machine learning model that is trained to mimic the input-output behavior of the DNN.

- ▶ Local approximations - Model Agnostic approaches
- ▶ Model translation

# Model-Agnostic approach for post-hoc explanations

## Different approaches

- ▶ Explanation by simplification (LIME)
- ▶ Feature Relevance Explanation (SHAP)
- ▶ Local Explanation
- ▶ Visual Explanation (PDP)

# Explanation by Simplification

## Principle

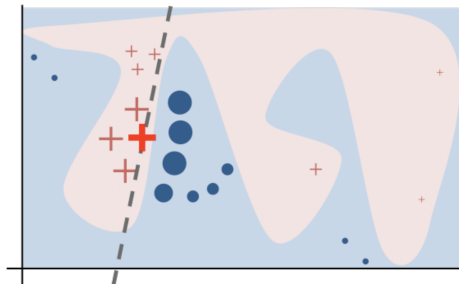
- ▶ A whole new system is rebuilt based on the trained model to be explained.
- ▶ The new system usually attempts :
  - ▶ at **optimizing its resemblance** to its antecedent functioning.
  - ▶ while **reducing its complexity**
  - ▶ and **keeping a similar performance score**.
- ▶ The new model is often easier to be implemented due to its reduced complexity



# LIME, Local Interpretable Model-agnostic Explanations (Ribeiro et al. KDD 2016)

One of the the most known contributions in this domain.

- The overall decision boundary is complex
- In the neighborhood of a single decision, the boundary is simple
- A single decision can be explained by auditing the black box around the given instance and learning a *local* decision.



Ribeiro et al - Why Should I Trust You ? : Explaining the Predictions of Any Classifier. KDD 2016 : 1135-1144<sup>32</sup>

32. <https://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf>

# LIME, Local Interpretable Model-agnostic Explanations (Ribeiro et al. KDD 2016)

## Local explanations

Explainability by **segmenting the solution space** and giving explanations to less complex solution subspaces that are relevant for the whole model.

# LIME, Local Interpretable Model-agnostic Explanations (Ribeiro et al. KDD 2016)

## Two contributions

- ▶ **LIME** : locally approximating black-box classifier with interpretable classifier.
- ▶ **SP-LIME** : finding near-optimal set of explanations to evaluate overall model performance.

# LIME

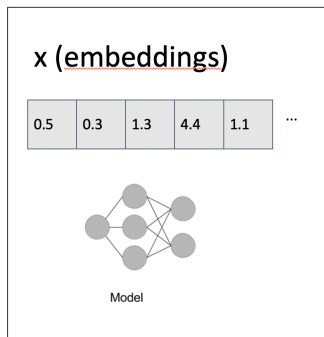
## Interpretable Data Representations

Interpretable explanations need to use a **representation that is understandable to humans**, regardless of the actual features used by the model.

## Examples

- ▶ Text : binary vector indicating the presence or absence of a word instead of word embedding.
- ▶ Image : binary vector indicating the presence or absence of super-pixels.

# LIME : Interpretable Representation



# LIME : Interpretable Representation

$x$  (3 color channels / pixel)



Model

$x'$  (contiguous superpixels)

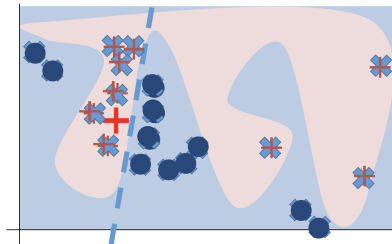


Human

10

# LIME : Principle

1. Sample points around  $x_i$
2. Use complex model to predict labels for each sample
3. Weigh samples according to distance to  $x_i$
4. Learn new simple model on weighted samples
5. Use simple model to explain



# LIME

## Setup

- ▶  $f$  : black-box classifier
- ▶  $X \in \mathbb{R}^d$  : feature space
- ▶  $Y_k \in \mathbb{R}$  : classification probability for class  $k$
- ▶  $X' \in \mathbb{R}^{d'}$  : interpretable feature space
- ▶  $G = \{g\}$  : family of interpretable functions.



# LIME

## Objective function

$$\xi(x) = \arg \min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

**Fidelity score**  
*(for local fidelity)*

**Complexity score**  
*(for interpretability)*

with  $\pi_x(z)$  a proximity measure between an instance  $z$  to  $x$ , used to define locality around  $x$ .

The explanation model for instance  $x$  is the model  $g$  (e.g. linear regression model) that minimizes loss  $L$  (e.g. mean squared error), which measures how close the explanation is to the prediction of the original model  $f$  while the model complexity  $\Omega$  is kept low (e.g. prefer fewer features).

# LIME

## Sparse Linear models

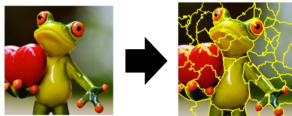
- ▶  $G$  : the class of linear models, such that  $g(z') = w_g^T \cdot z'$
- ▶  $\pi_x(z) = \exp \frac{-D(x,z)^2}{\sigma^2}$  : exponential kernel defined on some distance  $D$

## Objective function

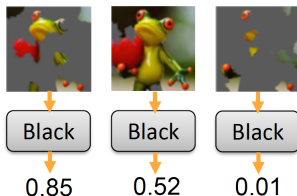
$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) (f(z) - g(z'))^2$$

# LIME : Intuition by example

- ▶ Given a data point you want to explain



- ▶ Sample at the nearby : each image is represented as a set of superpixels

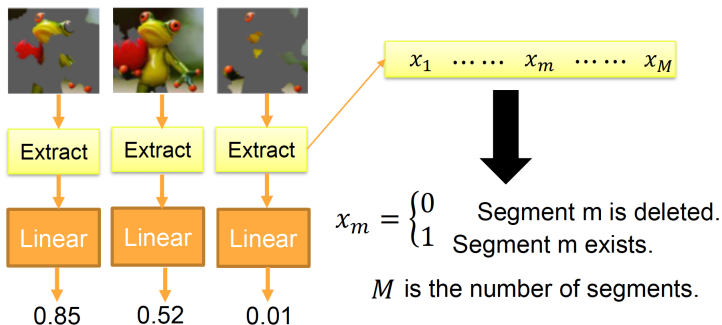


Randomly delete some segments.

Compute the probability of “frog” by black box

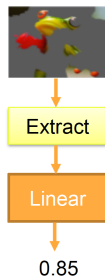
# LIME : Intuition by example

- Fit with linear (or interpretable model)



# LIME : Intuition by example

- ▶ Interpret the learned model



$$y = w_1x_1 + \dots + w_mx_m + \dots + w_Mx_M$$

$$x_m = \begin{cases} 0 & \text{Segment } m \text{ is deleted.} \\ 1 & \text{Segment } m \text{ exists.} \end{cases}$$

$M$  is the number of segments.

If  $w_m \approx 0$   $\Rightarrow$  segment  $m$  is not related to “frog”

If  $w_m$  is positive  $\Rightarrow$  segment  $m$  indicates the image is “frog”

If  $w_m$  is negative  $\Rightarrow$  segment  $m$  indicates the image is not “frog”

## LIME

---

**Algorithm 1** Sparse Linear Explanations using LIME

---

**Require:** Classifier  $f$ , Number of samples  $N$ **Require:** Instance  $x$ , and its interpretable version  $x'$ **Require:** Similarity kernel  $\pi_x$ , Length of explanation  $K$  $\mathcal{Z} \leftarrow \{\}$ **for**  $i \in \{1, 2, 3, \dots, N\}$  **do** $z'_i \leftarrow \text{sample\_around}(x')$  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$ **end for** $w \leftarrow \text{K-Lasso}(\mathcal{Z}, K) \triangleright$  with  $z'_i$  as features,  $f(z)$  as target**return**  $w$ 

---

## LIME

**Algorithm 1** Sparse Linear Explanations using LIME**Require:** Classifier  $f$ , Number of samples  $N$ **Require:** Instance  $x$ , and its interpretable version  $x'$ **Require:** Similarity kernel  $\pi_x$ , Length of explanation  $K$  $\mathcal{Z} \leftarrow \{\}$ **for**  $i \in \{1, 2, 3, \dots, N\}$  **do** $z'_i \leftarrow \text{sample\_around}(x')$ **Just collecting samples!** $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$ **end for** $w \leftarrow \text{K-Lasso}(\mathcal{Z}, K) \triangleright$  with  $z'_i$  as features,  $f(z)$  as target**return**  $w$

# LIME

---

## Algorithm 1 Sparse Linear Explanations using LIME

---

**Require:** Classifier  $f$ , Number of samples  $N$

**Require:** Instance  $x$ , and its interpretable version  $x'$

**Require:** Similarity kernel  $\pi_x$ , Length of explanation  $K$

$\mathcal{Z} \leftarrow \{\}$

**for**  $i \in \{1, 2, 3, \dots, N\}$  **do**

$z'_i \leftarrow \text{sample\_around}(x')$

$\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$  **Lasso "feature" selection**

**end for**

$w \leftarrow \text{K-Lasso}(\mathcal{Z}, K)$   $\triangleright$  with  $z'_i$  as features,  $f(z)$  as target

**return**  $w$

---

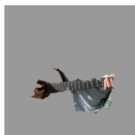


# LIME : Gaining insights from explanation

Explaining Google's Inception model



$$P(\text{Guitar}) = 0.32$$



$$P(\text{Dog}) = 0.24$$



$$P(\text{Hat}) = 0.21$$



# LIME : Gaining insights from explanation

Explaining Google's Inception model : predict wolf vs husky

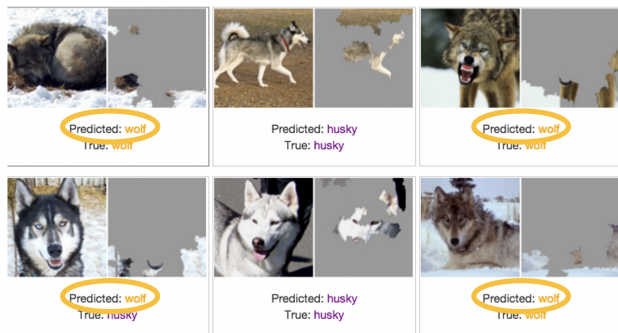
Train a neural network to predict **wolf** v. **husky**



Only 1 mistake!!!

# LIME : Gaining insights from explanation

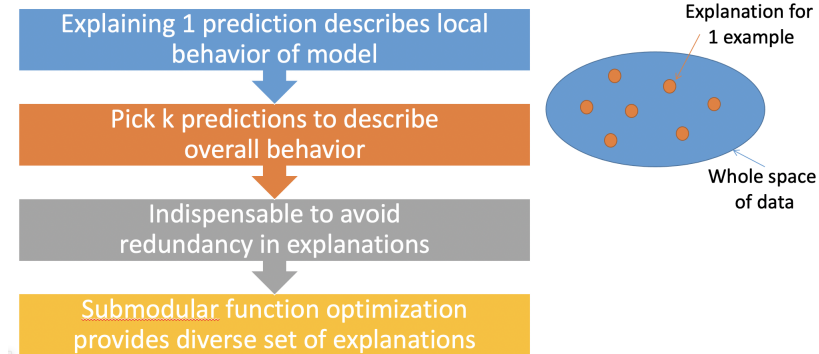
Explaining Google's Inception model : predict wolf vs husky



The model is just a good snow detector

# SP-LIME

Beyond explaining predictions : **explaining whole model**



# SP-LIME

## Question

How to choose a representative set of explanations (examples) to judge the model as a whole?

## Intuition

We want a set of examples that :

- ▶ Maximize the number of interpretable features used.
- ▶ Weight by importance each feature.

## SP-LIME

## Submodular pick principle

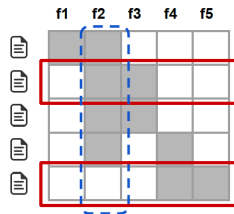


Figure 5: Toy example  $\mathcal{W}$ . Rows represent instances (documents) and columns represent features (words). Feature  $f_2$  (dotted blue) has the highest importance. Rows 2 and 5 (in red) would be selected by the pick procedure, covering all but feature  $f_1$ .

# SP-LIME

## Submodular pick principle

- ▶ Coverage for some set of examples  $V$  : sum of importances of features covered by those examples

$$c(V, \mathcal{W}, I) = \sum_{j=1}^{d'} 1_{[\exists i \in V: \mathcal{W}_{ij} > 0]} I_j$$

with  $I$  the importance function

- ▶ Find the set  $V$ , subject to budget  $B$  that achieves highest coverage.

$$\text{Pick}(\mathcal{W}, I) = \underset{V, |V| \leq B}{\text{argmax}} c(V, \mathcal{W}, I)$$

# LIME : Advantages

- ▶ Model-agnostic.
- ▶ Works for tabular data, text, images.
- ▶ The fidelity measure (how well the interpretable model approximates the black box predictions) gives us a good idea of how reliable the interpretable model.
- ▶ Use of interpretable features.
- ▶ Many implementations are available.
- ▶ Some theoretical Analysis of LIME are available<sup>33</sup>

---

33. <https://arxiv.org/pdf/2102.06307.pdf>



# LIME : Disadvantages

- ▶ Definition of the neighborhood
- ▶ Instability of explanations : the explanations of two very close points can vary greatly<sup>34</sup>.
- ▶ LIME explanations can be manipulated by the data scientist to hide biases.<sup>35</sup>

---

34. Alvarez-Melis, David, and Tommi S. Jaakkola. "On the robustness of interpretability methods." arXiv preprint arXiv :1806.08049 (2018)

35. Slack, Dylan, et al. "Fooling lime and shap : Adversarial attacks on post hoc explanation

# Outline

Motivations and Overview

XAI and Deep Learning

**Explanation with examples**

Explainable deep models

Conclusion

# Example-based explanation approaches

## Principle

Example-based explanation methods select **particular instances of the dataset** to explain the behavior of machine learning models or to explain the underlying data distribution.

- ▶ Mostly model-agnostic.
- ▶ Make sense if we can represent an instance of the data in a **humanly understandable way**.
- ▶ Help humans construct mental models of the machine learning model and the data the machine learning model has been trained on.
- ▶ Help to understand complex data distributions.

# Example-based explanation approaches

## Motivations from Human reasoning

- ▶ **Use of examples (prototypes)** is fundamental in human reasoning in the development of effective strategies for tactical decision-making (Newell and Simon, 1972)<sup>a</sup>; (Cohen et al, 96)<sup>b</sup>
- ▶ **Case-based reasoning** : a new situation can be well-represented by the summarized experience of previously solved problems (Kolodner et al, 92)<sup>c</sup>

a. Human Problem Solving,

<https://exhibits.stanford.edu/feigenbaum/catalog/kj570jz8641>

b. Metarecognition in time-stressed decision making : Recognizing, critiquing, and correcting  
<https://journals.sagepub.com/doi/10.1177/001872089606380203>

c. An Introduction to Case-Based Reasoning [http://alumni.media.mit.edu/~jorkin/generals/papers/Kolodner\\_case\\_based\\_reasoning.pdf](http://alumni.media.mit.edu/~jorkin/generals/papers/Kolodner_case_based_reasoning.pdf)

**MMD-critic** (Kim et al, 2016) Examples are not  
enough, learn to criticize! Criticism for  
Interpretability<sup>36</sup>

---

36. <https://papers.nips.cc/paper/2016/hash/5680522b8e2bb01943234bce7bf84534-Abstract.html>

# MMD-critic : Prototypes and Criticism

## Prototype

A prototype is a data instance that is representative of all the data.

## Criticism

A criticism is a data instance that is not well represented by the set of prototypes.

- ▶ provide insights together with prototypes, especially for data points which the prototypes do not represent well.

# MMD-critic

## Principle

- ▶ Use of maximum mean discrepancy (MMD) as a measure of similarity between points and potential prototypes.
- ▶ Selection of prototypes that maximize the MMD.
- ▶ Selection of criticism samples : samples that are not well-explained by the prototypes using a **regularized witness function** score.

# MMD-critic

## Principle

- ▶ Select the number of prototypes and criticisms you want to find.
- ▶ Find prototypes with greedy search. Prototypes are selected so that the distribution of the prototypes is close to the data distribution.
- ▶ Find criticisms with greedy search. Points are selected as criticisms where the distribution of prototypes differs from the distribution of the data.

## Needs

- ▶ **Kernel function** : to estimate the data densities
  - ▶ MMD : maximum mean discrepancy
- ▶ **witness function** : use to measure how different two distributions are at a particular data point.
- ▶ **Search strategy** for good prototypes and criticisms : simple **greedy search**.



# MMD-critic

## MMD

- ▶ Measures the discrepancy between two distributions.
- ▶ We want to evaluate whether the prototypes distribution differs from the data distribution.

$$MMD^2 = \frac{1}{m^2} \sum_{i,j=1}^m k(z_i, z_j) - \frac{2}{mn} \sum_{i,j=1}^{m,n} k(z_i, x_j) + \frac{1}{n^2} \sum_{i,j=1}^n k(x_i, x_j)$$

$k$  : kernel function that measures the similarity of two points

$m$  : the number of prototypes  $z$ ,  $n$  : number of data points.

- ▶ kernel : radial basis function

$$k(x, x') = \exp(-\gamma \|x - x'\|^2)$$

# MMD-critic

## Algorithm for finding prototypes

- ▶ Start with an empty list of prototypes.
- ▶ While the number of prototypes is below the chosen number  $m$ 
  - ▶ For each point in the dataset, check how much  $MMD^2$  is reduced when the point is added to the list of prototypes. Add the data point that minimizes the  $MMD^2$  to the list.
- ▶ Return the list of prototypes.

# MMD-critic

## Witness Function

Measure how much two density estimates differ at a particular point.

$$witness(x) = \frac{1}{n} \sum_{i=1}^n k(x, x_i) - \frac{1}{m} \sum_{j=1}^m k(x, z_j)$$

- ▶ First term : average proximity between point  $x$  and the data
- ▶ Second term : average proximity between point  $x$  and the prototypes

To find criticisms, we look for extreme values of the witness function in both negative and positive directions

- ▶ close to zero : the density function of the data and the prototypes are close together
- ▶ negative witness : the prototype distribution overestimates the data distribution
- ▶ positive witness : the prototype distribution underestimates the data distribution

**Criticisms are points with high absolute value in the witness function.**

# MMD-critic for interpretability

- ▶ Interpretable prediction model, i.e. nearest prototype model

$$\hat{f}(x) = \operatorname{argmax}_{i \in S} k(x, x_i)$$

**We select the prototype  $i$  from the set of prototypes  $S$  that is closest to the new data point.**

- ▶ MMD-critic to make any machine learning model globally explainable.
  - ▶ Find prototypes and criticisms with MMD-critic.
  - ▶ Train a machine learning model as usual.
  - ▶ Predict outcomes for the prototypes and criticisms with the machine learning model.
  - ▶ Analyse the predictions : In which cases was the algorithm wrong? Now you have a number of examples that represent the data well and help you to find the weaknesses of the machine learning model.

# MMD-critic : example

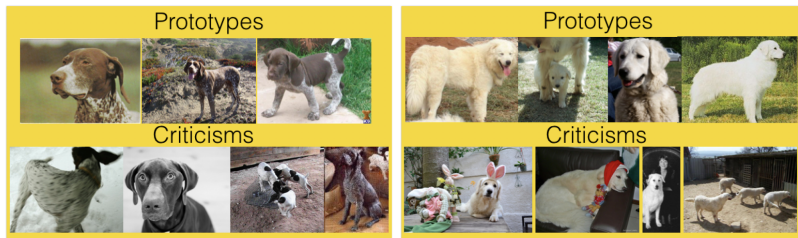


Figure 2: Learned prototypes and criticisms from Imagenet dataset (two types of dog breeds)

# Prototypes and criticisms : Advantages

- ▶ MMD-critic works with density estimates of the data : works with any type of data and any type of machine learning model.
- ▶ Easy to implement.
- ▶ MMD-critic is very flexible : it can be used to understand complex data distributions, to build an interpretable machine learning model or to shed light on the decision making of a black box machine learning model.

# Prototypes and criticisms : Disadvantages

- ▶ Choice of the number of prototypes and criticisms : How ?
- ▶ How do we select a kernel and its scaling parameter ?
- ▶ Take all the features as input, disregarding the fact that some features might not be relevant for predicting the outcome of interest.

# Counterfactual Visual Explanations



# Counterfactual Visual Explanation

(Goyal et al) Counterfactual Visual Explanations, ICML 2019<sup>37</sup>

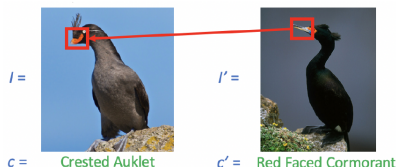


Figure 1. Our approach generates counterfactual visual explanations for a query image  $I$  (left) – explaining why the example image was classified as class  $c$  (*Crested Auklet*) rather than class  $c'$  (*Red Faced Cormorant*) by finding a region in a distractor image  $I'$  (right) and a region in the query  $I$  (highlighted in red boxes) such that if the highlighted region in the left image looked like the highlighted region in the right image, the resulting image  $I^*$  would be classified more confidently as  $c'$ .

# Counterfactual Visual Explanation

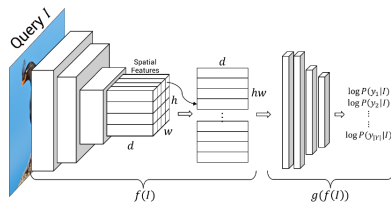


Figure 3. We decompose a CNN as a spatial feature extractor  $f(I)$  and a decision network  $g(f(I))$  as shown above.

The neural network is traditionally decomposed into two functional components – a spatial feature extractor and a decision network.

# Counterfactual Visual Explanation

- ▶ Given an image  $I$  and a distractor image  $I'$ , we want a transformation  $T$  from  $I$  to  $I^* = T(I; I')$  such that  $I^*$  appears to be an instance of class  $c'$  to the trained model  $g(f())$ .
- ▶ Idea to perform this transformation : replacing regions in image  $I$  with regions in image  $I'$

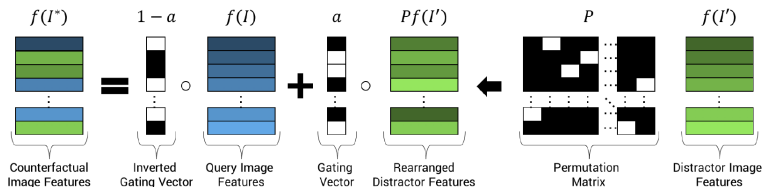


Figure 2. To parameterize our counterfactual explanations, we define a transformation that replaces regions in the query image  $I$  with those from a distractor  $I'$ . Distractor image features  $f(I')$  are first rearranged with a permutation matrix  $P$  and then selectively replace entries in  $f(I)$  according to a binary gating vector  $a$ . This allows arbitrary spatial cells in  $f(I')$  to replace arbitrary cells in  $f(I)$ .

# Counterfactual Visual Explanation

## Minimum-edit counterfactual problem

Minimizing the number of edits to transform  $I$  to  $I^*$  such that the predicted class for the transformed image features  $f(I^*)$  is the distractor class  $c'$ .

$$\begin{aligned} & \underset{P, \mathbf{a}}{\text{minimize}} \quad \|\mathbf{a}\|_1 \\ & \text{s.t.} \quad c' = \operatorname{argmax} g((\mathbf{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ Pf(I')) \\ & \quad \quad a_i \in \{0, 1\} \quad \forall i \text{ and } P \in \mathcal{P} \end{aligned}$$

$\mathcal{P}$  is the set of all  $hw \times hw$  permutation matrices.  $\mathbf{a}$ , a binary vector indicating whether to replace each spatial feature in image  $I$  with spatial features from image  $I'$ .

The paper proposes several relaxations of this objective to solve it.

# Outline

Motivations and Overview

XAI and Deep Learning

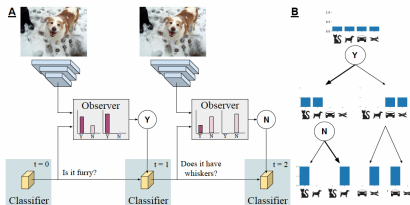
Explanation with examples

**Explainable deep models**

Conclusion

# Hierarchical classification

(Alaniz et al, 2019) XOC : explainable observer-classifier for explainable binary



**Figure 1.** Our Observer-Classifier Framework. **A.** For classifying an image into cat, dog, car, plane categories, the *observer* combines image features with a query message of the *classifier* at each time step and creates a binary response. The classifier, i.e. an LSTM, uses its hidden state as a query message. At every step, the state of the LSTM is updated with the binary response to improve classification accuracy. **B.** The corresponding decision making process as the underlying decision tree.

# Hierarchical classification

(Alaniz et al, 2019) XOC : explainable observer-classifier for explainable binary<sup>38</sup>

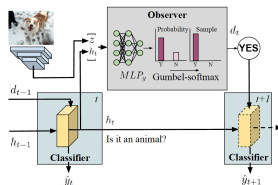


Figure 2. General Observer-Classifier (OC) Model: The classifier sends a query message  $h_t$  to the observer that the observer combines with image features  $z$  to construct a binary response  $d_t$  using its MLP and the Gumbel-softmax estimator. The response is fed into the classifier in the next time step, such that the predicted class distribution  $\hat{y}$  better fits the true class label.

Sequential interaction between two agents to solve an image-classification task by communicating

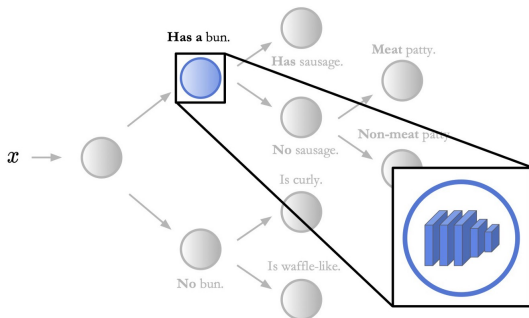
- ▶ The first agent, the observer, holds information about the image.
- ▶ The second agent, the classifier, predicts the associated ground-truth class using only the messages broadcast by the observer.

38. <https://www.arxiv-vanity.com/papers/1902.01780/>

# NBDT : Neural-backed Decision Tree (Wan et al, ICLR 2021)

See <sup>39</sup>

- ▶ Jointly improve both **accuracy** and **interpretability**. See <https://www.youtube.com/watch?v=fQ2eNFCsRIa&t=110s>



**Predictions are made via a decision tree, preserving high-level interpretability. However, each node in decision tree is a neural network making low-level decisions.**

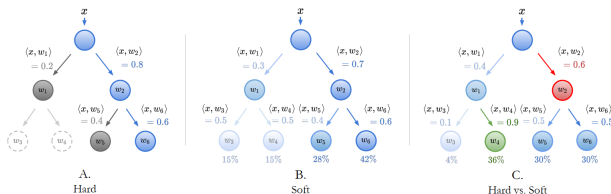
39. <https://research.alvinwan.com/neural-backed-decision-trees/>



# NBDT : Neural-backed Decision Tree (Wan et al, ICLR 2021)

Principle : Replacement of a network's final layer with a decision tree that :

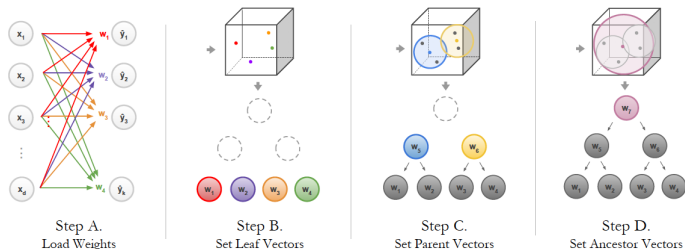
- ▶ Use path probabilities for inference to tolerate highly-uncertain intermediate decisions.
- ▶ Build a hierarchy from pre-trained model weights to lessen overfitting
- ▶ Train with a hierarchical loss to significantly better learn high-level decisions



**Figure 1: Hard and Soft Decision Trees.** **A. Hard:** is the classic "hard" oblique decision tree. Each node picks the child node with the largest inner product, and visits that node next. Continue until a leaf is reached. **B. Soft:** is the "soft" variant, where each node simply returns probabilities, as normalized inner products, of each child. For each leaf, compute the probability of its path to the root. Pick leaf with the highest probability. **C. Hard vs. Soft:** Assume  $w_4$  is the correct class. With hard inference, the mistake at the root (red) is irrecoverable. However, with soft inference, the highly-uncertain decisions at the root and at  $w_2$  are superseded by the highly certain decision at  $w_3$  (green). This means the model can still correctly pick  $w_4$  despite a mistake at the root. In short, soft inference can tolerate mistakes in highly uncertain decisions.

# NBDT : Neural-backed Decision Tree (Wan et al, ICLR 2021)

Principle : building induced hierarchy using model weights and labeling decision node with Wordnet (earliest share ancestor)



**Figure 2: Building Induced Hierarchies.** **Step A.** Load the weights of a pre-trained model's final fully-connected layer, with weight matrix  $W \in \mathbb{R}^{D \times K}$ . **Step B.** Take rows  $w_k \in W$  and normalize for each leaf node's weight. For example, the red  $w_1$  in A is assigned to the red leaf in B. **Step C.** Average each pair of leaf nodes for the parents' weight. For example,  $w_1$  and  $w_2$  (red and purple) in B are averaged to make  $w_5$  (blue) in C. **Step D.** For each ancestor, average all leaf node weights in its subtree. That average is the ancestor's weight. Here, the ancestor is the root, so its weight is the average of all leaf weights  $w_1, w_2, w_3, w_4$ .

# Outline

Motivations and Overview

XAI and Deep Learning

Explanation with examples

Explainable deep models

Conclusion

# A very fertile field

Many other approaches :

- ▶ Concept Activation Vectors
- ▶ Neuro-Symbolic models
- ▶ Visual Analytics Tools
- ▶ ...

# A very fertile field

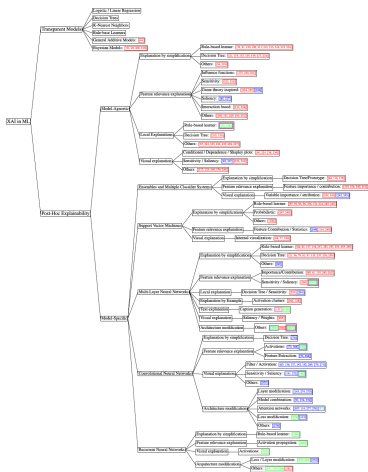


Fig. 6. Taxonomy of the reviewed literature and trends identified for explainability techniques related to different ML models. References listed in blue, green and red correspond to XAI techniques using image, text or tabular data, respectively. In order to build this taxonomy, the literature has been analyzed in depth to determine whether a post-hoc technique can be successfully applied to any ML model, even if e.g., explicitly mentions Deep Learning in its title and/or abstract. See references [109,110,111,112,117,118,120,129,131,133,135,139,149,156,158,160,161,165,167,169,171,173,177,179,182] (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

## Some explanation toolboxes are available

- ▶ AI Explainability 360 toolkit : intrinsic, post-hoc, local and global explainers.
  - ▶ <https://github.com/Trusted-AI/AIX360>
- ▶ InterpretML : intrinsic and post-hoc methods for Python and R.
  - ▶ <https://github.com/interpretml/interpret>
- ▶ Captum : library built for PyTorch models, focus on attribution methods.
  - ▶ <https://github.com/pytorch/captum>
- ▶ Fat Forensics : Python toolkit for evaluating Fairness, Accountability and Transparency of Artificial Intelligence systems.
  - ▶ <https://fat-forensics.org/>
- ▶ Google Pair : People + AI Guidebook with tools such as What-If Tool, Facets, LIT...
  - ▶ <https://pair.withgoogle.com/tools/>
- ▶ DALEX : post-hoc and model-agnostic explainers that allow local and global explanations.
  - ▶ <https://github.com/ModelOriented/DALEX>
- ▶ Shapash : Python library which aims to make machine learning interpretable and understandable by everyone, focus on visualizations.
  - ▶ <https://github.com/MAIF/shapash>
- ▶ ...

# The end : Non-Goals of XAI

## eXplainable Artificial Intelligence is not :

- ▶ About making ALL models explainable.
- ▶ About understanding EVERY SINGLE BIT about the model
- ▶ Against developing highly complex models.
- ▶ Only about gaining user trust or fairness.

Slide credit : Been Kim