Explainable AI Exercise

MLPM 2024

Florent Forest
Intelligent Maintenance and Operations Systems (IMOS), EPFL, Switzerland
florent.forest@epfl.ch
https://florentfo.rest

November 21, 2024

Terminology

XAI eXplainable Artificial Intelligence: field aiming to make AI systems more interpretable and trustworthy while maintaining their performance

Explainable Ability of a model to provide clear and understandable reasons for its decisions (*active*)

Interpretable Ability of a model to be understood and analyzed by human experts (*passive*).
"the ability to explain or to present in understandable terms to a human" [Doshi-Velez and Kim, 2017]

"the degree to which a human can understand the cause of a decision" [Miller, 2019]

Black-box model Non-interpretable model due to its complexity, opposed to **white-box** or **transparent**

Why XAI?

In many applications, **understanding** the model's predictions is as (or more) important than performance itself.

Why is this prediction wrong? Is the model correct for the right reasons? **How** did the model come to this conclusion? Why was my loan request denied? etc.

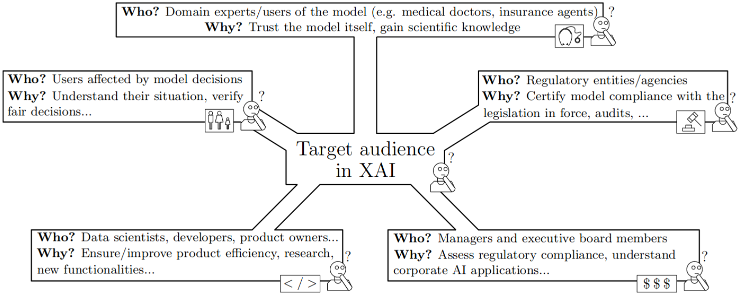
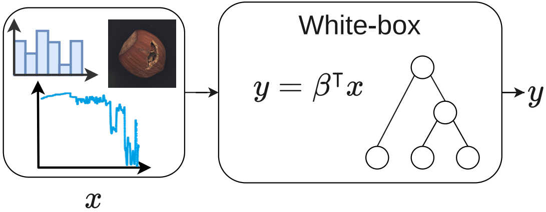


Figure 1: Target audience in XAI. Figure from [Arrieta et al., 2019].

Goals:

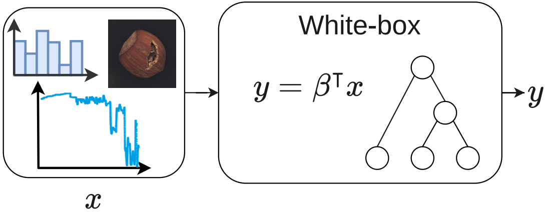
- **Trust**
- **Gain knowledge**
- **Understand decisions**
- **Improve** → fix/debug
- **Certify**, Assess compliance



The output is directly interpretable in terms of **weight coefficients** or **decision rules**.

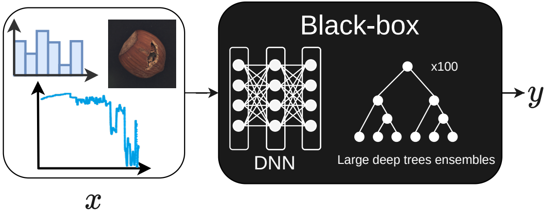
Performance ↓

Interpretability ↑



The output is directly interpretable in terms of **weight coefficients** or **decision rules**.

Performance ↓
Interpretability ↑



The output is the result of **non-linear** computations involving **millions** of parameters.

Performance ↑
Interpretability ↓

Taxonomies of XAI techniques

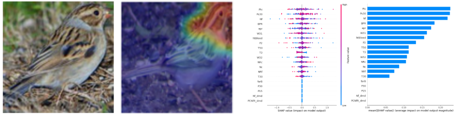
- Type of explanation
- Way of obtaining the explanation
- Type of data
- Global VS Local
- Model-agnostic VS Model-specific
- Post-hoc VS inherent/intrinsic/"by-design"
- Static VS Interactive
- etc.

Types of explanation

Feature attribution (a.k.a importance, relevance or saliency)

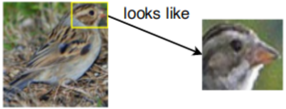
How do the input features affect the output?

- Most widely used type of XAI technique
- Perturbation-based: SHAP, LIME, RISE...
- Gradient or Propagation-based: Grad-CAM, Integrated Gradients, DeepLIFT, LRP...



Explanation-by-example

Case-based reasoning, Prototypes



Model simplification and rule extraction

Extracting a simple model (e.g. decision tree) or sets of logical rules to approximate a black-box model.

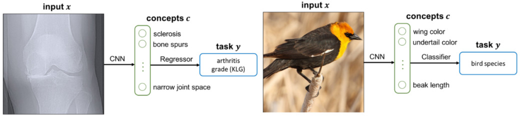
Counterfactual explanations

What is the smallest modification that would modify the model's outcome?

> You were denied a loan because your annual income was £30,000. If your income had been £45,000, you would have been offered a loan.

Concept explanations

TCAV, CBM, CEM...



Perturbation-based approaches

- Forward
- Perturb inputs (e.g. occlusion) $\rightarrow$ Measure impact on outputs
- Often requires many passes (inefficient)

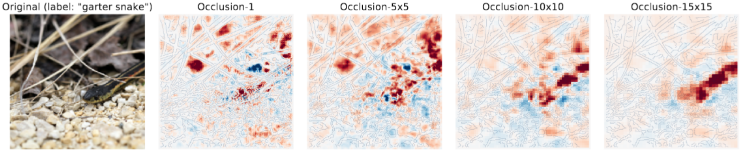
Propagation-based approaches

- Backward
- Back-propagate some importance signals from outputs to inputs
- Can be gradients or activation values
- Requires a single pass (efficient)

Occlude input features (replace with zero/mean/blur/etc.) and measure impact on model output:

$$\phi_i = f(\mathbf{x}) - f(\mathbf{x}_{[\mathbf{x}_i := b]}) \tag{1}$$

where ϕ_i is the importance of feature i, and b is a reference value.



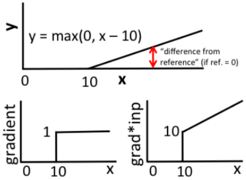
- Requires many evaluations $\rightarrow$ **computationally costly**!
- Depends on occlusion parameters

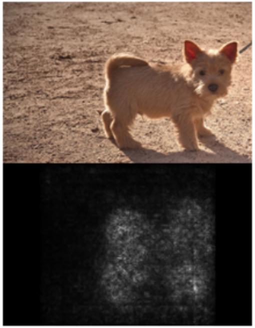
Extensions: RISE, D-RISE, masking methods for time series, graphs.

The **gradients** of output neurons with respect to **input dimensions** *at a given input* quantify the importance of each input **in the neighborhood of the current input**.

Let $\mathbf{x}$ be the current input and f the model function (e.g., score of a class):

- **Gradient**: $\left.\frac{\partial f}{\partial x}\right|_{x=\mathbf{x}}$
- **Input$\times$Gradient**: $\left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}} \odot \mathbf{x}$





References:

- [Baehrens et al., 2010], [Simonyan et al., 2014]
- Related: Deconvolutional networks, Guided Back-propagation

Limitation of perturbation- and propagation-based methods

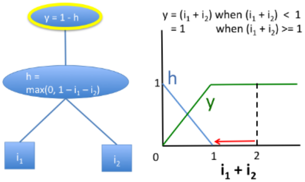
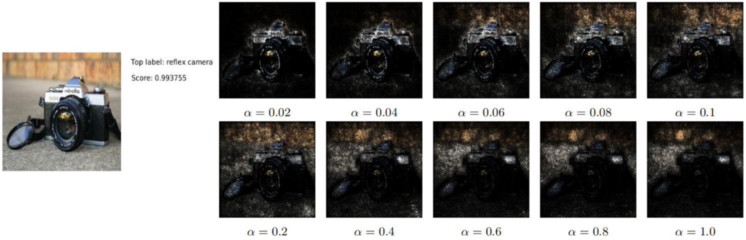


Figure 1. **Perturbation-based approaches and gradient-based approaches fail to model saturation**. Illustrated is a simple network exhibiting saturation in the signal from its inputs. At the point where $i_1 = 1$ and $i_2 = 1$, perturbing either i_1 or i_2 to 0 will not produce a change in the output. Note that the gradient of the output w.r.t the inputs is also zero when $i_1 + i_2 > 1$.

Feature attribution | Integrated Gradients

Solve the saturation issue by **integrating the gradients** between a *baseline* (e.g. zeros) and the current input:

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

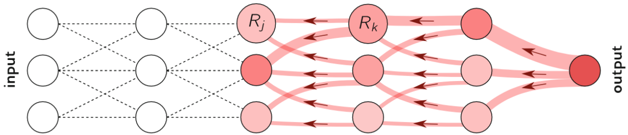


References:

- [Sundararajan et al., 2017]
- `distill.pub/2020/attribution-baselines/`

Relevance scores are propagated from the output to the input using **propagation rules**.

- Conservation properties
- Numerical stability
- Many different propagation rules, for different types of layers, activations and input ranges



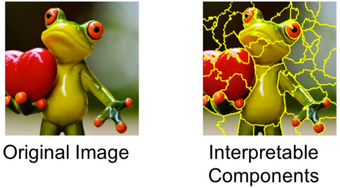
References:

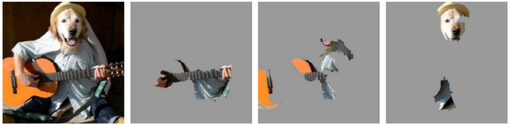
- [Bach et al., 2015], [Montavon et al., 2019]
- `github.com/chr5tphr/zennit`

LIME **locally approximates** the model f around a given input x using a **sparse linear model** fitted on **simplified inputs**.

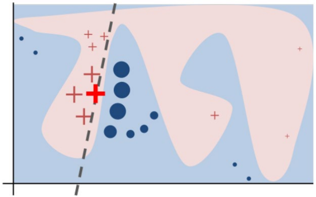
Steps:

1. Generate x' simplified version of x ("interpretable inputs")
2. Sample around x' in the simplified input space
3. Fit surrogate linear model (LASSO) weighted by a similarity kernel





(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*



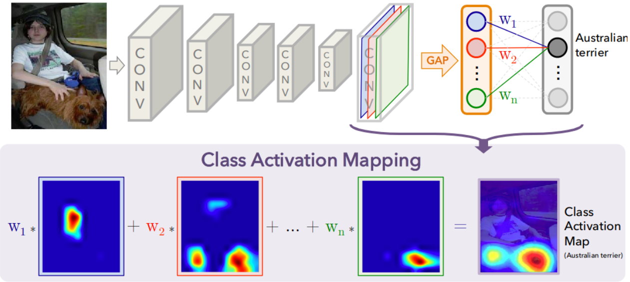
References:

- [Ribeiro et al., 2016]
- github.com/marcotcr/lime

Applying a **Global Average Pooling (GAP)** over the **last convolutional layer** in a CNN before the softmax allows to extract a **localization map** for a given class *C*:

$$S^c = \sum_k \underbrace{w_k^c}_{\text{class feature weights}} \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{A_{ij}^k}_{\text{feature map}}$$

$$S^c = \frac{1}{Z}\sum_i \sum_j \underbrace{\sum_k w_k^c A_{ij}^k}_{L^c_{\text{CAM}}}$$



Limitations: Low resolution, and requires specific architecture.

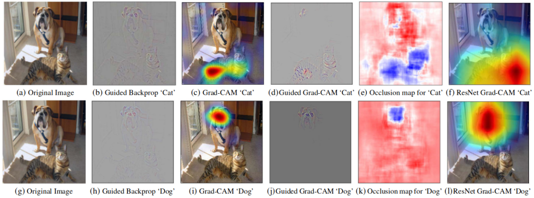
References:

- [Zhou et al., 2016]

Let A^k be feature maps (often last conv layer):

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

$$L^c_{\text{Grad-CAM}} = ReLU \underbrace{\left(\sum_k \alpha_k^c A^k \right)}_{\text{linear combination}}$$



(a) Original Image (b) Guided Backprop 'Cat' (c) Grad-CAM 'Cat' (d)Guided Grad-CAM 'Cat' (e) Occlusion map for 'Cat' (f) ResNet Grad-CAM 'Cat'

(g) Original Image (h) Guided Backprop 'Dog' (i) Grad-CAM 'Dog' (j)Guided Grad-CAM 'Dog' (k) Occlusion map for 'Dog' (l)ResNet Grad-CAM 'Dog'

Does not require GAP, but also limited to CNNs.

References:

- [Selvaraju et al., 2017]
- `github.com/jacobgil/pytorch-grad-cam`

Beyond feature attribution

While feature attribution is by far the most widely used XAI approach, is faces important shortcomings.

- Does not elucidate the decision-making process
- Local explanation
- Can be unreliable and misleading (sensitive to changes in the input, sometimes contradictory)
- Low-level features $\rightarrow$ hard to interpret

Other types of explanations or **inherently interpretable models** might be better suited.

References:

- [Rudin, 2019]

Exercise

Notebook on Moodle

References i

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2019).
Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI.
arXiv:1910.10045 [cs].

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015).
On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation.
PloS One, 10(7):e0130140.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. (2010).
How to Explain Individual Classification Decisions.
Journal of Machine Learning Research, 11(61):1803–1831.

Doshi-Velez, F. and Kim, B. (2017).
Towards A Rigorous Science of Interpretable Machine Learning.
arXiv:1702.08608 [cs, stat].

Miller, T. (2019).
Explanation in artificial intelligence: Insights from the social sciences.
Artificial Intelligence, 267:1–38.

Montavon, G., Binder, A., Lapuschkin, S., Samek, W., and Müller, K.-R. (2019).
Layer-Wise Relevance Propagation: An Overview.
In Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., and Müller, K.-R., editors, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700, pages 193–209. Springer International Publishing, Cham.
Series Title: Lecture Notes in Computer Science.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016).
”Why Should I Trust You?”: Explaining the Predictions of Any Classifier.
arXiv:1602.04938 [cs, stat].

Rudin, C. (2019).
Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.
Nature Machine Intelligence, 1(5):206–215.
Publisher: Nature Publishing Group.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017).
Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization.
In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626.
ISSN: 2380-7504.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014).
Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps.
arXiv:1312.6034 [cs].

Sundararajan, M., Taly, A., and Yan, Q. (2017).
Axiomatic Attribution for Deep Networks.
arXiv:1703.01365 [cs].

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016).
Learning Deep Features for Discriminative Localization.
In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929.
ISSN: 1063-6919.