Explanable AI
Interpretability of deep models

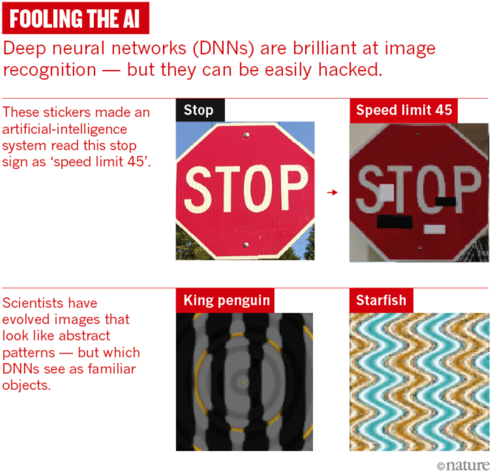
Ronan Sicre

QARMA, LIS

Interpretability is important for high stakes decisions.

Model understanding is absolutely critical in several domains -- particularly those involving *high stakes decisions*!

Interpretability is important for trustworthy DNNs.



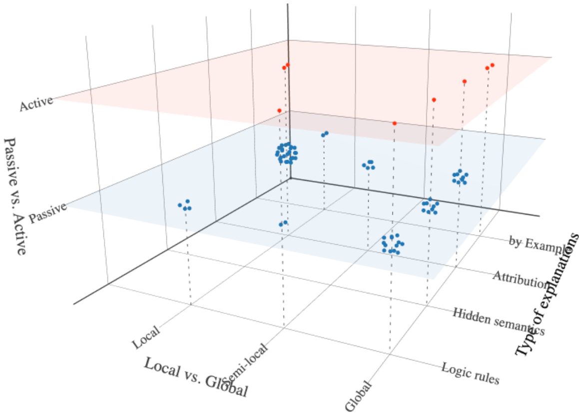
- Robustness and improvements
- Trust and understanding
- Security, legal necessity and responsibility

Dimensions of interpretability methods

Dimension 1 — Passive vs. Active Approaches	
Passive	Post hoc explain trained neural networks
Active	Actively change the network architecture or training process for better interpretability
Dimension 2 — Type of Explanations (in the order of increasing explanatory power)	
To explain a prediction/class by	
Examples	Provide example(s) which may be considered similar or as prototype(s)
Attribution	Assign credit (or blame) to the input features (e.g. feature importance, saliency masks)
Hidden semantics	Make sense of certain hidden neurons/layers
Rules	Extract logic rules (e.g. decision trees, rule sets and other rule formats)
Dimension 3 — Local vs. Global Interpretability (in terms of the input space)	
Local	Explain network's *predictions on individual samples* (e.g. a saliency mask for an input image)
Semi-local	In between, for example, explain a group of similar inputs together
Global	Explain the network *as a whole* (e.g. a set of rules/a decision tree)

[ZTLT20]

Dimensions of interpretability methods



[ZTLT20]

Attribution

	Local	Semi-Local	Global
Active (Transparency)	ExpO, DAPr, LFI-CAM	–	Dual-net (feature importance)
Passive (Post hoc)	LIME, MAPLE, Partial derivarives, DeconvNet, Guided backprop, Grad-CAM, Shapley values, Sensitivity analysis, Feature selector, Bias attribution	DeepLIFT, LRP, Integrated gradients, Feature selector, MAME	Feature selector, TCAV, ACE, SpRAy, MAME, DeepConsensus

[ZTLT20]

Post-hoc interpretation

Model agnostic attribution

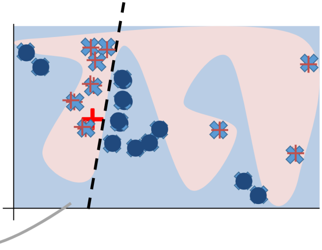
- **LIME** [RSG16]
- **Shapley** [SK10, AOG19]
- **Sensitivity analysis**: perturbation [PDS18, CCGD19, PPG20]
- ......

Saliency map

- **Gradient-based and backpropagation methods**: Gradient [AGGK18, SDBR15, BSH+10], Guidedbackprop [SDBR15], Grad-CAM [SCD+17]...
- **Discrete Gradient**: LRP [BBM+15, LTB+13, AMMS17], DeepLIFT [SGK17], intergrated Grad [STY17]
- **Adversarial perturbation based**: perceptual ball [ELR21]
- ......

LIME: Sparse Linear Explanation

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

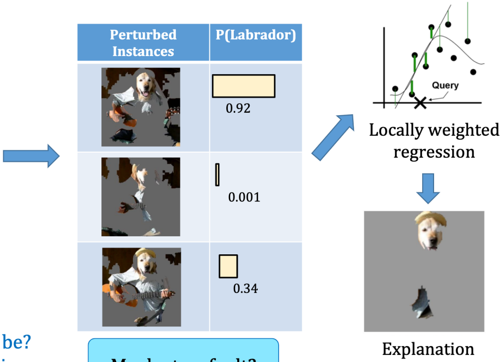


[RSG16]

LIME: examples

Original Image

P(labrador) = 0.21



LIME is quite customizable:

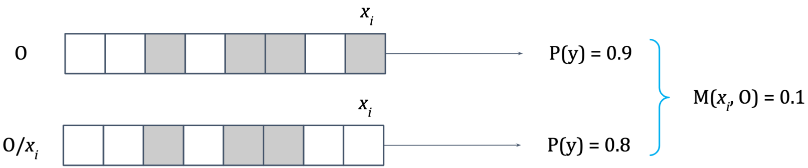
- How to perturb?
- Distance/similarity?
- How *local* you want it to be?
- How to express explanation

Maybe to a fault?

[RSG16]

Shapley

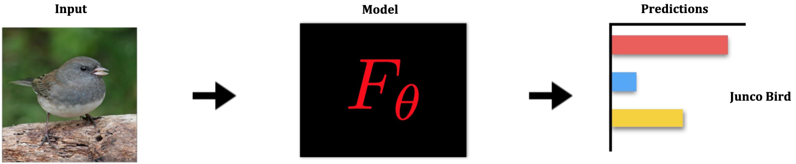
Marginal contribution of each feature towards the prediction, averaged over all possible permutations.



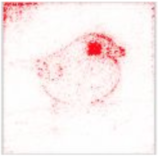
Fairly attributes the prediction to all the features.

[SK10, AOG19]

Saliency Map Overview



What parts of the input are most relevant for the model's prediction: 'Junco Bird'?



- **Feature Attribution**
- **'Saliency Map'**
- **Heatmap**

Class activation maps

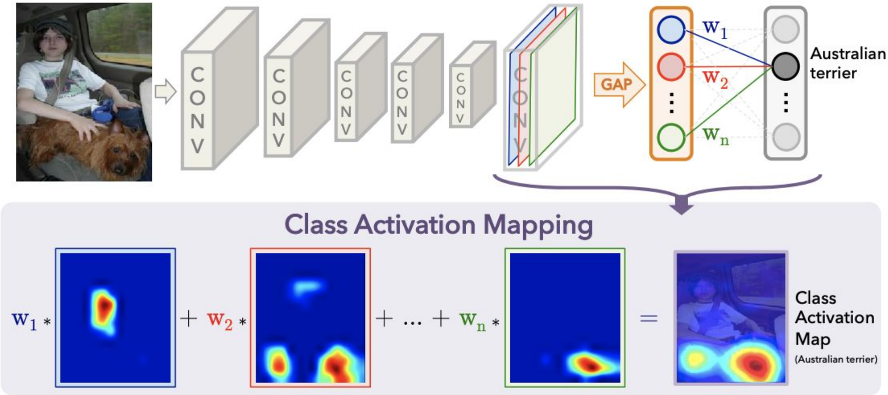


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

24

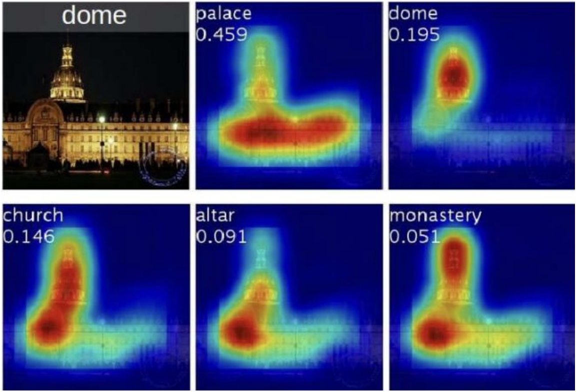
Class activation maps

2) Méthode CAM et ses dérivées

Class Activation Mapping :

Obtention d'une carte de saillance relativement à la classe choisie

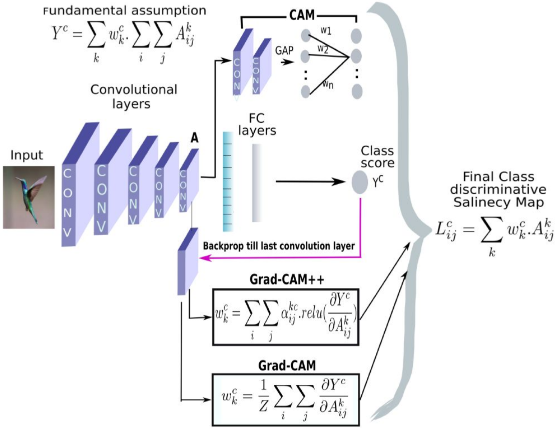
$$\rightarrow \quad L^c_{ij} = \sum_k w^c_k . A^k_{ij}$$



Class activation maps

2) Méthode CAM et ses dérivées

GradCAM et GradCAM++ :



→ L'expression de la carte de saillance a toujours la même forme, seule l'expression des poids change.

GradCAM :
$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

GradCAM++:
$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} . relu(\frac{\partial Y^c}{\partial A_{ij}^k})$$

avec

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \{\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\}}$$

Class activation maps

3) Autres méthodes

Ablation CAM :

- "Gradient-free" method

→ "Ablation" des features maps de la dernière couche convolutionnelle, et calcul du score obtenu pour la classe prédite.

- Nouvelle formule pour calculer les poids : $w_k^c = \frac{y^c - y_k^c}{y^c}$

- Carte de saillance Ablation-CAM : $L^c_{Ablation-CAM} = \text{ReLU}\left(\sum_k w_k^c A_k\right)$

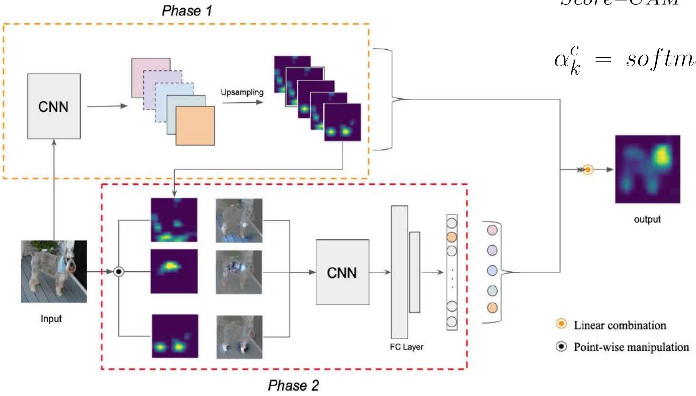
Class activation maps

3) Autres méthodes

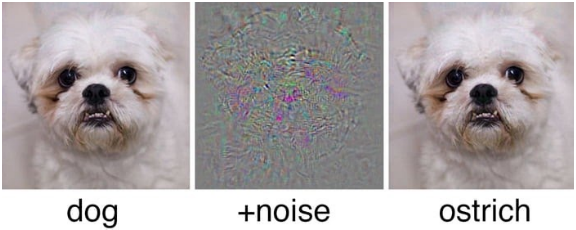
Score CAM :

$$L^c_{Score-CAM} = ReLU(\sum_k \alpha^c_k A^k_l)$$

$$\alpha^c_k = softmax(f^c(M_k) - f^c(X))$$



Perceptual ball



Adversarial Perturbation

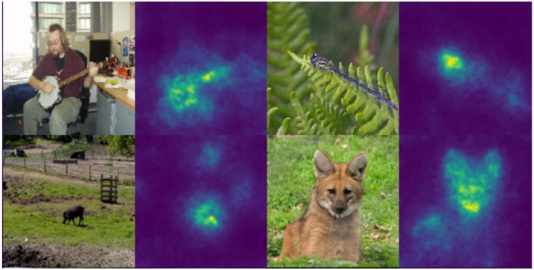
- Misclassification c$(f(\mathbf{x}+\mathbf{r})) \neq l_g$
- Small Distortion Norm ($\|\mathbf{r}\|_2$ or $\|\mathbf{r}\|_\infty$)

[ELR21]

Perceptual ball

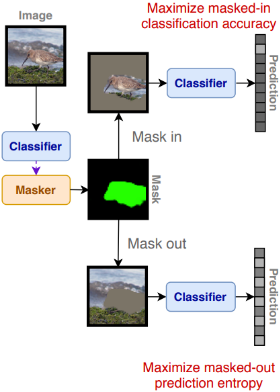
Generate adversarial perturbation

- Misclassification: $\ell(f(\mathbf{x}+\mathbf{r}), l_g) = f_{l_g}(\mathbf{x}+\mathbf{r}) - \max f_{l \neq l_g}(\mathbf{x}+\mathbf{r})$
- Small distortion: $\sum_i \|f^i(\mathbf{x}+\mathbf{r}) - f^i(\mathbf{x})\|_2 + \|\mathbf{r}\|_2$



[ELR21]

Masking-based saliency map method

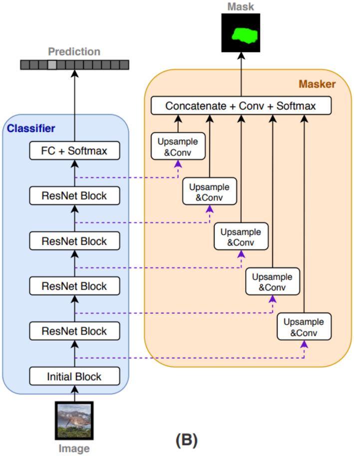


Loss function

- $L_{out}(f_{l_g}(\mathbf{x} \odot (1 - \mathbf{m})))$
- $L_{in}(f_{l_g}(\mathbf{x} \odot \mathbf{m}))$
- $R(\mathbf{m})$

[PPG20]

Masking-based saliency map method



[PPG20]

Resources and tools

Resources for free!:

- A Survey on Neural Network Interpretability
- Tutorial on Explaining ML Predictions: State-of-the-art, Challenges, and Opportunities - NeurIPS 2020 YT
- Tutorial on Interpretable Machine Learning - CVPR 2020

Some tools:

- Pytorch CAM-based interpretability methods
- Colah's blog
- Comparison CAM, SHAP, LIME
- TorchRay

Thank you!

QUESTIONS

Q&A

ANSWERS

References

[AGGK18] Julius Adebayo, Justin Gilmer, Ian J. Goodfellow, and Been Kim. Local explanation methods for deep neural networks lack sensitivity to parameter values. *ICLR Workshop*, 2018.

[AMMS17] Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Explaining recurrent neural network predictions in sentiment analysis. *EMNLP Workshop*, 2017.

[AOG19] Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pages 272–281. PMLR, 2019.

References (cont.)

[BBM+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 2015.

[BSH+10] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *J. MLR*, 2010.

[CCGD19] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. *ICLR*, 2019.

References (cont.)

[ELR21] Andrew Elliott, Stephen Law, and Chris Russell. Explaining classifiers using adversarial perturbations on the perceptual ball. In *CVPR*, 2021.

[LPOK21] Kwang Hee Lee, Chaewon Park, Junghyun Oh, and Nojun Kwak. Lfi-cam: Learning feature importance for better visual explanation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1355–1363, 2021.

References (cont.)

[LTB+13] Will Landecker, Michael D. Thomure, LuÃs M. A. Bettencourt, Melanie Mitchell, Garrett T. Kenyon, and Steven P. Brumby. Interpreting individual classifications of hierarchical networks. In *CIDM*, 2013.

[PASC+19] Gregory Plumb, Maruan Al-Shedivat, Angel Alexander Cabrera, Adam Perer, Eric Xing, and Ameet Talwalkar. Regularizing black-box models for improved interpretability. *arXiv preprint arXiv:1902.06787*, 2019.

[PDS18] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *BMVC*, 2018.

References (cont.)

[PPG20] Jason Phang, Jungkyu Park, and Krzysztof J Geras. Investigating and simplifying masking-based saliency methods for model interpretability. *arXiv preprint arXiv:2010.09750*, 2020.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *SIGKDD*, KDD '16, 2016.

References (cont.)

[SCD+17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *CVPR*, 2017.

[SDBR15] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *ICLR*, 2015.

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017.

References (cont.)

[SK10] Erik Strumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, 11:1–18, 2010.

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017.

[WC20] Maksymilian Wojtas and Ke Chen. Feature importance ranking for deep learning. *arXiv preprint arXiv:2010.08973*, 2020.

[WJL19] Ethan Weinberger, Joseph Janizek, and Su-In Lee. Learning deep attribution priors based on prior knowledge. *arXiv preprint arXiv:1912.10065*, 2019.

References (cont.)

[ZTLT20] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *arXiv preprint arXiv:2012.14261*, 2020.