Learning Objectives, Clever Hans, and Explainable AI

Grégoire Montavon [1,2]

[1]Dept. of Mathematics and Computer Science, Freie Universität Berlin
[2]BIFOLD – Berlin Institute for the Foundations of Learning and Data

CO@Work 2024, ZIB – 20 September 2024

Formulating the ML Problem

Formulating the ML Problem

input $\quad \boldsymbol{x} \in \mathbb{R}^d$

Formulating the ML Problem

input	$\boldsymbol{x} \in \mathbb{R}^d$
target	$t \in \mathbb{R}$

Formulating the ML Problem

input	$\boldsymbol{x} \in \mathbb{R}^d$
target	$t \in \mathbb{R}$
data distribution	$p(\boldsymbol{x}, t)$

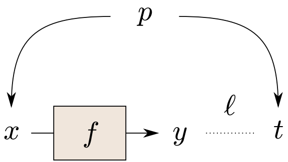
Formulating the ML Problem

$$\begin{aligned}
\text{input} \quad & \boldsymbol{x} \in \mathbb{R}^d \\
\text{target} \quad & t \in \mathbb{R} \\
\text{data distribution} \quad & p(\boldsymbol{x}, t) \\
\text{predictions} \quad & f(\boldsymbol{x}) \in \mathbb{R}
\end{aligned}$$

Formulating the ML Problem

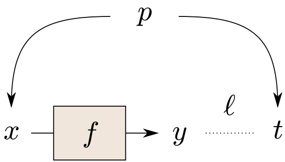
input	$\boldsymbol{x} \in \mathbb{R}^d$	
target	$t \in \mathbb{R}$	
data distribution	$p(\boldsymbol{x}, t)$	
predictions	$f(\boldsymbol{x}) \in \mathbb{R}$	
loss	$\ell(f(\boldsymbol{x}), t)$	(cost of outputing $f(x)$ given target t)

Formulating the ML Problem



input	$\boldsymbol{x} \in \mathbb{R}^d$	
target	$t \in \mathbb{R}$	
data distribution	$p(\boldsymbol{x}, t)$	
predictions	$f(\boldsymbol{x}) \in \mathbb{R}$	
loss	$\ell(f(\boldsymbol{x}), t)$	(cost of outputing $f(x)$ given target t)

Formulating the ML Problem

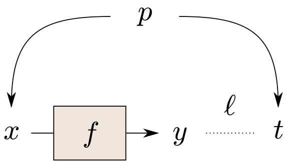


input	$\boldsymbol{x} \in \mathbb{R}^d$
target	$t \in \mathbb{R}$
data distribution	$p(\boldsymbol{x}, t)$
predictions	$f(\boldsymbol{x}) \in \mathbb{R}$
loss	$\ell(f(\boldsymbol{x}), t)$ (cost of outputing $f(\boldsymbol{x})$ given target t)

expected cost:

$$\int \ell(f(\boldsymbol{x}), t) dp(\boldsymbol{x}, t)$$

Formulating the ML Problem



input	$\boldsymbol{x} \in \mathbb{R}^d$
target	$t \in \mathbb{R}$
data distribution	$p(\boldsymbol{x}, t)$
predictions	$f(\boldsymbol{x}) \in \mathbb{R}$
loss	$\ell(f(\boldsymbol{x}), t)$ (cost of outputing $f(\boldsymbol{x})$ given target t)

Minimization of expected cost:

$$\arg\min_f \left\{ \int \ell(f(\boldsymbol{x}), t) dp(\boldsymbol{x}, t) \right\}$$

Formulating the ML Problem (cont.)

$$\arg\min_f \left\{ \int \ell(f(\boldsymbol{x}), t) dp(\boldsymbol{x}, t) \right\}$$

Formulating the ML Problem (cont.)

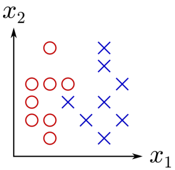
$$\arg\min_{f} \left\{ \int \ell(f(\boldsymbol{x}), t) dp(\boldsymbol{x}, t) \right\}$$

Caveat:

- This problem cannot be fully specified, because we do not know the true distribution p, and we only have access to a finite dataset:

$$\mathcal{D} = \{(\boldsymbol{x}_1, t_1), \ldots, (\boldsymbol{x}_N, t_N)\}$$

.



Formulating the ML Problem (cont.)

Addressing limited data:

Formulating the ML Problem (cont.)

Addressing limited data:

- Replace p with the empirical data distribution.

Formulating the ML Problem (cont.)

Addressing limited data:

- Replace p with the empirical data distribution.
- This gives the objective:

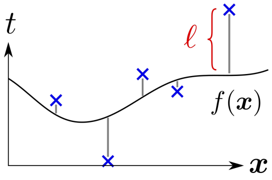
$$\arg\min_{f} \left\{ \frac{1}{N} \sum_{k=1}^{N} \ell(f(\boldsymbol{x}_k), t_k) \right\}$$

Formulating the ML Problem (cont.)

Addressing limited data:

- Replace p with the empirical data distribution.
- This gives the objective:

$$\arg\min_f \left\{ \frac{1}{N} \sum_{k=1}^{N} \ell(f(\boldsymbol{x}_k), t_k) \right\}$$

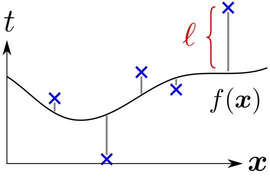


Formulating the ML Problem (cont.)

Addressing limited data:

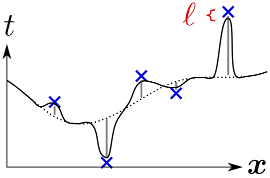
- Replace p with the empirical data distribution.
- This gives the objective:

$$\arg\min_f \left\{ \frac{1}{N} \sum_{k=1}^{N} \ell(f(\boldsymbol{x}_k), t_k) \right\}$$



Problem:

- The model f may only memorize the data and fail to make truthful predictions on the rest of p. (overfitting)



Formulating the ML Problem (cont.)

Addressing limited data (improved):

Formulating the ML Problem (cont.)

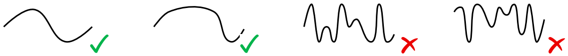
Addressing limited data (improved):

- Make a distinction between functions $f \in \mathcal{F}$ that are immune to overfitting (e.g. functions with few variations, classifiers with large margin) and functions that do not.

Formulating the ML Problem (cont.)

Addressing limited data (improved):

- Make a distinction between functions $f \in \mathcal{F}$ that are immune to overfitting (e.g. functions with few variations, classifiers with large margin) and functions that do not.



Formulating the ML Problem (cont.)

Addressing limited data (improved):

- Make a distinction between functions $f \in \mathcal{F}$ that are immune to overfitting (e.g. functions with few variations, classifiers with large margin) and functions that do not.

- One can then formulate the optimization problem as:

$$\arg\min_f \left\{ \frac{1}{N} \sum_{k=1}^{N} \ell(f(\boldsymbol{x}_k), t_k) \ ; \ f \in \mathcal{F} \right\}$$

Formulating the ML Problem (cont.)

Addressing limited data (improved):

- Make a distinction between functions $f \in \mathcal{F}$ that are immune to overfitting (e.g. functions with few variations, classifiers with large margin) and functions that do not.

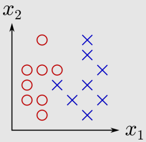
- One can then formulate the optimization problem as:

$$\arg\min_f \left\{ \frac{1}{N} \sum_{k=1}^{N} \ell(f(\boldsymbol{x}_k), t_k) \; ; \; f \in \mathcal{F} \right\}$$

Question:

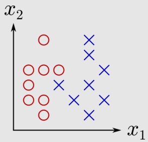
- How to specify $\mathcal{F}$?

Special Case of iid. Data



Scenario 1: Data sampled iid. according to the underlying distribution $p(\boldsymbol{x}, t)$.

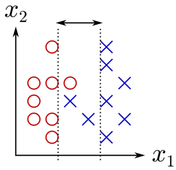
Special Case of iid. Data



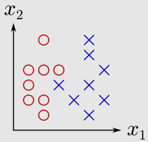
Scenario 1: Data sampled iid. according to the underlying distribution $p(\boldsymbol{x}, t)$.

Restrict $\mathcal{F}$ to the space of functions that cannot overfit, e.g. large-margin classifiers

$$\mathcal{F} = \{f : f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} \,;\, \|\boldsymbol{w}\| \leq 1/M\}.$$

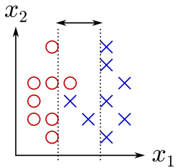


Special Case of iid. Data



Scenario 1: Data sampled iid. according to the underlying distribution $p(\boldsymbol{x}, t)$.

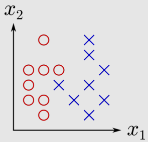
Restrict $\mathcal{F}$ to the space of functions that cannot overfit, e.g. large-margin classifiers



$$\mathcal{F} = \{f : f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}\,;\ \|\boldsymbol{w}\| \leq 1/M\}.$$

- Often reduces the gap between training and true error significantly (e.g. as measured by holdout validation).

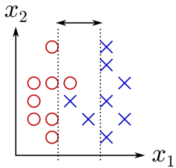
Special Case of iid. Data



Scenario 1: Data sampled iid. according to the underlying distribution $p(\boldsymbol{x}, t)$.

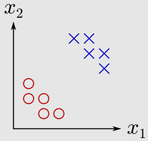
Restrict $\mathcal{F}$ to the space of functions that cannot overfit, e.g. large-margin classifiers

$$\mathcal{F} = \{f : f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}\,;\ \|\boldsymbol{w}\| \leq 1/M\}.$$



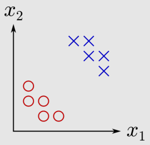
- Often reduces the gap between training and true error significantly (e.g. as measured by holdout validation).
- Also comes with theory (e.g. VC-theory) [Vapnik 2000].

More General Case



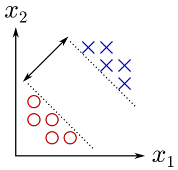
Scenario 2: Data sampled from a different distribution $q(\boldsymbol{x}) \neq p(\boldsymbol{x})$.

More General Case

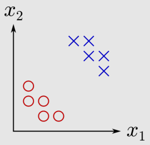


Scenario 2: Data sampled from a different distribution $q(\boldsymbol{x}) \neq p(\boldsymbol{x})$.

Observation: Restricting $\mathcal{F}$ to large-margin classifiers may fail to approximate the true decision boundary (here based on x_1).

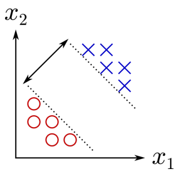


More General Case



Scenario 2: Data sampled from a different distribution $q(\boldsymbol{x}) \neq p(\boldsymbol{x})$.

Observation: Restricting $\mathcal{F}$ to large-margin classifiers may fail to approximate the true decision boundary (here based on x_1).



Hard to choose $\mathcal{F}$ without human intervention.

More General Case

Image source: Li et al. A Whac-A-Mole Dilemma (2022)

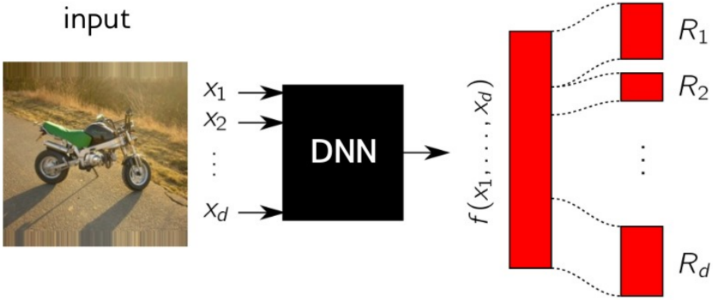
More General Case

Image source: Li et al. A Whac-A-Mole Dilemma (2022)

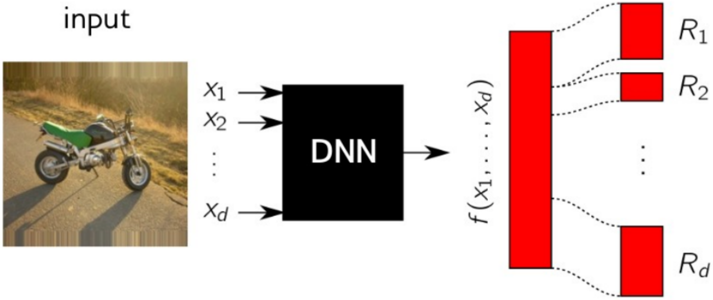
Hard to choose $\mathcal{F}$ at all.

Part 2: Explainable AI

Attribution of a Prediction to Input Features

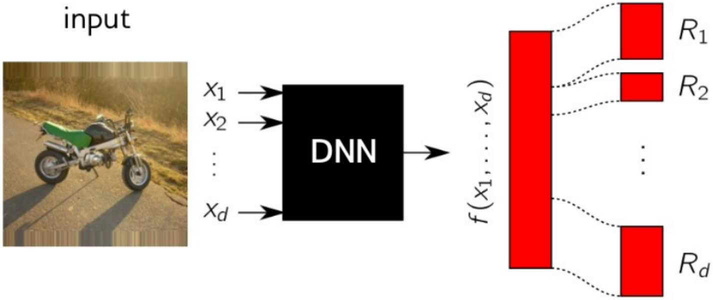


Attribution of a Prediction to Input Features



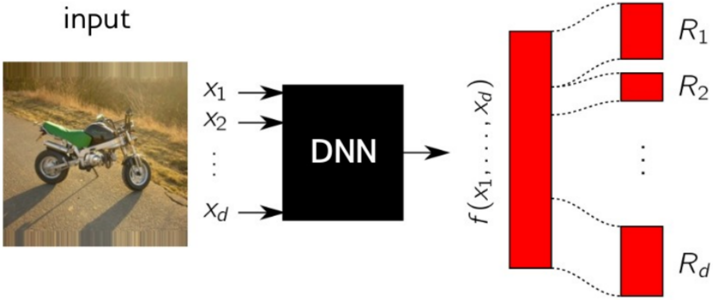
1. The data point $\boldsymbol{x} \in \mathbb{R}^d$ is fed to the ML model and we get a prediction $f(\boldsymbol{x}) \in \mathbb{R}$.

Attribution of a Prediction to Input Features



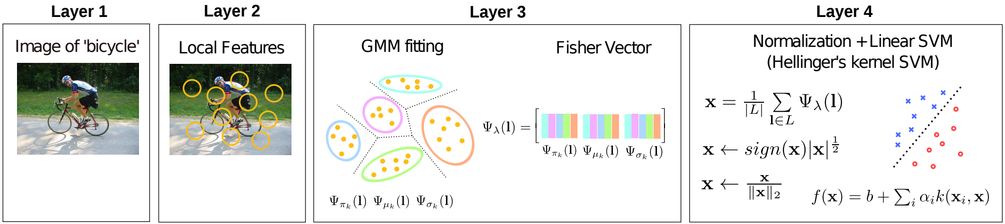
1. The data point $\boldsymbol{x} \in \mathbb{R}^d$ is fed to the ML model and we get a prediction $f(\boldsymbol{x}) \in \mathbb{R}$.
2. We explain the prediction by identifying the additive contribution of each input feature.

Attribution of a Prediction to Input Features



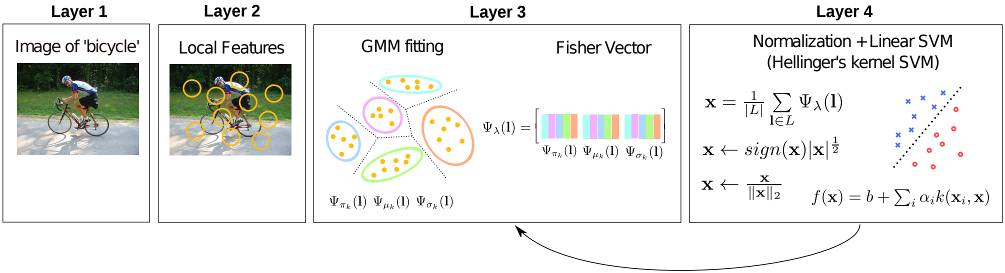
1. The data point $\boldsymbol{x} \in \mathbb{R}^d$ is fed to the ML model and we get a prediction $f(\boldsymbol{x}) \in \mathbb{R}$.
2. We explain the prediction by identifying the additive contribution of each input feature.
3. Important property of attribution: **conservation** ($\sum_{i=1}^{d} R_i = f(\boldsymbol{x})$).

Layer-Wise Relevance Propagation [Bach et al. PLOS 2015]



[Lapuschkin et al. CVPR 2016]

Layer-Wise Relevance Propagation [Bach et al. PLOS 2015]



Relevance Conservation

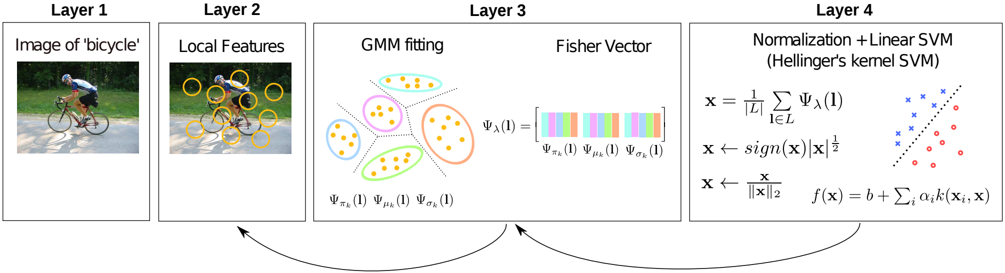
$$\sum_i R_i^{(3)} = f(\mathbf{x})$$

Redistribution Formula

$$\boxed{R_d^{(3)} = \sum_i \alpha_i y_i \phi(x_i)_d \phi(x)_d + \frac{b}{D}}$$

[Lapuschkin et al. CVPR 2016]

Layer-Wise Relevance Propagation [Bach et al. PLOS 2015]

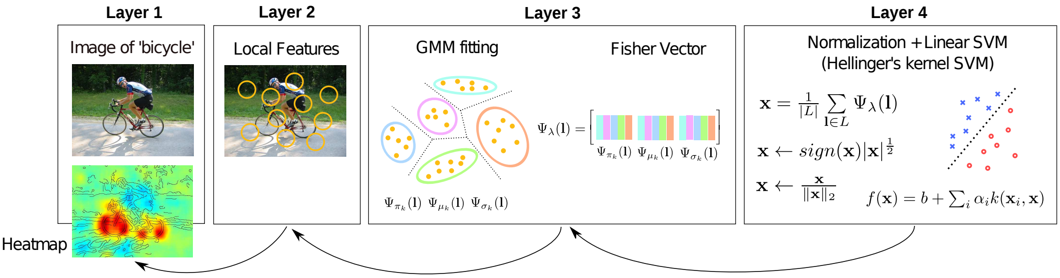


Relevance Conservation $\sum_i R_i^{(2)} = \sum_j R_j^{(3)}$ $\sum_i R_i^{(3)} = f(\mathbf{x})$

Redistribution Formula $\boxed{R_l^{(2)} = \sum_d R_d^{(3)} \frac{z_{ld}}{\sum_{l'} z_{l'd} + \epsilon \cdot \mathrm{sign}(\sum_{l'} z_{l'd})}}$ $\boxed{R_d^{(3)} = \sum_i \alpha_i y_i \phi(x_i)_d \phi(x)_d + \frac{b}{D}}$

[Lapuschkin et al. CVPR 2016]

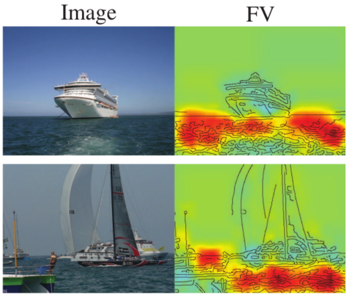
Layer-Wise Relevance Propagation [Bach et al. PLOS 2015]



Relevance Conservation	$\sum_i R_i^{(1)} = \sum_j R_j^{(2)}$	$\sum_i R_i^{(2)} = \sum_j R_j^{(3)}$	$\sum_i R_i^{(3)} = f(\mathbf{x})$
Redistribution Formula	$R_p^{(1)} = \sum_{l \in L(p)} \frac{R_l^{(2)}}{\lvert \mathrm{area}(l) \rvert}$	$R_l^{(2)} = \sum_d R_d^{(3)} \frac{z_{ld}}{\sum_{l'} z_{l'd} + \epsilon \cdot \mathrm{sign}(\sum_{l'} z_{l'd})}$	$R_d^{(3)} = \sum_i \alpha_i y_i \phi(x_i)_d \phi(x)_d + \frac{b}{D}$

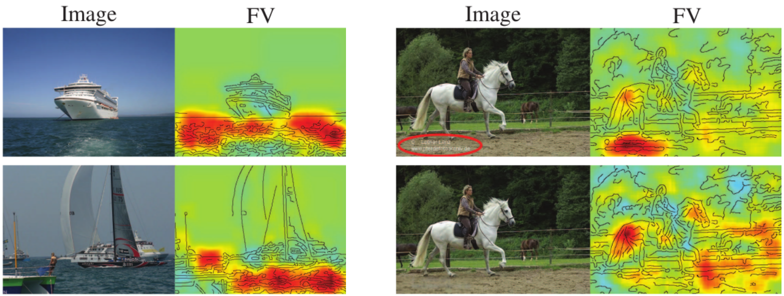
[Lapuschkin et al. CVPR 2016]

Layer-Wise Relevance Propagation



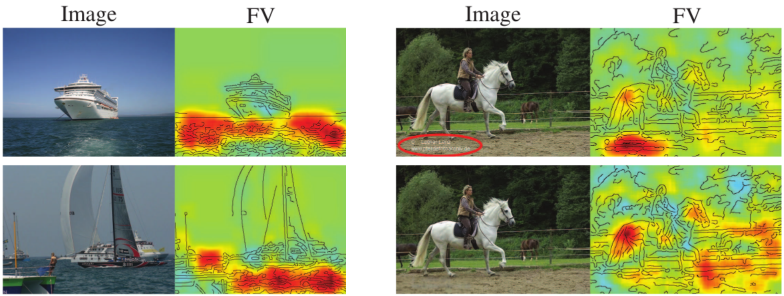
[Lapuschkin et al. CVPR 2016]

Layer-Wise Relevance Propagation



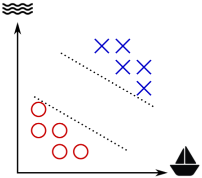
[Lapuschkin et al. CVPR 2016]

Layer-Wise Relevance Propagation

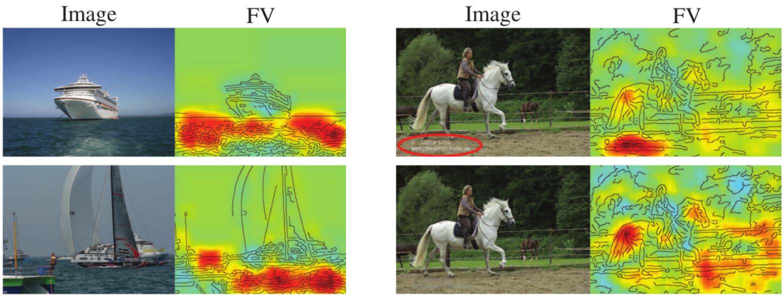


[Lapuschkin et al. CVPR 2016]

- Predictions are accurate but based on the wrong features (aka. the Clever Hans effect, cf. [Lapuschkin et al. NatComm 2019]).

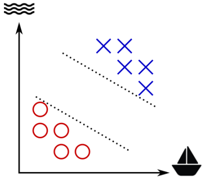


Layer-Wise Relevance Propagation



[Lapuschkin et al. CVPR 2016]

- Predictions are accurate but based on the wrong features (aka. the Clever Hans effect, cf. [Lapuschkin et al. NatComm 2019]). The model may start making errors when wrong features are missing.



Limits of 'Classical' Explanations

Limits of 'Classical' Explanations

Observation:

- Several concepts (including valid and invalid ones) are entangled in the same explanation.

Limits of ‘Classical’ Explanations

Observation:

- Several concepts (including valid and invalid ones) are entangled in the same explanation.

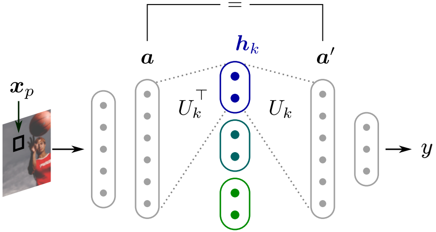
Question:

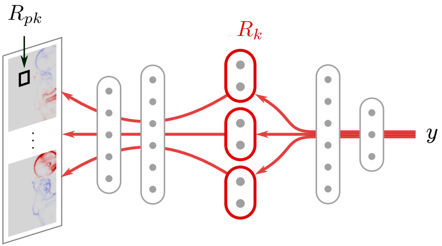
- Can we disentangle explanations into multiple distinct components (aka. concepts) so that they become more actionable? [Chormai et al. TPAMI 2024]

Explanation Subspaces [Chormai et al. TPAMI 2024]

Ideas:

- Extend the neural network with a 'virtual layer', which transforms the representation and back using an orthogonal matrix U.

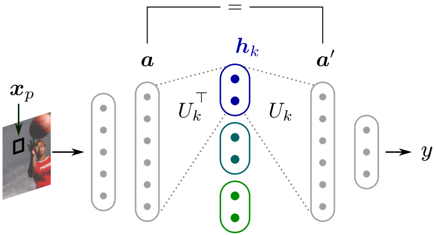


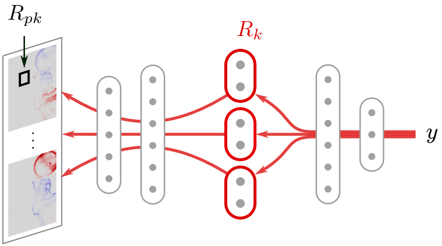


Explanation Subspaces [Chormai et al. TPAMI 2024]

Ideas:

- Extend the neural network with a 'virtual layer', which transforms the representation and back using an orthogonal matrix U.
- Decompose U into blocks $U = (U_1| \dots |U_K)$, where each block represents a subspace (one component of the explanation).





Optimizing Subspaces (DRSA) [Chormai et al. TPAMI 2024]

Notation:

- Neurons indexed by i
- Data points indexed by n
- Concepts indexed by k

Optimizing Subspaces (DRSA) [Chormai et al. TPAMI 2024]

Notation:

- Neurons indexed by i
- Data points indexed by n
- Concepts indexed by k
- Activations vector $\boldsymbol{a}$

Optimizing Subspaces (DRSA) [Chormai et al. TPAMI 2024]

Notation:

- Neurons indexed by i
- Data points indexed by n
- Concepts indexed by k
- Activations vector $\boldsymbol{a}$
- Response vector $\boldsymbol{c}$

Optimizing Subspaces (DRSA) [Chormai et al. TPAMI 2024]

Notation:

- Neurons indexed by i
- Data points indexed by n
- Concepts indexed by k
- Activations vector $\boldsymbol{a}$
- Response vector $\boldsymbol{c}$
- Mapping to concept U_k

Optimizing Subspaces (DRSA) [Chormai et al. TPAMI 2024]

Notation:

- Neurons indexed by i
- Data points indexed by n
- Concepts indexed by k
- Activations vector $\boldsymbol{a}$
- Response vector $\boldsymbol{c}$
- Mapping to concept U_k

Relevance of neuron i:

$$R_i = a_i \odot c_i$$

Optimizing Subspaces (DRSA) [Chormai et al. TPAMI 2024]

Notation:

- Neurons indexed by i
- Data points indexed by n
- Concepts indexed by k
- Activations vector $\boldsymbol{a}$
- Response vector $\boldsymbol{c}$
- Mapping to concept U_k

Relevance of neuron i:

$$R_i = a_i \odot c_i$$

Relevance of concept k:

$$R_k = \left((U_k^\top \boldsymbol{a})^\top (U_k^\top \boldsymbol{c}) \right)^+$$

Optimizing Subspaces (DRSA) [Chormai et al. TPAMI 2024]

Notation:

- Neurons indexed by i
- Data points indexed by n
- Concepts indexed by k
- Activations vector $\boldsymbol{a}$
- Response vector $\boldsymbol{c}$
- Mapping to concept U_k

Relevance of neuron i:

$$R_i = a_i \odot c_i$$

Relevance of concept k:

$$R_k = \left((U_k^\top \boldsymbol{a})^\top (U_k^\top \boldsymbol{c}) \right)^+$$

DRSA optimization objective:

$$\max_{(U_k)_k} \left\{ \operatorname{smin}_k \left\{ \operatorname{smax}_n \left\{ R_{kn} \right\} \right\} \right\}$$

Optimizing Subspaces (DRSA) [Chormai et al. TPAMI 2024]

Notation:

- Neurons indexed by i
- Data points indexed by n
- Concepts indexed by k
- Activations vector $\boldsymbol{a}$
- Response vector $\boldsymbol{c}$
- Mapping to concept U_k

Relevance of neuron i:

$$R_i = a_i \odot c_i$$

Relevance of concept k:

$$R_k = \left((U_k^\top \boldsymbol{a})^\top (U_k^\top \boldsymbol{c}) \right)^+$$

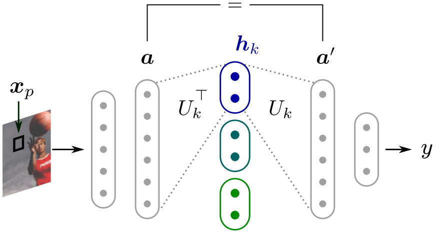
DRSA optimization objective:

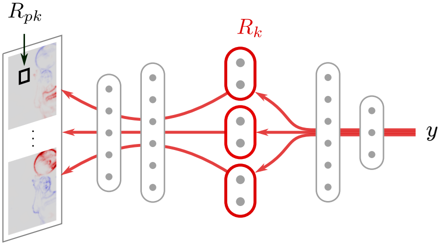
$$\max_{(U_k)_k} \left\{ \underset{k}{\text{smin}} \left\{ \underset{n}{\text{smax}} \left\{ R_{kn} \right\} \right\} \right\}$$

Focuses on what is *relevant* ($\neq$ representation learning).

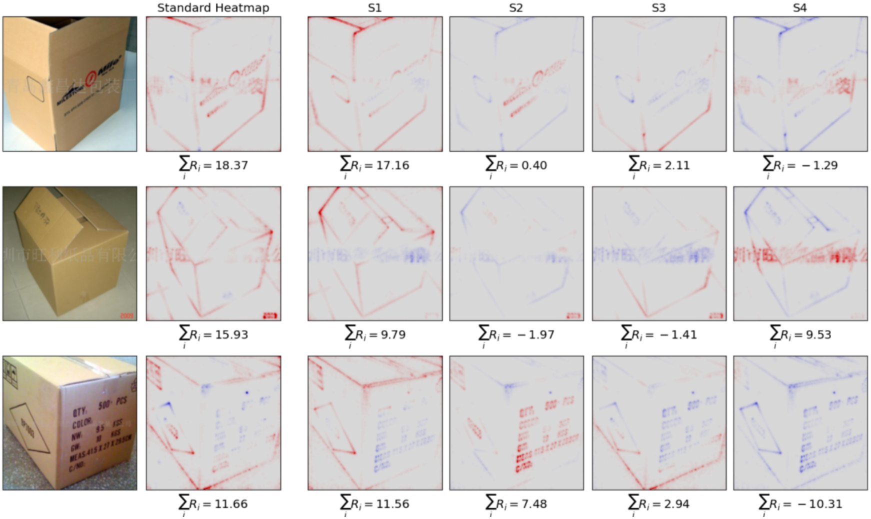
Explanation Subspaces [Chormai et al. TPAMI 2024]

Recap:

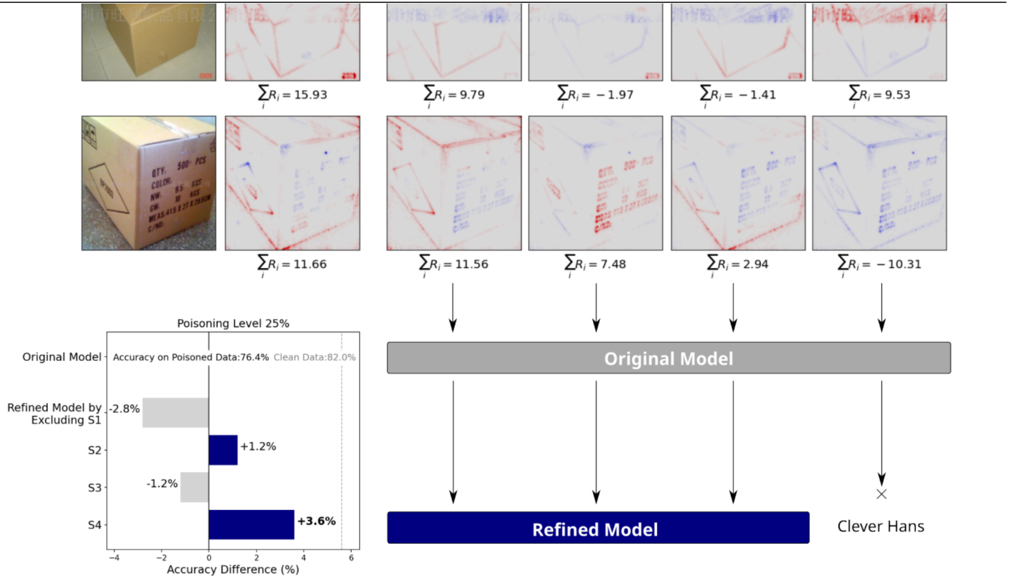




Inspecting Models with DRSA [Chormai et al. TPAMI 2024]



Fixing Models with DRSA [Chormai et al. TPAMI 2024]



Summary

Summary

- Machine learning comes with a well-defined cost-minimization formulation.

$$\arg\min_{f} \left\{ \int \ell(f(\boldsymbol{x}), t) dp(\boldsymbol{x}, t) \right\}$$

Summary

- Machine learning comes with a well-defined cost-minimization formulation.

$$\arg\min_f \left\{ \int \ell(f(\boldsymbol{x}), t) dp(\boldsymbol{x}, t) \right\}$$

- In practice, the minimization problem cannot be fully specified, because we do not have access to the true distribution $p(\boldsymbol{x}, t)$. An empirical approximation is:

$$\arg\min_f \left\{ \frac{1}{N} \sum_{k=1}^{N} \ell(f(\boldsymbol{x}_k), t_k) \right\}$$

Summary

- Machine learning comes with a well-defined cost-minimization formulation.

$$\arg\min_f \left\{ \int \ell(f(\boldsymbol{x}), t) dp(\boldsymbol{x}, t) \right\}$$

- In practice, the minimization problem cannot be fully specified, because we do not have access to the true distribution $p(\boldsymbol{x}, t)$. An empirical approximation is:

$$\arg\min_f \left\{ \frac{1}{N} \sum_{k=1}^{N} \ell(f(\boldsymbol{x}_k), t_k) \right\}$$

- Many functions f can fit the available data. Some may generalize better than others. Choosing a set of possible functions $\mathcal{F}$ is crucial, but difficult.

Summary

- Machine learning comes with a well-defined cost-minimization formulation.

$$\arg\min_f \left\{ \int \ell(f(\boldsymbol{x}), t) dp(\boldsymbol{x}, t) \right\}$$

- In practice, the minimization problem cannot be fully specified, because we do not have access to the true distribution $p(\boldsymbol{x}, t)$. An empirical approximation is:

$$\arg\min_f \left\{ \frac{1}{N} \sum_{k=1}^{N} \ell(f(\boldsymbol{x}_k), t_k) \right\}$$

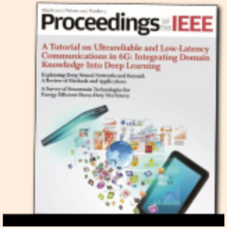
- Many functions f can fit the available data. Some may generalize better than others. Choosing a set of possible functions $\mathcal{F}$ is crucial, but difficult.
- Explainable AI (cf. [Samek et al. Proc IEEE 2021]) places the user in the loop. Doing so in an actionable way can be formulated as an own optimization (e.g. DRSA).

Our Review Paper

W Samek, G Montavon, S Lapuschkin, C Anders, KR Müller

Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications

Proceedings of the IEEE, 109(3):247-278, 2021



With the broader and highly successful usage of machine learning (ML) in industry and the sciences, there has been a growing demand for explainable artificial intelligence (XAI). Interpretability and explanation methods for gaining a better understanding of the problem-solving abilities and strategies of nonlinear ML, in particular, deep neural networks, are, therefore, receiving increased attention. In this work, we aim to: 1) provide a timely overview of this active emerging field, with a focus on " post hoc " explanations, and explain its theoretical foundations; 2) put interpretability algorithms to a test both from a theory and comparative evaluation perspective using extensive simulations; 3) outline best practice aspects, i.e., how to best include interpretation methods into the standard usage of ML; and 4) demonstrate successful usage of XAI in a representative selection of application scenarios. Finally, we discuss challenges and possible future directions of this exciting foundational field of ML.

Check our Website

Online demos, tutorials, code examples, software, etc.

References

[1] S. Bach et al. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLoS ONE* 10.7 (July 2015), e0130140.

[2] P. Chormai et al. "Disentangled Explanations of Neural Network Predictions by Finding Relevant Subspaces". In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2024). DOI: 10.1109/TPAMI.2024.3388275.

[3] S. Lapuschkin et al. "Analyzing Classifiers: Fisher Vectors and Deep Neural Networks". In: *CVPR*. IEEE Computer Society, 2016, pp. 2912–2920.

[4] S. Lapuschkin et al. "Unmasking Clever Hans Predictors and Assessing What Machines Really Learn". In: *Nature Communications* 10 (2019), p. 1096.

[5] W. Samek et al. "Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications". In: *Proc. IEEE* 109.3 (2021), pp. 247–278.

[6] V. Vapnik. *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science. Springer, 2000.