Entropy based explanation of N.N.

Santi Enrico
santi.enrico@spes.uniud.it

University of Udine

Feb. 2024

Presentation overview

XAI introduction

Introduction

AI based algorithms in recent years have become more and more popular.

$\downarrow$

Why ? **Because they work well.**

Their advancement enabled industry and academia to tackle problems that were thought intractable before.

Introduction

AI based algorithms in recent years have become more and more popular.

$\downarrow$

Why ? **Because they work well.**

Their advancement enabled industry and academia to tackle problems that were thought intractable before.

Problem : Such algorithms/models are usually included in the systems as black boxes, though, their answer is not always correct and usually they don't provide justification for their answer.

Introduction (cont.)

So we can't leverage the power of these black boxes in safety critical systems as we would do for other proven correct black box algorithms.

XAI (eXplainable AI) tries to overcome this limitation by polishing/opening these black boxes, more specifically by improving some of the model aspects :

- **Transparency** → Simulatability + Decomposition + Alg. transparency
- **Interpretability** → Answers to the question *Does the model provide an explanation of its process and an output justification ?*
- **Trustworthiness/confidence** → Given the model and the explanations *how much do users trust them ?* (e.g. self driving cars)

White boxes exist too

XAI techniques $\longrightarrow$ try to open black boxes.

But **white boxes exist too** (transparent by default models).

$\downarrow$

In general these are simpler models, which in many contexts tend to overfit, thus can't we restrict ourselves using just these white boxes.

White boxes exist too

XAI techniques $\longrightarrow$ try to open black boxes.

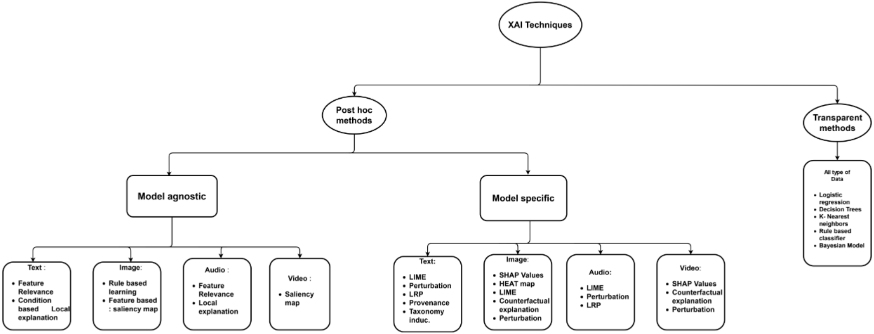
But **white boxes exist too** (transparent by default models).

$\downarrow$

In general these are simpler models, which in many contexts tend to overfit, thus can't we restrict ourselves using just these white boxes.

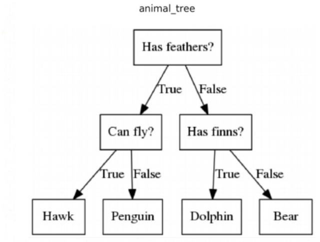
- Decision trees
- KNN
- Logistic regression
- ...

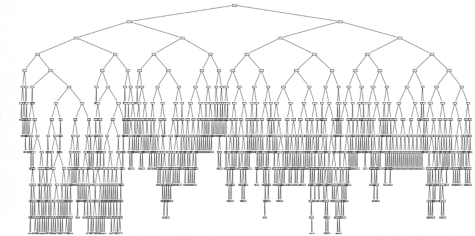
XAI taxonomy



More grey than white boxes

When white box models grow in size they tend to become more opaque by losing some transparency.

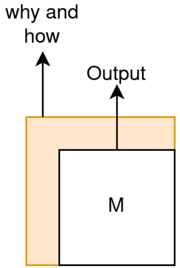




Consider for example simulatability or decomposition.

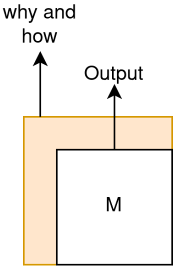
Post hoc methods

Post hoc methods → take a black box model and attach some additional components to explain it. These methods build on top of existing models.



Post hoc methods

Post hoc methods → take a black box model and attach some additional components to explain it. These methods build on top of existing models.



Can be further subdivided in :

- **Model specific** : They aim to explain a particular class of models by also leveraging their structure.
- **Model agnostic** : More general class of models which doesn't make assumptions on the structure of the model, thus they can only analyze I/O pairs.

An orthogonal subdivision

Instead of subdividing post hoc methods into model specific or agnostic we can subdivide the on the basis of their explanations :

- Local explanations → They explain only why a single sample produces such output (e.g. LIME). They don't scale in general.
- Global explanations → They explain the whole model or whole classes of samples (if we are in a classification task).

An orthogonal subdivision

Instead of subdividing post hoc methods into model specific or agnostic we can subdivide the on the basis of their explanations :

- Local explanations → They explain only why a single sample produces such output (e.g. LIME). They don't scale in general.
- Global explanations → They explain the whole model or whole classes of samples (if we are in a classification task).

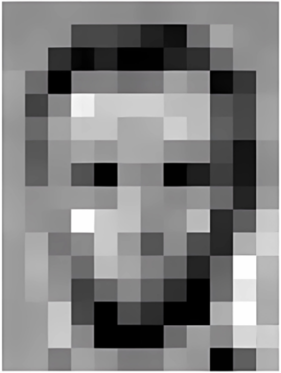
Explaining output in terms of input it may not be sufficient though.

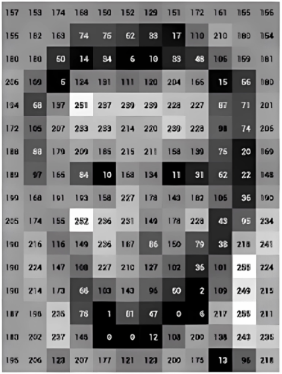
We need concepts

What do you see ?

157	153	174	168	150	152	129	151	172	161	156	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

We need concepts (cont.)





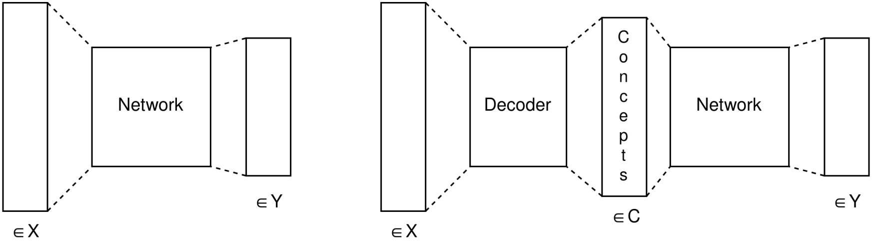
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

We need concepts (cont.)

Input features may not be human understandable, so **concept based methods** were introduced.

Concepts are high level features which are human understandable

Concept based methods take concepts as input. In particular given a classification task, a **Concept based classifier** is a function $h : C \rightarrow Y$ which takes as input concepts extracted by a decoder ($g : X \rightarrow C$).

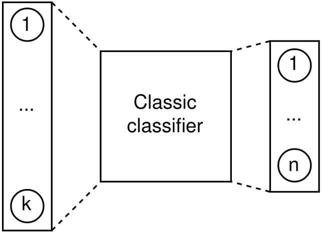


Entropy based explanation of N.N.

Model topology

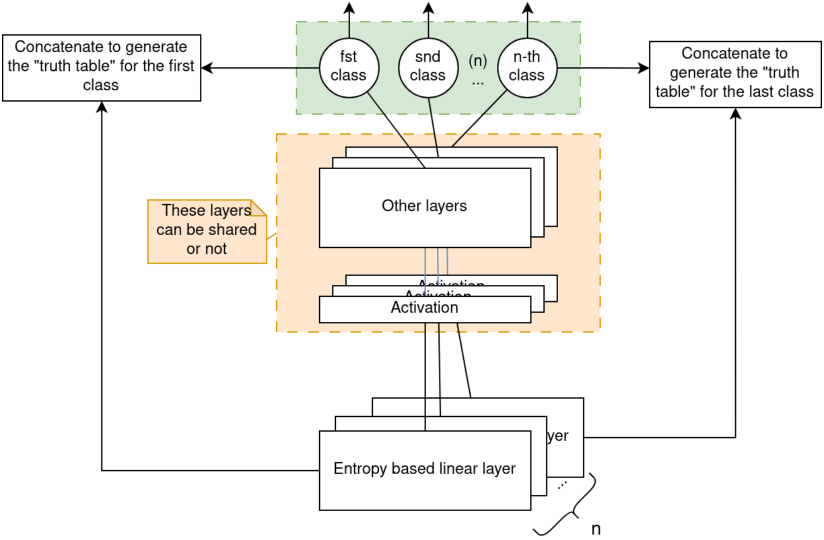
Context : **Classification of samples with k concepts in n classes**

The topology of the network (a concept based classifier) resembles a classic classifier, the main difference (aside the truth tables) is in the first layer, which now is a *entropy based linear layer* and in the fact that there are many n, one for each class, of them.



The goal of this network is to classify correctly the samples as well as characterize with a relatively short boolean formula all the n classes.

Model topology (cont.)



Model topology (cont.)

Note that we have up to the last layer a stack of n classifiers (if we don't merge them first in the middle layers).

The main component that drives the model explainability is the entropy linear layer which takes as input a concept tuple (sample) and computes two pieces of data :

- hidden states which the rest of the network will use to compute the output.
- a boolean concept tuple describing the sample by a boolean formula only on it's most relevant components.

Model topology (cont.)

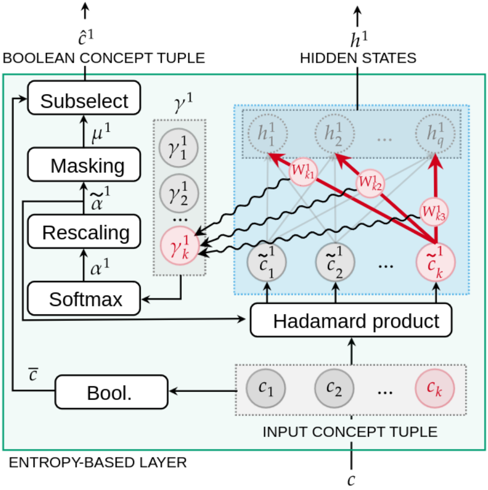
Note that we have up to the last layer a stack of n classifiers (if we don't merge them first in the middle layers).

The main component that drives the model explainability is the entropy linear layer which takes as input a concept tuple (sample) and computes two pieces of data :

- hidden states which the rest of the network will use to compute the output.
- a boolean concept tuple describing the sample by a boolean formula only on it's most relevant components.

The goal of each boolean concept tuple (one for each class) is to describe shortly in concept terms (relevant to the $i-th$ class) the sample considered.

Entropy linear layer



n of these characterize the first layer of the network. They are linear, so parameter wise they only need a weight matrix W^i and a bias vector b^i.

Entropy linear layer (cont.)

To compute the two outputs we first need to determine the attention which each class (i) has towards each concept (j).

For simplicity in the previous image such attention is depicted by the γ vector (they are not additional parameters) :

$$\gamma_j^i = \|W_j^i\|_1$$

Entropy linear layer (cont.)

To compute the two outputs we first need to determine the attention which each class (*i*) has towards each concept (*j*).

For simplicity in the previous image such attention is depicted by the γ vector (they are not additional parameters) :

$$\gamma_j^i = \|W_j^i\|_1$$

The attention each class i has towards the $j-th$ concept is the norm of the $j-th$ row of the weight matrix of the $i-th$ entropy linear layer.

A higher γ_j^i value means the concept j is more relevant for characterizing the class i, on the other hand $\gamma_j^i \rightarrow 0$ indicates that such concept is irrelevant in this context.

Entropy linear layer (cont.)

The attention vector of each layer is then normalized with a slightly modified softmax function :

$$\alpha_j^i = \frac{e^{\gamma_j^i/\tau}}{\sum_{l=1}^{k} e^{\gamma_l^i/\tau}} \qquad \textit{with } \tau \in \mathbb{R}^+$$

τ tells how to weight differently the concept difference (i.e. it drives the length of the boolean concept tuple).

The values are then just rescaled for numerical reasons :

$$\widetilde{\alpha}_j^i = \frac{\alpha_j^i}{\max_u \alpha_u^i}$$

Entropy linear layer (cont.)

Having the attention values for each class now we can now compute the Hadamard product of the attention values with the concept tuple :

$$\widetilde{c}^i = c \odot \widetilde{\alpha}^i$$

By doing so in $\widetilde{c}^i$ we will have that the less relevant concepts for the $i - th$ class will be sensibly smaller (in absolute value) than their counterparts in the input tuple c.

Entropy linear layer (cont.)

Having the attention values for each class now we can now compute the Hadamard product of the attention values with the concept tuple :

$$\widetilde{c}^i = c \odot \widetilde{\alpha}^i$$

By doing so in $\widetilde{c}^i$ we will have that the less relevant concepts for the $i-th$ class will be sensibly smaller (in absolute value) than their counterparts in the input tuple c.

As a final step the linear result of the linear layer (i.e. the hidden state) is calculated :

$$h^i = W^i \widetilde{c}^i + b^i$$

Entropy linear layer - the boolean concept tuple ?

Define a function $\mathbb{I}_{\geq \epsilon}(X) : \mathbb{R}^m \rightarrow \{0, 1\}^m$ that given a vector X it returns a vector in which each component is 0 if $x_i < \epsilon$ or 1 else.

To define the boolean interpretation for the tuple c, namely $\hat{c}^i$ we calculate :

$$\hat{c}^i := \xi(\mathbb{I}_{\geq \epsilon}(c), \mathbb{I}_{\geq \epsilon}(\widetilde{\alpha}^i))$$

where ξ returns a vector by selecting the components of the first vector in which the second vector in the respective position contains 1. The resulting vector is thus a 0 and 1 vector of length $\leq k$.

Training

Consider a training set C and a sample $c \in C$.
First a boolean representation of the network output is defined, namely $\bar{f}^i(c) = \mathbb{I}_{\geq \epsilon}(f^i(c))$ where $f^i(c)$ is the prediction for c to belong into the $i-th$ class (given by the $i-th$ entropy layer). An element of the $i-th$ truth table is made up of the concatenation (column-wise) of $\hat{c}^i$ and $\bar{f}^i(c)$.

The elements of the truth table generated in this way will be concatenated row-wise. The final $i-th$ truth table [1] will be expressed as :

$$T^i = (\hat{C}^i || \bar{f}^i)$$

1. Which is not a truth table in the classical sense

Where is the entropy ?

The entropy comes into play in the loss function :

$$\mathcal{L}(f, y, \alpha^1, \alpha^2, ..\alpha^n) = L(f, y) + \lambda \sum_{i=1}^{n} \mathcal{H}(\alpha^i)$$

Where $L(f, y)$ is any loss function used in a multi-label classification task (e.g. cross-entropy or KL divergence) and λ is an hyper-parameter balancing the accuracy of the model and the succinctness of the explanation given.

Where is the entropy ?

The entropy comes into play in the loss function :

$$\mathcal{L}(f, y, \alpha^1, \alpha^2, ..\alpha^n) = L(f, y) + \lambda \sum_{i=1}^{n} \mathcal{H}(\alpha^i)$$

Where $L(f, y)$ is any loss function used in a multi-label classification task (e.g. cross-entropy or KL divergence) and λ is an hyper-parameter balancing the accuracy of the model and the succinctness of the explanation given.

The second part of the loss seems a L1 Regularization over α^i ($\lambda \sum_{i=1}^{n} |(\alpha^i)|$).
But according to the results, L1 regularization wasn't penalizing enough the attention vectors to derive short explanations.

The explanations

How do we generate explanations ?

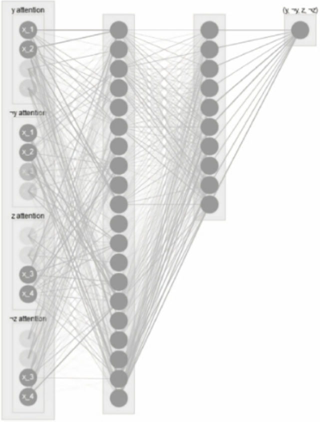
- Local (single sample) explanation : an entry in the truth table is a column wise concatenation of $\hat{c}^i$ and $\bar{f}^i(c)$. To derive an explanation for a sample need to conjoin the true concepts and the negated counterparts of the false ones. Repeating this process for all the classes will give us a n conjuctions explaining why the sample belongs or not to each class.
- Class explanation : provides an explanation of a whole class, namely it characterizes all the samples that will be classified as belonging to such class. To derive such explanation for a generic class i consider T_i and put in a disjunction all the local explanations for the samples positively labelled (i.e $\bar{f}^i(c) = 1$).

An example

Four concepts, four classes (two actually) :

Input (raw data)

x1	x2	x3	x4	y	¬y	z	¬z
0	0	0	0	0	1	0	1
0	1	0	0	1	0	0	1
1	0	0	0	1	0	0	1
1	1	0	0	0	1	0	1
0	0	0	0	0	1	0	1
0	0	0	1	0	1	1	0
0	0	1	0	0	1	1	0
0	0	1	1	0	1	1	0



Logic explanations

Class y

∀x: (x1 ∧ ¬x2) ∨ (x2 ∧ ¬x1) ↔ y

Class ¬y

∀x: (x1 ∧ x2) ∨ (¬x1 ∧ ¬x2) ↔ ¬y

Class z

∀x: x3 ∨ x4 ↔ z

Class ¬z

∀x: ¬x3 ∧ ¬x4 ↔ ¬z

The topology in this case seems a bit counter intuitive.

Conclusions

Conclusions

The idea is very intuitive and according to the results, the performances are in line with those of many white box models.

Conclusions

The idea is very intuitive and according to the results, the performances are in line with those of many white box models.

But...

- If the network become bigger, do explanations still hold ? recall that the explanations come only from the first layer...
- Can it scale ? we are repeating the input layer n times at best.

References

Plamen P. Angelov, Eduardo A. Soares, Richard Jiang, Nicholas I. Arnold, and Peter M. Atkinson.
Explainable artificial intelligence : an analytical review.
WIREs Data Mining and Knowledge Discovery, 11(5) :e1424, 2021.

Pietro Barbiero, Gabriele Ciravegna, Francesco Giannini, Pietro Lió, Marco Gori, and Stefano Melacci.
Entropy-based logic explanations of neural networks.
Proceedings of the AAAI Conference on Artificial Intelligence, 36(6) :6046–6054, June 2022.

Vaishak Belle and Ioannis Papantonis.
Principles and practice of explainable machine learning.
Frontiers in Big Data, 4, 2021.

Prashant Gohel, Priyanka Singh, and Manoranjan Mohanty.
Explainable ai : current status and future directions, 2021.

Timo Speith.
A review of taxonomies of explainable artificial intelligence (xai) methods.
In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 2239–2250, New York, NY, USA, 2022. Association for Computing Machinery.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan.
Axiomatic attribution for deep networks, 2017.

Kaiwen Xu, Kazuto Fukuchi, Youhei Akimoto, and Jun Sakuma.
Statistically significant concept-based explanation of image classifiers via model knockoffs, 2023.

Fin.