A GLIMPSE OF EXPLAINABLE AI, ENTROPY BASED

Foundations of Neural Networks

Santi Enrico
University of Udine
Academic year 2023/24

Contents

1 Introduction

AI based algorithms in recent years have become more and more popular and their advancement enabled industry and academia to tackle problems that were thought intractable before [1]. Such algorithms rely on models which are trained with a large training set and in many cases are used as black or opaque boxes [2], this is true for many complex models[1]. One of the major "problems" of using these models as black boxes though arises from the fact that their answer is not always perfect [1]. Indeed in many critical applications we can't leverage a quasi-perfect black box model as we'd leverage a 100% correct black box, such as a (formally proven correct) SAT solver. Related to the fact that the answer of a model is not 100% perfect we'd like also to be provided with some information on it. For example we'd like to know how such an answer was derived and why, this will help a human operator to make his/her own decision and check (if possible) the correctness of such answer.
In this way we could draw another parallelism between explainable AI models and SAT solvers. Consider an AI model providing a binary answer, having an explanation for such answer (as well as having the answer) could be thought as having a probabilistic[2] SAT solver returning us an assignment satisfying or not the SAT instance, in addition to the "problable" answer[3] to the instance. The assignment could be used to check the "problable" answer given, and in the satisfiable cases could be used to confirm the answer provided.

1.1 About this report

The goal of this report is to provide a glimpse of an overview to the XAI field and then to sum up the core and the ideas of [3]. For all the tests and benchmarks refer to [3], they won't be mentioned.

2 Overview of the current XAI state

2.1 XAI objectives

Some of the main objectives of explainable AI are the improvement of the following properties of a model[1]:

- **Transparency**: In [4] is defined as the combination of the following model properties:
  - Simulatability: indicates how easily the execution of the model can be simulated by a human, small models are clearly preferred in this case.
  - Decomposability: is related to how each aspect of the model (e.g. hyper-parameters, input data and the model components) plays a role in the model behaviour.
  - Algorithmic transparency: is the ability of the procedure carried out by the model (producing the output) to be understood by the user. As reported in [4], the requirement for a model satisfy such property is for the user to be able to inspect the model through a mathematical analysis.

[1] White box simpler models exist too.
[2] I.e. an algorithm returning the correct answer only in a fraction of the cases, similar to Monte Carlo algorithms.
[3] I.e. satisfiable or unsatisfiable.

- **Interpretability**: the capability of providing human understandable explanation on how the decision making process carried out by the model works and how the output can be justified.
- **Explainability**: is a form of interpretability targeted to the end user. It's similar to the interpretability concept though it focuses less on the internal functioning of the model and more on the decisions made. The jargon is usually also simplified.
- **Trustworthiness/Confidence**: is a property related to all the previous ones and describes the trust users put in decisions made by the AI system considered.

2.2 Black and white boxes

Not all AI models are equal and some are less opaque than others.
A first distinction in the XAI taxonomy is done on Post hoc methods and Transparent or ante-hoc methods. This latter category includes models satisfying transparency by default, such as decision trees, kNN, Bayesian networks and logistic regression. Transparency is not sufficient to guarantee the exaplainaibility of a model though, in addition when the model starts to grow, some aspects of the transparency tend to become more opaque (think for example how *simulatability* changes when a decision tree grows). However, what actually precludes such models from being used for all ML problems is their accuracy[4], which in general on more complex tasks is lower compared to the one of more opaque methods such as deep neural networks [5].

For what concerns Post hoc methods, we can further classify them into model specific or model agnostic. In a nutshell model specific methods explain and bring transparency to a particular set of models leveraging their structure. On the other hand model agnostic are more generic and work for any AI model, in this case a major role is played by analysis of input/output pairs (since we can't make assumptions nor leverage a model architecture). An example of post-hoc methods are LIME methods (Local interpretable Model-agnostic explanations), which extract feature importance from the samples by perturbing real input samples 1. Post hoc methods are built on top of an existing model and in contrast to transparent models are not actually part of it. The overall XAI taxonomy is shown in figure 1.

Another orthogonal[5] subdivision of XAI techniques can be made on whether they provide local or global explanations of the model. Locally explaining a model means providing answers of why and how *a single* sample produced such result, and in general these techniques don't scale globally[6], i.e. they aren't capturing the overall model behaviour [5].
As briefly mentioned there exist local techniques which explain black box models by ranking the most relevant input features, though such features may lack high level human interpretability (think for example to single pixel intensities in an image). To tackle such issue *concept based methods* have been developed, in which instead of taking into account features (which considered on their own can present little to know meaning) they explain the output

[4] e.g. In a classification task a single decision tree would tend to overfit, while a well crafted neural network tries to limit such phenomena.
[5] With respect to post hoc and transparent methods.
[6] Small input perturbations could lead to very different explanations[4]

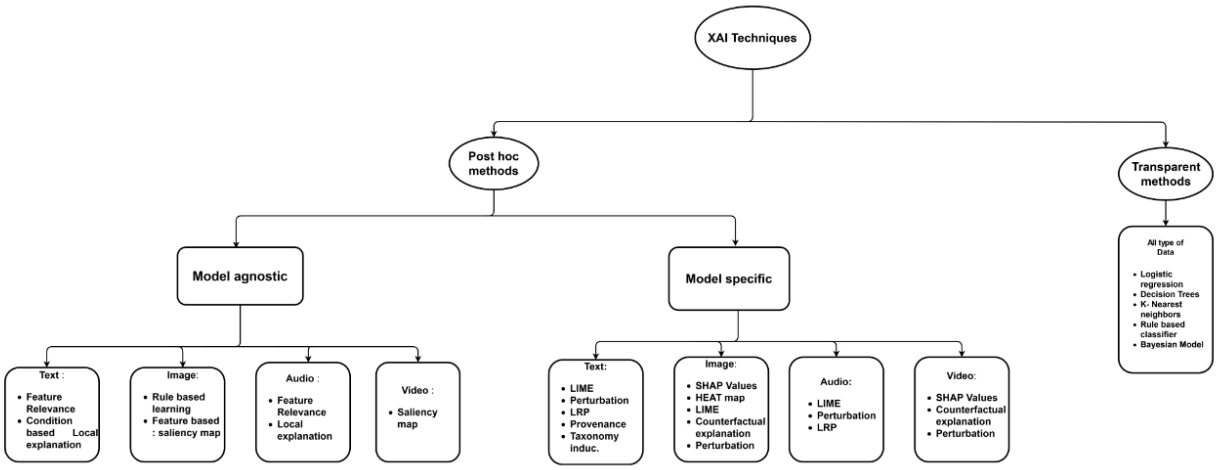

Fig. 1: Overview of the XAI taxonomy (taken from [1])

in term of higher level concepts. Few methods though manage to explain how the model uses such features in order to compute the output value.

3 Entropy and Neural Networks

As introduced in the previous section the models we deal with in realistic scenarios are those non transparent, since in complex scenarios they provide better performances. For this reason we will now sum up an explainable by design class of models (i.e. transparent) which have some constrains, but on the other hand, are more complex than the classic transparent methods mentioned.

3.1 Concept-based classifiers

Before diving into the model presented in [3] we need to introduce a particular type of classifiers, namely Concept-based classifiers, to do so we will start from a multi-class classification problem.
In a nutshell a multi-class classification task consists in given as input a vector of features describing some object, return the category in which the specified input belongs. In general the output is specified as a one hot encoding vector[7]. More formally, given $X \subseteq R^d$ and n possible classes we want to derive a function (classifier) $f : X \to Y$ minimizing the risk or true error over X. Y in this case is the the set of one hot vectors of size c. Alternatively If the problem is a multi-label classification problem $Y \subseteq \{0, 1\}^n$.

Concept-based classifier: A classifier in which both the input and output space are made up of human interpretable symbols.

When the input space of a classification problem is a non easily interpretable e.g. we have many non human interpretable features ranging into an infinite set (consider for example a black/white image) instead of training a classifier $f : X \to Y$, a decoder mapping the input

[7] I.e. a vector in which only one component is 1 and all the others are 0.

space to human-interpretable concepts $g : X \to C$ is first applied and only then the concept based classifier $h : C \to Y$ is used[6].

3.2 The model structure

Assuming to already have input concepts (i.e. high level features describing the instances) we will now present the overall structure of an entropy based network introduced in [3]. Consider to be given a classification task in which the inputs characterized by k concepts and we want to classify them into n classes. Figure 2 depicts the model used during the training phase of such classification task (for the actual production usage the components related to the truth tables generation won't concatenate data anymore).

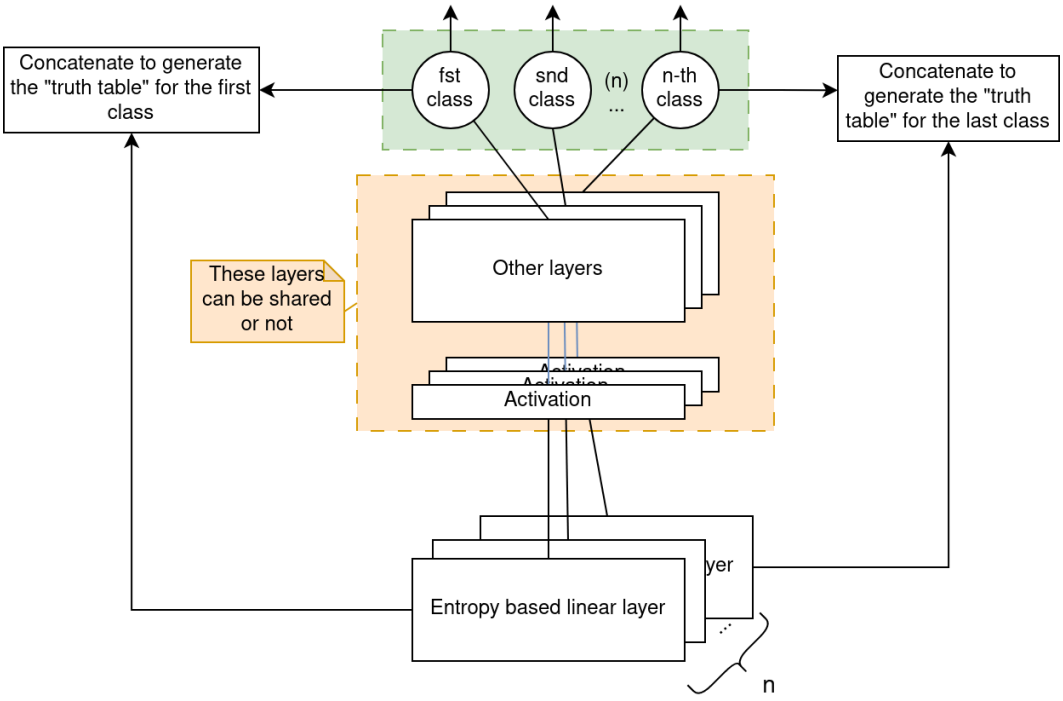

Fig. 2: Topology of an entropy based N.N.

The topology of the network can be summarized as:

- A series of n (one for each class) linear entropy based layers.
- A series of n activation layers.
- A series of n additional layers independent one from another or a series of shared additional layers (i.e. all entropy based linear layers will merge into a bigger network).
- An output layer of size n (or n single neurons connected independently to a subnetwork if the hidden layers are independent).

- n components responsible for concatenating the boolean concept tuple of each entropy based linear layer and the output of the neuron of the respective class (i.e. the $i-th$ component will concatenate the boolean concept tuple of entropy layer i with the output of the $i-th$ neuron in the output layer). These components are responsible for the generation of the n "truth tables" during the training phase.

We now focus on the entropy based linear layers and how the truth tables generated drive the model explainability. The idea behind entropy based layers is to induce the model into choosing a limited number of relevant concepts to characterize the output and provide a brief explanation (in accordance to Occam's razor).

Consider now a generic $(i-th)$ entropy based layer, what we want to learn in the training

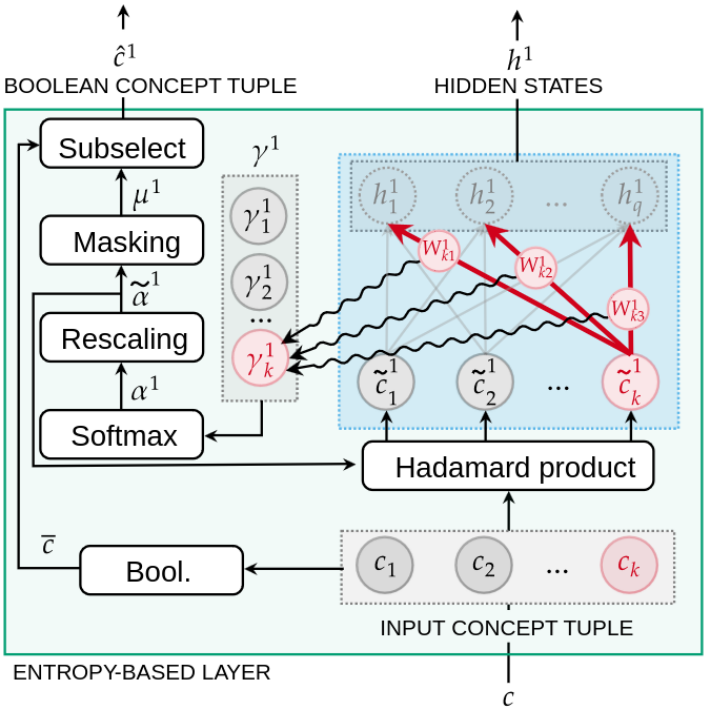

Fig. 3: A zoom into the entropy based layer referred to the first class (taken from [3]) in which the $k-th$ concept is highlighted.

phase are (since it's a linear layer) a weight matrix W^i and a bias vector b^i. Now consider the $j-th$ concept (see Figure 3), a first approximation of the relevance of the $j-th$ concept for the input to belong to the $i-th$ class is given by computing:

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

A higher γ_j^i value means the concept j is more relevant for characterizing the class i, on the other hand $\gamma_j^i \to 0$ indicates that such concept is irrelevant in this context. In order to select which concepts consider though, a slightly modified softmax function is calculated:

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

This the above function is just used to scale each γ^i into a discrete probability distribution. τ is referred as the temperature and can be intuitively used to select the number of concepts to consider (e.g. for very high values of τ all concepts are depicted as having the same relevance, while if τ goes to 0 only one concept will be disproportionately relevant). For numerical reasons α^i is transformed into $\widetilde{\alpha}^i$ and the Hadamard product with the concepts is then calculated:

$$\widetilde{c}^i = c \odot \widetilde{\alpha}^i_j \qquad\qquad with\ \ \widetilde{\alpha}^i_j = \frac{\alpha^i_j}{\max_u \alpha^i_u}$$

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 embeddings are calculated: $h^i = W^i\widetilde{c}^i + b^i$. The embeddings can then be further processed by a conjoined N.N. or n neural networks.
The entropy layer though, in addition of computing the above "linear" function, also computes the boolean interpretation for the tuple $c \in C$ given as input[8]. To do so we define a function $\mathbb{I}_{\geq\epsilon}(X) : \mathbb{R}^m \to \mathbb{R}^m$ that given a vector X it returns a vector in which each component is 0 if $x_i < \epsilon$ or 1 else. Finally 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 less or equal than k.
Now that we have seen what the model produces let's see how it's trained.

3.3 Training

Consider a training set C and a sample $c \in C$. First a boolean representation of the network output is defined, namely $\overline{f}^i(c) = \mathbb{I}_{\geq\epsilon}(f^i(c))$ 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 $\overline{f}^i(c)$. The process is repeated for all the samples in the training set, the elements of the truth table generated in this way will be concatenated row-wise. The final $i-th$ truth table[9] will be expressed as:

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

Having in mind how the truth tables are generated, recall that α^i is a probability vector of k weights depicting the relevance of the k concepts for the $i-th$ class[10]. We will denote the entropy of a probabilistic vector α^i by $\mathcal{H}(\alpha^i)$. Considered all this the loss function used is the following:

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

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

[8] C is a generic training set.
[9] Which is not a truth table in the classical sense
[10] The vector is calculated and used by the $i-th$ entropy layer

3.4 The explanations

We now have a trained network and the relative truth tables, but how do we generate explanations?
We first distinguish two kinds of explanations (similar to the local and global explanation concept mentioned in 2.2):

- Local (single sample) explanation: as mentioned above an entry in the truth table is a column wise concatenation of $\hat{c}^i$ and $\overline{f}^i(c)$. To derive an explanation for a (seen) 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 $\overline{f}^i(c) = 1$).

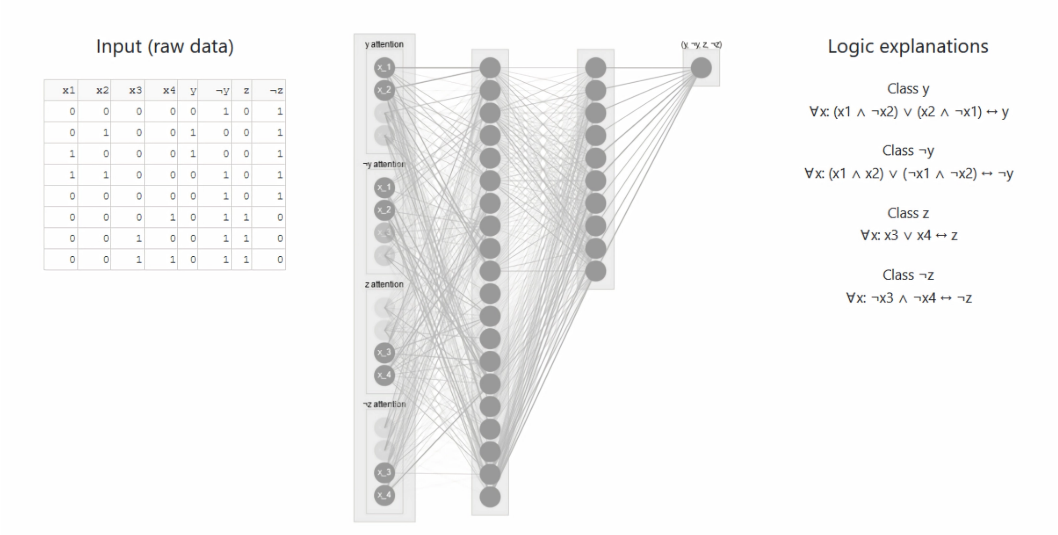

Fig. 4: Instance of an entropy based N.N. with $k = 4$ and $n = 4$ (taken from 3, the topology may be a bit misleading).

3.5 Usage

The trained model can then easily be used as every other N.N., given the input, the network produces the output and the reason why the input belongs or not to each class is explained by the class characterizing formula derived from the truth tables.

4 Conclusions

We conclude by recalling the importance of explainable and accurate methods in AI and how many fields could benefit from the merging of such classes of methods. In addition we mention a major specific advantage of the entropy based networks, namely by having a logic

(and concise) explanation on the reasons why the model gave such answer we can reverse engineer the process and tackle potential model vulnerabilities. This, in turn, could lead to a more conscious and wide diffusion of such AI models.

References

[1] P. Gohel, P. Singh, and M. Mohanty, "Explainable ai: current status and future directions," 2021.

[2] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold, and P. M. Atkinson, "Explainable artificial intelligence: an analytical review," *WIREs Data Mining and Knowledge Discovery*, vol. 11, no. 5, p. e1424, 2021. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1424

[3] P. Barbiero, G. Ciravegna, F. Giannini, P. Lió, M. Gori, and S. Melacci, "Entropy-based logic explanations of neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, p. 6046–6054, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1609/aaai.v36i6.20551

[4] V. Belle and I. Papantonis, "Principles and practice of explainable machine learning," *Frontiers in Big Data*, vol. 4, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fdata.2021.688969

[5] T. Speith, "A review of taxonomies of explainable artificial intelligence (xai) methods," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 2239–2250. [Online]. Available: https://doi.org/10.1145/3531146.3534639

[6] K. Xu, K. Fukuchi, Y. Akimoto, and J. Sakuma, "Statistically significant concept-based explanation of image classifiers via model knockoffs," 2023.